Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional

Unidad Zacatenco

Departamento de Control Automático

Humano en el bucle usando aprendizaje por reforzamiento

Tesis que presenta:

Carlos Armando Castillo Díaz

Para obtener el Grado de

Maestro en Ciencias

En la especialidad de

Control Automático

Director de Tesis:

Dr. Wen Yu Liu

Ciudad de México Agosto, 2022

Agradecimientos

- *A mis padres y hermanos, los cuales siempre me han apoyado, tolerado y motivado en las mejores y peores situaciones.*
- *Al Dr. Wen Yu Liu por haber compartido su conocimiento conmigo, además, por su apoyo, paciencia y tolerancia a lo largo del desarrollo de este trabajo.*
- *Agradezco al Consejo Nacional de Ciencia y Tecnología (CONACYT) por la beca que me fue otorgada.*

Resumen

En esta tesis se presentan sistemas de Humano en el bucle (HITL, por sus siglas en inglés) con lazos de control de retroalimentación internos y externos. Específicamente, nuestra formulación del problema considera que las leyes de control del lazo interno utilizan un algoritmo de aprendizaje por reforzamiento integral, de modo que el sistema general opera cerca de su comportamiento ideal como se desea en presencia de condiciones adversas debido a fallas y/o imprecisiones en el modelado. Se demuestra que el esquema de control adaptativo óptimo basado en IRL es un controlador dinámico con una estructura actor-crítico que tiene una memoria cuyo estado está dado por la función de costo o valor.

El aprendizaje por refuerzo es una herramienta de propósito general para diseñar controladores (políticas) óptimos y/o adaptables utilizando únicamente mediciones del estado, control y una señal de recompensa. Esto lo hace adecuado para sistemas que son difíciles de controlar utilizando metodologías de control convencionales.

Además, consideramos que las leyes de control del lazo externo existen debido al empleo de métodos de guía de alto nivel y/o cierre de bucle secuencial. Se considera que los humanos inyectan comandos directamente a la dinámica del bucle externo en respuesta a los cambios en el sistema, donde los comandos del bucle externo afectan la dinámica del bucle interno en respuesta a los comandos recibidos de los humanos así como en respuesta a los cambios en el sistema físico.

El sistema de control propuesto es evaluado en un péndulo invertido sobre un móvil, donde los resultados muestran que el aprendizaje por reforzamiento integral usado para compensar términos dinámicos en un robot junto con controladores clásicos PID responden de forma favorable en tareas de regulación y seguimiento de trayectoria. De igual manera que el lazo externo usando HITLC ayuda a reducir problemas con perturbaciones y/o problemas de modelado del sistema.

Abstract

In this thesis, Human in the Loop (HITL) systems with internal and external feedback control loops are presented. Specifically, our formulation of the problem considers that the inner loop control laws use an integral reinforcement learning algorithm, so that the overall system operates close to its ideal behavior as desired in the presence of adverse conditions due to faults and/or modeling inaccuracies. It is shown that the optimal adaptive control scheme based on IRL is a dynamic controller with a critical-actor structure that has a memory whose state is given by the cost or value function.

Reinforcement learning is a general purpose tool for designing optimal and/or adaptive controllers (polity) using only measurements of state, control, and a reward signal. This makes it suitable for systems that are difficult to control using conventional control methodologies.

Furthermore, we consider that the outer loop control laws exist due to the use of high-level guidance methods and/or sequential loop closure. Humans are considered to inject commands directly into the outer loop dynamics in response to changes in the system, where the outer loop commands affect the inner loop dynamics in response to commands received from humans as well as in response to changes in the system. changes in the physical system.

The proposed control system is evaluated in an inverted pendulum on a mobile, where the results show that integral reinforcement learning used to compensate dynamic terms in a robot together with classical PID controllers respond favorably in regulation and trajectory tracking tasks. In the same way that the outer loop using HITLC helps to reduce problems with disturbances and/or system modeling problems.

Índice general

Índice de figuras

Índice de tablas

Introducción

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 [1–3], las redes difusas [4, 5] y el aprendizaje por refuerzo [6].

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 con realimentación (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.

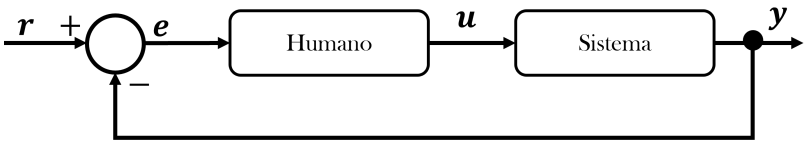


Figura 1: Human in the loop control

Los sistemas de HITLC ofrecen oportunidades interesantes para una amplia gama de

aplicaciones, incluida la gestión de energía [7], el cuidado de la salud [8] y los sistemas de automóviles [9]. 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 [10] 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 [11–13].

c) *La resiliencia y robustez del sistema.* Un sistema resiliente [14] es un sistema que se adapta a la incertidumbre cambiando su método de operaciones mientras continúa funcionando [15]. Sin embargo, un sistema robusto es el sistema que continúa funcionando en presencia de incertidumbre limitada sin ningún cambio en el sistema original [16]. El grado de robustez y resistencia del sistema controlado afecta el comportamiento humano mientras controla el sistema [17]. 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 [18].

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 [19] [20].

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 [7, 21]. Tomando como ejemplos: a)Human-in-the-plant, b)Human-in-the-controller, c)Human-machine-symbols, d)Human-in-loops

Antecedentes y Motivación

Para mostrar la contribución de la tesis, es necesario conocer de forma breve el estado del arte que se tiene sobre la observación de estados y la identificación paramétrica en sistemas de orden fraccionario. 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" [22]. 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) [23–26].

Existe un trabajo significativo en la literatura sobre esquemas de HITLCPs [27, 28] y sistemas robóticos [29–36]. El tema de HITL se ha tratado con cierta generalidad, aunque sin una perspectiva integral de sistemas y controles. En [10] se propone una taxonomía de "Human In The Loop Control Physical Systems" (HITLCPS), (figura Figura 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)***.

La investigación sobre el modelado humano comenzó utilizando el concepto de describir

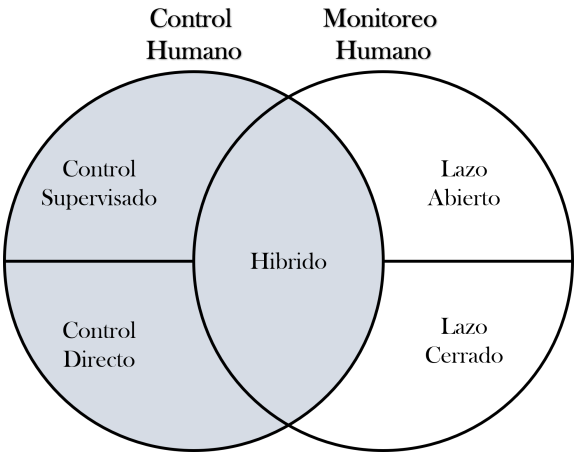


Figura 2: Taxonomía de aplicaciones de HITL [10]

la función del comportamiento humano en los trabajos de Tustin [37]. El modelo cuasi-lineal es uno de los primeros modelos humanos [38] propuestos por McRuer y Krendel [39], consta de una función descriptiva y una señal remanente que explica el comportamiento no lineal. McRuer y Krendel [40] 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 [41]. El modelo crossover, propuesto por McRuer y Graham [42], es un modelo humano prominente en aeronáutica [43, 44].

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

Teniendo en cuenta lo mencionado anteriormente, en está tesis se propone un observador PI fraccional, donde su construcción es basada en la propiedad IFAO, además el observador es robusto ante perturbaciones externas y para cierto tipo de sistemas también puede ser robusto ante incertidumbres paramétricas. Mientras que, el análisis de convergencia del error de observación se realiza utilizando el enfoque del acotamiento Mittag-Leffler en particular se demuestra que el error de observación es globalmente Mittag-Leffler acotado lo cual implica que el error de observación es uniformemente últimamente acotado.

Por otro lado, se introduce una nueva propiedad relacionada con la identificabilidad algebraica fraccional la cual permitirá traducir el problema de identificación paramétrica en un problema de observación de estado, de está forma, se puede aplicar el observador fraccional que se propone en está tesis para realizar la identificación paramétrica en linea de sistemas fraccionarios.

Finalmente, considerando como caso particular cuando el orden de derivación es entero, se muestra que la versión de orden entero del observador PI fraccional puede resolver los problemas de observación de estados e identificación paramétrica en sistemas de orden entero.

Objetivos

Diseñar un controlador PID usando el aprendizaje por reforzamiento integral como compensación de términos dinámicos en sistemas en un lazo interno. Además de proponer el diseño de un lazo externo tomando en cuenta al Humano para dar realimentación al sistema.

Para facilitar el cumplimiento del objetivo principal, este se divide en objetivos particulares. De tal forma que en la presente tesis los objetivos particulares son los

siguientes:

1. Investigar el estado del arte del aprendizaje por refuerzo, al igual que de Human in the loop en el control de sistemas lineales y no lineales para identificar los problemas específicos en el aprendizaje por refuerzo integral y las soluciones propuestas por otros autores.

2. Utilizar herramientas de aprendizaje reforzado y programación dinámica para resolver el problema del regulador cuadrático lineal para sistemas de tiempo continuo.

3. Dar una demostración de la convergencia del algoritmo de aprendizaje por reforzamiento integral (IRL, por sus siglas en inglés) que se propone.

4. Utilizar herramientas de aprendizaje reforzado y programación dinámica para resolver el problema del regulador cuadrático lineal para sistemas de tiempo discreto.

5. Comparar el desempeño del controlador PID e IRL de forma independiente, al igual que de forma conjunta, teniendo en cuenta caso con y sin perturbación en el sistema

6. Diseñar un método de control para el lazo externo haciendo uso de Human in the loop

7. Comparar los resultados de cada lazo teniendo en consideración la presencia y ausencia de perturbaciones

Estructura de la tesis

La tesis consta de cuatro capítulos, una conclusión general e ideas para un trabajo futuro, los capítulos se describen a continuación:

- **Capítulo 1. Preliminares.** Este capítulo presenta las ideas principales y los algoritmos del aprendizaje por refuerzo. Comenzamos con una discusión de MDP y luego nos enfocamos específicamente en una familia de técnicas conocidas como programación dinámica aproximada (o adaptativa) (ADP) o programación neurodinámica. Estos métodos son adecuados para el control de sistemas dinámicos, que es nuestro principal interés en el lazo interno. Además de enunciar un par de algoritmos necesarios para realizar la prueba de convergencia del algoritmo de IRL.

- **Capítulo 2. Aprendizaje por refuerzo para el diseño de control de lazo interno.**Este capítulo presenta un nuevo algoritmo basado en iteraciones de políticas que proporcionan un procedimiento de solución en línea para el problema de control óptimo para sistemas de tiempo continuo (CT), lineales e invariantes en el tiempo que tienen el modelo de espacio de estado $\dot{x}(t) = Ax(t)+Bu(t)$. Este es un algoritmo de aprendizaje adaptativo basado en el aprendizaje por refuerzo (RL) que converge a la solución de control óptima para el problema del regulador cuadrático lineal. Llamamos a esto un controlador adaptativo óptimo. El algoritmo de este capítulo proporciona un algoritmo de aprendizaje adaptativo óptimo en línea que resuelve el ARE en línea en tiempo real sin conocer la matriz A midiendo el estado y los datos de entrada $(x(t), u(t))$ a lo largo de las trayectorias del sistema. El algoritmo se basa en iteraciones de políticas y, como tal, tiene una estructura actor-crítico que consta de dos estructuras de aprendizaje adaptativo que interactúan.

- **Capítulo 3. Control adaptable en el lazo externo con Humano en el bucle.** En este capítulo se introduce el modelado del humano como un conjunto de ecuaciones diferenciales de coeficiente constante lineal.

Capítulo 1

Preliminares

El aprendizaje por refuerzo es un tipo de aprendizaje automático desarrollado en la Comunidad de Inteligencia Computacional en ingeniería informática. Tiene conexiones cercanas tanto con el control óptimo como con el control adaptativo. El aprendizaje por refuerzo se refiere a una clase de métodos que permiten el diseño de controladores adaptativos que aprenden en línea, en tiempo real, las soluciones a los problemas de control óptimo prescritos por el usuario.

Las estructuras actor-crítico que se muestran en la Figura 1.1 [45] son un tipo de algoritmos de aprendizaje por refuerzo. Estas estructuras brindan algoritmos de avance en el tiempo que se implementan en tiempo real donde un componente actor aplica una acción o política de control al entorno y un componente crítico evalúa el valor de esa acción. El mecanismo de aprendizaje respaldado por la estructura actor-crítico consta de dos pasos, a saber, la evaluación de la política, ejecutada por el crítico, seguida de la mejora de la política, realizada por el actor. El paso de evaluación de políticas se realiza observando desde el entorno los resultados de aplicar las acciones actuales

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

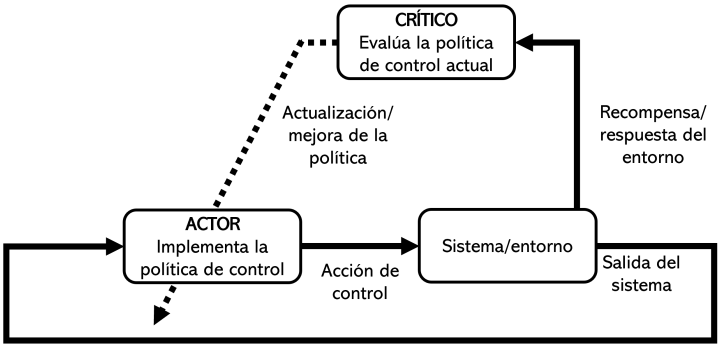


Figura 1.1: La configuración básica del problema para el control del aprendizaje.

Figura 1.2. 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 1.2 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.

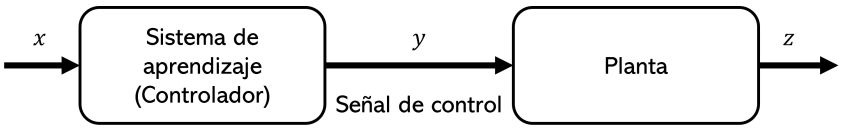


Figura 1.2: La configuración básica del problema para el control del aprendizaje.

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, además de 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 $F_W : X \rightarrow Y$, 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 [46]. Aprender el comportamiento de control apropiado implica determinar W para que la función de control resultante, F_W.

1.1. 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 [47,48]. 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.

1.1.1. 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 ([49, 50]) en los que un modelo del juego se utiliza para generar árboles de búsqueda. Muchos algoritmos de búsqueda heurística se han desarrollado en AI [50] 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 término 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 [51] y Barto [45]. Esto se ilustra en la Figura 1.3.

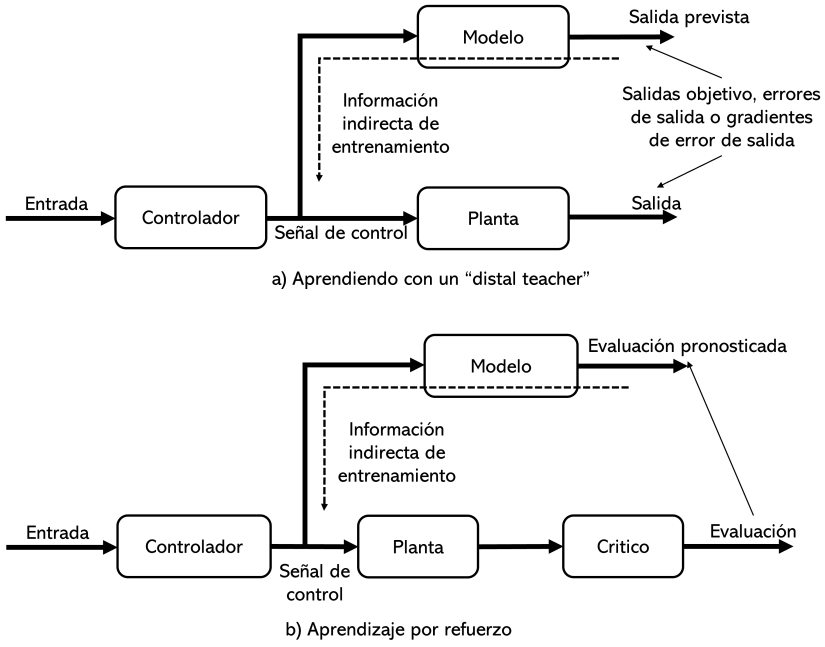


Figura 1.3: Un diagrama de bloques del método indirecto basado en gradientes para el control del aprendizaje.

1.1.2. 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 [52]. Tal método ha sido llamado "modelado inverso directo" [53] en la

literatura de aprendizaje conexionista y çoincidencia de entrada.[en] la literatura de control adaptativo [54]. 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 1.4. 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.

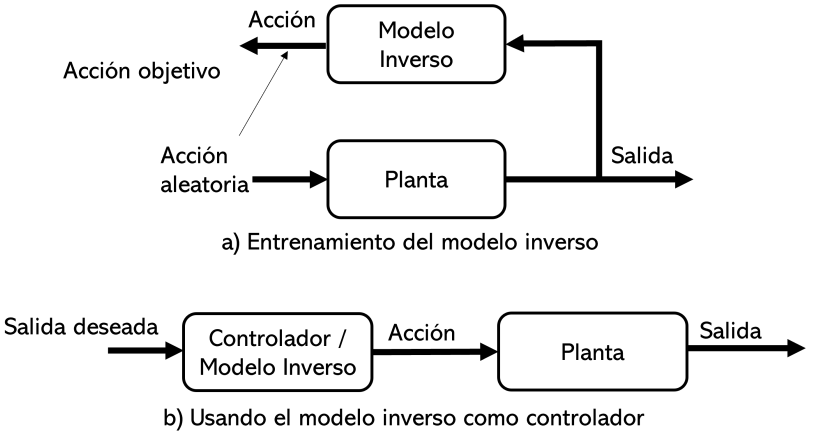


Figura 1.4: La parte (a) muestra la configuración utilizada para entrenar el modelo inverso, mientras que la parte (b) muestra cómo se utiliza el modelo inverso para controlar el proceso. [45]

1.2. Procesos de decisión de Markov

Considere el proceso de decisión de Markov (MDP, por sus siglas en inglés) (X, U, P, R), donde X es un conjunto de estados y U es un conjunto de acciones o controles (1.5). Las probabilidades de transición $P : X \times U \times X \to [0, 1]$ dan para cada estado $x \in X$ y acción $u \in U$ la probabilidad condicional $P^u_{x,x'} = \Pr\{x' \mid x, u\}$ de la transición al estado $x' \in X$ dado el MDP está en estado x y toma acción u. La función de costo $R : X \times U \times X \to \mathbb{R}$

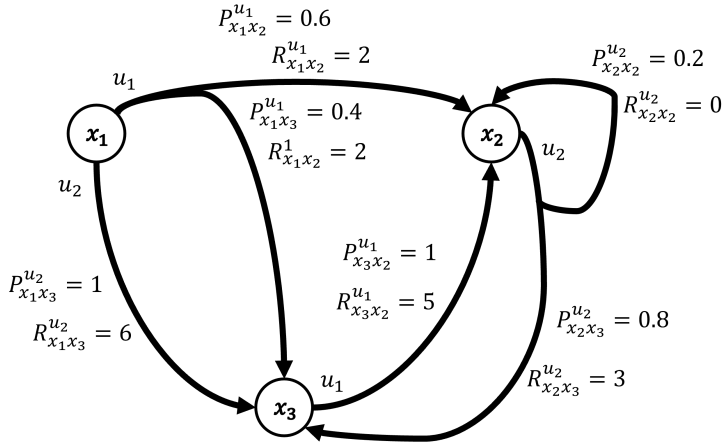


Figura 1.5: MDP mostrado como una máquina de estado finito con transiciones de estado controladas y costos asociados con cada transición

da el cost $R^u_{xx'}$ inmediato esperado después de la transición al estado $x' \in X$ dado que el MDP comienza en el estado $x \in X$ y toma la acción $u \in U$. La propiedad de Markov se refiere al hecho de que las probabilidades de transición $P^u_{x,x'}$ dependen únicamente del estado actual x y no del historial de cómo el MDP alcanzó ese estado.

El problema básico para MDP es encontrar un mapeo $\pi : X \times U \to [0, 1]$ que de para cada estado x y acción u la probabilidad condicional $\pi(x, u) = \Pr\{u \mid x\}$ de realizar la acción u dado que el MDP está en el estado x. Tal mapeo se denomina estrategia o política de control o acción de ciclo cerrado. La estrategia o política $\pi(x, u) = \Pr\{u \mid x\}$ se denomina estocástica o mixta si existe una probabilidad distinta de cero de seleccionar más de un control cuando se encuentra en el estado x. Podemos ver las estrategias mixtas como vectores de distribución de probabilidad que tienen como componente i la probabilidad de seleccionar la acción de control ith mientras se está en el estado $x \in X$. Si el mapeo $\pi : X \times U \to [0, 1]$ admite un solo control, con probabilidad 1, cuando en cada estado x, el mapeo se llama política determinista. Entonces, $\pi(x, u) = \Pr\{u \mid x\}$ corresponde a una función que mapea estados en controles $\mu(x) : X \to U$.

Los MDP que tienen estados finitos y espacios de acción se denominan MDP finitos.

1.2.1. Problemas de decisión secuencial óptima

Los sistemas dinámicos evolucionan causalmente a través del tiempo. Por lo tanto, consideramos problemas de decisión secuencial e imponemos un índice de etapa discreto k tal que el MDP toma una acción y cambia de estado en valores de etapa enteros no negativos k. Las etapas pueden corresponder al tiempo o, más generalmente, a secuencias de eventos. Nos referimos al valor de la etapa como el tiempo. Denote valores de estado y acciones en el tiempo k por x_k, u_k. MDP evoluciona en tiempo discreto.

A menudo es deseable que los sistemas de ingeniería humana sean óptimos en términos de conservación de recursos, tales como costo, tiempo, combustible y energía. Por lo tanto, la noción de optimización debe capturarse al seleccionar políticas de control para MDP. Defina, por lo tanto, un costo de etapa en el tiempo k por $r_k = r_k\left(x_k, u_k, x_{k+1}\right)$. Entonces $R^u_{xx'} = E\left\{r_k \mid x_k = x, u_k = u, x_{k+1} = x'\right\}$, con $E\{\cdot\}$ el operador de valor esperado. Se define un índice de rendimiento como la suma de los costos futuros en el intervalo de tiempo $[k, k+T]$

$$J_{k,T} = \sum_{i=0}^{T} \gamma^t r_{k+1} = \sum_{i=k}^{k+T} \gamma^{l-k} r_t$$

donde $0 \leq \gamma < 1$ es un factor de descuento que reduce el peso de los costos incurridos en el futuro.

El uso de MDP en los campos de la inteligencia computacional y la economía generalmente considera r_k como una recompensa incurrida en el momento k, también conocida como utilidad, y $J_{k,T}$ como un rendimiento descontado, también conocido como recompensa estratégica. En cambio, nos referimos a costos de etapa y costos futuros descontados para ser consistentes con los objetivos en el control de sistemas dinámicos. Por conveniencia podemos llamar a r_k la utilidad.

Considere que un agente selecciona una política de control $\pi_k\left(x_k, u_k\right)$ y la usa en cada etapa k del MDP. Estamos principalmente interesados en políticas estacionarias, donde las probabilidades condicionales $\pi_k\left(x_k, u_k\right)$ son independientes dé k. Entonces $\pi_k(x, u) = \pi(x, u) = \Pr\{u \mid x\}$, para todos los k. Las políticas deterministas no estacionarias tienen la

forma $\pi = \{\mu_0, \mu_1, \ldots\}$, donde cada entrada es una función $\mu_k(x) : X \to U; k = 0, 1, \ldots$. Las políticas deterministas estacionarias son independientes del tiempo, por lo que $\pi = \{\mu, \mu, \ldots\}$.

Seleccione una política estacionaria fija $\pi(x, u) = \Pr\{u \mid x\}$. Luego, el MDP de 'bucle cerrado' se reduce a una cadena de Markov con espacio de estado X. Es decir, las probabilidades de transición entre estados se fijan sin mayor libertad de elección de acciones. Las probabilidades de transición de esta cadena de Markov están dadas por

$$P_{x,r'} \equiv P^{\pi}_{x,r'} = \sum_{u} \Pr\{x' \mid x, u\} \Pr\{u \mid x\} = \sum_{u} \pi(x, u) P^{u}_{x,r'} \tag{1.1}$$

donde se utiliza la identidad de Chapman-Kolmogorov [55].

Bajo el supuesto de que la cadena de Markov corresponde a cada política, con probabilidades de transición dadas en (1.1), es ergódica, se puede demostrar que cada MDP tiene una política óptima determinista estacionaria [56, 57]. Una cadena de Markov es ergódica si todos los estados son recurrentes positivos y aperiódicos. Entonces, para una política dada existe una distribución estacionaria $P_{\pi}(x)$ sobre X que da la probabilidad de estado estacionario de que la cadena de Markov esté en el estado x.

El valor de una política se define como el valor esperado condicional del costo futuro cuando comienza en el estado x en el tiempo k y sigue la política $\pi(x, u)$ a partir de entonces

$$V^{\pi}_{k}(x) = E_{\pi}\{J_{k,T} \mid x_k = x\} = E_{\pi}\left\{\sum_{i=k}^{k+T} \gamma^{i-k} r_i \mid x_k = x\right\}$$

Aquí, $E_{\pi}\{\cdot\}$ es el valor esperado dado que el agente sigue la política $\pi(x, u)$.$V^{\pi}(x)$ se conoce como la función de valor para la política $\pi(x, u)$. Indica el valor de estar en el estado x dado que la política es $\pi(x, u)$.

Un objetivo principal de MDP es determinar una política $\pi(x, u)$ para minimizar el costo futuro esperado

$$\pi^*(x, u) = \arg\min_{\pi} V^{\pi}_{k}(x) = \arg\min_{\pi} E_{\pi}\left\{\sum_{i=k}^{k+T} \gamma^{i-k} r_i \mid x_k = x\right\}$$

Esta política se denomina la política óptima, y el valor óptimo correspondiente se da como

$$V_k^*(x) = \min_\pi V_k^\pi(x) = \min_\pi E_\pi \left\{ \sum_{i=k}^{k+T} \gamma^{i-k} r_i \mid x_k = x \right\}$$

Recursión hacia atrás

Usando la identidad de Chapman-Kolmogorov [58] y la propiedad de Markov podemos escribir el valor de la política $\pi(x, u)$ como

$$V_k^\pi(x) = E_\pi \left\{ J_k \mid x_k = x \right\} = E_\pi \left\{ \sum_{i=k}^{k+T} \gamma^{i-k} r_i \mid x_k = x \right\} \tag{1.2}$$

$$V_k^\pi(x) = E_\pi \left\{ r_k + \gamma \sum_{i=k+1}^{k+T} \gamma^{i-(k+1)} r_i \mid x_k = x \right\} \tag{1.3}$$

$$V_k^\pi(x) = \sum_u \pi(x, u) \sum_{x'} P_{xx'}^u \left[R_{xx'}^u + \gamma E_\pi \left\{ \sum_{i=k+1}^{k+T} \gamma^{i-(k+1)} r_i \mid x_{k+1} = x' \right\} \right] \tag{1.4}$$

Por lo tanto, la función de valor para la política $\pi(x, u)$ satisface

$$V_k^\pi(x) = \sum_u \pi(x, u) \sum_{x'} P_{xx'}^u \left[R_{xx'}^u + \gamma V_{k+1}^\pi(x') \right] \tag{1.5}$$

Esta ecuación proporciona una recursión hacia atrás para el valor en el tiempo k en términos del valor en el tiempo $k + 1$

1.2.2. Programación dinámica

El costo óptimo se puede escribir como

$$V_k^*(x) = \min_\pi V_k^\pi(x) = \min_\pi \sum_u \pi(x, u) \sum_{x'} P_{xx'}^u \left[R_{xx'}^u + \gamma V_{k+1}^\pi(x') \right] \tag{1.6}$$

El principio de optimalidad de Bellman ([59]) establece que “Una política óptima tiene la propiedad de que sin importar cuáles hayan sido las acciones de control previas, los

controles restantes constituyen una política óptima con respecto al estado resultante de esos controles previos". Por lo tanto, podemos escribir

$$V_k^*(x) = \min_{\pi} \sum_u \pi(x,u) \sum_{x'} P_{xx'}^u \left[R_{xx'}^u + \gamma V_{k+1}^* (x') \right] \tag{1.7}$$

Supongamos que ahora se aplica un control arbitrario u en el momento k y la política óptima se aplica a partir del momento $k+1$. Entonces el principio de optimalidad de Bellman dice que el control óptimo en el momento k está dado por

$$\pi^* (x_k = x, u) = \arg\min_{\pi} \sum_u \pi(x,u) \sum_{x'} P_{xx'}^u \left[R_{xx'}^u + \gamma V_{k+1}^* (x') \right] \tag{1.8}$$

Bajo el supuesto de que la cadena de Markov correspondiente a cada política, con las probabilidades de transición dadas en (1.1), es ergódica, cada MDP tiene una política óptima determinista estacionaria. Entonces podemos minimizar de manera equivalente a la expectativa condicional sobre todas las acciones u en el estado x. Por lo tanto,

$$V_k^*(x) = \min_{u} \sum_{x'} P_{xx'}^u \left[R_{xx\ primo}^u + \gamma V_{k+1}^* (x') \right] \tag{1.9}$$

$$u_k^* = \arg\min_{u} \sum_{x'} P_{xx'}^u \left[R_{xx'}^u + \gamma V_{k+1}^* (x') \right] \tag{1.10}$$

La recursividad hacia atrás (1.7), (1.9) forma la base de la programación dinámica (DP, por sus siglas en inglés) ([59]), que brinda métodos fuera de línea para trabajar hacia atrás en el tiempo para determinar las políticas óptimas ([60]). DP es un procedimiento fuera de línea para encontrar el valor óptimo y las políticas óptimas que requiere el conocimiento de la dinámica completa del sistema en forma de probabilidades de transición $P_{x,x'}^u = \Pr\{x' \mid x, u\}$y costos esperados$R_{xx'}^u = E\{r_k \mid x_k = x, u_k = u, x_{k+1} = x'\}$.

1.2.3. 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, que ahora desarrollamos.

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

$$J_k = \sum_{l=0}^{\infty} \gamma^l r_{k+1} = \sum_{i=k}^{\infty} \gamma^{l-k} r_i \tag{1.11}$$

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

$$V^{\pi}(x) = E_{\pi}\left\{J_k \mid x_k = x\right\} = E_{\pi}\left\{\sum_{i=k}^{\infty} \gamma^{i-k} r_t \mid x_k = x\right\} \tag{1.12}$$

Usando (1.5) con $T = \infty$ se puede ver que la función de valor para la política $\pi(x, u)$ satisface la ecuación de Bellman

$$V^{\pi}(x) = \sum_{u} \pi(x, u) \sum_{x'} P^{u}_{xx'} \left[R^{u}_{xx'} + \gamma V^{\pi}\left(x'\right)\right] \tag{1.13}$$

La importancia de esta ecuación es que en ambos lados aparece la misma función de valor, lo cual se debe a que se utiliza el costo de horizonte infinito. Por lo tanto, la ecuación de Bellman (1.13) puede interpretarse como una ecuación de consistencia que debe ser satisfecha por la función de valor en cada etapa de tiempo. Expresa una relación entre el valor actual de estar en el estado x y el valor de estar en el siguiente estado x' dado que se usa la política $\pi(x, u)$.

La ecuación de Bellman (1.13) es el punto de partida para desarrollar una familia de algoritmos de aprendizaje por refuerzo para encontrar políticas óptimas mediante el uso de experiencias causales recibidas paso a paso en el tiempo. La ecuación de optimización de Bellman 1.7 implica el operador 'mínimo', por lo que no contiene ninguna política

específica $\pi(x, u)$. Su solución se basa en conocer la dinámica, en forma de probabilidades de transición. Por el contrario, la forma de la ecuación de Bellman es más simple que la de la ecuación de optimización y es más fácil de resolver. La solución a la ecuación de Bellman produce la función de valor de una política específica $\pi(x, u)$. Como tal, la ecuación de Bellman se adapta bien al método actor-crítico de aprendizaje por refuerzo que se muestra en la figura 1.1. Posteriormente, se muestra que la ecuación de Bellman proporciona métodos para implementar la crítica de la figura 1.1, que se encarga de evaluar el desempeño de la política actual específica. Quedan por poner en marcha dos ingredientes clave. En primer lugar, se muestra que los métodos conocidos como iteración de políticas e iteración de valores utilizan la ecuación de Bellman para resolver problemas de control óptimo en el tiempo. En segundo lugar, al aproximar la función de valor en (1.13) mediante una estructura paramétrica, estos métodos pueden implementarse en línea utilizando algoritmos de identificación de sistemas de control adaptativo estándar, como los mínimos cuadrados recursivos.

En el contexto del uso de la ecuación de Bellman (1.13) para el aprendizaje por refuerzo, $V^{\pi}(x)$ puede considerarse como un rendimiento previsto, $\sum_u \pi(x, u) \sum_{x'} P^u_{x'} R^u_{xx'}$ la recompensa de un paso observada y $V^{\pi}(x')$ una estimación actual del comportamiento futuro. Tales nociones pueden capitalizarse en la discusión posterior sobre el aprendizaje de diferencias temporales, que las utiliza para desarrollar algoritmos de control adaptativo que pueden aprender un comportamiento óptimo en línea en aplicaciones en tiempo real.

Si el MDP es finito y tiene N estados, entonces la ecuación de Bellman (1.13) es un sistema de N ecuaciones lineales simultáneas para el valor $V^{\pi}(x)$ de estar en cada estado x dada la política actual $\pi(x, u)$. El valor óptimo de horizonte infinito satisface

$$V^*(x) = \min_{\pi} V^{\pi}(x) = \min_{\pi} \sum_u \pi(x, u) \sum_{x'} P^u_{xx'} \left[R^u_{xx'} + \gamma V^{\pi}(x') \right] \tag{1.14}$$

El principio de optimización de Bellman produce la ecuación de optimización de Bellman

$$V^*(x) = \min_{\pi} V^{\pi}(x) = \min_{\pi} \sum_u \pi(x, u) \sum_{x'} P^u_{xx'} \left[R^u_{xx'} + \gamma V^*(x') \right] \tag{1.15}$$

De manera equivalente, bajo el supuesto de ergodicidad en las cadenas de Markov correspondientes a cada política, la ecuación de optimización de Bellman se puede escribir como

$$V^*(x) = \min_u \sum_{x'} P^u_{xx'} \left[R^u_{xx'} + \gamma V^* \left(x' \right) \right] \tag{1.16}$$

Esta ecuación se conoce como ecuación de Hamilton-Jacobi-Bellman (HJB) en sistemas de control. Si el MDP es finito y tiene N estados, entonces la ecuación de optimización de Bellman es un sistema de N ecuaciones no lineales para el valor óptimo $V^*(x)$ de estar en cada estado. El control óptimo está dado por

$$u^* = \arg\min_u \sum_{x'} P^u_{x'} \left[R^u_{xx'} + \gamma V^* \left(x' \right) \right] \tag{1.17}$$

1.3. Propiedad de Markov

Idealmente, en un problema de aprendizaje por refuerzo se prefiere una señal de estado que resuma las sensaciones pasadas de manera compacta, pero de tal manera que se retenga toda la información relevante. Tal señal que logra retener toda la información relevante se dice que es Markov, o se dice que tiene la propiedad de Markov.

Considere cómo un entorno general podría responder en el momento $t + l$ a la acción realizada en el momento t. En el caso causal más general, esta respuesta puede depender de todo lo que ha sucedido antes. En este caso, la dinámica se puede definir solo especificando la distribución de probabilidad conjunta completa:

$$\Pr \left\{ S_{t+1} = s', R_{t+1} = r \mid S_0, A_0, R_1, \ldots, S_{t-1}, A_{t-1}, R_t, S_t, A_t \right\}$$

para todos los r, s' y todos los valores posibles de los eventos pasados: $S_0, A_0, R_1, \ldots, S_{t-1}, A_{t-1}, R_t, S_t, A_t$. Se dice que una señal de estado posee la propiedad de Markov si el siguiente estado y la recompensa obtenida a través de la dinámica del entorno recibida por el agente en el momento $t + 1$, dependen únicamente del estado del sistema y de la acción realizada por el agente RL en el momento tiempo t. El agente

puede mantener una estimación de la dinámica del entorno inherente a través de las Probabilidades de Transición de Estado, que se pueden definir mediante:

$$p\left(s', r \mid s, a\right) = \Pr\left\{S_{t+1} = s', R_{t+1} = r \mid S_t = s, A_t = a\right\} \tag{1.18}$$

Para todo r, s', s y a.

Si un entorno tiene la propiedad de Markov, entonces su dinámica de un paso (1.18) nos permite predecir el siguiente estado y la siguiente recompensa esperada dado el estado y la acción actuales. Se puede demostrar que, iterando esta ecuación, se pueden predecir todos los estados futuros y las recompensas esperadas a partir del conocimiento únicamente del estado actual, así como sería posible dada la historia completa hasta el momento actual. También se deduce que los estados de Markov proporcionan la mejor base posible para elegir acciones. Es decir, la mejor política para elegir acciones en función de un estado de Markov es tan buena como la mejor política para elegir acciones en función de historias completas.

1.4. Evaluación de políticas y mejora de políticas

Dada una política actual $\pi(x, u)$, su valor (1.12) se puede determinar resolviendo la ecuación de Bellman (1.13). Este procedimiento se conoce como evaluación de políticas.

Además, dado el valor de alguna política $\pi(x, u)$, siempre podemos usarlo para encontrar otra política que sea mejor, o al menos no peor. Este paso se conoce como mejora de políticas. Específicamente, suponga que $V^{\pi}(x)$ satisface (1.13). Luego defina una nueva política $\pi'(x, u)$ por

$$\pi'(x, u) = \arg\min_{\pi(x,)} \sum_{x'} P^{u}_{xx'} \left[R^{u}_{xx'} + \gamma V^{\pi}\left(x'\right)\right] \tag{1.19}$$

Entonces se puede demostrar que $V^{\pi'}(x) \leq V^{\pi}(x)$ [45, 56]. Se dice que la política determinada como en (2.33) es codiciosa con respecto a la función de valor $V^{\pi}(x)$.

En el caso especial de que $V^{\pi'}(x) = V^{\pi}(x)$ en (2.33), entonces $V^{\pi'}(x), \pi'(x,u)$ satisface (1.16) y (1.17); por lo tanto, $\pi'(x,u) = \pi(x,u)$ es la política óptima y $V^{\pi'}(x) = V^{\pi}(x)$ el valor óptimo. Es decir, una política óptima, y solo una política óptima, es codiciosa con respecto a su propio valor. En inteligencia computacional, codicioso se refiere a cantidades determinadas mediante la optimización en horizontes cortos o de un solo paso, independientemente de los impactos potenciales en el futuro lejano.

Ahora consideremos algoritmos que intercalan repetidamente los siguientes dos procedimientos.

Evaluación de políticas mediante la ecuación de Bellman

$$V^{\pi}(x) = \sum_{u} \pi(x,u) \sum_{x'} P^{u}_{xx'} \left[R^{u}_{xx'} + \gamma V^{\pi}(x') \right], \quad \text{para todos } x \in S \subseteq X \tag{1.20}$$

Mejora de políticas

$$\pi'(x,u) = \arg\min_{\pi(x,)} \sum_{x'} P^{u}_{xx'} \left[R^{u}_{xx'} + \gamma V^{\pi}(x') \right], \quad \text{para todos } x \in S \subseteq X \tag{1.21}$$

donde S es un subespacio adecuadamente seleccionado del espacio de estado, que se discutirá más adelante. Llamamos a una aplicación de (1.20) seguida de una aplicación de (1.21) un paso. Esta terminología contrasta con la etapa de tiempo de decisión k definida anteriormente.

En cada paso de dichos algoritmos, obtenemos una política que no es peor que la política anterior. Por lo tanto, no es difícil probar la convergencia en condiciones bastante moderadas al valor óptimo y la política óptima. La mayoría de estas demostraciones se basan en el teorema del punto fijo de Banach. Note que (1.16) es una ecuación de punto fijo para $V^{*}(\cdot)$. Luego, las dos ecuaciones (1.20) y (1.21) definen un mapa asociado que puede mostrarse en condiciones moderadas como un mapa de contracción [56, 61, 62], que converge a la solución dé (2.20).

El resultado es una familia de algoritmos de control adaptativo que convergen en soluciones de control óptimas. Dichos algoritmos son de la clase actor-crítico de los sistemas de aprendizaje por refuerzo, que se muestran en la figura 1.1. Allí, un agente

crítico evalúa la política de control actual utilizando métodos basados en 1.20. Una vez completada esta evaluación, la acción es actualizada por un agente actor basado en 1.21.

1.4.1. Iteración de políticas

Un método de aprendizaje por refuerzo para usar (1.20) y (1.21) para encontrar el valor óptimo y la política óptima es la iteración de políticas.

Algoritmo 1 Iteración de políticas

Seleccione una política inicial admisible $\pi_0(x, u)$

repeat $j = 0$:

Evaluación de políticas (actualización de valor)

$$V_j(x) = \sum_n \pi_j(x, u) \sum_{x'} P^u_{xx'} \left[R^u_{xx'} + \gamma V_j\left(x'\right)\right], \quad \text{for all } x \in X \tag{1.22}$$

Mejora de políticas (actualización de políticas)

$$\pi_{j+1}(x, u) = \arg\min_{\pi(x,\cdot)} \sum_{x'} P^u_{xx'} \left[R^u_{xx'} + \gamma V_j\left(x'\right)\right], \quad \text{for all } x \in X \tag{1.23}$$

until hasta la convergencia

En cada paso j, el algoritmo de iteración de políticas determina la solución de la ecuación de Bellman (1.22) para calcular el valor $V_j(x)$ de usar la política actual $\pi_j(x, u)$. Este valor corresponde a la suma infinita (1.12) para la póliza actual. Entonces la política se mejora usando (1.23). Los pasos se continúan hasta que no hay cambio en el valor o la política.

Teniendo en cuenta que j no es el índice de tiempo o etapa k, sino un índice de iteración de paso de iteración de política. La iteración de políticas se puede implementar para sistemas dinámicos en línea en tiempo real mediante la observación de datos medidos a lo largo de las trayectorias del sistema. Se necesitan datos para varias veces k para resolver la ecuación de Bellman (1.22) en cada paso j

Si el MDP es finito y tiene N estados, entonces la ecuación de evaluación de políticas (1.22) es un sistema de N ecuaciones lineales simultáneas, una para cada estado. El algoritmo de iteración de políticas debe inicializarse adecuadamente para converger. Se debe seleccionar la política inicial $\pi_0(x, u)$ y el valor V_0 para qué $V_1 \leq V_0$. Las pólizas iniciales que garantizan esto se denominan admisibles. Entonces, para cadenas de Markov finitas con N estados, la iteración de políticas converge en un número finito de pasos, menor o igual a N, porque solo hay un número finito de políticas [56].

Iteración de política iterativa

La ecuación de Bellman (1.22) es un sistema de ecuaciones simultáneas. En lugar de resolver directamente la ecuación de Bellman, se puede resolver mediante un procedimiento iterativo de evaluación de políticas. Nótese que (1.22) es una ecuación de punto fijo para $V_j(\cdot)$. Define el mapa iterativo de evaluación de políticas.

$$V_j^{i+1}(x) = \sum_u \pi_j(x, u) \sum_{x'} P_{xx'}^u \left[R_{xx'}^u + \gamma V_j^i(x') \right], \quad i = 1, 2, \ldots \tag{1.24}$$

que se puede demostrar que es un mapa de contracción en condiciones bastante suaves. Por el teorema del punto fijo de Banach, la iteración se puede inicializar en cualquier valor no negativo de $V_j^1(\cdot)$ y la iteración converge a la solución de (1.22). Bajo ciertas condiciones, esta solución es única. Una elección de valor inicial adecuada es la función de valor $V_{j-1}(\cdot)$ del paso anterior $j-1$. En una convergencia lo suficientemente cercana, establecemos $V_j(\cdot) = V_j^i(\cdot)$ y procedemos a aplicar (1.23).

El índice j en (1.24) se refiere al número de pasos del algoritmo de iteración de políticas. Por el contrario, i es un índice de iteración. La evaluación iterativa de políticas (1.24) debe compararse con la recurrencia hacia atrás en el tiempo (1.5) para el valor de horizonte finito. En (1.5), k es el índice de tiempo. Por el contrario, en (1.24), i es un índice de iteración. La programación dinámica se basa en (1.5) y procede hacia atrás en el tiempo.

1.4.2. Iteración de valor

Un segundo método para utilizar (1.21) y (1.20) en el aprendizaje por refuerzo es la iteración de valor.

Algoritmo 2 Iteración de políticas

Seleccione una política inicial admisible $\pi_0(x, u)$

repeat$j = 0$:

Actualización de valor

$$V_{j+1}(x) = \sum_u \pi_j(x, u) \sum_{x'} P^u_{xx'} \left[R^u_{xx'} + \gamma V_j\left(x'\right)\right], \quad \text{para todo } x \in S_j \subseteq X \tag{1.25}$$

Mejora de políticas

$$\pi_{j+1}(x, u) = \arg\min_{\pi(x,-)} \sum_{x'} P^u_{xx'} \left[R^u_{xx'} + \gamma V_j\left(x'\right)\right], \quad \text{para todo } x \in S_j \subseteq X \tag{1.26}$$

until hasta la convergencia

Podemos combinar la actualización del valor y la mejora de la política en una ecuación para obtener la forma equivalente para la iteración del valor

$$V_{j+1}(x) = \min_\pi \sum_u \pi(x, u) \sum_{x'} P^u_{xx'} \left[R^u_{xx'} + \gamma V_j\left(x'\right)\right], \quad \text{para todo } x \in S_j \subseteq X \tag{1.27}$$

o, de manera equivalente, bajo el supuesto de ergodicidad, en términos de políticas deterministas

$$V_{j+1}(x) = \min_u \sum_{x'} P^u_{xx'} \left[R^u_{xx'} + \gamma V_j\left(x'\right)\right], \quad \text{para todo } x \in S_j \subseteq X \tag{1.28}$$

Teniendo en cuenta que (1.25) es una simple recursión de un paso, no un sistema de ecuaciones lineales como lo es (1.22) en el algoritmo de iteración de políticas. De hecho,

la iteración de valor usa una iteración de (1.24) en su paso de actualización de valor. No encuentra el valor correspondiente a la política actual, pero toma solo una iteración hacia ese valor. Nuevamente, j no es el índice de tiempo, sino el índice de paso de iteración de valor.

Comparando la iteración de valores (1.27) con la programación dinámica (1.7). DP es un procedimiento hacia atrás en el tiempo para encontrar políticas de control óptimas y, como tal, no se puede implementar en línea en tiempo real. Por el contrario, en secciones posteriores, mostramos cómo implementar la iteración de valor para sistemas dinámicos en línea en tiempo real mediante la observación de datos medidos a lo largo de las trayectorias del sistema. Se necesitan datos de múltiples veces k para resolver la actualización (1.25) para cada paso j.

La iteración de valor estándar toma el conjunto de actualizaciones como $S_j = X$, para todos los j. Es decir, el valor y la política se actualizan para todos los estados simultáneamente. Los métodos de iteración de valores asincrónicos realizan las actualizaciones solo en un subconjunto de los estados en cada paso. En el caso extremo, las actualizaciones se pueden realizar en un solo estado en cada paso.

1.4.3. Función Q

El valor esperado condicional en (1.9)

$$Q_k^*(x,u) = \sum_{x'} P_{xx'}^u \left[R_{xx'}^u + \gamma V_{k+1}^*(x')\right] = E_\pi \left\{r_k + \gamma V_{k+1}^*(x') \mid x_k = x, u_k = u\right\} \quad (1.29)$$

se conoce como la función Q óptima [63,64]. El nombre proviene de 'función de calidad'. La función Q también se denomina función de acción-valor [6]. La función Q es igual al rendimiento esperado por realizar una acción arbitraria u en el momento k en el estado x y luego seguir una política óptima. La función Q es una función del estado actual x y también de la acción u. En términos de la función Q, la ecuación de optimización de Bellman tiene la forma particularmente simple

$$V_k^*(x) = \min_u Q_k^*(x,u) \tag{1.30}$$

$$u_k^* = \arg\min_u Q_k^*(x,u) \tag{1.31}$$

Dada alguna política fija $\pi(x,u)$ define la función Q para esa política como

$$Q_k^\pi(x,u) = E_\pi\left\{r_k + \gamma V_{k+1}^\pi(x') \mid x_k = x, u_k = u\right\} = \sum_{x'} P_{xx'}^u\left[R_{xx'}^u + \gamma V_{k+1}^\pi(x')\right] \tag{1.32}$$

donde (1.5) se usa. Esta función es igual al rendimiento esperado por realizar una acción arbitraria u en el momento k en el estado x y luego seguir la política existente $\pi(x,u)$.

Teniendo en cuenta que $V_k^\pi(x) = Q_k^\pi(x,\pi(x,u))$, por lo tanto (1.32) se puede escribir como la recursión hacia atrás en la función Q

$$Q_k^\pi(x,u) = \sum_{x'} P_{xx'}^u\left[R_{xx'}^u + \gamma Q_{k+1}^\pi(x',\pi(x',u'))\right]. \tag{1.33}$$

La función Q es una función bidimensional (2D) tanto del estado actual x como de la acción u. Por el contrario, la función de valor es una función unidimensional del estado. Para MDP finito, la función Q se puede almacenar como una tabla de búsqueda 2D en cada par de estado/acción. Tenga en cuenta que la minimización directa en (1.8) y (1.9) requiere el conocimiento de las probabilidades de transición de estado, que corresponden a la dinámica del sistema y los costos. Por el contrario, la minimización en (1.30) y (1.31) requiere conocer solo la función Q y no la dinámica del sistema.

1.5. Q-Learning

La programación dinámica es un enfoque basado en modelos para resolver problemas de aprendizaje por refuerzo [65]. 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:

$$\begin{aligned}\mathcal{Q}(x_t, u_t) = E\big[r_t(x_t, u_t, \ X_{t+1}) \\ +\gamma r_{t+1}\left(X_{t+1}, \arg\max_{u_{t+1}} \mathcal{Q}(X_{t+1}, u_{t+1}), X_{t+2}\right) \\ +\gamma^2 r_{t+2}\left(X_{t+2}, \arg\max_{u_{t+2}} \mathcal{Q}(X_{t+2}, u_{t+2}), X_{t+3}\right) + \ldots\big]\end{aligned} \tag{1.34}$$

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:

$$\mathcal{Q}^*(x_t, u_t) = E\left[r_t(x_t, u_t, X_{t+1}) + \gamma \max_{u_{t+1}} \mathcal{Q}^*(X_{t+1}, u_{t+1})\right] \forall x_t, u_t \tag{1.35}$$

Se garantiza que ejecutar $\arg_{\max_u \mathcal{Q}^*}(x,u)\ldots$ 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 $\mathcal{Q}^*$ [64]. 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:

$$\mathcal{Q}(x_t, u_t) \xleftarrow{\alpha} r(x_t, u_t, x_{t+1}) + \gamma \max_{u_{t+1}} \mathcal{Q}(x_{t+1}, u_{t+1}) \tag{1.36}$$

donde α es una tasa de aprendizaje (o tamaño de paso) entre 0 y 1 que controla la convergencia. La flecha en la Ecuación 1.36 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.

$$\begin{gathered} A \xleftarrow{\alpha} B, \text{ is equivalent to,} \\ A := (1-\alpha)A + \alpha B, \quad \alpha \in [0,1] \end{gathered} \tag{1.37}$$

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 [63]:

- 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.

1.6. Aprendizaje de diferencias temporales

Los métodos de diferencia temporal pueden considerarse técnicas de aproximación estocástica en las que la ecuación de Bellman (1.13), o sus variantes (1.22) se reemplazan por su evaluación a lo largo de una ruta de muestra única del MDP. Entonces, la ecuación de Bellman se convierte en una ecuación determinista que permite la definición de un error de diferencia temporal.

La ecuación (1.5) se usó para escribir la ecuación de Bellman (1.13) para el valor del horizonte infinito (1.12). De acuerdo con (1.3)-(1.5), una forma alternativa para la ecuación de Bellman es

$$V^{\pi}(x_k) = E_{\pi}\{r_k \mid x_k\} + \gamma E_{\pi}\{V^{\pi}(x_{k+1}) \mid x_k\} \tag{1.38}$$

Esta ecuación forma la base para el aprendizaje de diferencias temporales.

El aprendizaje por refuerzo de diferencias temporales utiliza una ruta de muestra, a saber, la trayectoria del sistema actual, para actualizar el valor. Entonces, (1.38) se reemplaza por la ecuación determinista de Bellman

$$V^{\pi}(x_k) = r_k + \gamma V^{\pi}(x_{k+1}) \tag{1.39}$$

lo cual es válido para cada conjunto de experiencia de datos observados (x_k, x_{k+1}, r_k) en cada etapa de tiempo k, este conjunto consiste en el estado actual x_k, el costo observado incurrido r_k y el siguiente estado x_{k+1}. El error de diferencia temporal se define como

$$e_k = -V^{\pi}(x_k) + r_k + \gamma V^{\pi}(x_{k+1}) \tag{1.40}$$

y la estimación actual para la función de valor se actualiza para que el error de diferencia temporal sea pequeño. En el contexto del aprendizaje de diferencias temporales, la interpretación de la ecuación de Bellman se muestra en la Figura 1.6, donde $V^{\pi}(x_k)$ puede considerarse como un rendimiento o valor predicho, r_k como la recompensa de un paso observada y $\gamma V^{\pi}(x_{k+1})$ como una estimación actual del valor futuro. La ecuación de Bellman se puede interpretar como una ecuación de coherencia que se cumple si la estimación actual del valor predicho $V^{\pi}(x_k)$ es correcta. Los métodos de diferencia temporal actualizan la estimación del valor predicho $\hat{V}^{\pi}(x_k)$ para reducir el error de diferencia temporal.

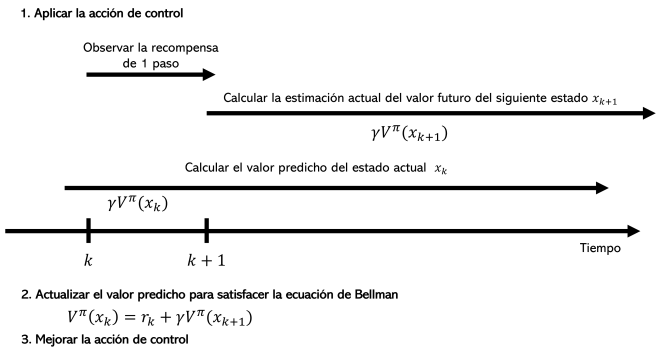


Figura 1.6: Interpretación de la diferencia temporal de la ecuación de Bellman.

Capítulo 2

Aprendizaje por refuerzo para el diseño de control de lazo interno

Este capítulo da una formulación del algoritmo *aprendizaje por refuerzo integral* (IRL, por sus siglas en inglés) para sistemas no lineales de tiempo continuo, que no depende de la dinámica de deriva del sistema $f(x)$. El algoritmo se presentó por primera vez en [66–69]. La estructura del algoritmo IRL permite el desarrollo de un algoritmo IRL adaptativo en línea que converge a la solución del problema de control óptimo en tiempo real midiendo datos a lo largo de las trayectorias del sistema. No se requiere conocimiento de la dinámica de deriva del sistema $f(x)$. A esto lo llamamos un algoritmo de control óptimo adaptativo, ya que es un controlador adaptativo que converge a la solución de control óptima. Este controlador IRL adaptativo tiene una estructura actor-crítico y es una combinación híbrida entre un controlador de tiempo continuo y una estructura de aprendizaje que opera con base en datos muestreados discretos del sistema. El controlador de tiempo continuo es dinámico y tiene una memoria cuyo estado es el valor o costo.

2.1. Control adaptativo óptimo usando integral aprendizaje por refuerzo para sistemas lineales

Las aplicaciones de RL en control de retroalimentación para sistemas dinámicos de tiempo continuo $\dot{x} = f(x) + g(x)u$ se han retrasado. Esto se debe al hecho de que la ecuación de Bellman para sistemas DT $V(x_k) = r(x_k, u_k) + \gamma V(x_{k+1})$ no depende de la dinámica del sistema, mientras que la ecuación de Bellman $0 = r(x, u) + (\nabla V)^T (f(x) + g(x)u)$ para sistemas CT depende de la dinámica del sistema $f(x), g(x)$.

La técnica de iteración de la política de aprendizaje por refuerzo integral resuelve el problema del regulador cuadrático lineal para sistemas de tiempo continuo en línea en tiempo real, utilizando solo un conocimiento parcial sobre la dinámica del sistema (es decir, no es necesario conocer la dinámica de deriva A del sistema). Este es en efecto un esquema de control adaptativo directo (es decir, no se emplea ningún procedimiento de identificación del sistema) para sistemas lineales parcialmente desconocidos que converge a la solución de control óptima.

2.1.1. Solución crítica adaptativa de tiempo continuo

Se mostrará el enfoque de aprendizaje por refuerzo integral (IRL) para resolver en línea el problema del regulador cuadrático lineal (LQR) [69] sin utilizar el conocimiento de la matriz del sistema A. IRL resuelve la ecuación algebraica de Riccati en línea en tiempo real, sin conocer la matriz del sistema A, midiendo datos $(x(t), u(t))$ a lo largo de las trayectorias del sistema.

Consideramos el sistema dinámico lineal invariante en el tiempo descrito

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{2.1}$$

con estado $x(t) \in \mathbb{R}^n$, entrada de control $u(t) \in \mathbb{R}^m$ y (A, B) estabilizable. A este sistema se asocia la función de coste cuadrática de horizonte infinito

$$V\left(x\left(t_0\right), t_0\right) = \int_{t_0}^{\infty} \left(x^T(\tau)Qx(\tau) + u^T(\tau)Ru(\tau)\right) d\tau \tag{2.2}$$

con $Q \geq 0, R > 0$ tal que $\left(Q^{1/2}, A\right)$ es detectable. El problema de control óptimo LQR requiere encontrar la política de control que minimice el costo

$$u^*(t) = \arg \min_{\substack{u(t) \\ u \leq i \leq s > 0}} V\left(t_0, x\left(t_0\right), u(t)\right) \tag{2.3}$$

La solución de este problema de control óptimo, determinada por el principio de optimización de Bellman, viene dada por la retroalimentación de estado $u(t) = -Kx(t)$ dada por

$$K = R^{-1}B^T P \tag{2.4}$$

donde la matriz P es la única solución definida positiva de la ecuación algebraica de Riccati (ARE)

$$A^T P + PA - PBR^{-1}B^T P + Q = 0 \tag{2.5}$$

Bajo la condición de detectabilidad para $\left(Q^{1/2}, A\right)$, la solución semidefinida positiva única del ARE determina un controlador de lazo cerrado estabilizador dado por (2.4).

Se sabe que la solución del problema de optimización de horizonte infinito se puede obtener utilizando el método de programación dinámica. La siguiente ecuación diferencial de Riccati debe ser resuelta

$$\begin{aligned} -\dot{P} &= A^T P + PA - PBR^{-1}B^T P + Q \\ P\left(t_f\right) &= P_{tf} \end{aligned} \tag{2.6}$$

Su solución convergerá a la solución del ARE como $t_f \to \infty$

Algoritmo de iteración de políticas usando refuerzo integral

El algoritmo se basa en una estructura actor-crítico y consta de una iteración de dos pasos, a saber, la actualización del crítico y la actualización del actor. La actualización de

la estructura crítica da como resultado el cálculo del costo de horizonte infinito asociado con un controlador estabilizador dado. Los parámetros del actor (es decir, la matriz de ganancia de retroalimentación del controlador K) se actualizan luego en el sentido de reducir el costo en comparación con la política de control actual.

Sea K una ganancia de retroalimentación de estado estabilizadora para (2.1) tal que $\dot{x} = (A-BK)x$ es un sistema estable en lazo cerrado. Entonces, el costo o valor cuadrático de horizonte infinito correspondiente está dado por

$$V(x(t)) = \int_t^\infty x^T(\tau)\left(Q + K^T R K\right) x(\tau) d\tau = x^T(t) P x(t) \tag{2.7}$$

donde P es la solución definitiva positiva simétrica real de la ecuación matricial de Lyapunov

$$(A - BK)^T P + P(A - BK) = -\left(K^T R K + Q\right) \tag{2.8}$$

Entonces, $V(x(t))$ sirve como función de Lyapunov para (2.1) con ganancia de controlador K. La función de valor (2.25) se puede escribir de la siguiente forma.

Forma de refuerzo integral de la función de valor: Ecuación IRL Bellman

$$V(x(t)) = \int_t^{t+T} x^T(\tau)\left(Q + K^T R K\right) x(\tau) d\tau + V(x(t+T)) \tag{2.9}$$

Denota $x(t)$ por x_t y escribe la función de valor como $V(x_t) = x_t^T P x_t$. Entonces, con base en la ecuación IRL Bellman (2.9), se puede escribir el siguiente algoritmo RL.

Algoritmo 3 Algoritmo de iteración de política de aprendizaje de refuerzo integral

Seleccione $\mu^{(0)}(x(t)) \in \Psi(\Omega)$ como política admisible.

$$x_t^T P_t x_t = \int_t^{t+T} x_t^T \left(Q + K_l^T R K_t\right) x_\tau d\tau + x_{t+T}^T P_t x_{t+T} \tag{2.10}$$

$$K_t + 1 = R^{-1} B^T P_t \tag{2.11}$$

Las ecuaciones (2.10) y (2.11) formulan un nuevo algoritmo de iteración de políticas para sistemas de tiempo continuo. Se requiere una ganancia de control de estabilización

inicial K_1. Tenga en cuenta que la implementación de este algoritmo no involucra la matriz de la planta A.

Escribir la función de costo como en (2.9) es la clave para el método de control adaptativo óptimo desarrollado en este capítulo. Esta ecuación tiene la misma forma que la ecuación de Bellman para sistemas de tiempo discreto. De hecho, es una ecuación de Bellman para sistemas de TC que se puede usar en lugar de la ecuación de Bellman en términos de la función hamiltoniana.

$$\rho(x(t), t, T) \equiv \int_t^{t+T} x^T(\tau) \left(Q + K^T R K\right) x(\tau) d\tau \tag{2.12}$$

el refuerzo integral, y (2.9) la forma de refuerzo integral de la función de valor. Entonces, (2.10), (2.11) es la forma IRL de iteraciones de políticas para sistemas CT.

Prueba de convergencia

Los siguientes resultados establecen la convergencia del algoritmo IRL (2.10), (2.11).

Lema 1. *Suponiendo que el sistema $x = A_i x$, con $A_i = A - BK_i$, es estable, resolver para P_i en (2.10) es equivalente a encontrar la solución de la ecuación subyacente de Lyapunov*

$$A_l^T P_l + P_l A_l = -\left(K_l^T R k_l + Q\right) \tag{2.13}$$

Demostración. Como A_i es una matriz estable y $K_l^T R K_l + Q > 0$ entonces existe una solución única de la ecuación de Lyapunov (2.13), $P_{yo} > 0$. Además, dado que $V_l(x_t) = x_t^T P_t x_t, \forall x_t$ es una función de Lyapunov para el sistema $x = A_i x$ y

$$\frac{d\left(x_t^T P_l x_t\right)}{dt} = x_t^T \left(A_l^T P_l + P_l A_l\right) x_t = x_t^T \left(K_t^T R K_l + Q\right) x_t \tag{2.14}$$

entonces, $\forall T > 0$ la solución única de la ecuación de Lyapunov satisface

$$\int_t^{t+T} x_\tau^T \left(Q + K_l^T R K_l\right) x_\tau d\tau = -\int_t^{t+T} \frac{d\left(x_\tau^T P_l x_\tau\right)}{d\tau} d\tau = x_t^T P_l x_l - x_{t+T}^T P_l x_t + T \tag{2.15}$$

es decir (2.10). Es decir, siempre que el sistema $\dot{x} = A_l x$ sea asintóticamente estable, la solución de (2.10) es la única solución de (2.13.

□

Lema 2. *Suponga que la política de control K_i se estabiliza en la iteración i con $V_i(x_t) = x_t^T P_i x_t$ el valor asociado. Entonces, si se usa (2.11) para actualizar la política de control, la nueva política de control K_{i+1} se estabiliza.*

$$A_l^T P_l + P_l A_l = -\left(K_l^T R k_l + Q\right) \tag{2.16}$$

Demostración. Tomando la función de costo definida positiva $V_i(x_t)$ como una función candidata de Lyapunov para las trayectorias de estado generadas al usar el controlador K_{i+1}. Tomando la derivada de $V_i(x_t)$ a lo largo de las trayectorias generadas por K_{i+1} se obtiene

$$\begin{aligned} V_l(x_t) =& x_t^T \left[P_l(A - BK_{l+1}) + (A - BK_{l+1})^T P_l\right] x_t \\ =& x_t^T \left[P_l(A - BK_l) + (A - BK_l)^T P_l\right] x_t \\ &+ x_t^T \left[P_l B(K_l - k_{l+1}) + (K_l - k_{l+1})^T B^T P_l\right] x_t \end{aligned} \tag{2.17}$$

El segundo término, usando la actualización dada por (2.11) y completando los cuadrados, puede escribirse como

$$\begin{aligned} & x_t^T \left[K_{l+1}^T R(k_l - K_{l+1}) + (K_l - K_{l+1})^T R K_{l+1}\right] x_t \\ & \quad = x_t^T \left[-(K_l - K_{l+1})^T R(K_l - K_{l+1}) - K_{l+1}^T R K_{l+1} + K_t^T R K_l\right] x_t \end{aligned}$$

Usando (1), el primer término en (2.17) se puede escribir como $-x_t^T \left[K_l^T R K_l + Q\right] x_t$ y sumando los dos términos se obtiene

$$\dot{V}_l(x_t) = -x_t^T \left[(K_l - K_{l+1})^T R(K_l - K_{l+1})\right] x_t - x_t^T \left[Q + K_{l+1}^T R K_{l+1}\right] x_t \tag{2.18}$$

Por lo tanto, bajo los supuestos iniciales de la configuración del problema $Q \geq 0, R > 0, V_l(x_t)$ es una función de Lyapunov que demuestra que la política de control actualizada $u = -K_{i+1}x$, con K_{i+1} dado por (2.11), es estable.

□

Observación 1. *Con base en el Lema 2, se puede concluir que si la política de control inicial dada por K_1 se está estabilizando, entonces todas las políticas obtenidas usando la iteración (2.10), (2.11) se están estabilizando para cada iteración i.*

Denotamos por $\mathrm{Ric}\,(P_i)$ *la función matricial definida como*

$$\mathrm{Ric}\,(P_l) = A^T P_l + P_l A + Q - P_l B R^{-1} B^T P_l \tag{2.19}$$

y sea Ric_{P_i} *denotemos la derivada de Fréchet de* $\mathrm{Ric}\,(P_i)$ *tomada con respecto a* P_i*. La función matricial* Ric'_{P_i} *evaluada en una matriz dada* M *es dado por*

$$\mathrm{Ric}'_{p_i}(M) = \left(A - BR^{-}1B^T P_t\right)^T M + M\left(A - BR^{-1}B^T P_t\right) \tag{2.20}$$

Lema 3. *La iteración (2.10), (2.11) es equivalente al método de Newton*

$$P_l = P_t - 1 - \left(\mathrm{Ric}'_{P_{i-1}}\right)^{-1} \mathrm{Ric}\,(P_{l-1}) \tag{2.21}$$

Demostración. Las ecuaciones (2.13) y (2.11) se pueden escribir de forma compacta como

$$A_l^T P_l + P_l A_l = -\left(P_{l-1} B R^{-1} B^T P_{l-1} + Q\right) \tag{2.22}$$

Restando $A_t^T P_{l-1} + P_{l-1} A_l$ en ambos lados obtenemos

$$\begin{aligned} A^T (P_l - P_{l-1}) + (P_t - P_{l-1}) A_l = &-\left(P_{l-1} A + A^T P_{l-1}\right. \\ &\left. - P_{l-1} B R^{-1} B^T P_{l-1} + Q\right) \end{aligned} \tag{2.23}$$

□

2.2. Aprendizaje por refuerzo integral (IRL) para sistemas de tiempo continuo no lineales

Considere el sistema dinámico afín en la entrada invariante en el tiempo dado por

$$\dot{x}(t) = f(x(t)) + g(x(t))u(x(t)), \quad x(0) = x_0 \tag{2.24}$$

con el estado $x(t) \in \mathbb{R}^n, f(x(t)) \in \mathbb{R}^n, g(x(t)) \in \mathbb{R}^{n\times m}$ y la entrada de control $u(t) \in U \subset \mathbb{R}^m$. Se supone que $f(0) = 0$, que $f(x) + g(x)u$ es una continua de Lipschitz en un conjunto $\Omega \subseteq \mathbb{R}^n$ que contiene el origen, y que el sistema dinámico es estabilizable en Ω, es decir, existe una función de control continua $u(t) \in U$ tal que el sistema en lazo cerrado es asintóticamente estable en Ω.

Notamos aquí que aunque la estabilidad asintótica global está garantizada en un caso de sistema lineal, generalmente es difícil de garantizar en un entorno de problema de sistema no lineal de tiempo continuo general. Esto se debe a la posible naturaleza no suave de la dinámica del sistema no lineal. En los puntos donde existen discontinuidades en $\dot{x}$, también existirán discontinuidades del gradiente de la función de costo. Por esta razón, la discusión aquí se restringe al caso en el que se busca la estabilidad asintótica solo en una región $\Omega \subseteq \mathbb{R}^n$ en la que la función de costo es continuamente diferenciable.

Definimos un costo integral de horizonte infinito asociado con la entrada de control $\{u(\tau); \tau \geq t\}$. como

$$V(x(t)) = \int_t^\infty r(x(\tau), u(\tau))d\tau = \int_t^\infty \left(Q(x) + u^T Ru\right) d\tau \tag{2.25}$$

donde $x(\tau)$ denota la solución de (2.24) para la condición inicial $x(t) \in \Omega$ y entrada $\{u(\tau); \tau \geq t\}$. El integrando de costo se toma como $r(x, u) = Q(x) + u^T Ru$ con $Q(x)$ definido positivo (es decir, $\forall x \neq 0, Q(x) > 0$ y $x = 0 \Rightarrow Q(x) = 0$) y $R \in \mathbb{R}^{m\times m}$ una matriz definida positiva simétrica.

Definición 1. *(Política (estabilizadora) admisible) [70] Una política de control $u(t) = \mu(x)$ se define como admisible con respecto a (2.25) en Ω, denotada por $\mu \in \Psi(\Omega)$, si $\mu(x)$ es continuo en $\Omega, \mu(0) = 0, \mu(x)$ estabiliza (2.24) en Ω y $V(x_0)$ es finito $\forall x_0 \in \Omega$.*

La función de costo o valor asociado con cualquier política de control admisible $\mu(t) = \mu(x(t)) \in \Psi(\Omega)$ es

$$V^\mu(x(t)) = \int_t^\infty r(x(\tau), \mu(x(\tau)))d\tau \tag{2.26}$$

donde $V^{\mu}(x)$ es C^1. Usando la fórmula de Leibniz, se encuentra que la versión infinitesimal de (2.26) es la siguiente.

Ecuación de Tiempo-Continuo de Bellman

$$0 = r(x, \mu(x)) + (\nabla V x^{\mu})^T (f(x) + g(x)\mu(x)), \quad V^{\mu}(0) = 0 \tag{2.27}$$

Aquí, ∇V_x^{μ} (un vector columna) denota el gradiente de la función de costo V^{μ} con respecto a x. Eso es $\nabla V_x^{\mu} = \partial V^{\mu}/\partial x$. La ecuación (2.27) es una ecuación de Bellman para sistemas no lineales de tiempo continuo (CT, por sus siglas en inglés), que, dada la política de control $\mu(x) \in \Psi(\Omega)$, puede resolverse para el valor $V^{\mu}(x)$ asociado a él. Dado que $\mu(x)$ es una política de control admisible, si $V^{\mu}(x)$ satisface (2.27), con $r(x, \mu(x)) \geq 0$, entonces se puede demostrar que $V^{\mu}(x)$ es una función de Lyapunov para el sistema (2.24) con política de control $\mu(x)$.

Ahora se puede formular el problema de control óptimo: dado el sistema de tiempo continuo (2.24), el conjunto $u \in \Psi(\Omega)$ de políticas de control admisibles y el funcional de costo de horizonte infinito (2.25), encuentre una política de control admisible tal que el valor (2.26) se minimiza.

Definiendo el hamiltoniano

$$H(x, u, V_x) = r(x(t), u(t)) + (\nabla V_x)^T (f(x(t)) + g(x(t))u(t)) \tag{2.28}$$

la función de costo óptima $V^*(x)$ satisface la ecuación de Hamilton-Jacobi-Bellman (HJB)

$$0 = \min_{u \in \Psi(\Omega)} [H(x, u, \nabla V_x^*)] \tag{2.29}$$

Suponiendo que el mínimo en el lado derecho de esta ecuación existe y es único, entonces la función de control óptima es

$$u^*(x) = -\frac{1}{2} R^{-1} g^T(x) \nabla V_x^* \tag{2.30}$$

Insertando esta política de control óptimo en el hamiltoniano obtenemos la formulación de la ecuación HJB

$$0 = Q(x) + (\nabla V_x^*)^T f(x) - \frac{1}{4} (\nabla V_x^*)^T g(x) R^{-1} g^T(x) \nabla V_x^*, \quad V^*(0) = 0 \tag{2.31}$$

Esta es una condición necesaria para la función de costo óptima. Para el caso de un sistema lineal con un costo funcional cuadrático, el equivalente de esta ecuación HJB es la ecuación algebraica de Riccati.

Para encontrar la solución de control óptima para el problema, se puede resolver la ecuación HJB (2.31) para la función de costo y luego sustituir la solución en (2.30) para obtener el control óptimo. Sin embargo, resolver la ecuación HJB es generalmente difícil y es posible que no existan soluciones analíticas. Resolver explícitamente el HJB también requiere un conocimiento completo de la dinámica del sistema $f(x), g(x)$.

2.2.1. Iteraciones de políticas de aprendizaje por refuerzo integral

El desarrollo de métodos RL, como la iteración de políticas y la iteración de valores para sistemas CT, ha retrasado su desarrollo para sistemas de tiempo discreto. Esto se debe a que la ecuación de Bellman de CT (2.27) no comparte ninguna de las propiedades beneficiosas de la ecuación de Bellman de Tiempo Discreto (TD, por sus siglas en inglés) de la sección, que es la siguiente

Ecuación de DT Bellman

$$V(x_k) = r(x_k, u_k) + \gamma V(x_{k+1}) = Q(x_k) u_k^T R u_k + \gamma V(x_{k+1}) \tag{2.32}$$

con k el índice de tiempo discreto y $r \leq 1$ el factor de descuento. Específicamente, la dinámica $(f(\cdot), g(\cdot))$ no aparece en la ecuación de DT Bellman, mientras que sí aparece en la ecuación de CT Bellman (2.27). Esto dificulta la formulación de algoritmos como Q-learning, que no requieren conocimiento de la dinámica del sistema. Además, en la ecuación de DT Bellman hay dos ocurrencias de la función de valor, evaluadas en diferentes

momentos k y $k+1$. Esto permite la formulación de iteración de valor, o programación dinámica heurística, para sistemas DT. Sin embargo, con solo una ocurrencia del valor en la ecuación de CT Bellman, no está del todo claro cómo formular cualquier tipo de procedimiento de iteración de valor.

Con base en la ecuación de Bellman de CT (2.27), se podría escribir un algoritmo de iteración de políticas para los sistemas de CT de la siguiente manera. Índice i es el número de paso de iteración

Algoritmo 4 Algoritmo de iteración de políticas para sistemas de tiempo continuo

Seleccione $\mu^{(0)}(x(t)) \in \Psi(\Omega)$ como política admisible.

Actualización de valor

(Paso de evaluación de políticas) Resuelva el valor $V^{\mu^{(n)}}(x(t))$ utilizando la ecuación de CT Bellman

$$0 = r\left(x, \mu^{(i)}(x)\right) + \left(\nabla V_x^{\mu^{(i)}}\right)^T \left(f(x) + g(x)\mu^{(i)}(x)\right) \text{ con } V^{\mu^{(i)}}(0) = 0$$

(Paso de mejora de la política) Actualizar la política de control usando

$$\mu^{(i+1)} = \arg\min_{u \in \Psi(\Omega)} \left[H\left(x, u, \nabla V_x^{\mu^{(n)}}\right)\right]$$

El algoritmo PI (Iteración de Políticas, por sus siglas en inglés) resuelve la ecuación HJB no lineal mediante iteraciones en ecuaciones lineales en la función de valor gradiente. Se demostró que el algoritmo converge en [70–72]. El algoritmo PI tiene una característica no deseada, ya que requiere un conocimiento completo de la dinámica del sistema $(f(\cdot), g(\cdot))$ para resolver la ecuación de CT Bellman en cada paso de iteración.

Algoritmo de iteración de política de aprendizaje de refuerzo integral

Para mejorar el Algoritmo 6, ahora presentamos una nueva formulación de la ecuación de Bellman de CT basada en el aprendizaje por refuerzo integral (IRL). El formulario

IRL permite el desarrollo de nuevos algoritmos de iteración de políticas para sistemas CT. Esto conduce a su vez a un algoritmo adaptativo en línea que resuelve el problema de control óptimo sin utilizar el conocimiento de la dinámica de deriva $f(x)$. Dada una política admisible y un intervalo de tiempo de integración $T > 0$, escriba la función de valor (2.26) como la siguiente forma equivalente [66, 67].

Forma de refuerzo integral de la función de valor: Ecuación IRL Bellman

$$V^{\mu}(x(t)) = \int_{t}^{t+T} r(x(\tau), \mu(x(\tau)))d\tau + V^{\ mu}(x(t+T)) \tag{2.33}$$

Tenga en cuenta que esta forma no contiene la dinámica del sistema $(f(\cdot), g(\cdot))$. El lema 4.1 muestra que esta ecuación es equivalente a la ecuación de Bellman (2.27) en el sentido de que ambas ecuaciones tienen la misma solución. Por lo tanto, el uso de la ecuación IRL Bellman (2.32 permite la formulación de algoritmos PI de tiempo continuo que comparten las características beneficiosas de los algoritmos PI de tiempo discreto. Integrando

$$\rho(x(t), t, t+T) = \int_{t}^{t+T} r(x(\tau), \mu(x(\tau)))d\tau \tag{2.34}$$

se conoce como *refuerzo integral* en el intervalo de tiempo $[t, t+T]$.

Sea $\mu^{(0)}(x(t)) \in \Psi(\Omega)$ una política admisible, seleccione $T > 0$ tal que, si $x(t) \in \Omega$, luego también $x(t+T) \in \Omega$. La existencia de tal período de tiempo $T > 0$ está garantizada por la admisibilidad de $\mu^{(0)}(\cdot)$ en Ω. Defina el siguiente algoritmo PI basado en la ecuación IRL Bellman.

El algoritmo 5 ofrece una nueva formulación para el algoritmo de iteración de políticas que permite la solución del problema de control óptimo sin necesidad de conocer la dinámica de deriva $f(x)$.

La ecuación IRL Bellman (2.35) es una versión discretizada de $V^{\mu^{(i)}}(x(t)) = \int_{f}^{\infty} r\left(x(\tau), \mu^{(i)}(x(\tau))\right) d\tau$ y puede verse como una ecuación de Lyapunov para sistemas no lineales. También se refiere a ella como

Algoritmo 5 Algoritmo de iteración de política de aprendizaje de refuerzo integral

Seleccione $\mu^{(0)}(x(t)) \in \Psi(\Omega)$ como política admisible.

Actualización de valor

(Paso de evaluación de la política) Resuelva el valor $V^{\mu^{(t)}}(x(t))$ utilizando la ecuación IRL Bellman

$$V^{\mu^{(1)}}(x(t)) = \int_t^{t+T} r\left(x(s), \mu^{(t)}(x(s))\right) ds + V^{\mu(t)}(x(t+T)) \text{ with } V^{\mu^{(t)}}(0) = 0 \quad (2.35)$$

(Paso de mejora de la política) Actualice la política de control utilizando

$$\mu^{(t+1)} = \arg\min_{w \in \Psi(\Omega)} \left[H\left(x, u, \nabla V_x^{u^{(t)}}\right)\right]$$

que explícitamente es

$$\mu^{(t+1)}(x) = -\frac{1}{2} R^{-1} g^T(x) \nabla V_x^{\mu^{(t)}} \quad (2.36)$$

$$LE\left(V^{\mu^{(t)}}(x(t))\right) \triangleq \int_t^{t+T} r\left(x(s), \mu^{(i)}(x(s))\right) ds + V^{\mu^{(i)}}(x(t+T)) - V^{\mu^{(i)}}(x(t)) \quad (2.37)$$

con $V^{\mu^{(i)}}(0) = 0$.

Convergencia de la iteración de políticas IRL

Ahora se muestra que el algoritmo IRL PI converge a la política de control óptima $\mu^* \in \Psi(\Omega)$ con el costo correspondiente $V^*(x_0) = \min_\mu \left(\int_0^\infty r(x(\tau), \mu(x(\tau)))d\tau\right)$.

2.3. Simulación

Considere el modelo dinámico del robot manipulador en el espacio cartesiano [73]

$$M(q)\ddot{x} + C(q,\dot{q})\dot{x} + F_c(\dot{q}) + G(q) + \tau_d = \tau + K_h f_h \tag{2.38}$$

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 número 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

$$\bar{M}\ddot{x}_m + \bar{B}\dot{x}_m + \bar{K}x_m = K_h f_h + \bar{l}\left(x_d\right) \equiv l\left(f_h, x_d\right) \tag{2.39}$$

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 $\bar{l}\left(x_d\right)$ es una entrada dependiente de la trayectoria.

2.3.1. Péndulo

El péndulo invertido es un sistema inherentemente inestable con una dinámica altamente no lineal. Este es un sistema que pertenece a la clase de sistemas mecánicos subactuados que tienen menos entradas de control que el grado de libertad. Esto hace que la tarea de control sea más desafiante, convirtiendo al sistema de péndulo invertido en un punto de referencia clásico para el diseño, prueba, evaluación y comparación de

diferentes técnicas de control clásicas y contemporáneas. Al ser un sistema intrínsecamente inestable, el péndulo invertido se encuentra entre los sistemas más difíciles y es uno de los problemas clásicos más importantes. El control de péndulo invertido ha sido un interés de investigación en el campo de la ingeniería de control. Por su importancia, se trata de una elección de sistema dinámico para analizar su modelo dinámico y proponer una ley de control.

En general, el problema de control consiste en obtener modelos dinámicos de sistemas y utilizar estos modelos para determinar leyes o estrategias de control para lograr la respuesta y el rendimiento deseados del sistema. La simplicidad del algoritmo de control, además de garantizar la estabilidad y robustez en el sistema de circuito cerrado, es una tarea desafiante en situaciones reales. La mayoría de los sistemas dinámicos, como los sistemas de potencia, los sistemas de misiles, los sistemas robóticos, el péndulo invertido, los procesos industriales, los circuitos caóticos, etc., son de naturaleza altamente no lineal. El control de tales sistemas es una tarea desafiante.

El control proporcional-integral-derivativo (PID) brinda la solución más simple y, sin embargo, la más eficiente para varios problemas de control del mundo real. Tanto las respuestas transitorias como las de estado estacionario son atendidas con su funcionalidad de tres términos (es decir, P, I y D). Desde su invención, la popularidad del control PID ha crecido enormemente.

Para este caso se propuso el siguiente diseño de inner loop, en el cual se tienen agregadas dos ganancias para facilitar la comparación los resultados de cada controlador

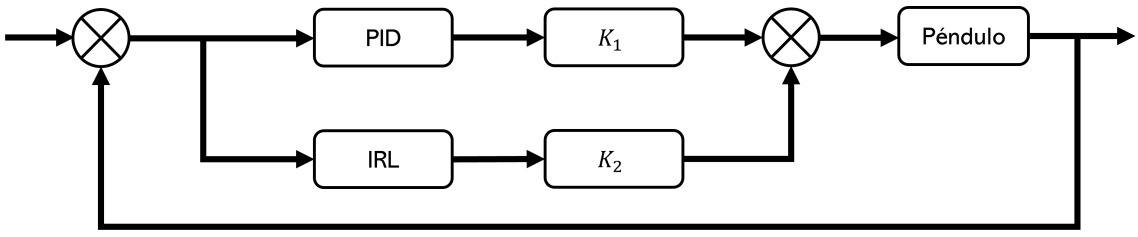


Figura 2.1: Lazo interno con PID+IRL

La dinámica del sistema carro-péndulo se expresa por

$$M(q)\dot{q} + C(q,\dot{q})\dot{q} + G(q) = Bu \tag{2.40}$$

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.

$$\dot{x} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ \frac{m\,sin(q)(g\,cos(q)-l\dot{q})}{M+m-m\,cos(q)^2} \\ \frac{sin(q)(g(M+m)-l\,m\,cos(q)\dot{q}))}{l(M+m-m\,cos(q))} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{M+m-m\,cos(q)^2} \\ \frac{cos(q)}{l(M+m-m\,cos(q))} \end{bmatrix} u \tag{2.41}$$

Donde:

$$\begin{aligned} x &= (x_c, \dot{x}_c, q, \dot{q}_c)^T \\ u &= F \end{aligned} \tag{2.42}$$

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

La Tabla 2.1 muestra los valores de los parámetros de control del algoritmo PDI+IRL

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.

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, mín e_x. Las acciones están definidas como el par aplicado al péndulo

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

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

Tabla 2.1: Parámetros usados para la simulación numérica

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:

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

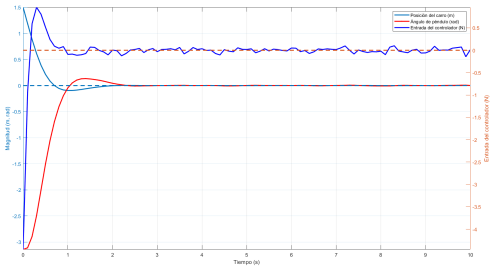
la recompensa estará definida de la forma:

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

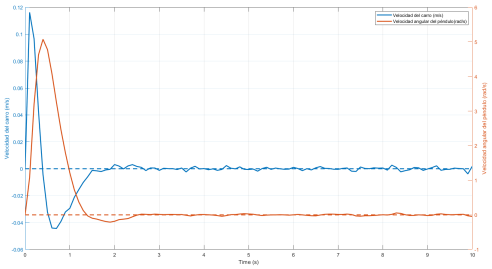
donde $\tilde{q} = q_d - q$, y la mayor recompensa se produce cuando el error $\tilde{q}$ y la velocidad $\dot{q}$ son cero, es decir, cuando el péndulo se encuentra sobre la vertical superior y su velocidad es cero. Además, en la función de recompensa, la velocidad se encuentra escalada por un factor 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(2.2, 3.5, 2.4) se presentan las gráficas de salida para el ángulo deposición y velocidad.

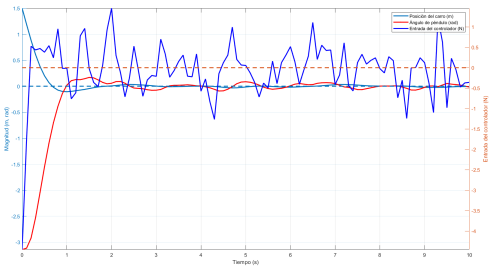
Para el control del péndulo invertido haciendo uso de solamente el PID se analiza para casos de uso de perturbación y sin perturbación es ejecutado. El péndulo se estabiliza en posición vertical y el carro alcanza la posición deseada en el caso que no hay perturbaciones, al momento de tomar en cuenta la perturbación en el sistema se puede visualizar que este no es capaz de estabilizar el péndulo y el carro.



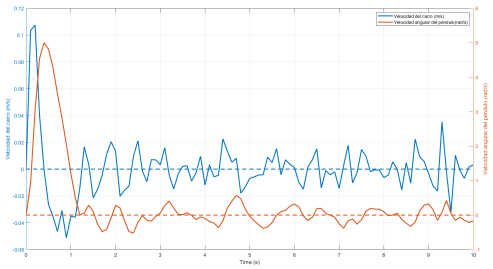
(a) Controlador PD sin perturbación



(b) Velocidad del sistema con un control PD sin perturbación



(c) Controlador PD con perturbación

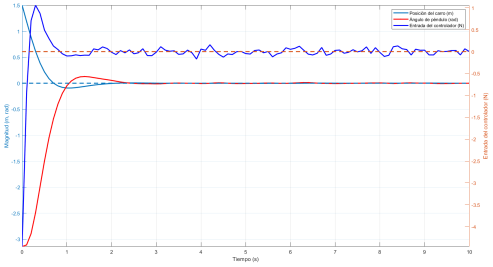


(d) Velocidad del sistema con un control PD con perturbación

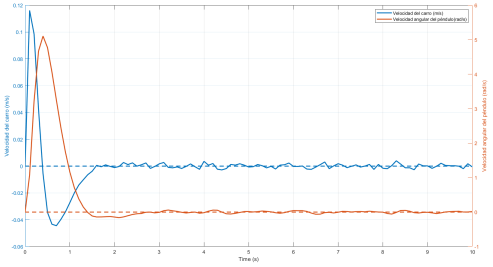
Figura 2.2: Gráficas de posición, ángulo y control, usando el controlador PID con y sin perturbaciones

Para el control del péndulo invertido haciendo uso de solamente el IRL se analiza para casos de uso de perturbación y sin perturbación es ejecutado. El péndulo se estabiliza en posición vertical y el carro alcanza la posición deseada en el caso que no hay perturbaciones, al momento de tomar en cuenta la perturbación en el sistema se

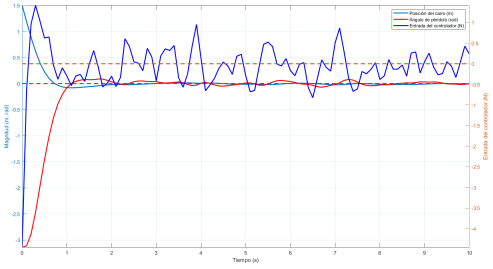
puede visualizar que este es capaz de estabilizar de una mejor manera que el PID, pero de igual manera se pueden ver algunas oscilaciones al tratar de llegar a la posición deseada.



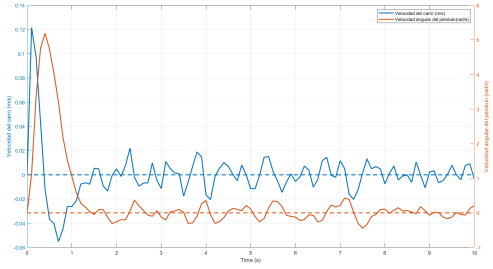
(a) Controlador IRL sin perturbación



(b) Velocidad del sistema con un control IRL sin perturbación



(c) Controlador IRL sin perturbació

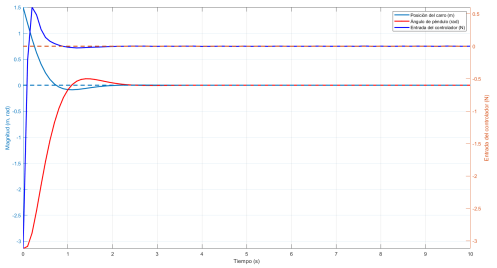


(d) Velocidad del sistema con un control IRL sin perturbación

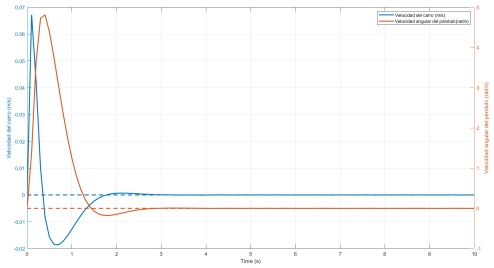
Figura 2.3: Gráficas de posición, ángulo y control, usando el controlador IRL con y sin perturbacioness

Para el control del péndulo invertido no lineal, control PID+IRL, se ha implementado una estrategia de control para un control óptimo. Se analiza para casos de uso de perturbación y sin perturbación es ejecutado. El IRL se alimenta directamente con todos los estados del sistema que se pueden obtener para la medición. El diseño de IRL se realiza utilizando el espacio de estado del modelo. Matlab se utiliza para desarrollar los modelos y analizar las salidas de respuesta. El ajuste de los controladores para lograr el control más óptimo se logra mediante el método de prueba y error. Los resultados de la

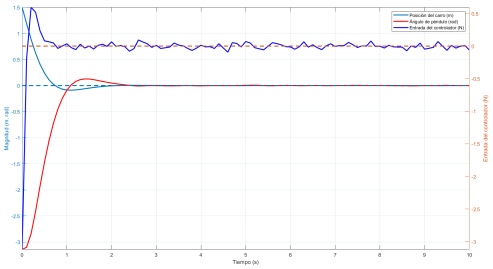
simulación justifican la efectividad relativa del uso del método de control IRL. El péndulo se estabiliza en posición vertical y el carro alcanza la posición deseada incluso en presencia de perturbaciones. El estudio de las respuestas de los controles muestra que el rendimiento del método de controles PID + IRL es superior al que usa solo PID. Esto concluye con la afirmación de que, de todos los esquemas de control realizados, IRL + PID es efectivo, robusto y simple.



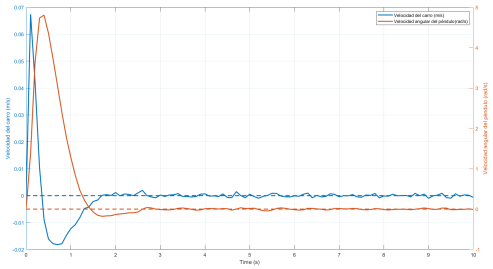
(a) Controlador PID+IRL sin perturbación



(b) Velocidad del sistema con un control PID+IRL sin perturbación



(c) Controlador PID+IRL con perturbación



(d) Velocidad del sistema con un control PID+IRL con perturbación

Figura 2.4: Gráficas de posición, ángulo y control, usando el controlador PID junto con el IRL como compensador con y sin perturbaciones

2.3.2. Conclusión

Este capítulo presenta un controlador adaptativo de tiempo continuo, basado en la iteración de políticas, que se adapta en línea para aprender la política de control óptimo de tiempo continuo sin usar el conocimiento sobre la dinámica de deriva del sistema no lineal. El controlador se basa en la forma de aprendizaje de refuerzo integral (IRL) de la ecuación de Bellman de CT, que no contiene la dinámica del sistema y comparte las propiedades beneficiosas de las ecuaciones de Bellman de tiempo discreto.

Se estableció la convergencia del algoritmo de iteración de la política IRL, bajo la condición de un controlador estabilizador inicial, a la solución del problema de control óptimo. Se proporcionó un método práctico para implementar el algoritmo basado en la aproximación de función de valor (VFA, por sus siglas en inglés). También se proporcionó prueba de convergencia para el algoritmo de iteración de políticas IRL utilizando VFA, teniendo en cuenta el error de aproximación.

Capítulo 3

Control adaptable en el lazo externo con Humano en el bucle

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 [1–3], las redes difusas [4, 5] y el aprendizaje por refuerzo [6].

3.1. 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 [74] que contribuyen a que los eventos no sucedan simultáneamente. Los sistemas con retrasos existen ampliamente en ingeniería, biología, física, investigación de operaciones y economía [74].

En [75], 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 [76] 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 [75].

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 [74]. En [77], 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 [74].

3.2. Modelado del humano

3.2.1. 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 [39].

La figura 3.1 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 [78] 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 $c(t)$.

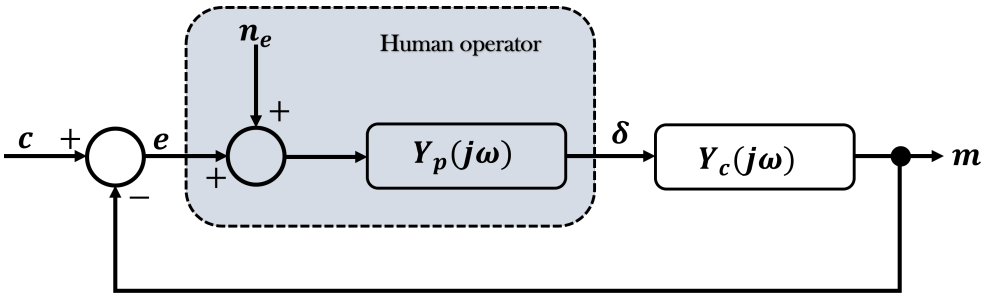


Figura 3.1: Una representación cuasi-lineal de función descriptiva del operador humano [42]

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.

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 [79]. 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.

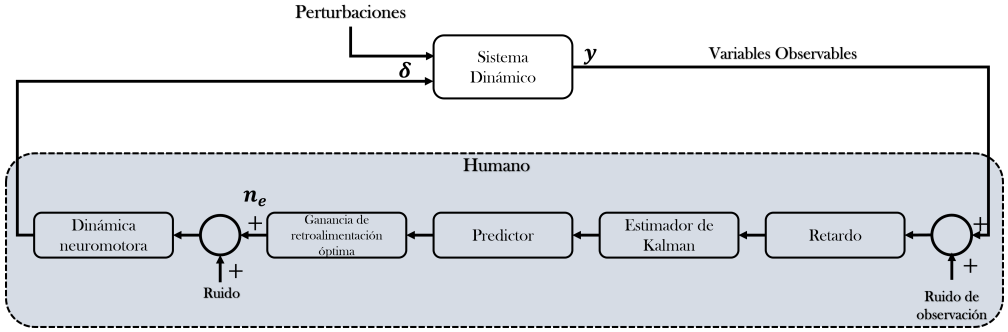


Figura 3.2: El modelo de control óptimo del operador humano [42]

3.2.2. 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).

La figura 3.2 muestra la estructura básica de 3.2. 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 3.2 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 3.2 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.

3.2.3. 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 [40,80].

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

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 [81].

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 2.39 para minimizar el esfuerzo de control humano f_h y optimizar el rendimiento del seguimiento en función de la tarea.

3.3. Método de control de bucle externo específico de la tarea

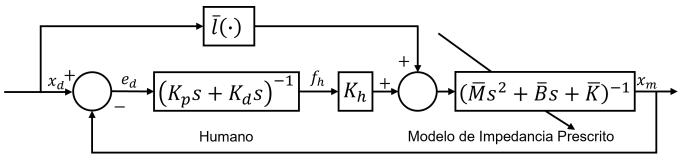


Figura 3.3: Interfaz hombre-robot en el bucle externo específico de la tarea

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 2.39 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 2.39 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

$$\bar{x} = \begin{bmatrix} x_m^T & \dot{x}_m^T \end{bmatrix}^T \in \mathbb{R}^{2n} \tag{3.5}$$

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 $K_h = 1$).

Observación 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 $Q = \text{diag}\,(Q_d, Q_h)$ 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 2.39, 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

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

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 2.39. Por otro lado, basado en el modelo humano 3.17, 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 [82], la función de transferencia humana de e_d a f_h se consideró como

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

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 2.39. 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

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

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 [69]

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

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 $K = \left[\bar{M}^{-1}K_q\bar{M}^{-1}\right] = -R^{-1}B^TP$, lo que da parámetros desconocidos del modelo de impedancia 2.39.

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.

3.4. 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 [66, 67, 83–86]. En este escrito, se utiliza el algoritmo integral RL (IRL) fuera de política [66, 67, 85, 86] 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) [87,88]. 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 [66, 67] 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 [86]

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

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 [85] y [86] 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 [86].

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.

3.5. Simulación

En esta sección, el diseño del controlador de bucle de tareas externo se muestra en la Fig. 2. En esta sección, los parámetros del modelo de impedancia del robot prescrito que se dan en (2.39) están optimizados para ayudar al ser humano a realizar una tarea determinada con el mínimo esfuerzo. y para minimizar un error de seguimiento.

Ahora se presentan los resultados del método de controlador de bucle externo propuesto.

Algoritmo 6 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

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

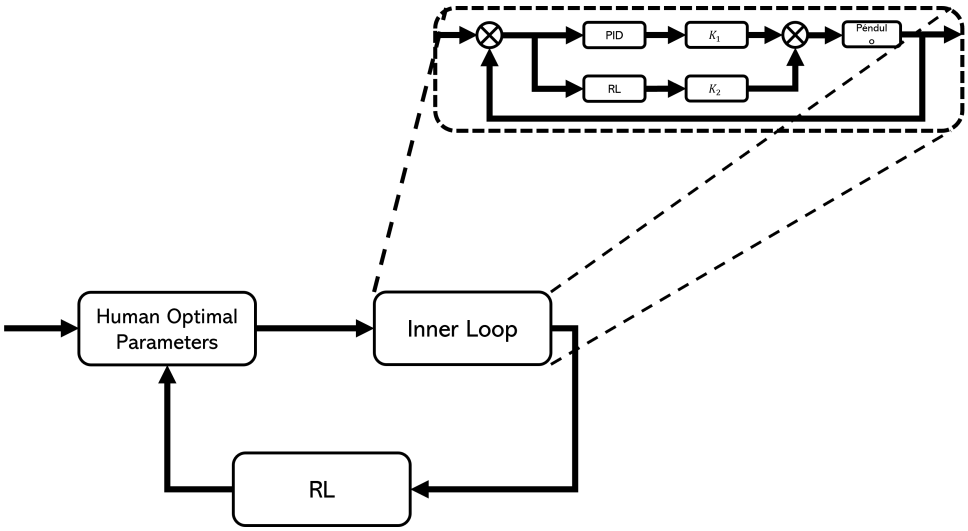
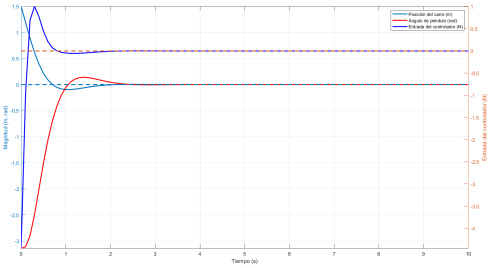
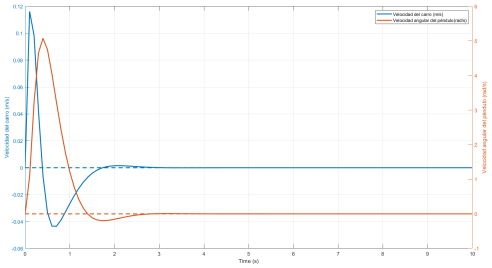


Figura 3.4: Método general de diseño de control de dos bucles para el sistema Human in the loop

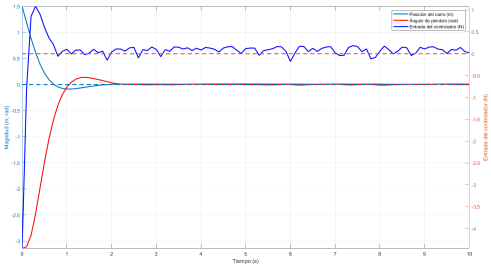
En el cual se visualiza que a pesar de haber una perturbación presente, tanto el inner loop como el humano aportan a la estabilización del sistema en este caso el péndulo



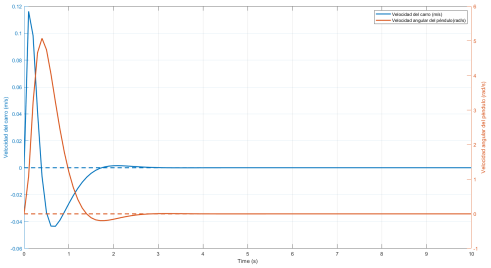
(a) Sistema con HITLC sin perturbación



(b) Velocidad del sistema con HITLC sin perturbación



(c) Sistema con HITLC con perturbación



(d) Velocidad del sistema con HITLC con perturbación

Figura 3.5: Gráficas de posición, ángulo y control, usando el lazo interno y externo con y sin perturbacioness

Capítulo 4

Conclusión General

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

Se presenta un nuevo método de diseño de control HITL 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.

Bibliografía

[1] 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.

[2] 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.

[3] 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.

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

[5] 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.

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

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

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

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

[14] 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.

[15] 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.

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

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

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

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

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

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

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

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

[40] 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.

[41] 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.

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

[43] 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.

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

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

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

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

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

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

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

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

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

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

[54] 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.

[55] Jack Hachigian. Collapsed markov chains and the chapman-kolmogorov equation. *The Annals of Mathematical Statistics*, 34(1):233–237, 1963.

[56] Dimitri P Bertsekas and John N Tsitsiklis. Neuro-dynamic programming: an overview. In *Proceedings of 1995 34th IEEE conference on decision and control*, volume 1, pages 560–564. IEEE, 1995.

[57] R. M. Wheeler and K. S. Narendra. Decentralized learning in Markov chains. *IEEE Transactions on Automatic Control*, 31:519–526, 1986.

[58] Jack Karush. On the chapman-kolmogorov equation. *The Annals of Mathematical Statistics*, 32(4):1333–1337, 1961.

[59] Richard Bellman. Dynamic programming and stochastic control processes. *Information and Control*, 1(3):228–239, 1958.

[60] Frank L. Lewis, Draguna Vrabie, and Kyriakos G. Vamvoudakis. Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers. *IEEE Control Systems Magazine*, 32(6):76–105, Dec 2012.

[61] Warren B. Powell. *Approximate Dynamic Programming - Solving the Curses of Dimensionality*. John Wiley & Sons, New York, 2011.

[62] 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.

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

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

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

[66] 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.

[67] 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.

[68] Draguna Vrabie, Kyriakos G. Vamvoudakis, and Frank L. Lewis. *Optimal Adaptive Control and Differential Games by Reinforcement Learning Principles*. Institution of Engineering and Technology, January 2012.

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

[70] Randal W Beard, George N Saridis, and John T Wen. Galerkin approximations of the generalized hamilton-jacobi-bellman equation. *Automatica*, 33(12):2159–2177, 1997.

[71] RJ Leake and Ruey-Wen Liu. Construction of suboptimal control sequences. *SIAM Journal on Control*, 5(1):54–63, 1967.

[72] Murad Abu-Khalaf and Frank L. Lewis. Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network hjb approach. *Automatica*, 41(5):779–791, 2005.

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

[74] 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.

[75] 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.

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

[77] 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.

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

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

[80] 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.

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

[82] 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.

[83] 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.

[84] 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.

[85] 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.

[86] 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.

[87] 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.

[88] 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.