



UNIVERSIDAD DE LA REPÚBLICA
FACULTAD DE INGENIERÍA



Optimización del Almacenamiento de Energía en Redes Eléctricas

TESIS PRESENTADA A LA FACULTAD DE INGENIERÍA DE LA
UNIVERSIDAD DE LA REPÚBLICA POR

Agustin Castellano

EN CUMPLIMIENTO PARCIAL DE LOS REQUERIMIENTOS
PARA LA OBTENCIÓN DEL TÍTULO DE
MAGISTER EN INGENIERÍA ELÉCTRICA.

DIRECTOR DE TESIS

Dr. Juan Andrés Bazerque Universidad de la República

TRIBUNAL

Dr. Mario Ibarburu IIMPI, Facultad de Ingeniería, UdelaR

Dr. Enrique Mallada Johns Hopkins University

Dr. Fernando Paganini Universidad ORT

DIRECTOR ACADÉMICO

Dr. Juan Andrés Bazerque Universidad de la República

Montevideo
lunes 26 julio, 2021

Optimización del Almacenamiento de Energía en Redes Eléctricas, Agustín Castellano.

ISSN 1688-2806

Esta tesis fue preparada en L^AT_EX usando la clase iietesis (v1.1).

Contiene un total de 78 páginas.

Compilada el lunes 26 julio, 2021.

\DocumentURL

There are no solutions, only trade-offs.

THOMAS SOWELL

Esta página ha sido intencionalmente dejada en blanco.

Agradecimientos

A Juan, por tu tiempo y tu pacencia, por todo lo que me enseñaste, por mostrarme tu forma rigurosa de pensar y ver las cosas, por nuestros desencuentros que han ayudado a forjar mi carácter.

A Pablo, por tus consejos, por ser un jefe piola.

A Andrés y Fernando, por las reuniones y discusiones en las que aprendí un montón.

A la gente de UTE, en especial Rodrigo, Camila, Manolo y Alfonso, por las nutritivas charlas e intercambios en el piso 8.

A mis amigos y colegas del instituto, por hacer más ameno mi pasaje por acá.

A Quique, por aceptarme y por lo que se viene.

A la Fing y a la UdelaR, por permitirme formarme, por su Educación pública y de calidad.

A Mauri, Santi, Mamá, Papá y Ceci, por todo el Amor que me han dado a lo largo de los años.

Gracias.

Esta página ha sido intencionalmente dejada en blanco.

A mis viejos.

Esta página ha sido intencionalmente dejada en blanco.

Abstract

Storage technologies are a fundamental component of electric power systems, ranging from slow, high capacity systems (such as hydro reservoirs) to fast, high ramping, low capacity systems (such as utility-scale batteries). In this thesis we deal with optimizing the operation of energy storage systems from a broad perspective, focusing on both the large- and low-scale systems. Operating a grid with storage is a difficult task, in the sense that i) there is inherent uncertainty from the stochastic variables involved (such as the demand and renewable energy available), and ii) storage dynamics couple decisions across time, implying that actions must be taken with respect to some global goal. Accordingly, we formulate the optimal dispatch problem as a multi-stage dynamic programming problem, subject to various control and state constraints.

We study both these cases and consider their applications on the Uruguayan grid. In the case of hydro-reservoirs, we model the cost-to-go functions as convex-quadratic in the reservoirs. This leads to an approximate dynamic programming algorithm which at each stage samples state-cost pairs and fits convex-quadratic functions in a recursive manner. We implement this efficiently via modern optimization solvers, and our results show that the control policy learned in this fashion exceeds the performance of a naïve myopic policy.

We also consider the operation of a bulk battery storage system in a single-bus model of the Uruguayan grid. In this regard, we consider learning the controller via Q-learning, the quintessential algorithm in the field of Reinforcement Learning. With no prior information on the transition model and on the stochastic variables involved, we obtain an agent that makes hourly decisions based on the state of the system, namely the state of charge of the battery, the time of day and the forecasted wind and demand. We train the controller with real data of three winters, and obtain a policy that operates the system with good performance, charging the battery—even at expenses of fuel generation—when generation is cheap and renewable energies abundant, and turning that surplus back to the grid when demand peaks.

Esta página ha sido intencionalmente dejada en blanco.

Resumen

Las tecnologías de almacenamiento energético son un componente fundamental de los sistemas eléctricos de potencia. Entre ellas se comprenden los sistemas lentos, de alta capacidad (como las represas hidroeléctricas), hasta los rápidos, de menor capacidad y con alta rampa de carga (tales como las baterías). Esta tesis versa sobre la optimización de la operación de estos sistemas desde una perspectiva amplia, abocándonos a los sistemas de gran y pequeño porte. Operar una red eléctrica con almacenamiento es una tarea compleja, en el sentido que i) existe incertidumbre inherente al problema debido a las variables estocásticas involucradas (como la demanda y la energía renovable disponible), y ii) la dinámica del almacenamiento acopla las decisiones en el tiempo, implicando que las acciones deben ser tomadas con respecto a algún objetivo global. En concordancia con esto, formulamos el problema de despacho óptimo en el marco de la programación dinámica en varias etapas, sujeto a variadas restricciones de control y estado.

Estudiamos ambos escenarios y consideramos su aplicación en el sistema eléctrico uruguayo. Para el caso de las represas, modelamos las funciones de valor del problema como cuadráticas convexas en el nivel de los embalses. Esto da lugar a un algoritmo de programación dinámica aproximada, que en cada etapa toma muestras de parejas estado-costos y ajusta las cuadráticas de manera recursiva. Implementamos este algoritmo de manera eficiente usando solvers de optimización modernos, y nuestros resultados muestran que la política de control aprendida supera en desempeño a una política miope.

Asimismo consideramos la operación de un banco de baterías en un modelo uninodal de la red uruguaya. El controlador se aprende a partir de Q-learning, uno de los algoritmos más conocidos del campo de *Reinforcement Learning*. Este controlador aprende a tomar decisiones de forma horaria, sin información previa de la dinámica de transición del sistema o de la estadística de las variables aleatorias involucradas, teniendo acceso únicamente al estado del sistema, a saber, el estado de carga de la batería, la hora del día y el pronóstico de demanda y energía renovable para la hora siguiente. Entrenamos a este controlador usando datos reales de tres inviernos uruguayos, y obtenemos una política que opera al sistema con buen desempeño, cargando la batería —incluso a expensas de generación térmica— cuando el costo de generación es barato y la energía renovable abundante, y que vuelca este excedente a la red cuando la demanda es alta.

Esta página ha sido intencionalmente dejada en blanco.

Tabla de contenidos

Agradecimientos	III
Resumen	VII
1. Introducción	1
1.1. Organización del documento	3
2. Programación Dinámica Aproximada Cuadrática para el despacho hidrotérmico	5
2.1. Introducción	5
2.2. Modelado: el caso uruguayo	8
2.2.1. Aportes	10
2.2.2. Incorporación de memoria al estado hidrológico	11
2.2.3. Estimación del modelo Markoviano	12
2.3. Aproximaciones cuadráticas	15
2.3.1. Motivación	15
2.4. Algoritmia	15
2.4.1. Pasada hacia atrás	15
2.4.2. Pasada hacia adelante	18
2.4.3. Implementación	21
2.5. Resultados	22
2.5.1. Pasada hacia atrás	22
2.5.2. Pasada hacia adelante	22
3. Aprendiendo a operar un banco de baterías	27
3.1. Modelado	27
3.1.1. Estado del sistema	29
3.2. Objetivo	30
3.3. Aprendiendo de los datos	31
3.3.1. Función de costo	31
3.3.2. Demanda y Renovable	33
3.4. Algoritmo	37
3.4.1. Q-learning episódico	39
3.4.2. Evaluando la política aprendida	39
3.5. Simulaciones	40
3.5.1. Discusión sobre los resultados	42

Tabla de contenidos

3.6. Operación con pronósticos exactos	44
4. Discusión y trabajo a futuro	49
Referencias	51
Índice de tablas	57
Índice de figuras	58

Capítulo 1

Introducción

Los sistemas de almacenamiento energético (SAEs) son uno de los componentes fundamentales de los sistemas eléctricos de potencia. Han estado presentes en las redes eléctricas desde la instalación de la primera represa, sobre fines del Siglo XIX [1]. Más recientemente se han incorporado nuevos tipos, gracias al desarrollo tecnológico de baterías, bancos de condensadores y masas rotantes [2].

Una de las características fundamentales de los sistemas eléctricos de potencia es que, en condiciones normales de operación, la generación debe equipararse instante a instante con la demanda. De no cumplirse esto, el desbalance de la red generalmente se traduce en niveles de tensión y frecuencia que se desvían de sus parámetros nominales, pudiendo desembocar en el colapso de la red [3, 4]. Los sistemas de almacenamiento energético (SAEs) abren nuevas posibilidades en este ámbito, pudiendo asistir en la generación y en la inyección o absorción de potencia rápidamente.

Los SAEs se vuelven particularmente importantes en redes eléctricas con alta penetración de energías renovables, como es el caso de la uruguayana. El recurso solar tiene su mayor incidencia al mediodía, mientras que el eólico presenta gran variabilidad y en promedio mayor penetración en la madrugada [5] (ver a modo de ejemplo la Figura 1.1). Por otro lado, la demanda eléctrica es baja durante la madrugada y el día y en general alcanza su máximo en la tarde/noche [6]. En este sentido, existe un desfase entre la disponibilidad de los recursos renovables y la necesidad de consumo. Incorporar almacenamiento —en particular baterías, con su escala de tiempo rápida— permite trasladar excedentes de generación a momentos del día en que más se necesiten, evitando así descartar estos excedentes o venderlos a países limítrofes a bajo costo. Este enfoque, apodado *arbitraje de energía*, tiene consecuencias en la planificación del sistema eléctrico, pudiéndose diferir inversiones en expansión de la matriz de generación. También se puede mitigar inversiones en redes de transmisión o distribución, instalando almacenamiento en nodos de la red con alta demanda, evitando así la sobrecarga de las líneas. Los SAEs de escala rápida, que permiten actuar en pasos de tiempo corto inyectando potencia considerable, permiten a su vez ayudar a controlar la estabilidad de la red mediante regulación de frecuencia [7, 8] y tensión [9].

Caben varias interrogantes referidas a los SAEs. Algunas de ellas refieren al

Capítulo 1. Introducción

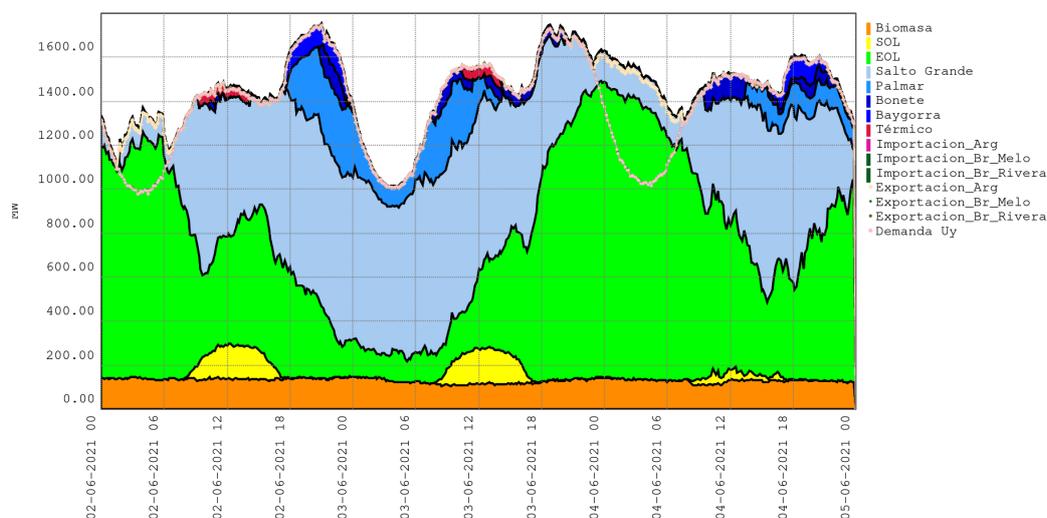


Figura 1.1: Despacho de generación por fuentes para el sistema eléctrico uruguayo, del 2 al 4 de Junio de 2021. Los tipos de generación son: solar (amarillo), eólica (verde), hidráulica (celestes y azules), térmica (rojo) y biomasa (naranja). El trazo punteado color piel corresponde a la demanda nacional, los demás trazos a las exportaciones a Brasil y Argentina. La demanda de cada día presenta tres zonas a destacar: un pico durante la tarde-noche, un valle durante la madrugada y una meseta durante el día. Notar la gran penetración de fuentes renovables (en particular eólica). Prácticamente en todas las horas el despacho se hace al 100 % con fuentes renovables, teniendo que asistir con generación térmica únicamente sobre el mediodía del 2 y 3 de Junio. Fuente: ADME [13].

dimensionado [10] (qué tipo de sistema poner, de qué capacidad y con qué rampa de carga), *ubicación* [10] (en qué parte(s) de la red instalarlo(s)), *operación* [11] (cómo operarlo) y *modelado* [12] (cómo se modela la dinámica del almacenamiento, si incluye pérdidas o filtraciones, degradación). Durante este trabajo nos centramos en lo referente a la operación de los sistemas de almacenamiento.

Estudiamos dos tipos de sistemas, que se encuentran en sendos extremos del espectro (ver Figura 1.2 para una comparación entre SAEs). En primer lugar, tratamos las represas hidroeléctricas con embalses, que son sistemas con almacenamiento de gran porte. Se encuentran asociados a una escala inherentemente *lenta*, habiendo embalses (como Rincón del Bonete, por ejemplo) con constantes de tiempo de varios meses. Tiene sentido entonces considerar el problema de despacho como un problema donde se toman decisiones semana a semana, por ejemplo, y donde se cuenta con datos de los aportes esperados (el caudal hídrico en las represas) y de la demanda promedio para la semana. En segundo lugar, en el otro extremo se encuentran los bancos de baterías, de menor capacidad que las represas pero con la opción de inyectar o absorber potencia rápidamente. Están asociadas a una escala de tiempo más rápida, del orden de minutos o incluso segundos. Al considerar el despacho de estos sistemas tiene sentido tomar en cuenta la variabilidad de los recursos renovables (el viento, por ejemplo) y la demanda.

Uno de los grandes desafíos referidos a la operación del almacenamiento es que las variables y decisiones resultan *acopladas* en el tiempo. Decidir usar toda la

1.1. Organización del documento

energía almacenada en este instante puede ser beneficioso a corto plazo, pero detrimental al considerar lo que podría pasar en los instantes siguientes. El problema se complejiza aún más cuando se consideran variables *estocásticas*, como el viento, el recurso solar o la demanda, que agregan incertidumbre a las predicciones. El marco ideal para trabajar en este sentido es la programación dinámica estocástica [14, 15], donde en general se busca optimizar cierto costo durante varias etapas del problema, sujeto a restricciones y transiciones probabilísticas del estado del sistema.

Cuando se consideran variables estocásticas, la programación dinámica en su forma más pura requiere conocer sus distribuciones de probabilidad, para así poder calcular valores esperados. En muchos casos, sin embargo, estas distribuciones son desconocidas, debiendo entonces recurrir a alguna técnica de solución aproximada. Una solución puede ser armar un modelo estocástico suficientemente descriptivo del comportamiento de estas variables aleatorias y aproximar los valores esperados por sorteos Monte Carlo (Capítulo 2). Otro enfoque consiste en dispensar completamente de los modelos y aprender puramente en base a datos y recompensas (Capítulo 3).

1.1. Organización del documento

La Tesis se subdivide en cuatro capítulos. En el Capítulo 2 (cuya versión abreviada se publicó en [16]) presentamos el problema de operación de embalses interconectados, con aplicación al caso uruguayo. Planteamos el problema de optimización de varias etapas y horizonte fijo, y buscamos resolverlo con el algoritmo de *programación dinámica hacia atrás* [17]. Para evitar la Maldición de la Dimensionalidad [18] recurrimos a aproximar la función de valor del sistema a cada paso por una cuadrática convexa en el vector del nivel de los lagos. La incertidumbre del sistema a cada etapa consiste de los *aportes* o el caudal hídrico que ingresa a cada represa. Para lidiar con esta incertidumbre introducimos una variable de estado adicional que resume las condiciones hidrológicas del sistema. El problema de programación dinámica resultante queda subdividido en una *pasada hacia atrás* donde se toman muestras y se ajustan funciones cuadráticas, y una *pasada hacia adelante* donde se simulan costos de operación sobre varias trayectorias. Se contrastan los resultados contra los de una política miope obteniendo un desempeño superior.

En el Capítulo 3 (versión abreviada publicada en [19]) nos centramos en almacenamiento en el otro extremo del espectro, considerando un banco de baterías en un modelo uninodal de la red uruguaya. Aquí abordamos el problema de optimizar la operación del banco ante la incertidumbre de la demanda y de la potencia eólica disponible. Usamos datos de consumo, energía renovable y costo de generación de los inviernos de 2017 a 2019 en Uruguay, y formulamos el problema como uno de horizonte infinito con costos descontados. Resolvemos este problema usando Q-learning, uno de los algoritmos más conocidos del campo de aprendizaje por recompensas (*Reinforcement Learning*). Para ello determinamos una única función de costo cuadrática para el generador equivalente del sistema, usando los precios nodales en cada uno de los generadores de la red. Los datos de viento y demanda de

Capítulo 1. Introducción

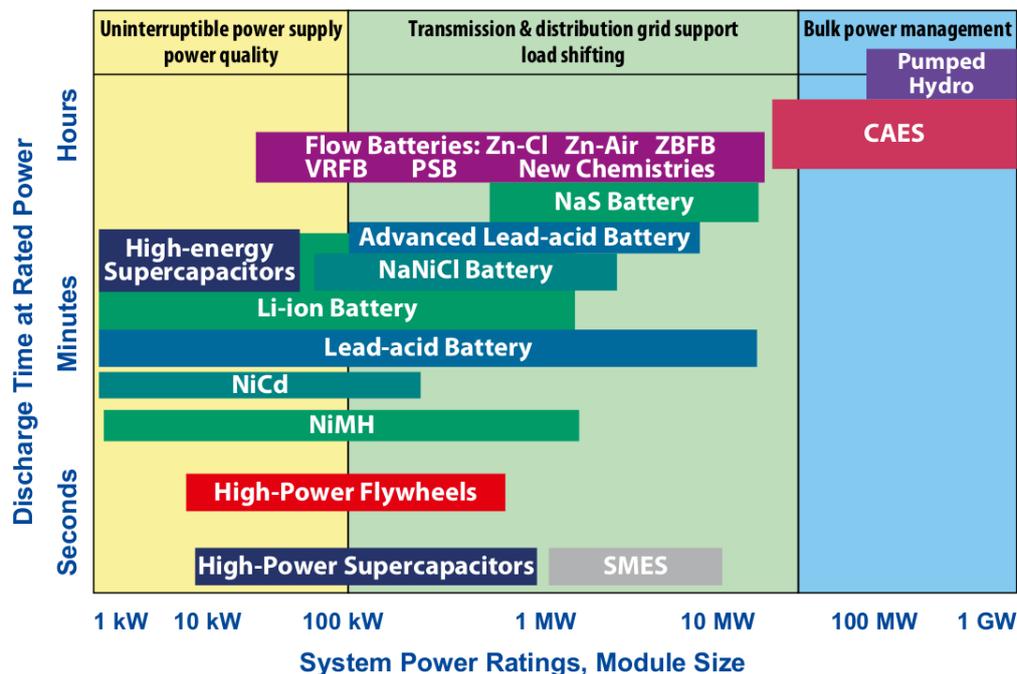


Figura 1.2: Comparación de distintos sistemas de almacenamiento energético. Los sistemas se agrupan según la potencia instalada (abscisas) y el tiempo que le lleva descargarse completamente (ordenadas). Fuente: EPRI [20].

cada día se discretizan y se procesan, de forma de acercarse más a una hipótesis de modelo markoviano. Realizamos un entrenamiento basado en *episodios*, corriendo al agente durante varios días y luego randomizando su estado inicial, a modo de favorecer la exploración del algoritmo. El agente entrenado con Q-learning logra aprovechar los excedentes de generación, operando la red a un costo menor que en el caso sin baterías. Luego de presentar varios ejemplos de despacho, el capítulo termina con una discusión sobre los resultados y pistas sobre el desempeño obtenido. Contrastamos los resultados aprendidos con este enfoque en base a datos versus el despacho hecho por un controlador que, además de conocer la función de transición de la batería, tiene acceso a un pronóstico perfecto de viento y demanda.

Concluimos en el Capítulo 4, resumiendo brevemente el contenido de la tesis y detallando el trabajo actual y posibles líneas de investigación a futuro.

Capítulo 2

Programación Dinámica Aproximada Cuadrática para el despacho hidrotérmico

Este capítulo se centra en los sistemas de almacenamiento eléctrico de mayor porte —las represas. A lo largo del mismo nos centramos en la operación de un sistema *hidrotérmico*, con una única central térmica y varias represas, algunas de ellas interconectadas en cascada bajo el curso de un mismo río. El problema de despacho puede formularse como un problema de programación dinámica estándar, de horizonte fijo y con restricciones en cada paso. Resolveremos este problema aproximando la función de valor de cada etapa por una función cuadrática convexa en el estado del sistema.

Como explicaremos más adelante, *entrenar* al sistema consiste en resolver varios programas cuadráticos (QP) con restricciones lineales y un programa semidefinido positivo (SDP) para cada etapa del problema. Luego de entrenar, obtenemos un controlador de forma *implícita*, que decide el control a aplicar en cada etapa mediante la resolución de un QP.

Aplicamos el algoritmo propuesto para el caso del sistema hidrotérmico uruguayo, y comparamos el desempeño de nuestro controlador frente a una política de base que no considera las dinámicas de transición, obteniendo resultados superiores.

Hemos publicado una versión abreviada de este capítulo en [16].

2.1. Introducción

Considerar el problema de operación de un sistema hidroeléctrico, compuesto por (posiblemente varias) represas interconectadas, y plantas de generación térmica. Dado un horizonte de operación K , e índices de tiempo $k = 0, \dots, K - 1$, el objetivo del operador es conseguir una secuencia de controles $\mathbf{u} = \{u_k\}_{k=0}^{K-1}$ de forma tal de minimizar el costo de operación del sistema

Capítulo 2. Programación Dinámica Aproximada Cuadrática para el despacho hidrotérmico

$$\sum_{k=0}^{K-1} g_k(x_k, u_k, w_k) \quad (2.1)$$

donde $g_k : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \in \mathbb{R}$ es una función que representa el costo asociado a cada etapa (pudiendo contemplar el costo de generación térmica el costo de importación o ganancia por exportación de energía, penalizaciones por no cumplir con la demanda requerida, etc.), $x_k \in \mathbb{R}^n$ es el estado del sistema (el nivel de cada una de las represas), $u_k \in \mathbb{R}^m$ es el vector de controles (el turbinado y vertido en cada represa, las consignas de generación térmica, la decisión de arrancar o frenar máquinas térmicas, decisiones sobre importación o exportación de energía, etc.), y $w_k \in \mathbb{R}^p$ es el ruido del sistema (las lluvias en cada represa).

Veamos este problema un poco más de cerca. Al comienzo del horizonte ($k = 0$), el operador únicamente tiene acceso al estado inicial del sistema x_0 y al vector inicial de ruidos w_0 . Luego de decidir el control a aplicar u_0 , el sistema evoluciona hacia un nuevo estado x_1 siguiendo la dinámica de transición que, en su forma más genérica es

$$x_{k+1} = f_k(x_k, u_k, w_k). \quad (2.2)$$

En términos generales, en el paso k se decide u_k teniendo acceso únicamente a x_k y w_k . Diremos entonces que se trabaja *a ruido visto*: el operador tiene pleno conocimiento de todo el ruido del paso al comienzo del intervalo. Esto es válido en el entendido de que se tiene un buen pronóstico de las variables aleatorias involucradas: los aportes en la represa para toda la semana, la demanda, etc. El control a elegir debe contemplar por un lado el costo instantáneo de la etapa actual $g_k(x_k, u_k, w_k)$ y por otro lado el costo *esperado* a futuro, sin conocimiento previo de los ruidos futuros w_{k+1}, \dots, w_{K-1} . Amen de parecer un poco engorroso, dado un estado inicial x el objetivo es obtener la secuencia de controles de forma de minimizar la siguiente medida de costo

$$\mathbb{E}_0 \left[\min_{u_0} g_0(x_0, u_0, w_0) + \mathbb{E}_1 \left[\min_{u_1} g_1(x_1, u_1, w_1) + \mathbb{E}_2 [\dots] \right] \mid x_0 = x \right] \quad (2.3)$$

s. to:

$$\begin{aligned} x_{k+1} &= f_k(x_k, u_k, w_k) \quad \forall k \\ u_k &\in \mathcal{U}_k(x_k, w_k) \quad \forall k \end{aligned}$$

donde \mathbb{E}_k denota la esperanza respecto al ruido w_k . Nótese que estas esperanzas están anidadas, esto se debe a que en el paso k los valores de x_k y w_k están fijos, y el control debe decidirse minimizando el costo de la etapa y el *costo esperado* de los tiempos futuros, donde se toman las esperanzas con respecto a los ruidos futuros [21]. En cada paso el vector de control está restringido al conjunto $\mathcal{U}_k(x_k, w_k)$, que representa restricciones de caja, tales como que los turbinados y vertidos en cada represa son no negativos y acotados por arriba, la generación térmica es no negativa. Detallaremos estas restricciones más adelante.

2.1. Introducción

Cuando el sistema es lineal y los costos son cuadráticos, la solución a (2.3) es un controlador lineal y se obtiene a través de la Ecuación de Ricatti [22]. Este tipo de controladores LQR (regulador lineal-cuadrático, por sus siglas en inglés) trabajan con la hipótesis de que los ruidos son Gaussianos de media nula y sin restricciones [22]. Como no es nuestro caso, nos enfocaremos en otras técnicas.

El problema anidado (2.3) admite una recursión muy similar a la ecuación de Bellman [14, p.23]

$$V_k(x_k) = \mathbb{E} \left[\min_{u_k \in \mathcal{U}_k(x_k, w_k)} \{g_k(x_k, u_k, w_k) + V_{k+1}(f_k(x_k, u_k, w_k))\} \right] \quad (2.4)$$

donde V_k es la función de valor del paso k (*cost-to-go*), que representa el costo esperado de operar el sistema desde un estado x_k , comenzando en la semana k y yendo hasta el final del horizonte. Cabe destacar que el orden en que se realizan el mínimo y el valor esperado en (2.4) es al revés que en la formulación clásica [23]. La diferencia radica en que en este caso trabajamos a *ruido visto*, mientras que en la formulación clásica el control se decide únicamente en función de x_k , y el ruido se revela después.

Para resolver (2.4) en el paso k es menester conocer las funciones de valor del paso $k+1$. Para obtener todas las funciones de valor lo usual es realizar una *pasada hacia atrás*, comenzando en la última etapa y decrementando hasta $k=0$, resolviendo (2.4) sucesivamente para cada posible estado x_k . Esto se muestra en el Algoritmo 1. La utilidad de la Ecuación de Bellman radica en que un problema en general complejo como (2.3) puede descomponerse en subproblemas como (2.4). Luego, el costo esperado en el problema de horizonte completo (2.3) coincidirá con evaluar la función de costo de la etapa 0 para el vector $x_0 = x$, es decir $V_0(x)$.

Datos: Funciones de costo g_k ; conjunto de estados \mathcal{X} ; conjuntos de ruidos \mathcal{W}_k

Resultado: Funciones de valor $V_k(\cdot) \forall k$

Inicializar $V_K(\cdot) = 0$

for $k = K - 1, \dots, 0$ **do**

for $x \in \mathcal{X}$ **do**

for $w \in \mathcal{W}_k$ **do**

$\hat{\beta}_k(x, w) = \min_{u \in \mathcal{U}_k(x, w)} \{g_k(x, u, w) + V_{k+1}(f_k(x, u, w))\}$

end

$V_k(x) = \frac{1}{|\mathcal{W}_k|} \sum_{w \in \mathcal{W}_k} \hat{\beta}_k(x, w)$

end

end

Algoritmo 1: Programación dinámica discreta clásica

Resolver (2.4) de forma exacta es, en general, extremadamente complejo. En primer lugar, muy pocas veces se conoce la estadística del ruido w_k por lo que se

Capítulo 2. Programación Dinámica Aproximada Cuadrática para el despacho hidrotérmico

Represas	Estados posibles
1	400
2	160×10^3
3	64×10^6
4	25×10^9
5	10×10^{12}

Tabla 2.1: Ilustración de la Maldición de la Dimensionalidad: cantidad de valores que puede tomar la variable de estados en función del número de represas, usando $m = 20$ discretizaciones por componente del vector de estados.

vuelve imposible tomar el valor esperado. Solucionaremos esto considerando distintas realizaciones de ruido y promediando los valores obtenidos [24]. En segundo lugar, el paso de minimización no es trivial. En efecto, el enfoque clásico combinatorio consiste en discretizar el espacio de estados y el espacio de los controles e ir resolviendo secuencialmente (2.4) de manera exhaustiva, para cada estado y para cada valor posible de ruido. Esto escala muy mal cuando la dimensión de los espacios (ya sean de estados o controles) es grande, o cuando se quiere mejorar la precisión introduciendo una discretización más fina. Este inconveniente, comúnmente apodado la *Maldición de la Dimensionalidad* [18] ha catalizado el surgimiento de técnicas aproximadas para resolver (2.4) [18], [25], [15, Cap.6].

El siguiente ejemplo [25] sirve para ilustrar la Maldición de la Dimensionalidad. Admitamos como estado expandido del sistema al nivel de las represas y los aportes, o sea $s_t = [x_t, w_t]$. Si el sistema tiene n represas y se elige discretizar cada componente del vector de estados en m intervalos, entonces habrá m^{2n} posibles estados en cada etapa del problema. Suponiendo $m = 20$ intervalos de discretización, la Tabla 2.1 refleja cómo crece exponencialmente la cantidad de estados posibles cuando el número de represas aumenta.

2.2. Modelado: el caso uruguayo

El caso de estudio para este trabajo será el sistema hidroeléctrico uruguayo. Uruguay cuenta con cuatro represas, tres de ellas ubicadas en la cuenca del Río Negro —interconectadas en cascada— y una en Salto Grande, como se evidencia en la figura 2.1. En la Tabla 2.2 y la Figura 2.2 se resumen y visualizan los embalses y la capacidad de turbinado de cada una de ellas. Ordenando las represas como Bonete, Baygorria, Palmar, Salto, obtenemos un vector $x_k \in \mathbb{R}^4$. El vector de controles $u_k = [r_k^\top, s_k^\top, t_k]^\top \in \mathbb{R}_+^9$ recolecta el turbinado ($r_k \in \mathbb{R}_+^4$) y vertido ($s_k \in \mathbb{R}_+^4$) en cada represa y la generación térmica total ($t_k \in \mathbb{R}_+$). La dinámica de transición de estados viene dada por

$$x_{k+1} = f(x_k, u_k, w_k) = x_k + B(r_k + s_k) + w_k \quad (2.5)$$

donde B es la matriz que interconecta espacialmente las represas

2.2. Modelado: el caso uruguayo



Figura 2.1: Represas de Uruguay. Salto Grande en el Norte, y las represas de Río Negro, de este a oeste Bonete, Baygorria, Palmar. Bonete (marcada en rojo) contiene el embalse más grande, y por ende la mayor capacidad de reserva.

Represa	Embalse (hm^3)	Turbinado ($hm^3/semana$)
Bonete	8200	411
Baygorria	348	500
Palmar	1300	830
Salto Grande	1900	2510

Tabla 2.2: Represas de Uruguay con su embalse nominal y capacidad de turbinado [26]. $1hm^3 \sim 1$ hectómetro cúbico.

$$B = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \quad (2.6)$$

y w_k es el vector de aportes para cada represa, es decir, las lluvias en hm^3 . Las tres primeras columnas de la matriz B modelan el flujo de agua en la cuenca del Río Negro. Para la primera columna, por ejemplo, lo que se turбина y vierte en Bonete en el paso k entra a Baygorria en el paso $k + 1$.

Emplearemos pasos semanales y un horizonte de planeamiento anual $k = 0, 1, \dots, K - 1$ con $K = 52$ semanas en un año aproximadamente.

- El costo de la etapa es el costo de generación térmica. Tomaremos la hipótesis de que es lineal

$$g_k(x_k, u_k, w_k) = g(u_k) = c \cdot t_k \quad (2.7)$$

con $c = 200USD/MWh$.

Capítulo 2. Programación Dinámica Aproximada Cuadrática para el despacho hidrotérmico

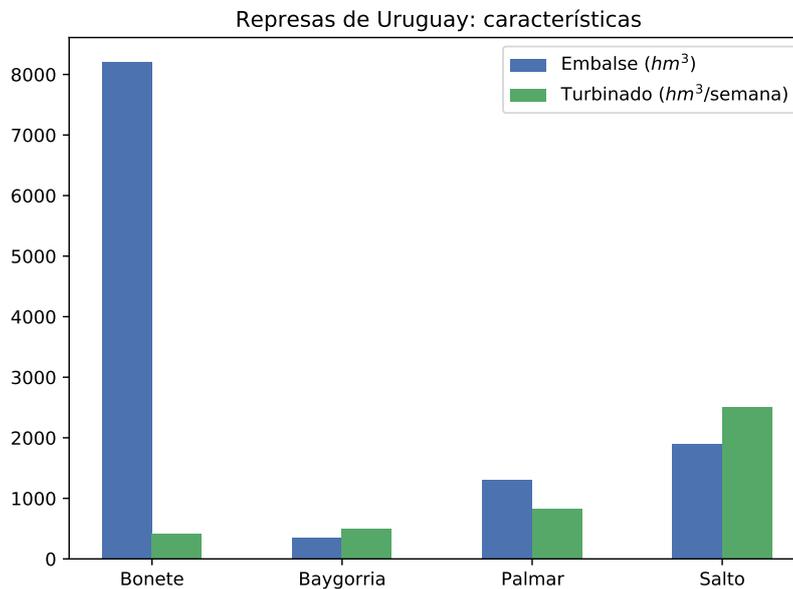


Figura 2.2: Visualización del embalse y capacidad de turbinado para las represas de Uruguay. Notar cómo la principal capacidad de almacenamiento se da en Rincón del Bonete.

- La generación de cada represa es lineal con el volumen de agua turbinada, a través de un coeficiente energético único a la represa.
- Las restricciones a cumplir en cada paso son:
 - La dinámica del sistema (2.5)
 - Balance entre generación y demanda
 - Restricciones de caja para los turbinados $0 \leq r_k \leq r_{max}$
 - Positividad de los vertidos $0 \leq s_k$
 - Restricciones de caja para el estado siguiente $0 \leq x_{k+1} \leq x_{max}$
 - Positividad de la generación térmica $0 \leq t_k$

Como señalábamos en 2.1, todas las restricciones exceptuando la dinámica del sistema serán recolectadas como una restricción sobre el control $u_k \in \mathcal{U}_k(x_k, w_k)$

2.2.1. Aportes

Contamos con datos de aportes semanales en las tres represas de mayor embalse —Bonete, Palmar, Salto. Estos datos fueron recolectados desde 1909 a 2013, por lo que se cuenta con 105 años (o “crónicas”). En la figura 2.3 mostramos ejemplos de estas crónicas.

2.2. Modelado: el caso uruguayo

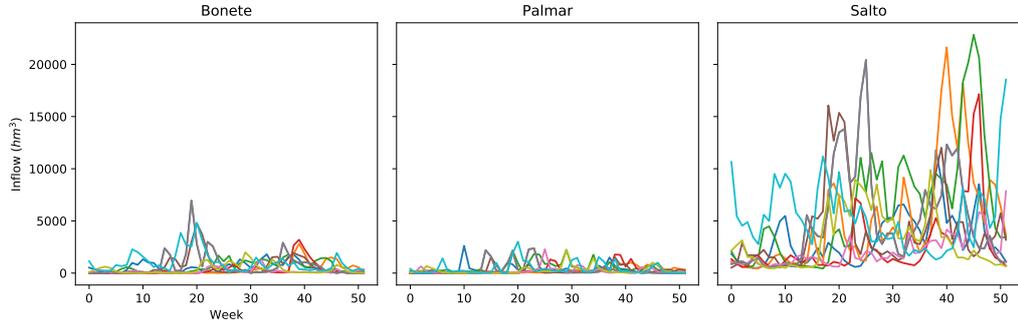


Figura 2.3: Ejemplo de aportes históricos (“crónicas”) en las tres principales represas de Uruguay.

2.2.2. Incorporación de memoria al estado hidrológico

Los aportes se encuentran correlacionados, tanto temporalmente —si una semana llueve mucho, es probable que la siguiente también— como espacialmente —si dos represas se encuentran próximas entre sí, es de esperar que los aportes sean similares. Una forma posible de capturar la correlación temporal inherente al sistema es incorporar una *variable de estado hidrológica* e_k , que resuma las condiciones hidrológicas del sistema, condicionando la distribución de los aportes. Esta variable puede, por ejemplo, tomar valores discretos y obedecer una dinámica markoviana

$$\mathcal{P}_{ee'}^{(k)} = P(e_{k+1} = e' \mid e_k = e). \quad (2.8)$$

La idea es separar el estado hidrológico e del nivel de los lagos x y resolver el valor esperado de la ecuación de Bellman en dos pasos. Luego de decidir discretizar el estado hidrológico en cinco valores posibles, podemos calcular una función de valor distinta $V_{k,e}(x)$ para cada e posible. Luego, fijada la etapa k y fijado $e_k = e$, el costo futuro puede estimarse como un promedio ponderado de las funciones de valor del paso siguiente, donde los ponderadores son la probabilidad de transicionar a cada estado hidrológico. Esto se ilustra en la siguiente recursión de Bellman generalizada.

$$V_{k,e}(x) = \mathbb{E}_{w|e} \left[\min_{u \in \mathcal{U}_k(x,w)} \left\{ g_k(x, u, w) + \sum_{e'} \mathcal{P}_{ee'} \cdot V_{k+1,e'}(f_k(x, u, w)) \right\} \right] \quad (2.9)$$

La diferencia con (2.4) radica en que el costo futuro es una combinación lineal de las funciones de valor del paso y estado hidrológico siguiente, y que el valor esperado de los ruidos se toma condicionado al estado hidrológico actual.

En los modelos usados en el sistema eléctrico uruguayo, cuando se toma una única variable de estado hidrológica, se suelen usar cinco estados discretos para ella, indexados de 0 a 4, con 0 correspondiente a un estado “muy seco” y 4 correspondiente a “muy húmedo” [27]. En uno de estos modelos se define una magnitud

Capítulo 2. Programación Dinámica Aproximada Cuadrática para el despacho hidrotérmico

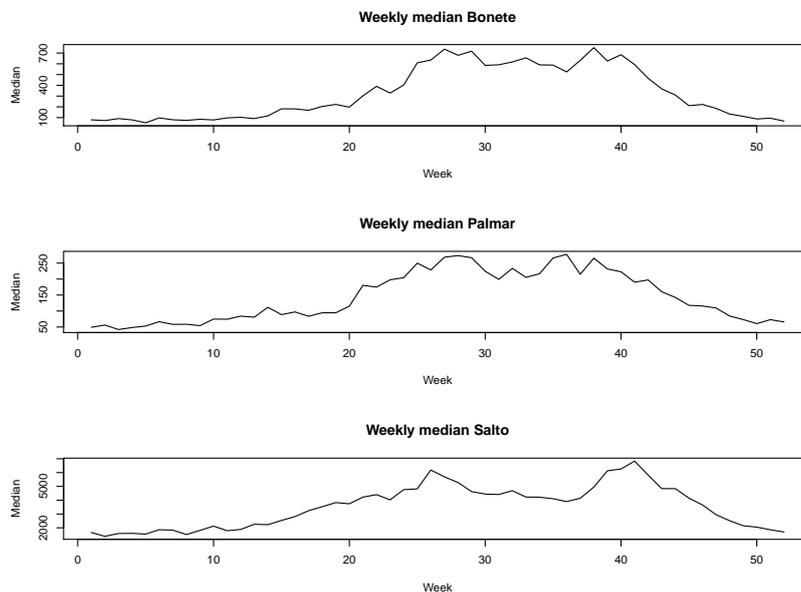


Figura 2.4: Mediana de aportes semanales (en hm^3) para cada una de las tres represas. Para cada semana la mediana se computa a lo largo de los 105 años de crónicas (1909–2013).

proporcional a la energía entrante al sistema en cada semana multiplicando el caudal incremental a cada central por un coeficiente energético que aproxima la cantidad de energía obtenible si se turбина un m^3 de aporte, en la central y en las que están agua abajo de ella. Durante el resto de este trabajo nos referiremos a este modelo como *Modelo de Energía Entrante*, y lo abreviaremos como MEE.

Como elección de diseño, y en concordancia con lo realizado por los actores del sistema eléctrico nacional, en el trabajo [16] realizamos un agrupamiento de los aportes respetando los cinco valores de e_k , con transiciones semanales a través de una cadena de Markov homogénea —es decir, una sola para todo el año. Luego de entrenar políticas con este modelo, realizamos corridas usando las series históricas y obtuvimos en la práctica mejores resultados que con el MEE. Luego ahondaremos en eso.

2.2.3. Estimación del modelo Markoviano¹

Como mencionamos anteriormente, contamos con datos de aportes w_k en las tres represas de mayor porte (Bonete, Palmar, Salto) para cada una de las 52 semanas del año, para 105 años. Deseamos incorporar la variable hidrológica e_k a nuestro modelo, y a partir de los datos estimaremos la dinámica markoviana de la misma. Como elección de diseño, e_k podrá tomar cinco valores discretos (de cero

¹Si bien participé en la discusión de los métodos, el trabajo que se detalla en esta subsección fue realizado principalmente por Camila Martínez y Andrés Ferragut, coautores de [16]. Se incluye resumidamente en esta tesis por completitud, ya que es utilizado en secciones posteriores.

2.2. Modelado: el caso uruguayo

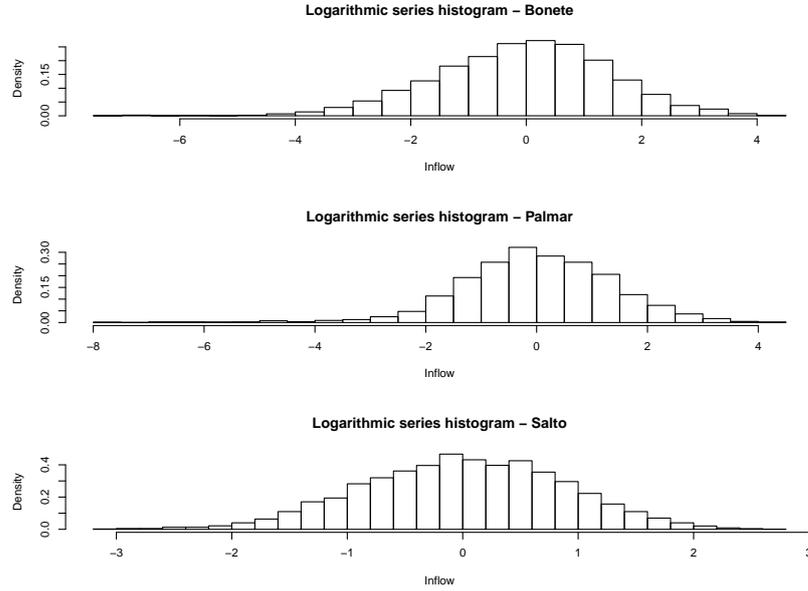


Figura 2.5: Histograma con los aportes de las series normalizadas. Cada serie se divide por su mediana semanal (Figura 2.4) y luego se le aplica una transformación logarítmica.

a cuatro) y la dinámica de transición será invariante en el tiempo, es decir, estará completamente determinada por la matriz $\mathcal{P} \in \mathbb{R}^{5 \times 5}$ de entradas

$$\mathcal{P}_{ee'} = P(e_{k+1} = e' \mid e_k = e) \quad (2.10)$$

La diferencia de (2.8) con (2.10) es que la primera depende del tiempo. Con nuestro enfoque tenemos que ajustar 20 parámetros ($\mathcal{P} \in \mathbb{R}^{5 \times 5}$ y las filas suman 1) usando $105 \times 52 = 5460$ aportes $\in \mathbb{R}^3$. En cambio, el MEE (que, como ya explicamos, usa una matriz para cada semana del año), estima los 20 parámetros de cada matriz usando 105 aportes. Esperamos con este enfoque reducir el *overfitting* a los datos.

Para agrupar los aportes procedemos de la siguiente forma. Cada serie anual se divide por su mediana semanal, de forma de remover las variaciones estacionales. Luego se toma el logaritmo de las series normalizadas. Luego de estos pasos, las series son aproximadamente Gaussianas. En las figuras 2.4 y 2.5 mostramos la mediana de los aportes y la distribución luego de las transformaciones.

Para definir los grupos o *clusters* a los que corresponde cada aporte empleamos la técnica de Análisis por Componente Principal (*Principal Component Analysis*, o PCA) [28, 29]. En resumidas cuentas, se busca el vector de características \mathbf{y} a través del cual se maximiza la variabilidad de los datos. Si llamamos \mathbf{W} a la matriz cuyas filas contienen a los datos log-normalizados (de dimensión $(105 \times 52) \times 3$, tantas filas como aportes semanales y tres columnas por las represas de mayor aporte), basta con resolver

$$\mathbf{y} = \arg \max_{\mathbf{y}} \frac{\mathbf{y}^\top \mathbf{W}^\top \mathbf{W} \mathbf{y}}{\mathbf{y}^\top \mathbf{y}} \quad (2.11)$$

Capítulo 2. Programación Dinámica Aproximada Cuadrática para el despacho hidrotérmico

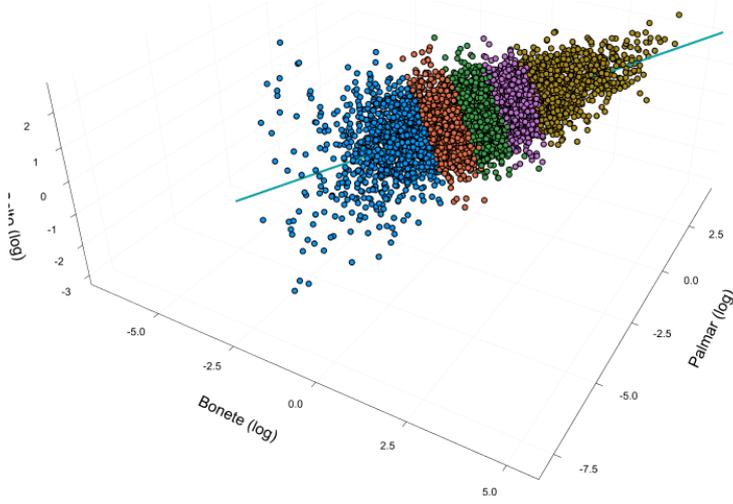


Figura 2.6: Clusters resultantes al aplicar PCA. Los clusters están asociados a la hidráulica del estado hidrológico: el cluster azul tiene aportes pequeños y se corresponde con un estado *muy seco*, mientras que el cluster amarillo tiene aportes grandes y se corresponde con un estado *muy húmedo*

donde $\mathbf{W}^T \mathbf{W}$ es la covarianza de los datos. El objetivo de (2.11) se conoce como el cociente de Rayleigh, y es sabido que su solución coincide con descomponer espectralmente la matriz de correlación y tomar el vector propio asociado al valor propio más grande [30]. Una vez hecho esto, cada aporte es proyectado al eje dado por este vector. Luego de hacer esta proyección, elegimos 5 clusters y asignamos cada punto (vector de aporte semanal) a uno de los clusters, de forma tal que cada agrupamiento contenga aproximadamente 20% de los datos. Esto se ilustra en la Figura 2.6. Estos agrupamientos tienen significado físico: el cluster 0 se corresponde con aportes pequeños (y por ende un estado hidrológico *muy seco*), mientras que en el otro extremo el cluster 4 posee aportes grandes (y estado hidrológico *muy húmedo*).

Finalmente, para estimar la matriz de transición \mathcal{P} empleamos un enfoque frecuentista, contando el número de transiciones entre dos clusters. En particular, la probabilidad \mathcal{P}_{ij} de transicionar entre el cluster i y el j se calcula como

$$\mathcal{P}_{ij} = \frac{\sum_t \mathbb{1}_{\{e_{t-1}=i, e_t=j\}}}{\sum_t \mathbb{1}_{\{e_t=i\}}} \quad (2.12)$$

donde e_t representa el estado a tiempo t , y la suma se realiza a lo largo de todo el set de datos.

De aquí en más los datos de aportes están clasificados por semana y por cluster. Para una semana k y estado hidrológico e tenemos el conjunto de aportes $\mathcal{W}_{k,e}$. Muestrearemos de estos conjuntos al resolver versiones aproximadas de (2.9). Ahondaremos en ello en la Sección 2.4.1.

2.3. Aproximaciones cuadráticas

Como ya hemos comentado en la Sección 2.1, (2.9) no se puede resolver en forma cerrada por la presencia de restricciones y el desconocimiento de la estadística del ruido. La solución que emplearemos consiste en aproximar cada función de valor $V_{k,e}$ por una función $\tilde{V}_{k,e}$ cuadrática en el vector de estados x . El algoritmo, que presentaremos más adelante, consta en cada paso de dos etapas: una *etapa de muestreo* y una *etapa de ajuste*. En la primera etapa se consiguen parejas estado-costo resolviendo una versión aproximada de (2.9). En la segunda etapa se ajusta una función cuadrática a los datos obtenidos. Comenzaremos la sección comentando la motivación detrás de esta elección, para luego pasar a presentar más en detalle el algoritmo desarrollado.

2.3.1. Motivación

La motivación para este planteo surgió de observar datos de corridas de la División de Planificación del Abastecimiento de UTE. Nos fueron provistos datos de corridas de paso semanal y horizonte de varios años, donde el costo del despacho térmico se modela como lineal. En estas simulaciones los niveles de los lagos se discretizan uniformemente en cinco o diez valores posibles. En la Figura 2.7 se aprecia el *cost-to-go* en función del nivel de Bonete, y en la Figura 2.8 el *cost-to-go* en función de dos de los cuatro lagos. De apreciar estas figuras, decidimos tomar un modelo cuadrático definido positivo para el costo.

El hecho de contar con funciones cuadráticas definidas positivas presenta la ventaja de que vuelve tratable a la Ecuación de Bellman. En efecto, la *etapa de muestreo* consistirá en resolver varios programas cuadráticos con restricciones lineales [31, p.152], mientras que en la *etapa de ajuste* cada función aproximada $\tilde{V}_{k,e}$ se obtendrá resolviendo un programa semidefinido positivo [32].

2.4. Algoritmia

2.4.1. Pasada hacia atrás

Como mencionamos en la sección anterior, para cada paso k y estado hidrológico e el algoritmo se subdivide en dos etapas: una *etapa de muestreo* y una *etapa de ajuste*.

Etapas de muestreo En la primera etapa se consiguen parejas estado-costos resolviendo una versión aproximada de (2.9). El espacio de estados se discretiza (de manera uniforme, por ejemplo), y para cada posible punto se plantea el siguiente problema de optimización para distintos escenarios w_i .

$$\hat{\beta}_{k,e}(x, w_i) = \min_{u \in \mathcal{U}_k(x, w_i)} \left\{ g_k(x, u, w_i) + \sum_{e'} \mathcal{P}_{ee'} \cdot \tilde{V}_{k+1, e'}(f_k(x, u, w_i)) \right\} \quad (2.13)$$

Capítulo 2. Programación Dinámica Aproximada Cuadrática para el despacho hidrotérmico

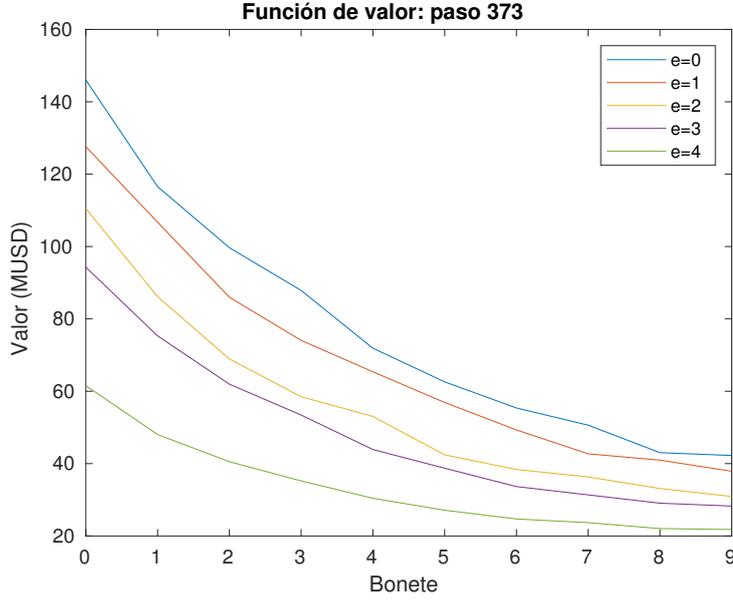


Figura 2.7: Funciones de valor (*cost-to-go*) en función del nivel de agua de Bonete para cada uno de los posibles estados hidrológicos. Notar cómo el costo esperado es más alto para estados hidrológicos más secos y decrece a medida que la reserva aumenta.

donde destacamos que para el paso k se emplean las aproximaciones cuadráticas del paso $k + 1$. Los ruidos se sortean con reposición de los conjuntos $\mathcal{W}_{k,e}$ obtenidos con la clusterización descrita en la Sección 2.2.3. Repitiendo (2.13) para M distintos ruidos w_i y promediando se obtiene una pareja estado-costo $(x, \beta(x))$.

$$\beta_{k,e}(x) = \frac{1}{M} \sum_{i=1}^M \hat{\beta}_{k,e}(x, w_i) \quad (2.14)$$

Cabe destacar que la serie de problemas (2.13) debe repetirse para cada punto de la grilla de estados x y para cada posible estado hidrológico e . El problema de optimización (2.13) es un programa cuadrático con restricciones lineales [31, p.152]. El objetivo es cuadrático, ya que consta de la suma de un término lineal (el costo de la etapa) y una combinación convexa de términos cuadráticos (el valor esperado del paso siguiente).

Etapas de ajuste Luego de contar con varias parejas se procede a la etapa de ajuste, donde se ajusta una función cuadrática positiva a dichos puntos. Podemos expresar la función de valor aproximada como

$$\tilde{V}_{k,e}(x) = x^\top P_{k,e} x + q_{k,e}^\top x + r_{k,e} \quad (2.15)$$

donde $r_{k,e} \in \mathbb{R}$, $q_{k,e} \in \mathbb{R}^n$, $P_{k,e} \in \mathbb{R}^{n \times n}$ y $P_{k,e} \succeq 0$.

Lo que sigue a continuación se repite para cada etapa k y para cada estado hidrológico e . Omitimos los subíndices k, e para reducir notación. Para ajustar

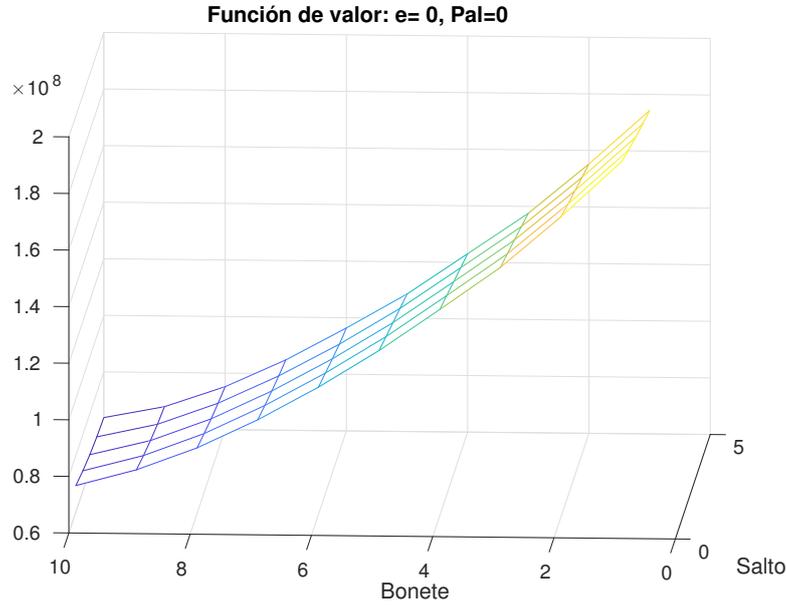


Figura 2.8: *Cost-to-go* en función del nivel de Bonete y Salto, para Palmar vacío y el estado hidrológico más seco. Notar cómo la mayor variabilidad en el costo se da en la dirección de Bonete.

estos parámetros empleamos el criterio de minimización del error cuadrático medio entre la función a aproximar y los datos. Como resultado de la *etapa de muestreo* explicada anteriormente, tenemos N parejas estado-costo de la forma (x^s, β^s) con $s = 1, \dots, N$. El problema a resolver es

$$\begin{aligned} \min_{P,q,r} \sum_{s=1}^N \left(x^{s\top} P x^s + q^\top x^s + r - \beta^s \right)^2. \quad (2.16) \\ \text{s. t.: } P \succeq 0 \end{aligned}$$

Para llevar a una formulación aceptada por un solver de optimización, podemos escribir (2.16) equivalentemente como

$$\min_{P,q,r,t} \|t\|_2^2 \quad (2.17)$$

$$\text{s. t.: } P \succeq 0$$

$$x^{s\top} P x^s + q^\top x^s + r - \beta^s = t_s \quad s = 1, \dots, N \quad (2.18)$$

donde introducimos $t \in \mathbb{R}^N$, y t_s es la s -ésima entrada del vector. (2.17) resulta ser un programa semidefinido positivo [32]. Resumimos el enfoque que acabamos de presentar en el Algoritmo 2. De igual forma que el algoritmo de programación dinámica clásica, éste trabaja realizando una pasada hacia atrás. Estando en el paso k , se fija un estado hidrológico e y se obtiene la cuadrática aproximada $\tilde{V}_{k,e}$, siguiendo las etapas de muestreo y ajuste ya mencionadas. Esto se repite

```

Result: Funciones de valor aproximadas  $\tilde{V}_{k,e} \forall k, e$ 
Data: Grilla de estados  $\mathcal{X} = \{x^1, \dots, x^N\}$ ; sets de aportes  $\mathcal{W}_{k,e} \forall k, e$ 
Inicializar  $V_{K,e}(\cdot) = 0 \quad \forall e$ 
for  $k = K - 1, \dots, 0$  do
  for  $e = 0, \dots, 4$  do
    /* Etapa de muestreo */
    for  $s = 1, \dots, N$  do
      Fijar estado  $x^s$ 
      for  $i = 1, \dots, M$  do
        Muestrear escenario de aportes  $w_i \sim \text{Unif}(\mathcal{W}_{k,e})$ 
         $\hat{\beta}_{k,e}(x^s, w_i) =$ 
         $\min_u \left\{ g_k(x^s, u, w_i) + \sum_{e'} \mathcal{P}_{ee'} \cdot \tilde{V}_{k+1,e'}(x') \right\}$ 
       $\beta_{k,e}(x^s) = \frac{1}{M} \sum_{i=1}^M \hat{\beta}_{k,e}(x^s, w_i)$ 
    /* Etapa de ajuste */
     $\{P_{k,e}, q_{k,e}, r_{k,e}\} =$ 
     $\operatorname{argmin}_{P \succeq 0, q, r} \frac{1}{N} \sum_{s=1}^N (x^{s\top} P x^s + q^\top x^s + r - \beta_{k,e}^s)^2$ 

```

Algoritmo 2: Programación dinámica aproximada cuadrática (PDAC)

para todos los estados hidrológicos, para luego dar un paso hacia atrás. El costo computacional del algoritmo consiste en resolver $N \times M$ programas cuadráticos y un programa semidefinido para cada paso k y estado hidrológico e posible. A pesar de la estructura secuencial del algoritmo, se puede optimizar el tiempo de cómputo paralelizando la etapa de muestreo (la resolución de (2.13)).

2.4.2. Pasada hacia adelante

De las cuadráticas obtenidas se hereda de forma natural una política de operación para el sistema. Esta política $\pi = \{\mu_0, \dots, \mu_{K-1}\}$ elige a tiempo k el vector de control $u_k = \mu_k(x_k, e_k, w_k)$ para estados y ruidos dados de forma implícita, resolviendo el paso de Bellman

$$u_k = \mu_k(x_k, e_k, w_k) = \operatorname{argmin}_{u \in \mathcal{U}_k(x_k, w_k)} \left\{ g_k(x_k, u, w_k) + \sum_{e'} \mathcal{P}_{e_k e'} \cdot \tilde{V}_{k+1,e'}(x_{k+1}) \right\} \quad (2.19)$$

$$x_{k+1} = f(x_k, u, w_k)$$

que es nuevamente un programa cuadrático. Es deseable que esta política logre nuestro objetivo original, minimizar el costo de operación del sistema para estados iniciales x y e dados:

Result: costos de operación $costo(x, e)$ para $x \in \mathcal{X}_0$
Data: Funciones de valor $\tilde{V}_{k,e} \forall k, e$
 Conjunto de estados iniciales \mathcal{X}_0
for $x_0 \in \mathcal{X}_0$ **do**
 for $e_0 \in \{0, \dots, 4\}$ **do**
 Sortear estados hidrológicos $\mathbf{e} = [e_0, \dots, e_{K-1}]$ según (2.10)
 Sortear aportes $\mathbf{w} = [w_0, \dots, w_{K-1}]$
 for $i = 1, \dots, Q$ **do**
 for $k = 0, \dots, K - 1$ **do**
 Aplicar $u_k = \mu_k(x_k, w_k)$ resolviendo (2.19)
 $C_Q \leftarrow C_Q + g_k(x_k, u_k, w_k)$
 $costo(x_0, e_0) \simeq \frac{1}{Q} \sum_{i=1}^Q C_Q$

Algoritmo 3: PDAC: pasada hacia adelante

$$costo\ total(x, e) = \mathbb{E} \left[\sum_{k=0}^{K-1} g_k(x_k, u_k, w_k) \mid x_0 = x, e_0 = e \right] \quad (2.20)$$

Si las aproximaciones cuadráticas son buenas, este costo debería ser muy similar al predicho por la función de valor aproximada a tiempo cero $\tilde{V}_{0,e}$. Nuestros experimentos (como veremos más adelante) marcan que esta cuadrática generalmente subestima el costo total (2.20). Para predecir de forma más certera el costo de operación bajo la política aprendida realizamos en paralelo *pasadas hacia adelante*. Para ello comenzamos sorteando una trayectoria de estados hidrológicos $\mathbf{e} = [e_0, e_1, \dots, e_{K-1}]$ comenzando desde $e_0 = e$ y usando la dinámica de (2.10). Similarmente sorteamos una crónicas de aportes $\mathbf{w} = [w_0, \dots, w_{K-1}]$ donde el aporte a tiempo k se elije de $w_k \in \mathcal{W}_{k,e_k}$. Luego, comenzando desde $k = 0$ el algoritmo resuelve secuencialmente (2.19), recolectando en cada paso el costo de la etapa $g_k(x, u, w)$ y transicionando hacia el nuevo estado x' . Esto se repite hasta el final del horizonte, con lo que se obtiene una muestra del costo de operación. Esto puede repetirse para Q trayectorias distintas de estado hidrológico y aportes. Luego, promediando los costos se obtiene una buena aproximación de la esperanza en (2.20). Resumimos este procedimiento en el Algoritmo 3.

El Algoritmo puede modificarse ligeramente para conseguir el *costo total* únicamente como función de los lagos. Esto puede lograrse sorteando equiprobablemente el estado hidrológico inicial e_0 de cada trayectoria. En este caso el costo para un estado x_0 se puede comparar, por ejemplo, con el promedio de los valores funcionales de las cuadráticas $\frac{1}{|\mathcal{E}|} \sum_{e \in \mathcal{E}} \tilde{V}_{0,e}(x_0)$.

Contendientes para la política

Interesa conocer qué tan buena es la política de operación aprendida. Para ello, podemos comparar su desempeño —en relación al costo obtenido en (2.20)—

Capítulo 2. Programación Dinámica Aproximada Cuadrática para el despacho hidrotérmico

contra el desempeño de una política *miope*. Esta última opera en cada instante minimizando únicamente el costo instantáneo de la etapa. Esto quiere decir que si hay agua disponible, usará la necesaria en pos de incurrir en costo nulo. A cada paso, entonces, la política miope consigue el control

$$\begin{aligned}\mu_k^{miope}(x_k, w_k) &= \operatorname{argmin}_{u \in \mathcal{U}_k(x_k, w_k)} g_k(x_k, u, w_k) \\ x_{k+1} &= f(x_k, u, w_k)\end{aligned}\quad (2.21)$$

El problema (2.21) resulta ser un programa lineal que en principio podría admitir varias soluciones. En efecto, cuando la reserva disponible es grande, es posible turbinar agua como para cubrir la demanda y además incurrir en vertido (descartar agua). A pesar de ser óptimo para (2.21), vertidos innecesarios provocarán que en el futuro no se cuente con agua, lo cual incrementará el costo total. Para evitar este inconveniente consideramos una política miope más razonable que penaliza el vertido

$$\begin{aligned}\mu_k^{miope}(x_k, w_k) &= \operatorname{argmin}_{u \in \mathcal{U}_k(x_k, w_k)} \left\{ g_k(x_k, u, w_k) + \epsilon u^\top S u \right\} \\ x_{k+1} &= f(x_k, u, w_k)\end{aligned}\quad (2.22)$$

donde $\epsilon > 0$ y la matriz S selecciona los términos de vertido del control, es decir, es una matriz diagonal que contiene unos en las coordenadas correspondientes al vertido y sus demás entradas son nulas.

Además de comparar nuestra política contra la miope, podemos estimar una cota inferior para el desempeño óptimo. Esto lo logramos asumiendo conocidas al principio del problema la trayectoria completa de aportes y estados hidrológicos $\mathbf{w} = [w_0, \dots, w_{K-1}]$ y $\mathbf{e} = [e_0, \dots, e_{K-1}]$. Luego la secuencia de decisiones óptima $\mathbf{u} = [u_0, \dots, u_{K-1}]$ (y el costo asociado) se obtiene resolviendo el problema de K etapas:

$$\begin{aligned}\mathbf{u}^{LB}(x, \mathbf{w}) &= \operatorname{argmin}_{\mathbf{u} \in \mathbf{U}(x, \mathbf{w})} \sum_{k=0}^{K-1} g_k(x_k, u_k, w_k) \\ x_{k+1} &= f_k(x_k, u_k, w_k) \quad k = 0, \dots, K-1 \\ x_0 &= x\end{aligned}\quad (2.23)$$

donde $\mathbf{u} \in \mathbf{U}(x, \mathbf{w})$ indica que $u_k \in \mathcal{U}_k(x_k, w_k)$ para todo k , con los conjuntos $\mathcal{U}_k(x_k, w_k)$ descritos anteriormente. El problema (2.23) es un programa lineal y obtiene la secuencia de decisiones \mathbf{u} de todo el horizonte, sabiendo de antemano los aportes y los estados hidrológicos por los que el sistema va a transitar. Esto se contrasta con nuestro algoritmo, que en el tiempo k solo tiene acceso al aporte w_k y al estado e_k actual.

Intuitivamente, el costo obtenido con (2.23) promediando sobre las trayectorias de ruido \mathbf{w} es menor que (2.20) promediando sobre la probabilidad de los estados hidrológicos iniciales, porque se cuenta con un pronóstico perfecto sobre todos los aportes futuros.

2.4.3. Implementación

Todo el código, desde el modelo hidrotérmico uruguayo hasta los algoritmos fue implementado en `Python`. Entre los módulos importados destacamos `cvxpy` [33, 34], librería usada para resolver problemas de optimización. Este módulo permite comunicación con solvers estándar de problemas de optimización; en nuestro caso empleamos `GUROBI` [35] para resolver los programas cuadráticos de la etapa de muestreo y `MOSEK` [36] para el ajuste cuadrático.

Preprocesamiento de los datos

Durante el trabajo surgieron varios problemas durante la implementación. En particular, la elección de los solvers llevó bastante tiempo. Muchos de los resolvers incluidos por defecto en `cvxpy` fallaban en algunas instancias, mientras que otros convergían en varias ocasiones a soluciones que calificaban como subóptimas.

Afinar la etapa de ajuste fue particularmente trabajosa. Recordemos que esta consiste en ajustar cuadráticas a las parejas estado-costo (x, β) , resolviendo el programa semidefinido positivo (SDP)

$$\min_{P \succeq 0, q, r} \frac{1}{N} \sum_{s=1}^N \left(x^{s\top} P x^s + q^\top x^s + r - \beta^s \right)^2. \quad (2.24)$$

Todos los solvers gratuitos y capaces de resolver SDPs de `cvxpy` fallaban al ser inicializados. Luego de conseguir una licencia estudiantil, comenzamos a usar `MOSEK`. Éste, sin embargo, generalmente marcaba que (2.24) no tenía solución. Luego de inspeccionar los datos, concluimos que esto podía deberse a un problema de escala: mientras el vector de estado (el nivel de los lagos en hm^3) tiene valores del orden de 10^3 , los costos de operación β (en dólares) alcanzan valores entre $10^6 - 10^8$ dependiendo de la etapa. Esta discordancia numérica impedía al solver hallar una solución.

Decidimos entonces reescalar los datos y resolver un problema equivalente a (2.24). De aquí en más, los estados se normalizan respecto a su máximo valor $\bar{x} = \max_{s \in \{1, \dots, N\}} \|x^s\|_\infty$ y a los costos se les resta la media muestral μ_β y se divide este valor entre la desviación muestral σ_β . Con estas modificaciones planteamos el siguiente problema

$$\min_{\tilde{P} \succeq 0, \tilde{q}, \tilde{r}} \frac{1}{N} \sum_{s=1}^N \left(\frac{x^{s\top}}{\bar{x}} \tilde{P} \frac{x^s}{\bar{x}} + \tilde{q}^\top \frac{x^s}{\bar{x}} + \tilde{r} - \frac{\beta^s - \mu_\beta}{\sigma_\beta} \right)^2 \quad (2.25)$$

que pudo ser resuelto sin inconvenientes por `MOSEK`. Resta únicamente encontrar el cambio de variable que transforma la cuadrática dada por $\tilde{P}, \tilde{q}, \tilde{r}$ en la cuadrática original y que garantice la equivalencia entre (2.25) y (2.24). Veremos que esto es sencillo. Multiplicando el objetivo de (2.25) por σ_β^2 y desarrollando obtenemos el problema equivalente

$$\min_{\tilde{P} \succeq 0, \tilde{q}, \tilde{r}} \frac{1}{N} \sum_{s=1}^N \left(x^{s\top} \frac{\sigma_\beta}{\bar{x}^2} \tilde{P} x^s + \frac{\sigma_\beta}{\bar{x}} \tilde{q}^\top x^s + \sigma_\beta \tilde{r} + \mu_\beta - \beta^s \right)^2 \quad (2.26)$$

Capítulo 2. Programación Dinámica Aproximada Cuadrática para el despacho hidrotérmico

Luego es claro que el cambio de variable necesario es

$$\begin{cases} P = \frac{\sigma_\beta}{\bar{x}^2} \tilde{P} \\ q = \frac{\sigma_\beta}{\bar{x}} \tilde{q} \\ r = \sigma_\beta \tilde{r} + \mu_\beta \end{cases} \quad (2.27)$$

2.5. Resultados

En esta sección presentamos algunos resultados obtenidos al emplear los algoritmos desarrollados. Comenzamos con la fase de *entrenamiento*, donde obtenemos las parejas estado-costo y ajustamos cuadráticas. Proseguimos con corridas hacia adelante, simulando el costo de operación del sistema. Finalizamos presentando, quizás los resultados de mayor interés, que constan de comparar el desempeño de nuestra política con otra entrenada usando el modelo de estado hidrológico MEE.

2.5.1. Pasada hacia atrás

Para el muestreo de parejas estado-costo particionamos el espacio de estado en una grilla. Para ello, dividimos el embalse de Bonete en N_0 intervalos equiespaciados y las demás represas en tres intervalos. El costo $\beta_{k,e}$ para cada uno de estos puntos se obtuvo promediando (2.13) para $M = 10$ escenarios de aportes distintos. En la figura 2.9 se aprecia un corte de la sábana cuadrática obtenida, para la semana 40 y estado hidrológico $e_{40} = 1$. En esa corrida discretizamos Bonete en diez valores ($N_0 = 10$). Observamos que el costo del sistema parece depender principalmente de Bonete.

Presentamos una figura con los resultados de otra corrida en la Figura 2.10. En ella observamos una sábana cuadrática para la semana 48 con $N_0 = 5$. Notar cómo, en este caso, el nivel del embalse de la otra represa (Salto) influye más en el costo (cf. Figura 2.9). Esto se debe principalmente a que al estar más cerca del horizonte, tener un segundo lago lleno disminuye apreciablemente el *cost-to-go*.

2.5.2. Pasada hacia adelante

Podemos comparar el costo de operación predicho por la función de valor $V_{0,e_0}(x_0)$ con el costo simulado *hacia adelante* $\sum_{k=0}^{K-1} g_k$, comenzando desde x_0 y e_0 y siguiendo la política aprendida (ver Sección 2.4.2). En la Figura 2.11 se contrastan el costo predicho y simulado en función del embalse inicial de Bonete, arrancando desde un estado hidrológico intermedio ($e_0 = 2$). El costo simulado se obtiene corriendo 105 trayectorias distintas y promediando los resultados obtenidos. Además se presenta una cota inferior para el costo óptimo, que se obtiene resolviendo el programa lineal de K etapas descrito en (2.23). Nuestros experimentos muestran que las predicciones $\tilde{V}_{0,e}$ son optimistas. Observar en todos los casos cómo el costo decrece a medida que la capacidad de reserva aumenta.

2.5. Resultados

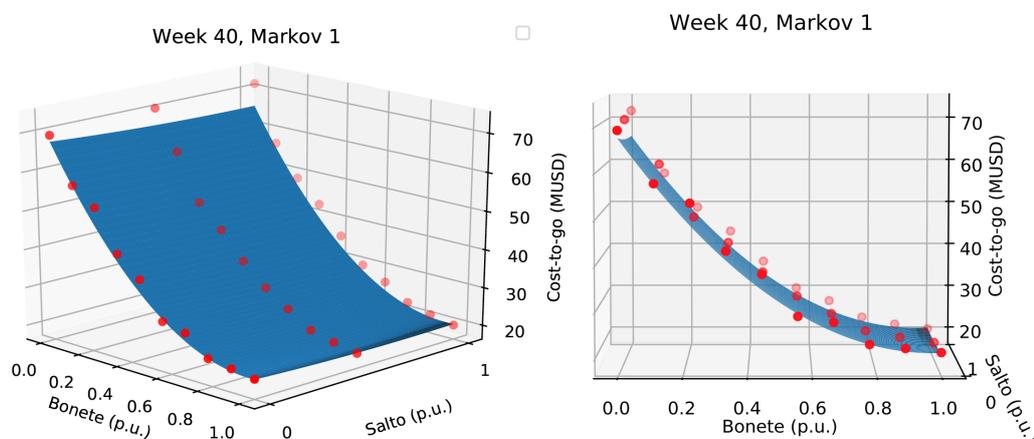


Figura 2.9: Ajuste cuadrático para la semana 40 y estado hidrológico seco ($e_{40} = 1$). La grilla de estados contiene $N_0 = 10$ para Bonete y tres puntos para las otras tres represas. En rojo: parejas estado-costo (muestreadas con (2.13)–(2.14)). En azul: sábana cuadrática (obtenida usando (2.16)). El *cost-to-go* es menor cuanto más reserva haya y depende principalmente de Bonete (la represa con mayor embalse).

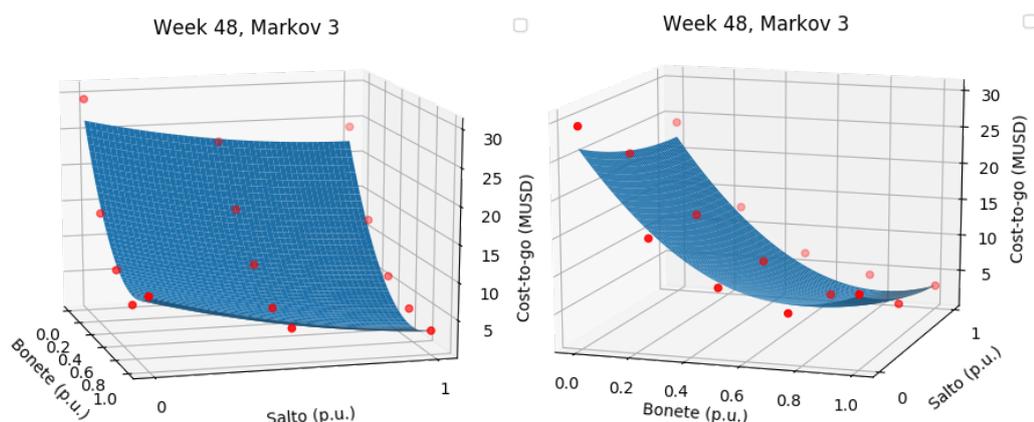


Figura 2.10: Ajuste cuadrático para la semana 48 y estado hidrológico húmedo ($e_{48} = 3$). La grilla de estados contiene $N_0 = 5$ puntos para Bonete y tres puntos para las otras tres represas. En rojo: parejas estado-costo (muestreadas con (2.13)–(2.14)). En azul: sábana cuadrática (obtenida usando (2.16)). El *cost-to-go* depende principalmente de Bonete. Sin embargo, en este caso la variabilidad en Salto también afecta el costo (comparar con la Figura 2.9). Esto se debe a que se está cerca del fin del horizonte.

La política entrenada generalmente obtiene un mejor desempeño que la política miope (ver (2.22)), en particular cuando se cuenta con almacenamiento inicial, tal como se muestra en la Figura 2.12.

Durante la pasada hacia atrás pueden aprenderse distintas políticas variando la fineza en la discretización del espacio de estados. La figura 2.13 muestra el costo de operación obtenido por políticas que tienen distinta discretización en Bonete

Capítulo 2. Programación Dinámica Aproximada Cuadrática para el despacho hidrotérmico

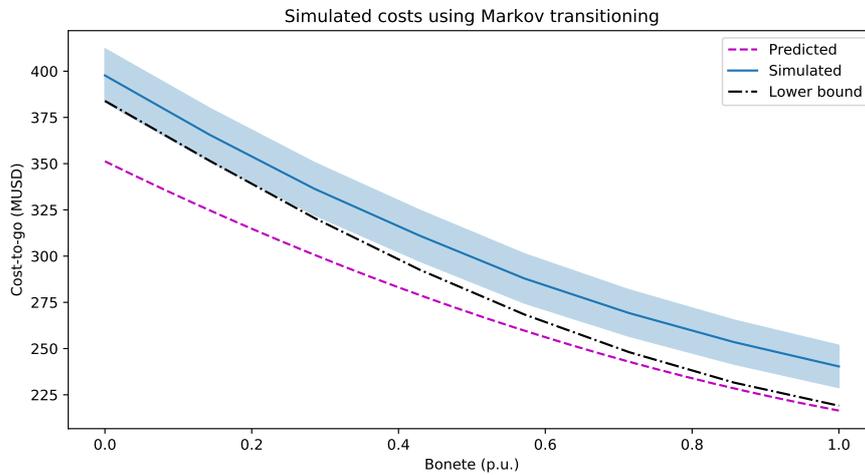


Figura 2.11: Costo de operación anual en función del nivel inicial de Bonete para $e_0 = 2$. En violeta: costo predicho por la cuadrática $\tilde{V}_{0,2}$. En azul sólido: costo promedio para $T = 105$ trayectorias. En azul sombreado: intervalo dado por $\pm\sigma/\sqrt{T}$ siendo σ la desviación muestral. En negro: cota inferior obtenida con (2.23)

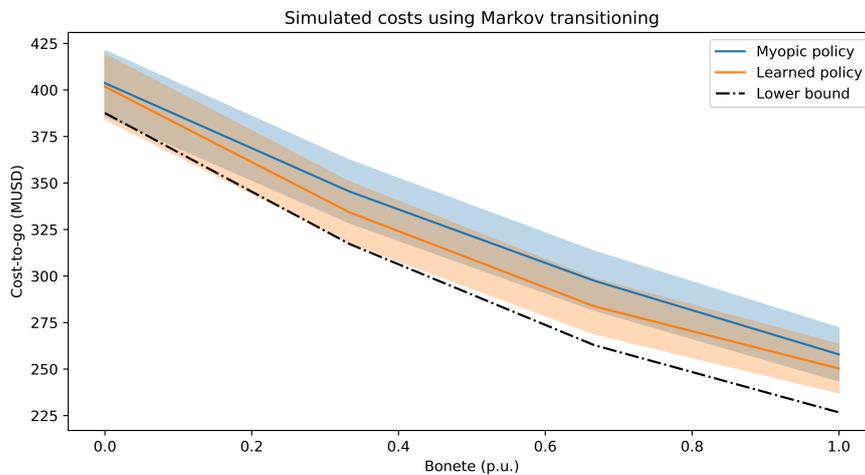


Figura 2.12: Comparación entre el desempeño de la política miope (azul) y la política aprendida (naranja) en función del nivel inicial de Bonete, junto a una cota inferior para el costo. Nuestra política consigue un 4% de reducción en costo respecto a la miope cuando Bonete está a medio llenar.

(N_0). En este experimento, el costo obtenido fue menor usando $N_0 = 10$.

Simulación usando crónicas históricas

Evaluar el desempeño de nuestra política es una difícil tarea. Hasta ahora, los experimentos presentados muestran que nuestra política es preferible ante la

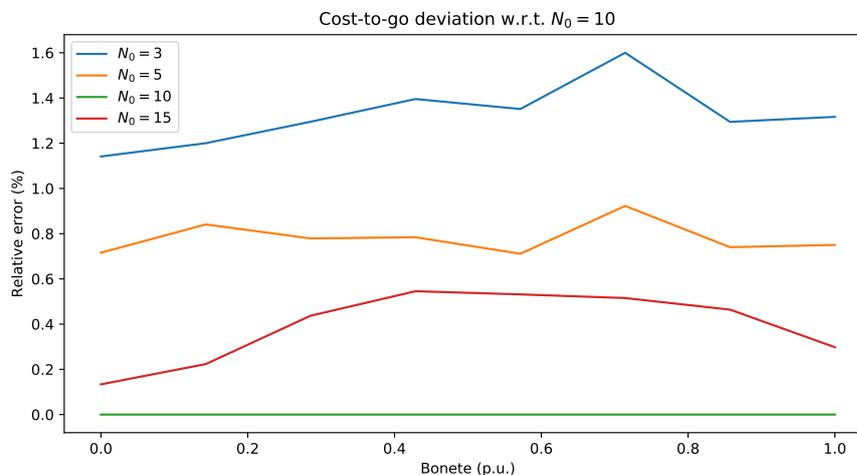


Figura 2.13: Desviación relativa en el costo simulado hacia adelante para distintas políticas, en función del nivel inicial de Bonete. Cada trazo corresponde al costo de una política entrenada con un paso de discretización de Bonete (N_0) distinto. Se usa como base $N_0 = 10$. Observar que las diferencias en costo no son sustanciales.

miope. Sin embargo, en algunos casos está lejos de la cota inferior para el costo (quizás el óptimo esté muy lejos de esta cota).

Más arduo aún es evaluar qué tan bueno es el agrupamiento de aportes descrito en la Sección 2.2.3, que dio lugar a los estados hidrológicos e_k y a la matriz de transición (2.10). Recordamos que el modelo de energía entrante (MEE) emplea, al igual que en nuestro caso, cinco niveles de discretización, pero con matrices variantes en el tiempo —una para cada semana del año. Una manera indirecta de comparar nuestra clusterización con la del MEE es la siguiente: entrenar dos políticas en la pasada hacia atrás, una con cada modelo. La diferencia esencial radica en las lluvias a sortear al resolver (2.13), y en los ponderadores del costo futuro en esa ecuación. Luego de entrenadas, en la pasada hacia adelante se obtiene el costo de operación simulando sobre las *crónicas históricas*. Es decir, se simula usando las lluvias del año 1909, 1910, etc. La figura 2.14 muestra el costo obtenido al recorrer las crónicas históricas. Como puede apreciarse, la política entrenada con la clusterización propuesta (PCA) supera a la entrenada con el MEE. Cabe aclarar que la única diferencia entre estas dos formulaciones es que al entrenar las cuadráticas con MEE se usan matrices variantes en el tiempo (y por ende cambian los ponderadores del valor futuro del sistema en (2.9)).

Capítulo 2. Programación Dinámica Aproximada Cuadrática para el despacho hidrotérmico

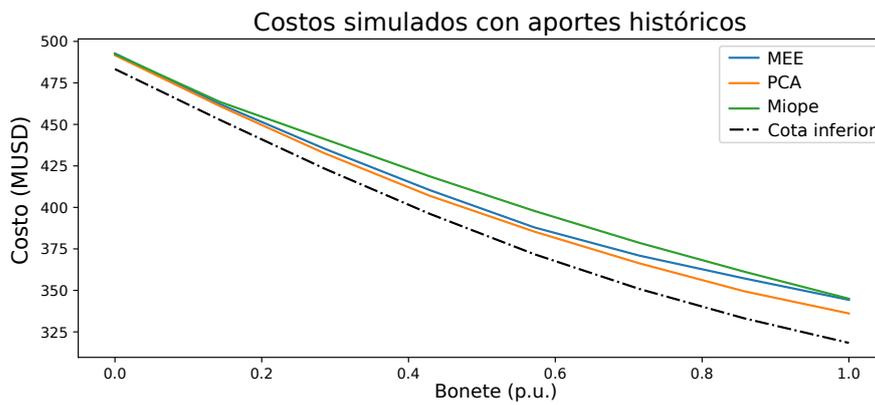


Figura 2.14: Costo de operación promedio en función del nivel inicial de Bonete. Las trayectorias de aportes coinciden con los datos históricos. La política entrenada con el modelo descrito en 2.2.3 (PCA) obtiene en el mejor caso una reducción del 2,4% en costo frente a una entrenada con el modelo de energía entrante (MEE).

Capítulo 3

Aprendiendo a operar un banco de baterías

En este capítulo abordamos el problema de controlar un banco de baterías en una red eléctrica sencilla. Trabajaremos sobre la red eléctrica uruguaya, usando datos reales de demanda, energía renovable y costos de generación. Asumiendo que las líneas de transmisión tienen capacidad ociosa, un modelo natural a emplear es el de una red uninodal. El control a diseñar opera sobre la batería eligiendo a cada instante la potencia que ésta inyecta (o absorbe) a la red, para cumplir con el balance nodal y las restricciones de la batería, y buscando minimizar un costo de operación. Para obtener este controlador usaremos el algoritmo de Q-learning [37, 38], uno de los algoritmos más celebrados dentro del campo de Aprendizaje por Refuerzo (*Reinforcement Learning*).

Comenzamos presentando el modelo, justificando la elección para el estado del sistema. Continuamos con el objetivo a optimizar —minimizar (en valor esperado) el costo descontado de operación del sistema. Describimos luego los datos de demanda, energía renovable y precio usados, y cómo fueron preprocesados y usados para construir una función de costo. Resumimos el algoritmo de *Q-learning episódico* más adelante, y finalizamos presentando los resultados.

Una versión abreviada de los contenidos de este capítulo se presentó en [19].

3.1. Modelado

Consideramos el modelo uninodal de la red eléctrica de la Figura 3.1, donde una carga es alimentada por un generador convencional, una turbina eólica y donde también se encuentra una batería, que puede absorber o entregar energía a la red según necesario. Como se trata de un modelo uninodal, la turbina eólica pretende representar el agregado de generadores eólicos del parque, y el generador convencional pretende modelar la suma de los generadores térmicos e hidráulicos disponibles (generador *hidrotérmico*, de aquí en más). Asimismo, la carga representa la demanda total del sistema.

Para cada intervalo de tiempo a considerar $t = 0, 1, \dots$ la carga consume una

Capítulo 3. Aprendiendo a operar un banco de baterías

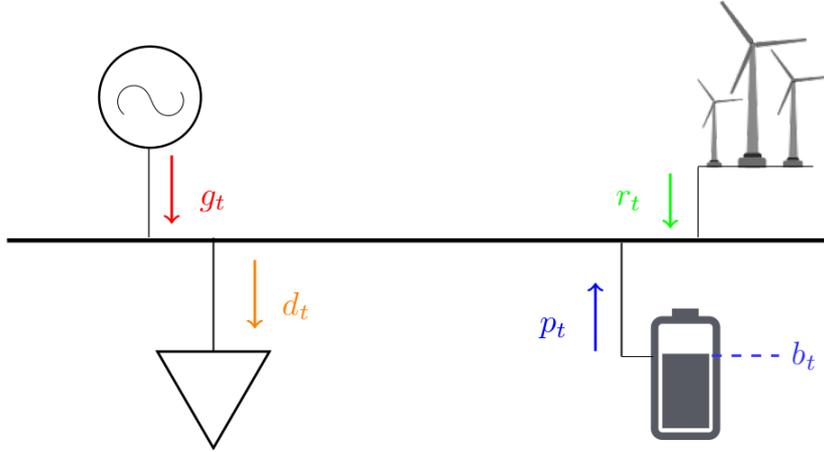


Figura 3.1: Modelo uninodal de la red. El generador (g_t), la renovable disponible (r_t), la demanda (d_t) y la potencia de la batería (p_t) se inyectan a la red.

potencia d_t , el generador convencional genera g_t , la renovable disponible es r_t y la batería inyecta al sistema una potencia p_t (que puede ser negativa si la batería está absorbiendo energía a la red). La ecuación fundamental que debe cumplir este sistema es el balance generación-demanda a cada instante, a saber

$$g_t + r_t + p_t - d_t = 0 \quad (3.1)$$

donde en principio se admite $g_t < 0$, en el caso en que la potencia renovable y la inyectada por la batería superen a la demanda, que se modela como exportación a países limítrofes. La batería se carga o descarga según el modelo de primer orden sin pérdidas

$$b_{t+1} = b_t - \Delta_T \cdot p_t \quad (3.2)$$

siendo Δ_T la duración del intervalo. Asumiremos de aquí en más que cada intervalo dura una hora. Hay restricciones inherentes al uso de la batería: su nivel de carga está acotado entre 0 y un valor máximo \bar{B} , y la potencia a inyectar o absorber en un intervalo está acotada por \bar{P}

$$0 \leq b_{t+1} \leq \bar{B} \quad (3.3)$$

$$|p_{t+1}| < \bar{P} \quad (3.4)$$

Para cada instante de tiempo hay un costo asociado del sistema $c(g_t)$ que es el costo incurrido por la generación hidrotérmica, que más adelante detallaremos. El objetivo es diseñar un controlador que a cada instante decida la potencia p_t a inyectar por la batería, en función del estado del sistema x_t . En la notación convencional de Teoría de Control, nuestro objetivo es obtener una política $\mu : \mathcal{X} \rightarrow \mathcal{U} : \mu(x_t) = p_t$.

3.1.1. Estado del sistema

En una primera aproximación, el estado del sistema podría incluir al nivel de la batería, la renovable y la demanda: $x_t = (b_t, r_t, d_t)$. Esta elección sin embargo es insuficiente. Su principal inconveniente es que la renovable y la demanda no siguen una dinámica Markoviana, algo esencial para aplicar Q-learning y otros algoritmos de programación dinámica. Esencialmente, una variable aleatoria X es Markoviana si cumple la propiedad de Markov [39, p.587]

$$P(X_{t+1} | X_t, X_{t-1}, \dots) = P(X_{t+1} | X_t) \quad \forall t.$$

Intuitivamente, el valor de la variable en el estado actual concentra toda la información del pasado, y es suficiente para determinar la estadística del paso siguiente. La variabilidad inherente al viento y a la demanda no cumplen esto: no alcanza con ver si en este instante la demanda es alta para predecirla en el instante siguiente.

El tiempo como estado

Una hipótesis razonable es suponer que la demanda y la renovable son Markovianas si incluimos el *tiempo* u hora del día en el estado. Introducimos entonces la variable τ_t que representa la hora del día en el instante t . Por ejemplo, $\tau_3 = 5$ quiere decir que en el instante de tiempo 3 la hora corresponde a las 5a.m. Esta nueva variable τ toma valores discretos entre 0 y 23, y es claro que es Markov: si en este instante son las k horas, en el instante siguiente serán las $k + 1$ horas.

$$P(\tau_{t+1} = k + 1 | \tau_t = k) = 1 \quad \forall k \in \{0, \dots, 22\} \quad (3.5)$$

En el caso particular en que actualmente sean las 23 horas, la hora siguiente será la hora 0:

$$P(\tau_{t+1} = 0 | \tau_t = 23) = 1 \quad (3.6)$$

La última propiedad de esta variable es su *cicloestacionariedad*. Cada $T = 24$ índices de tiempo las horas del día se repiten, y por ende

$$P(\tau_{t+1} | \tau_t) = P(\tau_{t+1+mT} | \tau_{t+mT}) \quad \forall m \in \mathbb{Z} \quad (3.7)$$

Cicloestacionariedad del viento y la demanda

Asumimos entonces que al incorporar el tiempo al estado, la renovable y la demanda resultan estacionarias

$$P(R_{t+1} | R_t, \tau_t) = P(R_{t+1} | R_t, \tau_t, R_{t-1}, \tau_{t-1}, \dots) \quad (3.8)$$

$$P(D_{t+1} | D_t, \tau_t) = P(D_{t+1} | D_t, \tau_t, D_{t-1}, \tau_{t-1}, \dots). \quad (3.9)$$

Incorporar la hora del día al estado del sistema resulta intuitivo. Fijando $D_t = 1200MW$, es razonable pensar que $P(D_{t+1} | D_t = 1200MW)$ varía dependiendo

Capítulo 3. Aprendiendo a operar un banco de baterías

de la hora del día. Si nos encontramos a las 13 horas, es de esperar que la demanda suba, mientras que si es medianoche lo razonable es que la demanda baje (comparar con la Figura 3.6, a ser presentada más adelante).

Vector de estados

Tenemos entonces como estado del sistema a la tupla

$$x_t = (b_t, r_t, d_t, \tau_t). \quad (3.10)$$

Con la incorporación del tiempo u hora del día al estado del sistema, la transición de estados resulta Markoviana y homogénea en el tiempo

$$P(x_{t+1} | x_t, p_t) = P(x_{t+1} | x_t, p_t, x_{t-1}, p_{t-1}, \dots)$$

3.2. Objetivo

Tal como se detallaba en la sección anterior, en cada etapa existe un costo asociado a la generación hidrotérmica $c(g_t)$. Relegamos las particularidades de esta función a la sección siguiente. Basta con asumir por ahora que este costo es cuadrático en la generación.

Nuestro objetivo es diseñar un controlador $\mu : \mathcal{X} \rightarrow \mathcal{A}$ que logre minimizar el costo acumulado de generación partiendo de cualquier estado inicial x . El costo acumulado esperado es la función de valor para dicho estado, a saber

$$V_\mu(x) = \mathbb{E}_\mu \left[\sum_{t=0}^{\infty} \gamma^t c(U_t) \mid X_0 = x \right] \quad (3.11)$$

donde la esperanza se toma respecto a la trayectoria inducida por la política, y donde el factor de descuento $\gamma \in (0, 1)$ se introduce para garantizar la convergencia de la suma infinita. Deseamos entonces obtener la política μ_\star que minimice (3.11) para cualquier estado inicial posible

$$\mu_\star = \underset{\mu \in \Pi}{\operatorname{argmin}} V_\mu(x) \quad \forall x \quad (3.12)$$

donde el mínimo se toma con respecto a un conjunto de políticas *admisibles* Π , a saber, aquellas que cumplen las restricciones de la red (3.3)–(3.4).

Tal como argumentábamos en el capítulo pasado (Sección 2.1), resolver (3.12) es extremadamente complejo. En particular, en este caso (a diferencia del capítulo anterior) no conocemos explícitamente la dinámica de transición del sistema. En efecto, para el estado $x_t = (b_t, r_t, d_t, \tau_t)$ solo conocemos la transición para el nivel de la batería b_t (dada la acción p_t) y para el tiempo τ_t , pero no así para las variables inherentemente aleatorias a nuestro problema: la energía renovable r_t y la demanda d_t . Un enfoque posible sería estimar estas probabilidades de transición (asumiendo un modelo autorregresivo [40, Cap.2] y encontrando los parámetros, por ejemplo) y luego aplicar algoritmos estándar de programación dinámica. Usaremos un enfoque

3.3. Aprendiendo de los datos

libre de modelo (*model-free*) y basado en datos, pero equivalente, en el sentido de que optimizaremos (3.12).

Para ello resulta útil introducir la función Q o función de valor del estado-acción asociada a (3.11)

$$Q_\mu(x, u) = \mathbb{E}_\mu \left[\sum_{t=0}^{\infty} \gamma^t c(U_t) \mid X_0 = x, U_0 = u \right]. \quad (3.13)$$

Intuitivamente (3.13) difiere de (3.11) en que se fija la primera acción a tomar u , y luego se sigue con la política μ . Resolveremos de forma aproximada (3.12) discretizando el espacio de estados y acciones y aprendiendo (3.13) para cada pareja posible, usando una versión del algoritmo de Q-learning [37], basada en episodios. Q-learning optimiza (3.12) la mismo tiempo que aprende las probabilidades de transición [15, Cap.6.5]. Antes de presentar el algoritmo, detallaremos la obtención de la función de costo $c(u_t)$ y haremos apreciaciones sobre los datos de demanda y renovable usados.

3.3. Aprendiendo de los datos

En este trabajo empleamos datos reales de generación, demanda y potencia eólica disponible de la red eléctrica uruguaya. Los datos fueron adquiridos de la página de ADME [13]. Nos concentramos en tres inviernos, comprendiendo los períodos entre Junio y Setiembre de los años 2017, 2018 y 2019. Elegimos centrarnos en una estación para que la estadística de las variables aleatorias —en particular la demanda y el viento— fuera similar en todos los casos. Esto torna más plausible nuestra hipótesis de dinámica Markoviana.

Comenzamos comentando los datos referentes a los costos de generación de la red, y como a partir de ahí obtuvimos un modelo para un generador equivalente. Procedemos luego a detallar el tratamiento de los datos de demanda y viento.

3.3.1. Función de costo

Nuestro insumo para obtener una función de costo son muestras diarias de los precios nodales de cada generador del parque uruguayo. Como estimador del costo de cada generador consideramos el promedio de estos precios a través de los tres inviernos; estos promedios se resumen en la Tabla 3.1 ordenados de menor a mayor. Reiteramos que en nuestro modelo uninodal deseamos representar el parque generador como un único generador equivalente. Luego de ordenar a los generadores del más barato al más costoso (asumiendo que el operador del sistema puede elegir qué máquina prender cuándo), obtenemos parejas de generación-costo para la máquina equivalente, que se ilustran en la Tabla 3.2.

Procedemos a obtener una función de costo para esta máquina. Por razones que detallaremos en breve, nos interesa ajustar una función de costo cuadrática (la otra opción natural es una función lineal a tramos). Dadas las parejas generación-

Capítulo 3. Aprendiendo a operar un banco de baterías

Central	Potencia instalada (MW)	Costo (USD/MWh)
Baygorria	108	0
Bonete	152	2.2
Palmar	333	41.0
Salto	945	81.9
Central Motores	104	122.1
Punta del Tigre	372	170.7
CC	530	182.2
La Tablada	200	210.2

Tabla 3.1: Generadores hidrotérmicos del Uruguay: potencia instalada según orden de costo. Notar cómo los generadores de menor costo son los hidroeléctricos.

Potencia (MW)	Costo (USD/MWh)
0	0.0
108	0.0
260	330.7
593	13980.2
1538	91365.9
1642	104068.9
2014	167572.2
2544	264125.5

Tabla 3.2: Parejas generación-costo para el generador equivalente Uruguayo.

costo (g_i, c_i) $i = 1, \dots, N$ que surgen de la Tabla 3.2, ajustamos una cuadrática positiva que pase por el origen, es decir

$$f(g) = ag^2, \quad (3.14)$$

según el criterio de mínimos cuadrados:

$$\min_{a \geq 0} \sum_{i=1}^N (ag_i^2 - c_i)^2 \quad (3.15)$$

Este problema se puede escribir equivalentemente en notación matricial como

$$\min_{a \geq 0} (a\mathbf{G} - \mathbf{c})^\top (a\mathbf{G} - \mathbf{c}) \quad (3.16)$$

donde $\mathbf{G} = [g_1^2 \ g_2^2 \ \dots \ g_N^2]^\top$ y $\mathbf{c} = [c_1 \ c_2 \ \dots \ c_N]^\top$. (3.16) es el conocido problema de regresión por mínimos cuadrados [41, Sec.3.2], cuya solución en forma cerrada es

$$a = \frac{\mathbf{c}^\top \mathbf{G}}{\mathbf{G}^\top \mathbf{G}}. \quad (3.17)$$

3.3. Aprendiendo de los datos

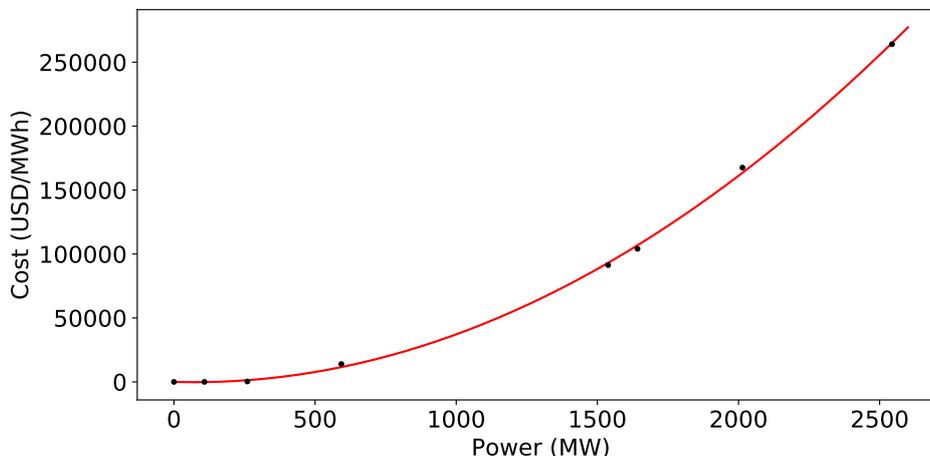


Figura 3.2: Función de costo para el generador equivalente. Los puntos negros son las parejas generación–costo de la Tabla 3.2

Algunas apreciaciones sobre la función de costo

La función de costo obtenida aparece graficada en la Figura 3.2. Cabe destacar que en este modelo se aceptan generaciones negativas, asociadas a un costo nulo (lo que indicaría exportación a países limítrofes gratuitamente). La elección de una función cuadrática no es caprichosa, fue tomada con la idea de observar un efecto en el controlador resultante. Dado un nivel fijo de generación g asociado a un costo $c(g)$, tenemos que la variación de costo ante una suba de generación Δg es creciente con g , a saber

$$\Delta c = c(g + \Delta g) - c(g) = a(g + \Delta g)^2 - ag^2 = a\Delta g(2g + \Delta g).$$

Por lo tanto, cuanto mayor sea la generación, más costoso resulta generar un incremento marginal. Esta intuición debería traducirse al comportamiento del agente: es de esperar que una buena política use la energía acumulada en momentos de alta demanda. Asimismo, en momentos de baja demanda quizás valga la pena incurrir en un costo extra de generación para cargar la batería (ya que generar un poco más resulta barato), y acumular esa energía para los picos de demanda. Esta sensibilidad asociada a la función cuadrática no se apreciaría si la función ajustada fuese lineal, o si fuese lineal a tramos y nos encontráramos dentro de un mismo tramo.

3.3.2. Demanda y Renovable

La demanda y la renovable se corresponden con procesos estocásticos cuya dinámica varía día a día y estación a estación. Por ejemplo, la estadística de la demanda en verano es inherentemente distinta a la de invierno: en verano los picos de consumo se dan cerca del mediodía (cuando hace mucho calor), mientras

Capítulo 3. Aprendiendo a operar un banco de baterías

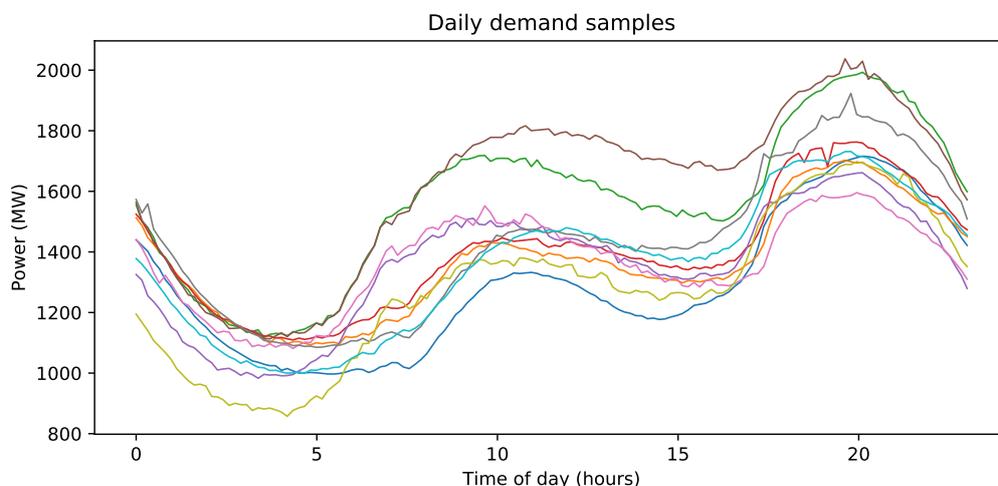


Figura 3.3: Ejemplos de demanda diaria para el invierno de 2019. Se muestran 10 días tomados al azar. Los perfiles de cada día son similares; se distinguen tres zonas: valle (de madrugada), meseta (durante el día), pico (en la noche).

que en invierno se dan en la tarde/noche. Para que nuestra hipótesis de *cicloes-tacionariedad* sea plausible, nos centraremos únicamente en crónicas de invierno. Consideramos entonces, tal como explicamos al inicio de la sección, los datos diarios de demanda y viento entre los meses de junio y setiembre para los años 2017, 2018 y 2019.

A modo de ejemplo, en las Figuras 3.3 y 3.4 presentamos algunas crónicas de demanda y viento para el invierno de 2019. Las crónicas de demanda son bastante similares entre sí: en todos los casos identificamos un *valle* en la madrugada (donde la demanda es mínima), una *meseta* durante el día (de demanda media) y un *pico* o máximo al caer la noche. Contrariamente, en los datos de viento se vislumbra mayor variabilidad. Notamos, sin embargo, que los momentos de mayor generación eólica se dan en la noche o en la madrugada.

En lo que sigue describimos cómo fueron preprocesados los datos. Antes de volcarnos al aprendizaje removimos datos espúreos y reescalamos las trayectorias de demanda y renovable.

Outliers

Durante dos días de estos inviernos ocurrieron apagones importantes. A modo de ejemplo, el 16 de junio de 2019 ocurrió un apagón masivo a nivel nacional [42]. Los perfiles de demanda y viento para este día son *outliers* que fueron descartados (ver Figura 3.5).

Reescalando los datos

Existen leves variaciones interanuales entre los perfiles medios de demanda y viento. En la Figura 3.6 representamos las curvas de demanda promedio para cada

3.3. Aprendiendo de los datos

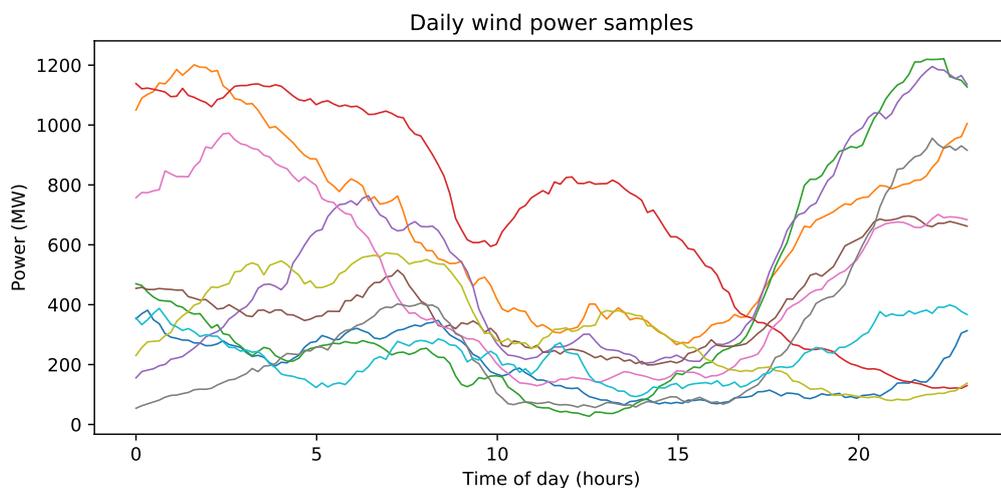


Figura 3.4: Perfiles diarios de viento para el invierno de 2019. Se muestran 10 días tomados al azar. La disponibilidad del viento es mucho más variable que la demanda (comparar con Figura 3.3).

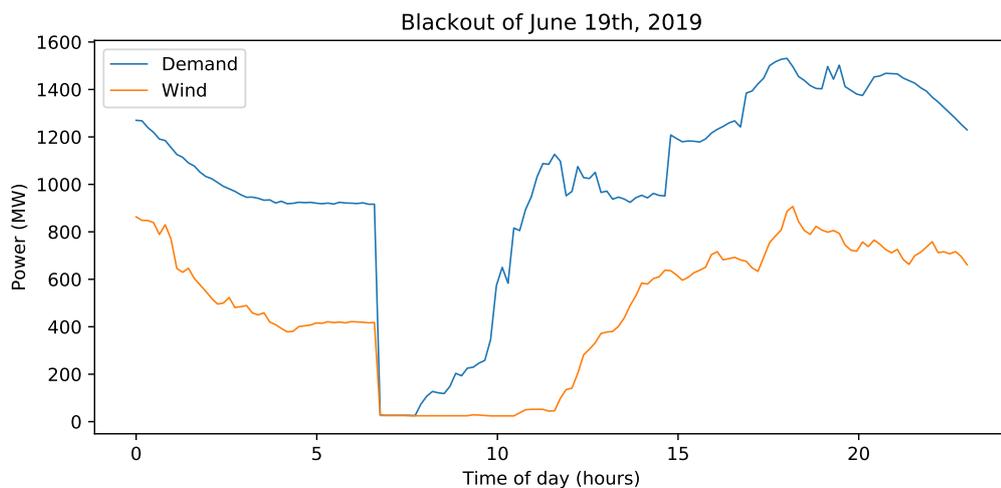


Figura 3.5: Apagón masivo del 16 de junio de 2019. Este día fue removido del dataset de entrenamiento.

uno de los inviernos. Asimismo, en la Figura 3.7 aparecen los perfiles medios de viento. La región sombreada entorno a cada curva corresponde a $\pm\sigma/\sqrt{N}$, donde σ es la desviación muestral y N la cantidad de datos (es decir, la cantidad de días del invierno correspondiente). Aquí puede verificarse lo que discutíamos anteriormente: el perfil de la demanda es aproximadamente similar en todos los días, mientras que el viento presenta gran variabilidad (cf. Figuras 3.3 y 3.4).

Podemos ver que hay variaciones año a año entre los perfiles medios de demanda. Para que nuestra hipótesis de un modelo Markoviano sea más plausible,

Capítulo 3. Aprendiendo a operar un banco de baterías

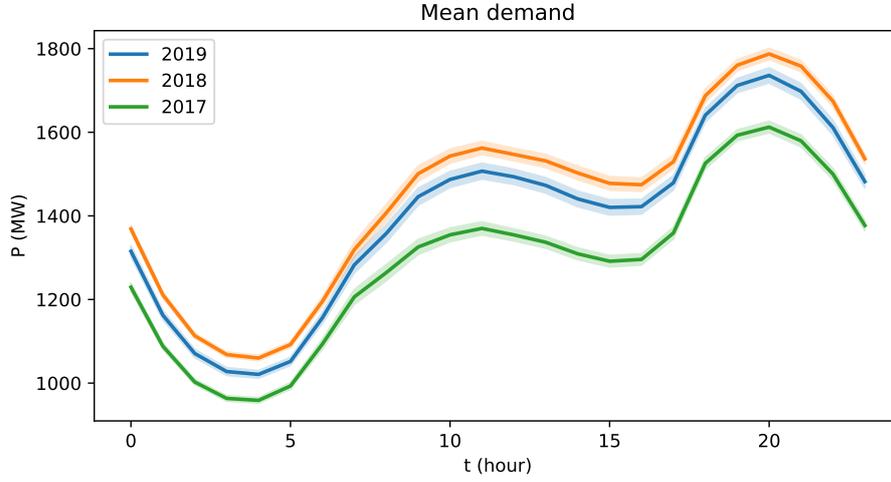


Figura 3.6: Demanda promedio para los inviernos de 2017, 2018 y 2019. En trazo sólido: demanda media. En trazo sombreado: desviación respecto a la media de $\pm\sigma/\sqrt{N}$, donde σ es la desviación muestral y N la cantidad de muestras (días del invierno). La varianza de la demanda es pequeña. La demanda crece en promedio cada año.

Tabla 3.3: Factores de corrección aplicados a cada serie temporal de demanda y viento.

Año	Demanda	Renovable
2019	1	1
2018	0.965	1.023
2017	1.082	1.440

procedemos a reescalar los perfiles diarios de demanda para asemejarlos a los del año 2019, según el criterio de mínimos cuadrados. Sea \mathbf{d}_0 la demanda promedio del año 2019, y \mathbf{d}_1 y \mathbf{d}_2 la demanda media de 2018 y 2017, respectivamente. Para $a = 0, 1, 2$ buscamos el coeficiente k que resuelva:

$$k_a = \underset{k}{\operatorname{argmin}} \|\mathbf{d}_0 - k\mathbf{d}_a\|^2 \quad (3.18)$$

cuya solución en forma cerrada es

$$k_a = \frac{\langle \mathbf{d}_0, \mathbf{d}_a \rangle}{\|\mathbf{d}_a\|^2} \quad (3.19)$$

donde $\langle \mathbf{d}_0, \mathbf{d}_a \rangle$ es el productor interno usual en \mathbb{R}^T .

Para el viento tomamos el mismo enfoque, buscando el coeficiente que asemeje la eólica anual promedio al promedio del año 2019. Resumimos los coeficientes obtenidos en la Tabla 3.3. Luego de reescalar los datos, observamos que la demanda neta —la resta de la demanda con la renovable disponible— es positiva en la gran mayoría de las horas. Para poder apreciar el fenómeno de arbitraje de energía que buscamos, duplicamos la potencia eólica disponible a cada hora.

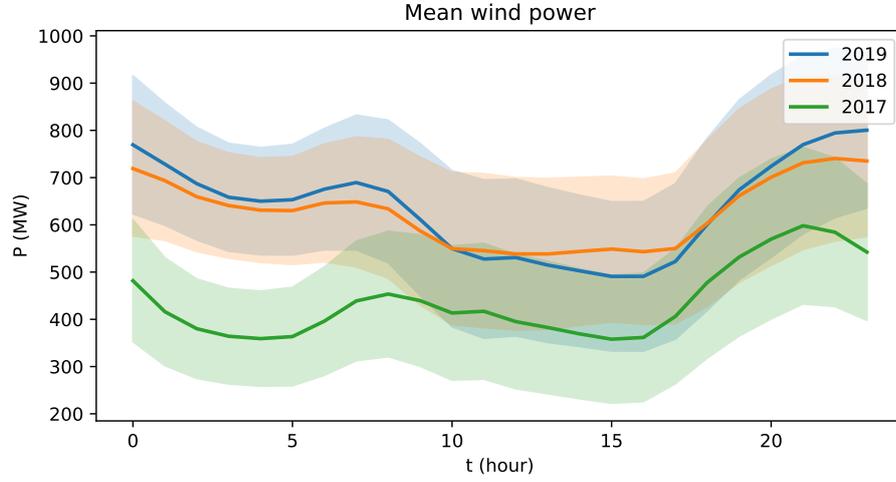


Figura 3.7: Potencia eólica promedio para los inviernos de 2017, 2018 y 2019. En trazo sólido: potencia media. En trazo sombreado: desviación respecto a la media de $\pm\sigma/\sqrt{N}$, donde σ es la desviación muestral y N la cantidad de muestras (días del invierno). Observar cómo la varianza del viento es grande (en comparación con la demanda, cf. Figura 3.6). En promedio, el invierno de 2017 fue menos ventoso que el de los otros años.

3.4. Algoritmo

En la Sección 3.2 detallamos que nuestro objetivo es obtener la política μ_* que minimice el costo de operación del sistema

$$\mu_* = \min_{\mu \in \Pi} V_\mu(x) = \mathbb{E}_\mu \left[\sum_{t=0}^{\infty} \gamma^t c(U_t) \mid X_0 = x \right] \quad \forall x.$$

Para aprender la política μ usaremos el algoritmo de Q-learning, que comienza con una tabla Q llena de ceros:

$$Q_\mu(x, u) = 0 \quad \forall x \in \mathcal{X}, \forall u \in \mathcal{U}$$

y, conforme el agente va actuando sobre el sistema, desde el estado x tomando una acción u y viendo el estado resultante x' , actualiza la entrada de la tabla Q de manera asíncrona según

$$Q_\mu(x, u) \leftarrow Q_\mu(x, u) + \alpha_t (c_t + \gamma \min_{u'} Q_\mu(x', u') - Q_\mu(x, u)). \quad (3.20)$$

Uno de los requisitos fundamentales para que el algoritmo converja a la política óptima es que *cada pareja* (x, u) *sea visitada y actualizada infinitas veces* [43]. Aunque esto en la práctica es irrealizable, empíricamente es una buena idea favorecer la *exploración* del algoritmo, llevándolo a visitar distintas parejas estado-acción. Esto lo lograremos de dos formas, a saber:

Capítulo 3. Aprendiendo a operar un banco de baterías

1. En vez de actuar en cada paso siguiendo la política μ , usaremos una política que con probabilidad ϵ toma una acción al azar. En la literatura esto se conoce como política ϵ -greedy.
2. Correremos episodios, inicializando aleatoriamente el estado del sistema, y dejando que el agente aprenda durante N días consecutivos. Luego de esto, arranca un nuevo episodio en otro estado al azar.

Política ϵ -greedy

Con respecto al primer punto, conforme se van actualizando las entradas de la tabla Q , la política aprendida μ opera en cada estado según

$$\mu(x) = \underset{u}{\operatorname{argmin}} Q_\mu(x, u) \quad (3.21)$$

es decir, en cada paso toma la acción que minimiza la entrada de la tabla Q , para el estado actual. Para favorecer la exploración hacia nuevas parejas estado-acción usamos una política ϵ -greedy. Dado un parámetro $\epsilon \in (0, 1)$, definimos la política μ_ϵ como

$$\mu_\epsilon(x) = \begin{cases} \mu(x) & p = 1 - \epsilon \\ U \sim \operatorname{Unif}(\mathcal{U}) & p = \epsilon \end{cases} \quad (3.22)$$

En resumidas cuentas, (3.22) sigue la política aprendida con alta probabilidad, o sorteando una acción al azar con baja probabilidad. Típicamente se usan valores pequeños de ϵ , del orden de 0,1. Este parámetro también puede ser variante en el tiempo ($\epsilon := \epsilon_t$), en cuyo caso es usual tomarlo monótono decreciente.

Episodios

Con respecto al segundo punto, para favorecer la exploración también haremos uso de episodios. Cada episodio arranca con el agente en un estado al azar del sistema —sorteando el nivel de la batería, la hora del día, y el día de arranque correspondiente a alguno de los inviernos de 2017, 2018 y 2019. El agente operará el sistema —actualizando su política— durante N días, y luego será enviado a un nuevo estado inicial al azar.

Con este enfoque, lo que estamos haciendo en la práctica es buscar una política que minimice el costo esperado para un horizonte igual a NT :

$$V_\mu^{ep}(x) = \mathbb{E}_\mu \left[\sum_{t=0}^{NT-1} \gamma^t c(U_t) \mid X_0 = x, U_t \sim \mu(\cdot | X_t) \right] \quad \forall x, \quad (3.23)$$

donde ahora la suma contempla NT pasos y N es un parámetro de diseño. Cabe destacar que maximizar (3.23) no es lo mismo que resolver (3.11). Esencialmente estamos despreciando la cola de la geométrica

$$\mathbb{E}_\mu \left[\sum_{t=NT}^{\infty} \gamma^t c(U_t) \right]. \quad (3.24)$$

3.4. Algoritmo

La hipótesis subyacente es que la influencia de γ hace que la cola (3.24) sea muy pequeña. En efecto, si el costo $c(u)$ está globalmente acotado superiormente por C , resulta

$$\mathbb{E}_\mu \left[\sum_{t=NT}^{\infty} \gamma^t c(U_t) \right] \leq \frac{\gamma^{NT}}{1-\gamma} C .$$

3.4.1. Q-learning episódico

En virtud de la sección anterior, lo que haremos será correr sucesivos episodios y entrenar así al agente. Cada episodio comienza con un estado inicial $x_0 = (b_0, d_0, r_0, \tau_0)$ (formado, recordemos, por el nivel de la batería, la demanda actual, la potencia eólica disponible y la hora del día en cuestión) sorteado de manera aleatoria. El nivel de la batería b_0 se toma uniformemente aleatorio dentro de sus posibles valores. La hora del día τ_0 se toma aleatoria entre 0 y $T - 1 = 23$. El día en cuestión se toma al azar dentro del conjunto de datos que contiene los tres inviernos. Si hay L días en cada invierno, el día inicial se elige uniformemente aleatorio sobre los primeros $L - N$ días (recordar que N es la cantidad de días en un episodio). De esta forma garantizamos que cada episodio se corre dentro de un mismo invierno.

Luego de randomizar el estado de arranque, se sigue la variante ϵ -greedy de la política aprendida hasta ahora μ , actualizando la tabla Q en cada paso. Luego de completar N días completos, se randomiza el estado de arranque y se repite lo antedicho. Esta etapa de entrenamiento se resume en el Algoritmo 4.

3.4.2. Evaluando la política aprendida

Una manera útil de visualizar el aprendizaje a lo largo del tiempo es graficando una *curva de aprendizaje*, donde se aprecie el desempeño del agente conforme transcurre el algoritmo. Para conseguir esta gráfica precisamos *evaluar* las políticas que van surgiendo del Algoritmo 4. Definiremos el *costo asociado a la política* μ como el costo esperado de un episodio, es decir

$$V_\mu := \mathbb{E}_{\mu, x_0} \left[\sum_{t=0}^{NT-1} \gamma^t c_t \right] \quad (3.25)$$

Notar que en este caso el valor esperado se toma respecto al estado inicial x_0 , que en nuestro caso es uniformemente aleatorio (ver Algoritmo 4). Calcular este costo en la práctica resultaría muy demandante computacionalmente —habría que correr un episodio arrancando de *todos* los estados iniciales posibles, y luego promediar. En vez de hacer esto sustituiremos (3.25) por una versión aproximada, usando sorteos Monte Carlo. Para ello sorteamos M estados iniciales $x^{(i)}$, $i = 1 \dots M$ de manera uniforme y obtenemos el costo promedio

$$V_\mu \simeq \frac{1}{M} \sum_{i=1}^M \left[\sum_{t=0}^{NT-1} \gamma^t c_t(\mu(x_t)) \mid X_0 = x^{(i)} \right] \quad (3.26)$$

```

Input: num_eps           /* episodios a correr */
 $\epsilon_t$              /* regla de exploración */
learning-rate  $\alpha_t$       /* learning-rate */
N                          /* días en un episodio */
Data: Datos de viento y demanda
Result: política  $\mu \approx \mu^*$ 
for episode = 1, 2, ..., num_eps do
    /* Randomizar estado inicial */
     $\tau_0 \sim \text{Unif}(0, T - 1)$ 
    Sortear  $\{(r_0, d_0), (r_1, d_1), \dots, (r_{N.T}, d_{N.T})\}$ 
     $b_0 \sim \text{Unif}(0, \overline{B})$ 
     $x_0 = (b_0, d_0, r_0, \tau_0)$ 
    for t = 0, 1, ..., N.T - 1 do
        /* correr N días */
         $u_t = \mu_{\epsilon_t}(x_t)$ 
         $b_{t+1} \leftarrow b_t - u_t$ 
        Revelar  $(r_{t+1}, d_{t+1})$  y costo  $c_t$ 
        Fijar  $x_{t+1} = (b_{t+1}, d_{t+1}, r_{t+1}, \tau_{t+1})$ 
         $q(x_t, u_t) \leftarrow q(x_t, u_t) + \alpha_t (c_t + \gamma \min_{u'} q(x_{t+1}, u') - q(x_t, u_t))$ 

```

Algoritmo 4: Q-learning episódico

La estrategia resultante entonces consiste en alternar instancias de entrenamiento —llamando al Algoritmo 4— con instancias de evaluación de la política aprendida. Cabe hacer una pequeña apreciación: en la etapa de evaluación se usa la política aprendida μ , y no su versión exploradora μ_ϵ . Resumimos la rutina principal en el siguiente algoritmo.

3.5. Simulaciones

La batería se modela con una capacidad de $\overline{B} = 1000MWh$ y sin restricciones de rampa de carga. Discretizamos los niveles de la batería cada $100MWh$. Como el paso Δ_T corresponde a una hora, esto induce una discretización para la potencia de la batería de $100MW$. Coherentemente, ésta será a su vez la discretización para la potencia demandada, generada y la disponible del viento. Resumimos estos y otros datos empleados en la Tabla 3.4. Cabe destacar que la capacidad de la batería empleada es enorme: los bancos de batería de mayor porte actualmente se encuentran en Australia, de $150MWh$ [44], y en California, de $250MWh$ [45], aunque hay planes de instalar a fines de 2021 un nuevo banco en Australia de $450MWh$ [46].

Recordando que el estado $x = (b, d, r, \tau)$ es una tupla compuesta por la carga de la batería, demanda, renovable y tiempo, tenemos que con estos valores la cardinalidad del espacio de estados es $|\mathcal{X}| = 10 \times 21 \times 15 \times 24 = 75600$. Si a esto le

```

Input:
num_eps                               /* episodios a correr */
 $\epsilon_t$                           /* regla de exploración */
 $\alpha_t$                               /* learning-rate */
N                                      /* días en un episodio */
M                                      /* sorteos Monte Carlo para evaluar */
for  $t = 1, \dots$  do
  /* entrenamiento */
  Q-learning episódico(num_eps,  $\epsilon_t$ ,  $\alpha_t$ , N)
  /* evaluación */
   $V_\mu \leftarrow 0$ 
  for  $m = 0, M - 1$  do
    /* Muestrear (3.26) */
     $x_0 = (b_0, d_0, r_0, \tau_0)$  aleatorio (como en Alg. 4).
     $V_\mu \leftarrow V_\mu + \sum_{t=0}^{NT-1} \gamma^t c_t(\mu(x_t))$ 
   $V_\mu \leftarrow V_\mu / M$ 

```

Algoritmo 5: Entrenador y evaluador

Parámetro	Nombre	Valor/descripción
Δ_T	paso de discretización	1 hora
T	período cicloestacionario	24
$c_t(g_t)$	función de costo	cuadrática
d_t, r_t	demanda y renovable	discretizadas cada 100MW
p_t	potencia inyectada a la batería	discretizada cada 100MW
\bar{B}	capacidad de la batería	1000MWh
\bar{P}	rampa de carga de la batería	1000MW
γ	factor de descuento	0.9
N	días en un episodio	3
num_eps	episodios por fase de entrenamiento	10^4
M	sorteos Monte Carlo en evaluación	10^4

Tabla 3.4: Resumen con las distintas variables y parámetros empleados.

agregamos que en cada estado hay $|\mathcal{U}(x)| = 10$ posibles acciones a tomar, resulta que la tabla Q tiene $|\mathcal{X}| \times |\mathcal{U}| = 756000$ entradas. Un tamaño considerable.

Inicializamos al agente con una tabla $q = 0$. Durante el entrenamiento se alternan dos etapas: la etapa de aprendizaje, donde se corre el Algoritmo 4 durante $\text{num_eps} = 10^4$, y la etapa de evaluación de la política aprendida, donde se estima el costo esperado de un episodio (3.26), promediando los costos sobre $M = 10^4$ trayectorias.

Para tener una medida del desempeño de la política aprendida la comparamos contra el costo de operación del sistema sin baterías. En este caso, la demanda

Capítulo 3. Aprendiendo a operar un banco de baterías

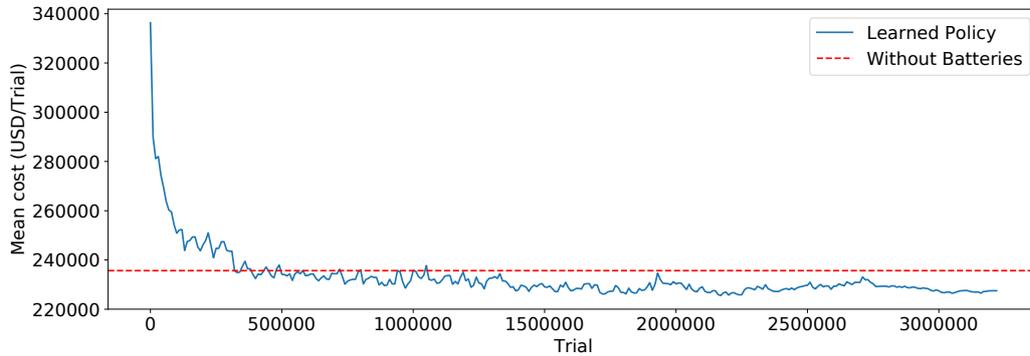


Figura 3.8: Curva de aprendizaje mostrando el costo promedio de un episodio en función del entrenamiento. En rojo punteado se muestra el costo promedio (exacto) que se obtiene con una política sin baterías. Luego de aproximadamente 500000 iteraciones, la política aprendida supera en desempeño a la política sin baterías.

neta debe ser cubierta en su totalidad por generación térmica, de acuerdo a (3.1). Este costo lo calculamos de forma exhaustiva, promediando el costo de un episodio para cualquier estado de arranque x_0 posible.

En la Figura 3.8 presentamos la curva de aprendizaje para el agente. A medida que transcurre el tiempo, el agente aprende a operar mejor la red. Luego de unos 5×10^5 episodios obtiene costos de operación menores al sistema sin baterías. El entrenamiento es lento, precisándose muchos episodios (más de 3×10^6) para conseguir una política medianamente buena. Para fijar ideas, correr 10^6 iteraciones desde un Python Notebook [47] en Google Colab [48] lleva unas 8 horas. Cada aproximadamente 5×10^5 iteraciones guardamos los valores de la tabla Q y ajustamos (a la baja) los valores de ϵ y α . Esto explica en parte la mayor suavidad de la curva en su extremo derecho. De todas formas, la gráfica es inherentemente ruidosa debido a ambas etapas del algoritmo: *entrenamiento* y *evaluación*. La varianza de Q-learning puede ser alta, en particular cuando se usa una muestra por update [49]. Otra fuente de ruido se debe a que la evaluación del costo se hace aproximando (3.25) por sorteos Monte Carlo (3.26).

En la Figura 3.9 presentamos el despacho de un día particular con el controlador obtenido. Esta política actúa, al menos en parte, como esperábamos: la batería se carga durante etapas de baja demanda (incluso a expensas de generación térmica, barata), y esta energía se vuelca a la red cuando la demanda aumenta.

3.5.1. Discusión sobre los resultados

Luego de mucho entrenar, logramos una política que supera a la operación sin baterías. Igualmente, viendo los ejemplos de despacho parece que ésta está lejos de ser optimal. El desempeño obtenido puede deberse a tres factores:

- falta de entrenamiento
- influencia de γ

3.5. Simulaciones

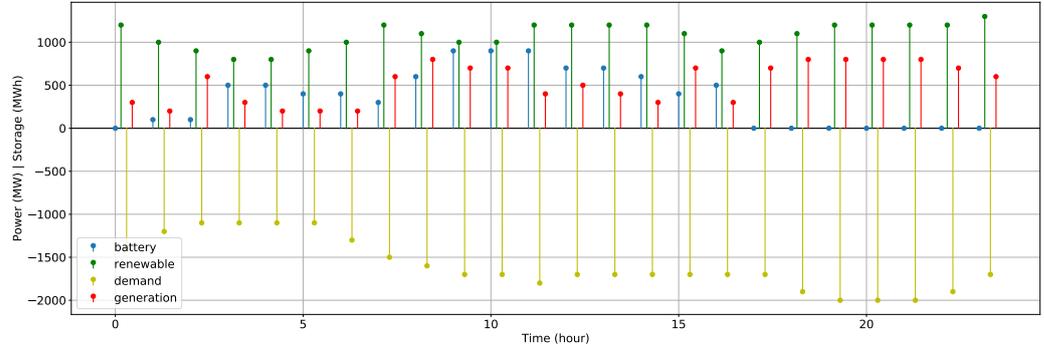


Figura 3.9: Ejemplo de despacho hora a hora con la política aprendida. En azul: nivel de la batería (MWh); en verde y amarillo: potencia eólica y demanda (MW); en rojo: generación térmica (MW). En el instante t , la suma de la renovable, generación térmica y demanda es igual a la diferencia del nivel de la batería entre t y $t + 1$. Notar cómo se carga la batería durante la mañana (poca demanda) a costas de generación extra, y se descarga en la tarde (mayor demanda).

- hipótesis de modelo Markoviano

Es posible que el agente precise entrenarse aún más. Con la discretización empleada la tabla Q tiene $|\mathcal{X}| \times |\mathcal{U}| = 7,56 \times 10^5$ entradas. Entrenando unos 3×10^6 episodios, y considerando que cada episodio consta de $N = 3$ días, se tiene que en promedio cada entrada de la tabla se actualiza unas 286 veces. Quizás no sea suficiente.

Por otro lado, uno de los comportamientos que esperábamos observar era que la batería se cargara durante los momentos de poca demanda, incluso a expensas de generación térmica. Esto se observa parcialmente en los ejemplos de despacho. Tal vez en este caso esté influyendo el factor de descuento γ , penalizando en menor medida los costos en instantes futuros y lejanos. Con el valor de γ empleado, el largo efectivo del horizonte es $\frac{1}{1-\gamma} = 10$ horas; los potenciales ahorros en gasto se diluyen en el factor de descuento.

Finalmente, estamos asumiendo que el estado del sistema es Markov. En la Sección 3.1.1 argumentábamos que esto puede ser poco realista para el viento y la demanda. Para paliar esto, comentamos que podíamos incorporar al estado las p realizaciones anteriores del viento y la demanda, y asumir un modelo Markoviano para esa transición, por ejemplo $P(R_{t+1} | (R_t, R_{t-1}, \dots, R_{t-p}), U_t)$.

Para analizar esto un poco más a fondo, estudiemos cuán bien se ajusta un modelo autorregresivo [50, p.578] a las series temporales de viento y demanda. Si encontramos que un modelo de orden p se ajusta bien a los datos, podemos concluir que asumir Markov con p componentes en el estado sería razonable. Con esta idea en mente, nos centramos entonces en las Figuras 3.10 y 3.11, donde se muestran la función de autocorrelación parcial [50, p.575] para el viento y la demanda del año 2019, respectivamente (los resultados son similares para las series de años anteriores). Como puede apreciarse en la primera figura, al parecer para el viento

Capítulo 3. Aprendiendo a operar un banco de baterías

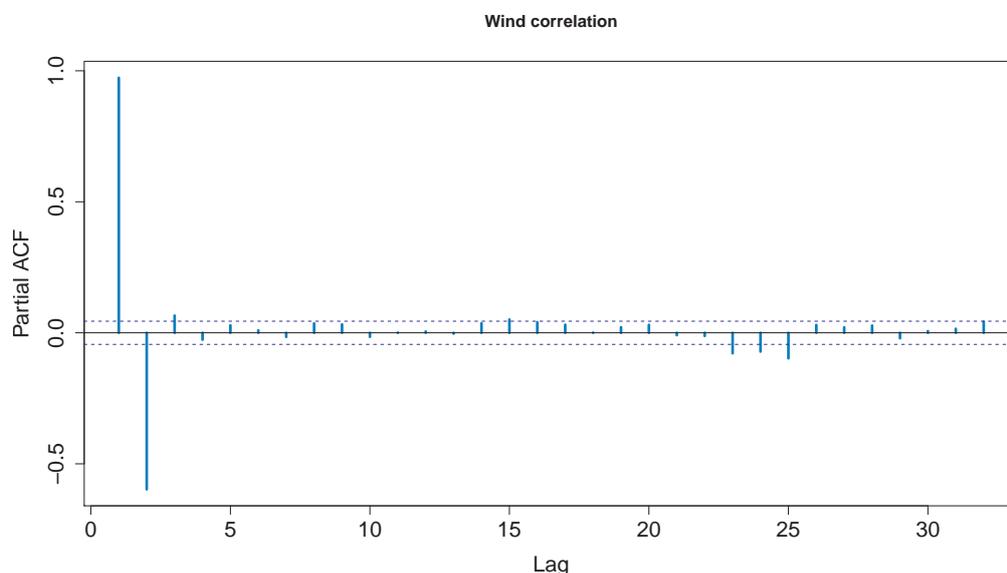


Figura 3.10: Autocorrelación parcial de la serie temporal de viento, año 2019. La autocorrelación es significativa para los *lags* de orden 1 y 2 y despreciable para el resto. Un modelo AR-2 se ajustaría bien a esta serie.

se ajustaría bien un modelo autorregresivo de orden 2. No es fácil concluir algo similar para la demanda.

3.6. Operación con pronósticos exactos

Finalizamos este capítulo con una prueba de concepto, estudiando el desempeño de un controlador que cuenta con un pronóstico *exacto* de la demanda y la renovable disponible para todo el día. Como veremos en breve, cuando se cuenta con pronósticos perfectos es posible resolver el problema de horizonte completo o varias etapas al comienzo del episodio y actuar en *lazo abierto*. De manera similar a lo que se vio en el Capítulo 2, el costo obtenido de esta forma es una cota inferior para el que se consigue con cualquier otro controlador (en particular con Q-learning), ya que al comienzo del intervalo se sabe con certeza la demanda y la potencia eólica disponible en toda la trayectoria. Los resultados que aparecen a continuación fueron presentados en la edición 2019 de Ingeniería DeMuestra [51].

Por lo que queda de este trabajo nos olvidamos de los episodios y de Q-learning, y nos centramos en el despacho u operación del sistema durante un día. Fijado el día, la hora $t = 0$ y el estado inicial del sistema x_0 (que en esta sección corresponde al nivel de la batería b_0 , la demanda inicial d_0 y la renovable inicial r_0), el controlador tiene acceso a un *pronóstico perfecto* de la demanda y la renovable durante todo el día, que recolectamos respectivamente en las variables $\mathbf{d}, \mathbf{r} \in \mathbb{R}^T$. El objetivo entonces es minimizar el costo (sin descuento) de operación en el horizonte diario, con paso horario:

3.6. Operación con pronósticos exactos

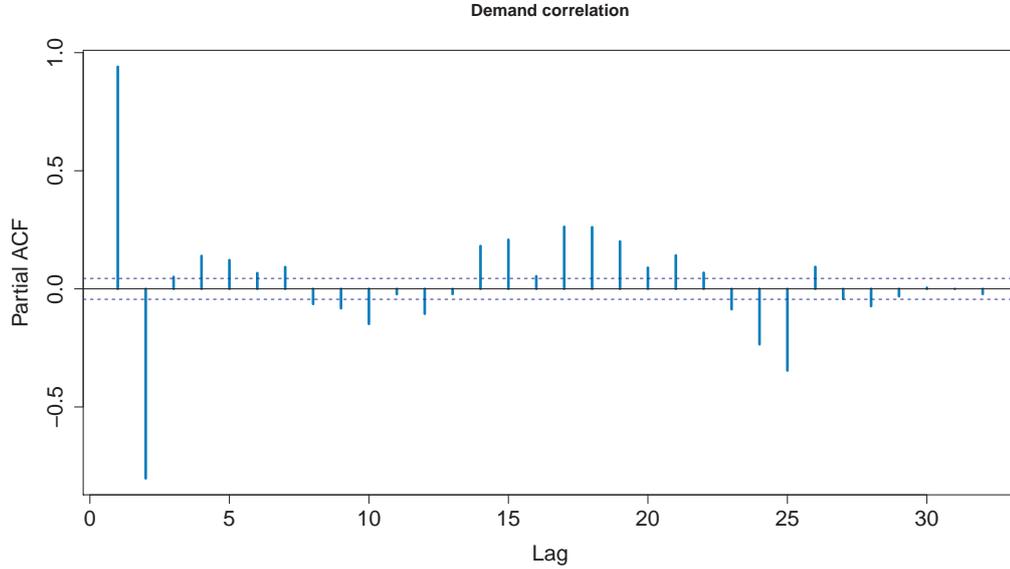


Figura 3.11: Autocorrelación parcial de la serie temporal de demanda, año 2019. Los lags de primer y segundo orden son los más significativos. Sin embargo, los lags de orden 15–20 son considerables. En este caso un modelo AR-2 no ajustaría bien.

$$\text{QP}(b_0, \mathbf{d}, \mathbf{r}) = \min_{\mathbf{g}, \mathbf{b}, \mathbf{p}} \mathbf{g}^\top \mathbf{A} \mathbf{g} \quad (3.27)$$

$$\text{s. a: } \mathbf{g} + \mathbf{r} + \mathbf{p} - \mathbf{d} = \mathbf{0} \quad (3.28)$$

$$b_{t+1} = b_t - p_t \quad t = 0, \dots, T-1 \quad (3.29)$$

$$\mathbf{0} \leq \mathbf{b} \leq \bar{B} \mathbf{1} \quad (3.30)$$

$$-\bar{P} \mathbf{1} < \mathbf{p} < \bar{P} \mathbf{1} \quad (3.31)$$

$$(b_0, d_0, r_0) = x_0 \quad (3.32)$$

donde expresamos el costo $\sum_{t=0}^{T-1} c(g_t)$ de (3.14)–(3.17) en forma matricial, con $\mathbf{A} = \text{diag}(a\mathbf{1})$ y $\mathbf{1} \in \mathbb{R}^T$ el vector de unos; (3.28) es el balance uninodal de (3.1) expresado en forma vectorial; (3.30) y (3.31) son las restricciones en la capacidad y rampa de carga de la batería; (3.32) fija el estado inicial del sistema. En este problema están fijos el nivel inicial de la batería b_0 , la demanda \mathbf{d} y la renovable \mathbf{r} de todo el horizonte, y se optimiza sobre el estado de carga de la batería, su potencia inyectada y la consigna de generación.

Las ecuaciones (3.27)–(3.32) constituyen un programa cuadrático con restricciones lineales [31, p.152] (de ahí la nomenclatura QP en (3.27)). Lo resolveremos usando el *Operator Splitting QP Solver* (OSQP) [52], disponible en Python a través de la librería *cvxpy* [33].

Podemos calcular el costo de operación promedio de un día como

Capítulo 3. Aprendiendo a operar un banco de baterías

$$V_{\text{QP}} = \mathbb{E}_{\pi_{\text{QP}}, x_0} \left[\sum_{k=0}^{T-1} c(g_k) \right] \quad (3.33)$$

donde QP hace referencia a que la política resulta de resolver (3.27)–(3.32) y la esperanza se toma respecto a la distribución del estado inicial x_0 y a la trayectoria inducida por la política. Con una distribución uniforme para x_0 , obtenemos (3.33) resolviendo (3.27)–(3.32) exhaustivamente para cada estado de arranque posible y promediando los objetivos (esto se logra cambiando x_0 en (3.32)). Incluso podemos estudiar cómo varía este costo en función de la capacidad nominal de la batería \bar{B} . Esto es lo que se muestra en la Figura 3.12, donde se aprecia el costo (3.33) en función de la capacidad de la batería (asumimos que no hay restricciones en la rampa de carga (3.31)). Junto a esta curva se muestra el costo de operación obtenido con la política *miope*, que usa inmediatamente, de ser posible, toda la energía que precise de la batería:

$$\pi_{\text{miope}}(x_t) = [d_t - r_t]_{b_t - \bar{B}}^{b_t} \quad (3.34)$$

donde $[z]_a^b = \min\{b, \max\{a, z\}\}$ es la proyección de z al conjunto $[a, b]$. La gran diferencia en el costo medio de estas dos políticas se explica por dos razones. En primer lugar, la política óptima aprovecha el hecho de que la función de costo es cuadrática. Aprende a incurrir en costos de generación extra en momentos de poca demanda, cargando así la batería. De esta forma, se consigue disminuir el costo de generación durante los picos nocturnos usando la energía acumulada. En segundo lugar, como el control QP tiene acceso a un pronóstico perfecto, se las arreglará para poder tomar siempre el excedente de renovable.

En las dos figuras que siguen aparecen ejemplos del despacho diario siguiendo la política óptima. Tanto en la Figura 3.13 como en la Figura 3.14 se observan comportamientos similares. Durante la madrugada se aprovecha el hecho de que la demanda neta es baja para cargar la batería, a costas de un poco de generación. La batería llega plenamente cargada al mediodía en ambos casos. Paulatinamente se descarga en el correr de la tarde, llegando a carga nula en la medianoche. En el pico de consumo nocturno no queda otra que incurrir en mucha generación térmica, pero asistida por la descarga de la batería. La consigna del generador es particularmente interesante para la Figura 3.14: se obtiene un perfil de generación prácticamente uniforme en el tiempo.

3.6. Operación con pronósticos exactos

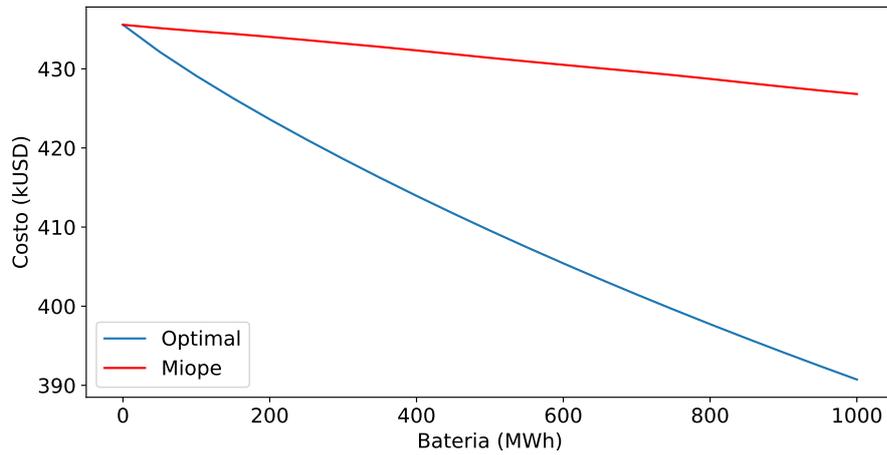


Figura 3.12: Costo medio de operar el sistema (3.33) en función de la capacidad instalada en la batería. En azul: costo obtenido con la política óptima, según (3.27)–(3.31). En rojo: costo con la política miope.

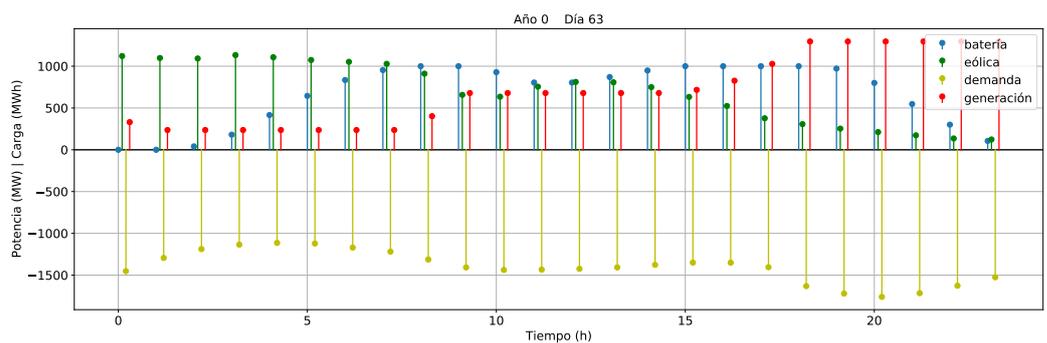


Figura 3.13: Primer ejemplo de despacho diario siguiendo la trayectoria óptima de (3.27)–(3.31). Las potencias están en MW y la carga de la batería en MWh . La suma de generación (rojo), renovable (verde) y demanda (amarillo) se equipara con la diferencia del nivel de la batería (azul) entre instantes sucesivos.

Capítulo 3. Aprendiendo a operar un banco de baterías

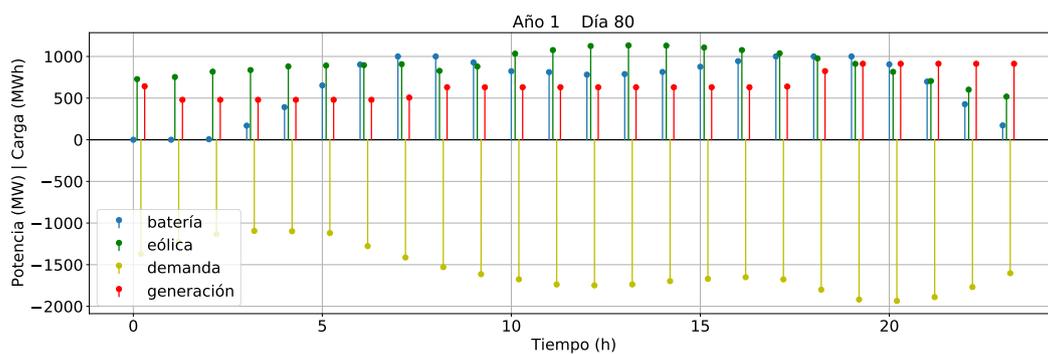


Figura 3.14: Segundo ejemplo de despacho diario siguiendo la trayectoria óptima de (3.27)–(3.31). Las potencias están en MW y la carga de la batería en MWh . La suma de generación (rojo), renovable (verde) y demanda (amarillo) se equipara con la diferencia del nivel de la batería (azul) entre instantes sucesivos.

Capítulo 4

Discusión y trabajo a futuro

En este trabajo estudiamos la operación de sistemas de almacenamiento energético en redes eléctricas. En lo que a capacidad de almacenamiento refiere, trabajamos en dos puntas del espectro: en una primera instancia estudiamos sistemas con represas hidroeléctricas, lentos y de gran porte. En la segunda nos centramos en las baterías, sistemas naturalmente más rápidos y de menor porte.

En ambos casos empleamos técnicas de programación dinámica aproximada para obtener los controladores. Una vez resuelto el problema, los controles quedan determinados implícitamente, debiendo recurrir a la resolución de un problema de optimización. La complejidad de este problema es, sin embargo, diferente para los dos enfoques usados. Para el caso de las represas, estimamos y ajustamos funciones de valor cuadráticas para cada etapa del sistema. Allí el controlador resultante se determina resolviendo un programa cuadrático, fijado el estado y el aporte. Para la operación del sistema con baterías, el entrenamiento o *aprendizaje* se basó en Q-learning. En parte debido a la gran aleatoriedad del recurso renovable, nos basamos en un enfoque libre de modelo. Este algoritmo cuenta con la ventaja de que, una vez finalizado el aprendizaje, la política de control es inmediata —sale de tomar la acción que maximiza la tabla Q para el estado actual. Aquí obtuvimos en la práctica resultados aceptables, con un agente que logró aprender una política mejor a la de un sistema sin baterías. Una de las críticas a este algoritmo es su lentitud de convergencia.

En relación al trabajo sobre el despacho hidrotérmico con represas, una posible extensión sería estudiar la incorporación de más variables al espacio de estados y acciones, y ver cómo reconciliar esto con la formulación cuadrática convexa. Por ejemplo ver cómo incorporar variables enteras, algo muy usado en esta área [53].

Para la operación del banco de baterías cabe destacar que emplear un modelo uninodal de la red facilitó el diseño del controlador, ya que la restricción de la red quedaba determinada implícitamente. Esto es, dada la inyección de la batería, la consigna del generador se obtenía en forma cerrada con la ecuación de balance de la red. A su vez, los límites de potencia a inyectar o absorber por la batería quedaban bien determinados en función de su estado de carga. En la medida en que el modelo se complejiza, considerando por ejemplo distintos nodos en la red, un modelo para el flujo de carga o almacenamiento distribuido, aparecen otras

Capítulo 4. Discusión y trabajo a futuro

restricciones a tener en cuenta, no tan sencillas de manejar: respetar los límites de potencia por las líneas, operar a niveles de tensión similares a los nominales, etc. Asimismo aumenta el espacio de acciones, que debe tener (al menos) una componente por generador y batería de la red.

Esta es una posible extensión de nuestro trabajo: considerar el problema de despacho óptimo en modelos más complejos de la red. Algunos trabajos en esta línea aplican técnicas de Control Predictivo por Modelo (*Model Predictive Control, MPC*) [54], aprovechando el conocimiento de la dinámica de transición del sistema y las restricciones de la red [55, 56, 57]. Uno de los inconvenientes de este enfoque es que el modelo debe conocerse certeramente. Una vía alternativa puede ser combinar técnicas de MPC con aprendizaje por refuerzo, como en [58]. De todas formas, cabe la pregunta de si existe la posibilidad de aprender controladores en este contexto dispensando de los modelos. En este sentido es crucial aprender rápidamente las restricciones del sistema, para evitar tomar acciones que lleven a condiciones inseguras de operación. *Aprender* las restricciones de un problema sobre la marcha, es una subárea de *Reinforcement Learning* en la que hay incipiente interés [59, 60]. Parte de nuestro trabajo más reciente [61, 62] se centra en ello —desarrollar algoritmos que den lugar a políticas que, durante la etapa de aprendizaje, aprendan rápidamente a respetar las restricciones del problema. Por el momento, sin embargo, nuestros aportes en el área consideran espacios de estado y acción discretos (al igual que los vistos en el Capítulo 3).

Otra de nuestras líneas de trabajo considera aprender directamente una representación funcional de la política de control [63, Cap.6]. Cuenta con la ventaja de que en este caso la política opera directamente sobre espacios continuos, evitando recurrir a las discretizaciones. Una avenida de estudio es como incorporar restricciones al aprendizaje, y así obtener, idealmente, políticas que i) operan sobre espacios continuos, y ii) rápidamente aprenden las restricciones del problema.

Referencias

- [1] Paul Breeze. *Power system energy storage technologies*. Academic Press, 2018.
- [2] Fu-Bao Wu, Bo Yang y Ji-Lei Ye. *Grid-Scale Energy Storage Systems and Applications*. Academic Press, 2019.
- [3] Michel Artenstein, Ricardo Franco, Alvaro Giusto, Pablo Monzón y Celia María Sena. “Estudios de estabilidad a corto plazo del sistema eléctrico uruguayo”. En: *Reporte técnico PDT 47/05* (2008). URL: <https://iie.fing.edu.uy/investigacion/grupos/ecsep/publicaciones/reportepdt.pdf>.
- [4] Agustin Castellano, Romina Silvano y Gustavo Viqueira. “Disparo automático de cargas por subfrecuencia”. En: *Tesis de grado, Universidad de la República (Uruguay)* (2017). URL: <https://www.colibri.udelar.edu.uy/jspui/handle/20.500.12008/20134>.
- [5] Chris Harris. *Electricity markets: pricing, structures and economics*. Vol. 328. John Wiley & Sons, 2006.
- [6] Georgios B Giannakis, Vassilis Kekatos, Nikolaos Gatsis, Seung-Jun Kim, Hao Zhu y Bruce F Wollenberg. “Monitoring and optimization for power grids: A signal processing perspective”. En: *IEEE Signal Processing Magazine* 30.5 (2013), págs. 107-128.
- [7] Yan Jiang, Richard Pates y Enrique Mallada. “Dynamic Droop Control in Low-inertia Power Systems”. En: *IEEE Transactions on Automatic Control* (2020), págs. 1-1. DOI: 10.1109/TAC.2020.3034198.
- [8] Yan Jiang, Eliza Cohn, Petr Vorobev y Enrique Mallada. “Storage-based frequency shaping control”. En: *IEEE Transactions on Power Systems* (2021).
- [9] Antonio Giannitrapani, Simone Paoletti, Antonio Vicino y Donato Zarrilli. “Optimal allocation of energy storage systems for voltage control in LV distribution networks”. En: *IEEE Transactions on Smart Grid* 8.6 (2016), págs. 2859-2870.

Referencias

- [10] Juan Andrés Bazerque. “Stochastic Optimization of Power Systems with Risk Constraints And Sparsely Distributed Storage”. En: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2018, págs. 3824-3828.
- [11] Lie Xu y Dong Chen. “Control and operation of a DC microgrid with variable generation and energy storage”. En: *IEEE transactions on power delivery* 26.4 (2011), págs. 2513-2522.
- [12] Rajni Kant Bansal, Pengcheng You, Dennice F Gayme y Enrique Mallada. “Storage Degradation Aware Economic Dispatch”. En: *arXiv preprint arXiv:2010.00683* (2020).
- [13] *ADME: Administración del Mercado Eléctrico*. <https://www.adme.com.uy>. [Online; visitado en Junio de 2021].
- [14] Dimitri P Bertsekas. *Dynamic programming and optimal control*. 3th edition. Vol. I.
- [15] Dimitri P Bertsekas. *Dynamic programming and optimal control*. 3th edition. Vol. II.
- [16] A. Castellano, C. Martínez, P. Monzón, J. Andrés Bazerque, A. Ferragut y F. Paganini. “Quadratic approximate dynamic programming for scheduling water resources: a case study”. En: *2020 IEEE PES Transmission Distribution Conference and Exhibition - Latin America (T D LA)* (2020), págs. 1-6. DOI: 10.1109/TDLA47668.2020.9326171.
- [17] Edsger W Dijkstra. “A note on two problems in connexion with graphs”. En: *Numerische mathematik* 1.1 (1959), págs. 269-271.
- [18] Warren B Powell. *Approximate Dynamic Programming: Solving the curses of dimensionality*. Vol. 703. John Wiley & Sons, 2007.
- [19] Agustin Castellano y Juan Andrés Bazerque. “Learning the operation of energy storage systems from real trajectories of demand and renewables”. En: *2020 IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT)*. 2020, págs. 1-5. DOI: 10.1109/ISGT45199.2020.9087648.
- [20] *Electric Power Research Institute (EPRI): Electricity Energy Storage Technology Options*. (accedido junio 2021). URL: <https://www.epri.com/research/products/000000000001020676>.
- [21] Tsvetan Asamov, Daniel F. Salas y Warren B. Powell. *SDDP vs. ADP: The Effect of Dimensionality in Multistage Stochastic Optimization for Grid Level Energy Storage*. 2016. arXiv: 1605.01521 [math.OC].
- [22] Eduardo D Sontag. *Mathematical control theory: deterministic finite dimensional systems*. Vol. 6. Springer Science & Business Media, 2013.

- [23] Dimitri P Bertsekas. *Dynamic programming and optimal control*. 4th edition. Vol. I. 2.
- [24] Dimitri P Bertsekas. *Reinforcement learning and optimal control*. Athena Scientific Belmont, MA, 2019.
- [25] Mario VF Pereira y Leontina MVG Pinto. “Multi-stage stochastic optimization applied to energy planning”. En: *Mathematical programming* 52.1-3 (1991), págs. 359-375.
- [26] *portal UTE: fuentes de generación*. (accedido diciembre 2020). URL: <https://www.ute.com.uy/institucional/infraestructura/fuentes-de-generacion>.
- [27] Gonzalo Casaravilla, Ruben Chaer y Pablo Alfaro. “SimSEE: Simulador de sistemas de energía eléctrica”. En: *Proyecto PDT 47/12. Technical Report 7, Universidad de la República (Uruguay). Facultad de Ingeniería. Instituto de Ingeniería Eléctrica, Number 7-Dec, Tech. Rep.* (2008).
- [28] Svante Wold, Kim Esbensen y Paul Geladi. “Principal component analysis”. En: *Chemometrics and intelligent laboratory systems* 2.1-3 (1987), págs. 37-52.
- [29] Hervé Abdi y Lynne J Williams. “Principal component analysis”. En: *Wiley interdisciplinary reviews: computational statistics* 2.4 (2010), págs. 433-459.
- [30] Shi Yu, Léon-Charles Tranchevent, Bart De Moor e Yves Moreau. *Kernel-based data fusion for machine learning*. Springer, 2013.
- [31] Stephen Boyd y Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.
- [32] Lieven Vandenberghe y Stephen Boyd. “Semidefinite programming”. En: *SIAM review* 38.1 (1996), págs. 49-95.
- [33] Steven Diamond y Stephen Boyd. “CVXPY: A Python-embedded modeling language for convex optimization”. En: *Journal of Machine Learning Research* 17.83 (2016), págs. 1-5.
- [34] Akshay Agrawal, Robin Verschueren, Steven Diamond y Stephen Boyd. “A rewriting system for convex optimization problems”. En: *Journal of Control and Decision* 5.1 (2018), págs. 42-60.
- [35] *Gurobi optimization*. <https://www.gurobi.com>. [Online; visitado en Febrero de 2021].
- [36] *Mosek ApS*. <https://www.mosek.com>. [Online; visitado en Febrero de 2021].
- [37] Christopher John Cornish Hellaby Watkins. “Learning from delayed rewards”. En: *Ph.D. Thesis – King’s College, Cambridge United Kingdom* (1989).

Referencias

- [38] Christopher JCH Watkins y Peter Dayan. “Q-learning”. En: *Machine learning* 8.3-4 (1992), págs. 279-292.
- [39] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. 2.^a ed. John Wiley & Sons, 2014.
- [40] Simon S Haykin. *Adaptive filter theory*. Pearson Education India, 2008.
- [41] Trevor Hastie, Robert Tibshirani y Jerome Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.
- [42] *Apagón eléctrico de Argentina, Paraguay y Uruguay de 2019 (Wikipedia)*. https://es.wikipedia.org/wiki/Apagon_electrico_de_Argentina,_Paraguay_y_Uruguay_de_2019. [Online; visitado en Febrero de 2021].
- [43] John N Tsitsiklis. “Asynchronous stochastic approximation and Q-learning”. En: *Machine learning* 16.3 (1994), págs. 185-202.
- [44] “Tesla builds world’s largest battery in Australian outback”. En: *LA Times* ((accedido junio 2021)). URL: <https://www.latimes.com/business/technology/la-fi-tn-tesla-australia-20171201-story.html>.
- [45] “LS Power Energizes Largest Battery Storage Project in the World, The 250 MW Gateway Project in California”. En: *Cision PR Newswire* ((accedido junio 2021)). URL: <https://www.greentechmedia.com/articles/read/ls-power-energizes-worlds-biggest-battery-near-san-diego-just-in-time-for-heatwave>.
- [46] “Australia Picks Massive Tesla-Supplied Battery to Ease Transmission Constraint”. En: *Greentech Media* ((accedido junio 2021)). URL: <https://www.greentechmedia.com/articles/read/australia-picks-massive-tesla-battery-to-ease-transmission-constraint>.
- [47] Fernando Pérez y Brian E. Granger. “IPython: a System for Interactive Scientific Computing”. En: *Computing in Science and Engineering* 9.3 (mayo de 2007), págs. 21-29. ISSN: 1521-9615. DOI: 10.1109/MCSE.2007.53. URL: <https://ipython.org>.
- [48] *Google Colab*. (accedido mayo 2021). URL: <https://www.colab.research.google.com>.
- [49] Csaba Szepesvári. “The asymptotic convergence-rate of Q-learning”. En: *NIPS*. Vol. 10. Citeseer. 1997, págs. 1064-1070.
- [50] Steven M Kay. *Fundamentals of statistical signal processing*. Prentice Hall PTR, 1993.
- [51] *Ingeniería de Muestra 2019*. (accedido mayo 2021). URL: <http://uri.pages.fing.edu.uy/idm.uy/2019/proyectos.html#IIE>.

- [52] Bartolomeo Stellato, Goran Banjac, Paul Goulart, Alberto Bemporad y Stephen Boyd. “OSQP: An operator splitting solver for quadratic programs”. En: *Mathematical Programming Computation* (2020), págs. 1-36.
- [53] Alberto Borghetti, Claudia D’Ambrosio, Andrea Lodi y Silvano Martello. “An MILP approach for short-term hydro scheduling and unit commitment with head-dependent reservoir”. En: *IEEE Transactions on power systems* 23.3 (2008), págs. 1115-1124.
- [54] Francesco Borrelli, Alberto Bemporad y Manfred Morari. *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017.
- [55] Le Xie, Yingzhong Gu, Ali Eskandari y Mehrdad Ehsani. “Fast MPC-based coordination of wind power and battery energy storage systems”. En: *Journal of Energy Engineering* 138.2 (2012), págs. 43-53.
- [56] Salem Batiyah, Roshan Sharma, Sherif Abdelwahed y Nasibeh Zohra-bi. “An MPC-based power management of standalone DC microgrid with energy storage”. En: *International Journal of Electrical Power & Energy Systems* 120 (2020), pág. 105949.
- [57] Josip Vasilj, Sebastien Gros, Damir Jakus y Mario Zanon. “Day-ahead scheduling and real-time economic MPC of CHP unit in microgrid with smart buildings”. En: *IEEE Transactions on Smart Grid* 10.2 (2017), págs. 1992-2001.
- [58] Mario Zanon y Sébastien Gros. “Safe reinforcement learning using robust MPC”. En: *IEEE Transactions on Automatic Control* (2020).
- [59] Sebastian Junges, Nils Jansen, Christian Dehnert, Ufuk Topcu y Joost-Pieter Katoen. “Safety-constrained reinforcement learning for MDPs”. En: *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer. 2016, págs. 130-146.
- [60] Santiago Paternain, Miguel Calvo-Fullana, Luiz FO Chamon y Alejandro Ribeiro. “Learning safe policies via primal-dual methods”. En: *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE. 2019, págs. 6491-6497.
- [61] Agustin Castellano, Juan Bazerque y Enrique Mallada. “Learning to be safe, in finite time”. En: *2021 American Control Conference (ACC)* (2021).
- [62] Agustin Castellano, Hancheng Min, Juan Bazerque y Enrique Mallada. “Learning to Act Safely with Limited Exposure and Almost Sure Certainty”. En: *arXiv preprint arXiv:2105.08748* (2021).

Referencias

- [63] Juan Bazerque. “Optimización del almacenamiento de energía en redes eléctricas”. En: *Reporte técnico: Proyecto ANII-FSE-1-2017-1-145060* (2020). URL: https://www.anii.org.uy/proyectos/FSE_1_2017_1_145060/optimizacion-del-almacenamiento-de-energia-en-redes-electricas.

Índice de tablas

2.1. Ilustración de la Maldición de la Dimensionalidad: cantidad de valores que puede tomar la variable de estados en función del número de represas, usando $m = 20$ discretizaciones por componente del vector de estados.	8
2.2. Represas de Uruguay con su embalse nominal y capacidad de turbinado [26]. $1hm^3 \sim 1$ hectómetro cúbico.	9
3.1. Generadores hidrotérmicos del Uruguay: potencia instalada según orden de costo. Notar cómo los generadores de menor costo son los hidroeléctricos.	32
3.2. Parejas generación-costo para el generador equivalente Uruguayo. .	32
3.3. Factores de corrección aplicados a cada serie temporal de demanda y viento.	36
3.4. Resumen con las distintas variables y parámetros empleados. . . .	41

Esta página ha sido intencionalmente dejada en blanco.

Índice de figuras

1.1. Despacho de generación por fuentes para el sistema eléctrico uruguayo, del 2 al 4 de Junio de 2021. Los tipos de generación son: solar (amarillo), eólica (verde), hidráulica (celestes y azules), térmica (rojo) y biomasa (naranja). El trazo punteado color piel corresponde a la demanda nacional, los demás trazos a las exportaciones a Brasil y Argentina. La demanda de cada día presenta tres zonas a destacar: un pico durante la tarde-noche, un valle durante la madrugada y una meseta durante el día. Notar la gran penetración de fuentes renovables (en particular eólica). Prácticamente en todas las horas el despacho se hace al 100 % con fuentes renovables, teniendo que asistir con generación térmica únicamente sobre el mediodía del 2 y 3 de Junio. Fuente: ADME [13].	2
1.2. Comparación de distintos sistemas de almacenamiento energético. Los sistemas se agrupan según la potencia instalada (abscisas) y el tiempo que le lleva descargarse completamente (ordenadas). Fuente: EPRI [20].	4
2.1. Represas de Uruguay. Salto Grande en el Norte, y las represas de Río Negro, de este a oeste Bonete, Baygorria, Palmar. Bonete (marcada en rojo) contiene el embalse más grande, y por ende la mayor capacidad de reserva.	9
2.2. Visualización del embalse y capacidad de turbinado para las represas de Uruguay. Notar cómo la principal capacidad de almacenamiento se da en Rincón del Bonete.	10
2.3. Ejemplo de aportes históricos (“crónicas”) en las tres principales represas de Uruguay.	11
2.4. Mediana de aportes semanales (en hm^3) para cada una de las tres represas. Para cada semana la mediana se computa a lo largo de los 105 años de crónicas (1909–2013).	12
2.5. Histograma con los aportes de las series normalizadas. Cada serie se divide por su mediana semanal (Figura 2.4) y luego se le aplica una transformación logarítmica.	13

Índice de figuras

2.6. Clusters resultantes al aplicar PCA. Los clusters están asociados a la hidraulicidad del estado hidrológico: el cluster azul tiene aportes pequeños y se corresponde con un estado <i>muy seco</i> , mientras que el cluster amarillo tiene aportes grandes y se corresponde con un estado <i>muy húmedo</i>	14
2.7. Funciones de valor (<i>cost-to-go</i>) en función del nivel de agua de Bonete para cada uno de los posibles estados hidrológicos. Notar cómo el costo esperado es más alto para estados hidrológicos más secos y decrece a medida que la reserva aumenta.	16
2.8. <i>Cost-to-go</i> en función del nivel de Bonete y Salto, para Palmar vacío y el estado hidrológico más seco. Notar cómo la mayor variabilidad en el costo se da en la dirección de Bonete.	17
2.9. Ajuste cuadrático para la semana 40 y estado hidrológico seco ($e_{40} = 1$). La grilla de estados contiene $N_0 = 10$ para Bonete y tres puntos para las otras tres represas. En rojo: parejas estado-costo (muestreadas con (2.13)–(2.14)). En azul: sábana cuadrática (obtenida usando (2.16)). El <i>cost-to-go</i> es menor cuanto más reserva haya y depende principalmente de Bonete (la represa con mayor embalse).	23
2.10. Ajuste cuadrático para la semana 48 y estado hidrológico húmedo ($e_{48} = 3$). La grilla de estados contiene $N_0 = 5$ puntos para Bonete y tres puntos para las otras tres represas. En rojo: parejas estado-costo (muestreadas con (2.13)–(2.14)). En azul: sábana cuadrática (obtenida usando (2.16)). El <i>cost-to-go</i> depende principalmente de Bonete. Sin embargo, en este caso la variabilidad en Salto también afecta el costo (comparar con la Figura 2.9). Esto se debe a que se está cerca del fin del horizonte.	23
2.11. Costo de operación anual en función del nivel inicial de Bonete para $e_0 = 2$. En violeta: costo predicho por la cuadrática $\hat{V}_{0,2}$. En azul sólido: costo promedio para $T = 105$ trayectorias. En azul sombreado: intervalo dado por $\pm\sigma/\sqrt{T}$ siendo σ la desviación muestral. En negro: cota inferior obtenida con (2.23)	24
2.12. Comparación entre el desempeño de la política miope (azul) y la política aprendida (naranja) en función del nivel inicial de Bonete, junto a una cota inferior para el costo. Nuestra política consigue un 4% de reducción en costo respecto a la miope cuando Bonete está a medio llenar.	24
2.13. Desviación relativa en el costo simulado hacia adelante para distintas políticas, en función del nivel inicial de Bonete. Cada trazo corresponde al costo de una política entrenada con un paso de discretización de Bonete (N_0) distinto. Se usa como base $N_0 = 10$. Observar que las diferencias en costo no son sustanciales.	25

2.14. Costo de operación promedio en función del nivel inicial de Bonete. Las trayectorias de aportes coinciden con los datos históricos. La política entrenada con el modelo descrito en 2.2.3 (PCA) obtiene en el mejor caso una reducción del 2,4 % en costo frente a una entrenada con el modelo de energía entrante (MEE). 26

3.1. Modelo uninodal de la red. El generador (g_t), la renovable disponible (r_t), la demanda (d_t) y la potencia de la batería (p_t) se inyectan a la red. 28

3.2. Función de costo para el generador equivalente. Los puntos negros son las parejas generación-costos de la Tabla 3.2 33

3.3. Ejemplos de demanda diaria para el invierno de 2019. Se muestran 10 días tomados al azar. Los perfiles de cada día son similares; se distinguen tres zonas: valle (de madrugada), meseta (durante el día), pico (en la noche). 34

3.4. Perfiles diarios de viento para el invierno de 2019. Se muestran 10 días tomados al azar. La disponibilidad del viento es mucho más variable que la demanda (comparar con Figura 3.3). 35

3.5. Apagón masivo del 16 de junio de 2019. Este día fue removido del dataset de entrenamiento. 35

3.6. Demanda promedio para los inviernos de 2017, 2018 y 2019. En trazo sólido: demanda media. En trazo sombreado: desviación respecto a la media de $\pm\sigma/\sqrt{N}$, donde σ es la desviación muestral y N la cantidad de muestras (días del invierno). La varianza de la demanda es pequeña. La demanda crece en promedio cada año. 36

3.7. Potencia eólica promedio para los inviernos de 2017, 2018 y 2019. En trazo sólido: potencia media. En trazo sombreado: desviación respecto a la media de $\pm\sigma/\sqrt{N}$, donde σ es la desviación muestral y N la cantidad de muestras (días del invierno). Observar cómo la varianza del viento es grande (en comparación con la demanda, cf. Figura 3.6). En promedio, el invierno de 2017 fue menos ventoso que el de los otros años. 37

3.8. Curva de aprendizaje mostrando el costo promedio de un episodio en función del entrenamiento. En rojo punteado se muestra el costo promedio (exacto) que se obtiene con una política sin baterías. Luego de aproximadamente 500000 iteraciones, la política aprendida supera en desempeño a la política sin baterías. 42

3.9. Ejemplo de despacho hora a hora con la política aprendida. En azul: nivel de la batería (MWh); en verde y amarillo: potencia eólica y demanda (MW); en rojo: generación térmica (MW). En el instante t , la suma de la renovable, generación térmica y demanda es igual a la diferencia del nivel de la batería entre t y $t + 1$. Notar cómo se carga la batería durante la mañana (poca demanda) a costas de generación extra, y se descarga en la tarde (mayor demanda). . . . 43

Índice de figuras

3.10. Autocorrelación parcial de la serie temporal de viento, año 2019. La autocorrelación es significativa para los <i>lags</i> de orden 1 y 2 y despreciable para el resto. Un modelo AR-2 se ajustaría bien a esta serie.	44
3.11. Autocorrelación parcial de la serie temporal de demanda, año 2019. Los lags de primer y segundo orden son los más significativos. Sin embargo, los lags de orden 15–20 son considerables. En este caso un modelo AR-2 no ajustaría bien.	45
3.12. Costo medio de operar el sistema (3.33) en función de la capacidad instalada en la batería. En azul: costo obtenido con la política óptima, según (3.27)–(3.31). En rojo: costo con la política miope.	47
3.13. Primer ejemplo de despacho diario siguiendo la trayectoria óptima de (3.27)–(3.31). Las potencias están en <i>MW</i> y la carga de la batería en <i>MWh</i> . La suma de generación (rojo), renovable (verde) y demanda (amarillo) se equipara con la diferencia del nivel de la batería (azul) entre instantes sucesivos.	47
3.14. Segundo ejemplo de despacho diario siguiendo la trayectoria óptima de (3.27)–(3.31). Las potencias están en <i>MW</i> y la carga de la batería en <i>MWh</i> . La suma de generación (rojo), renovable (verde) y demanda (amarillo) se equipara con la diferencia del nivel de la batería (azul) entre instantes sucesivos.	48

Esta es la última página.
Compilado el lunes 26 julio, 2021.
\DocumentURL