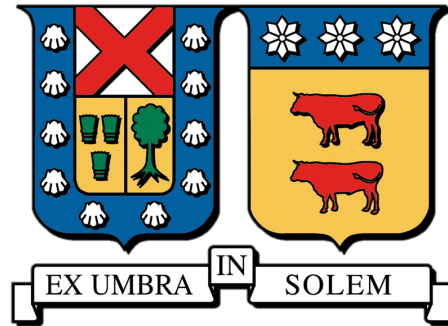


UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA

DEPARTAMENTO DE INGENIERÍA ELECTRÓNICA



IMPLEMENTACIÓN DE CONTROL PREDICTIVO
EN PLANTAS PV UTILIZANDO MILP

DIEGO ALEJANDRO CRUZ CARREÑO

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL ELECTRÓNICO

PROFESOR/A GUÍA: JUAN CARLOS AGÜERO

PROFESOR/ES CO-REFERENTE/S:

GONZALO CARVAJAL

ANGEL CEDEÑO

Agosto-2024



CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): Memoria o trabajo de título; Tesis de Postgrado;

Título del trabajo: IMPLEMENTACIÓN DE CONTROL PREDICTIVO EN PLANTAS PV UTILIZANDO MILP

Nombre del candidato(a): Diego A. Cruz Carreño

Carrera / Grado: Ingeniería Civil Electrónica

Campus: Casa Central Valparaíso ; Departamento: Electrónica

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Juan C. Aguiar, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL

El trabajo **NO contiene información que amerite confidencialidad** y puede ser publicado de inmediato en repositorio con acceso abierto.

El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (embargo) por:

6 meses; 12 meses; 2 años; 3 años; 5 años; 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 21/08/2025 ; Firma: [Firma]

Estudiante o Candidato(a):

Fecha: 07-07-2025 ; Firma: [Firma]

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.

Resumen

Este proyecto contempla el uso de control predictivo (MPC) aplicado a plantas fotovoltaicas(PV) y a su sistema de almacenamiento de energía en baterías (BESS), utilizando como algoritmo de optimización un modelo de programación lineal entera y mixta (MILP), donde el objetivo es maximizar el rédito económico de la planta fotovoltaica.

Temas como el modelado de plantas fotovoltaicas con sistema de bancos de batería son tratados en éste documento con el objetivo de aplicar un esquema de control predictivo utilizando programación lineal entera y mixta, esto con la ayuda del *solver* “Cplex”.

Para poder utilizar Cplex se utilizara el entorno de programación “MATLab”, el cual permite utilizar su *solver* para MILP como complemento. Además MATLab nos permite generar gráficos donde se pueden ver los resultados sencillamente.

BESS en Plantas Fotovoltaicas, Cplex, MATLab, programación lineal entera y mixta, modelo de control predictivo, optimización.

Abstract

This project involves the use of Model Predictive Control (MPC) applied to photovoltaic plants (PV) and their Battery Energy Storage System (BESS), using a Mixed Integer Linear Programming (MILP) model as the optimization algorithm, where the objective is to maximize the economic returns of the photovoltaic plant.

Topics such as the modeling of photovoltaic plants with battery storage systems are discussed in this document with the aim of applying a predictive control scheme using Mixed Integer Linear Programming (MILP), with the help of the “Cplex” solver.

To use Cplex, the MATLAB programming environment will be employed, which allows the use of its solver for MILP as a complement. Additionally, MATLAB allows us to generate graphs where the results can be easily visualized.

BESS in Photovoltaic Plants, Cplex, MATLAB, Mixed Integer Linear Programming, Predictive Control Model, Optimization.

Agradecimientos

Quiero agradecer profundamente, y como no podría ser de otra forma, a mi familia y mi novia, la que en este punto ya es parte de de mi familia. Ustedes siempre han estado ahí conmigo apoyándome durante todo este ciclo universitario, tanto en los momentos buenos como en los momentos mas malos. Las dificultades que hemos pasado juntos me han enseñado a perseverar incluso cuando pensaba que todo estaba perdido, enseñándome que esos momentos son pasajeros, incitando a que me levante y siga por el camino que elegí.

A mis amigos, aquellos que lo han sido incluso de antes de la universidad como a los que he conocido en esta institución. Ustedes me ayudaron a darme cuenta que no estoy solo, que todos tienen problemas y sufren situaciones parecidas. Gracias por los momentos alegres compartidos dentro y fuera de la universidad, por los momentos en que trasnochamos estudiando para aprovechar cada minuto antes de un certamen. Gracias por todo.

A mis profesores, quienes me han formado como profesional, siempre receptivos y pendientes de las preguntas que no haya logrado responder por mi mismo. En especial agradecer a mi profesor guía, quien ha tenido paciencia infinita y un nivel de comprensión admirable. Además de agradecer al centro ANID-Basal AC3E Project Grant AFB240002 y a los proyectos ANID-Fondecyt 1211630 y 3240181. Gracias.

Si no fuera por ustedes esto habrá sido imposible.

Tabla de Contenidos

1	Introducción	1
1.1	Estado del arte	2
1.1.1	Plantas Fotovoltaicas	2
1.1.2	Mercado de la energía	3
1.1.3	Sistema de almacenamiento de energía en baterías	5
1.1.4	Control predictivo basado en modelos (MPC)	5
1.1.5	Programación lineal entera mixta	7
1.1.6	MATLab y Cplex	8
2	Objetivos y Alcances	9
2.1	Objetivos específicos	9
3	Modelo de control predictivo para una planta PV	11
3.1	Especificación del problema	11
3.2	Problema de optimización	14
3.3	Variables binarias	17

3.4	Equivalencia de variables binarias	18
3.5	Reformulación de problema de optimización	20
3.6	Validación del modelo con dos variables binarias	22
3.6.1	Representación matricial	23
3.6.2	Sintaxis para simulación en MATLAB	25
3.6.3	Simulación	27
3.7	Desgaste de batería	29
3.7.1	Propuesta	30
3.8	Problema de optimización con tres variables binarias	32
3.8.1	Restricciones	32
3.8.2	Reformulación del problema de optimización	32
3.8.3	Representación matricial	34
3.8.4	Simulación	36
3.9	Conflicto en modelo con tres variables y solución	38
3.9.1	Propuesta	38
3.10	Problema de optimización con tres variables binarias y una entera	39
3.10.1	Restricciones	39
3.10.2	Reformulación del problema de optimización	41
3.10.3	Representación matricial	42
3.10.4	Simulación	44

3.11	Unificación de métodos	46
3.11.1	Propuesta	46
3.12	Problema de optimización con tres variables binarias y conteo de intervalos donde $P^{ES} \neq 0$	50
3.12.1	Restricciones	50
3.12.2	Reformulación del problema de optimización	50
3.12.3	Representación matricial	52
3.12.4	Simulación	54
3.13	Análisis comparativo de resultados	56
3.13.1	Factores de peso $c_{sw} = c_{bls} = 0$	56
3.13.2	Factores de peso $c_{sw} = 700, c_{bls} = 50$	57
3.13.3	Factores de peso $c_{sw} = 700, c_{bls} = 100$	59
3.13.4	Factores de peso $c_{sw} = 1400, c_{bls} = 50$	61
3.13.5	Factores de peso $c_{sw} = 1400, c_{bls} = 100$	63
4	Conclusiones y Recomendaciones	67
4.1	Conclusiones y resultados	67
4.2	Recomendaciones	68

Lista de Tablas

3.1	Parámetros de simulación	27
3.2	Tabla de verdad	30
3.3	Valores variables binarias para distintos P^{ES}	48

Lista de Figuras

1.1	Celda fotovoltaica	2
1.2	Precios del mercado energético	4
1.3	Diagrama de bloques MPC	5
3.1	Diagrama del modelo de control	12
3.2	Datos de simulación	27
3.3	Resultados de simulación con 165 períodos	28
3.4	Desbalance de potencia	29
3.5	Resultados de simulación con 165 períodos	37
3.6	Resultados de simulación con 165 períodos	45
3.7	Resultados de simulación con 165 períodos	55
3.8	E^{ES} para 2 variables, 4 variables y conteo de $P^{ES} \neq 0$	56
3.9	Comparación simulación para 2 variables, 4 variables y conteo $P^{ES} \neq 0$	57
3.10	P^{ES} para 2 y 4 variables	57
3.11	P^{ES} para 2 variables binarias, y conteo de $P^{ES} \neq 0$	58

3.12	P^{ES} para 4 variables, y conteo de $P^{ES} \neq 0$	58
3.13	Comparación simulación para 2 variables, 4 variables y conteo $P^{ES} \neq 0$	59
3.14	P^{ES} para 2 variables binarias, y conteo de $P^{ES} \neq 0$	59
3.15	P^{ES} para 4 variables, y conteo de $P^{ES} \neq 0$	60
3.16	Comparación rédito económico para 2 variables binarias y conteo $P^{ES} \neq 0$	61
3.17	P^{ES} para 2 variables binarias y 4 variables	61
3.18	P^{ES} para 4 variables, y conteo de $P^{ES} \neq 0$	62
3.19	Comparación rédito económico para 2 variables binarias y 4 variables	63
3.20	P^{ES} para 4 variables, y conteo de $P^{ES} \neq 0$	63
3.21	P^{ES} para 4 variables, y conteo de $P^{ES} \neq 0$	64
3.22	Comparación rédito económico para 4 variables, y conteo de $P^{ES} \neq 0$	65
3.23	Comparación rédito económico para 4 variables, y conteo de $P^{ES} \neq 0$	65
3.24	P^{ES} para 4 variables, y conteo de $P^{ES} \neq 0$	66

Capítulo 1

Introducción

En este capítulo entraremos en el contexto de la situación, y el porque de este trabajo. Se pondrá un poco de contexto en cuanto a la operación de las plantas fotovoltaicas y porqué son utilizadas como fuente de energía hoy en día. Por otro lado, y siguiendo la línea de las plantas fotovoltaicas (plantas PV), se hablara un poco acerca del mercado energético y como estas plantas participan de en él.

En consecuencia a lo anterior, se dará una breve introducción al uso de “Sistema de Almacenamiento de Energía en Baterías” (BESS) y en que medida resultan necesarios para tener una mayor impacto en el mercado energético antes mencionado.

A partir de lo anterior se verá en este documento el porque es necesario realizar un control sobre la BESS de la planta fotovoltaica, en cuanto a la carga y descarga, para maximizar las ganancias económicas recibidas de la venta de la energía generada.

1.1 Estado del arte

1.1.1 Plantas Fotovoltaicas

La energía solar es quizá el tipo de energía renovable mas conocida que existe, y las plantas fotovoltaicas son aquellas que aprovechan esta energía entregada por el sol. Esto lo hacen mediante paneles fotovoltaicos, los cuales están compuestos de una serie de celdas fotovoltaicas que reciben el impacto de fotones (rayos solares) en ellas generando corriente, lo que se conoce como “efecto fotoelectrico”.

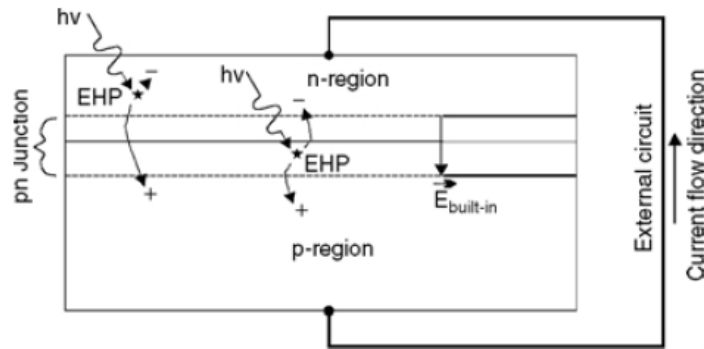


Figura 1.1: Celda fotovoltaica

Fuente: Imagen recuperada de [9].

Las celdas fotovoltaicas típicas suelen estar formadas de dos capas de semiconductor a las cuales se les llama “regiones” (como se ve en la figura 1.1). Los fotones impactan en el semiconductor generando una excitación en los electrones de este, ocasionando que salten de una región a otra, lo cual genera una diferencia de potencial entre ambas regiones, lo que finalmente genera una corriente continua, es decir, la energía solar es finalmente convertida en energía eléctrica. Posteriormente se utiliza un inversor para poder obtener la corriente alterna que esta lista para ser distribuida a la red, o ser almacenada en sistemas de almacenamiento en baterías.

Sin embargo, este tipo de energía renovable tiene el problema que depende mucho de las condiciones del clima en el momento de generación de energía, por lo que no les es posible mantener un suministro constante de energía en cada momento del día. Para solucionar esta problemática es que se busca predecir la energía generada por la planta en distintos tramos del día, para así buscar los momentos idóneos en que se deba almacenar la energía o suministrarla a la red.

1.1.2 Mercado de la energía

El mercado de la energía recientemente a ido transicionando de ser un mercado “regulado” a uno “no regulado” en varios países [1], es decir, en estos países ya no existe una entidad reguladora del mercado energético que pueda controlar los precios de esta o controlar los actores involucrados en la generación y distribución de la energía. Gracias a esto se genera una competencia entre los diferentes actores en este mercado, ocasionando que se busquen nuevas tecnologías que ayuden a generar un buen posicionamiento.

En estos mercados no regulados (como por ejemplo el mercado español) se “subasta” la demanda de energía donde cada participante del mercado informa la cantidad de energía que esta dispuesto a ofertar, además del precio por MW/h que será entregada. Al mismo tiempo que los compradores (generalmente distribuidoras de energía) informan la cantidad de energía que requieren en ese momento. Luego el precio final de la energía se establece en el ultimo precio ofertado con el que se supla la demanda de energía. Por ejemplo:

- Suponiendo que los participantes del mercado ofrecen los siguientes precios:
 - Participante A: 10MW a \$20 MW/h
 - Participante B: 20MW a \$30 MW/h
 - Participante C: 30MW a \$40 MW/h
 - Participante D: 40MW a \$50 MW/h
- El comprador u operador de mercado demanda 35MW.

En el ejemplo anterior, la demanda de 35MW se suple con toda la energía ofertada por A y B, además de parte de la energía ofertada por C. Como el ultimo actuador al que se le compra energía es al participante C el precio se establece según este lo requiera. Entonces toda la energía que se compre será al precio ofertado por el participante C, es decir \$40 MW/h, incluso a los participantes A y B, a pesar de que estos hayan ofertado a un precio menor, se les compra al precio ajustado. Estas cantidades de energía son comprometidas en periodos de tiempo que usualmente son de una o mas horas, como se puede observar en la figura 1.2.

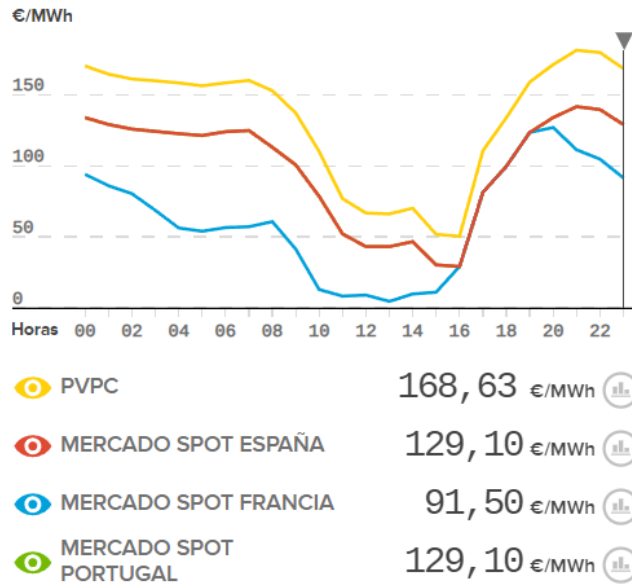


Figura 1.2: Precios del mercado energético

Imagen extraída de [3].

En la figura 1.2 se pueden ver los precios de la energía a distintas horas del día, donde es posible observar que a las horas de mayor consumo, es decir, antes de las diez de la mañana y después de las cuatro de la tarde, es cuando el precio de la energía se eleva.

Sin embargo, como esto se establece con anterioridad al momento de proveer efectivamente la energía, en ocasiones ocurre que la energía que entrega alguno de los participantes del mercado no es suficiente para cumplir con el compromiso, generando un imbalance de energía en la red. En estos casos, para cumplir con la energía requerida (35MW en el ejemplo) los operadores deben comprar energía disponible a otros participantes del mercado a un precio mayor al establecido con anterioridad (precio de imbalance). La diferencia del precio establecido y el precio de imbalance, es cargado como penalización a aquel participante que ha provocado el imbalance de energía en la red.

1.1.3 Sistema de almacenamiento de energía en baterías

Los sistemas de almacenamiento de energía en baterías, o por sus siglas “BESS” (del inglés “Battery Energy Storage Systems”), son tecnologías que, como su nombre indica, permiten el almacenamiento de energía para su uso posterior. Principalmente se utilizan para facilitar una gestión eficiente de la energía que se entrega a la red principal.

Las BESS cuentan con un sistema de control que permite optimizar el uso de las baterías, gestionando los momentos de carga y descarga de manera idónea, buscando, usualmente, el mayor rédito económico.

Este tipo de sistemas se utilizan particularmente en plantas generadoras de energías renovables, como es el caso de las plantas fotovoltaicas, ya que nos ayudan con los problemas.

Estos sistemas de control permiten minimizar las penalizaciones económicas vistas en la sección 1.1.2, o en su defecto maximizar las ganancias.

1.1.4 Control predictivo basado en modelos (MPC)

El control predictivo basado en modelos (MPC por sus siglas en inglés “Model Predictive Control”) es una estrategia de control avanzada que tiene la capacidad de gestionar múltiples variables al mismo tiempo. A diferencia de estrategias de control como el control PID (Proporcional-Integral-Derivativo), MPC se basa en un modelo que predice los estados futuros del sistema, resultando en un control con más información [2]. El diagrama de bloques de esta estrategia de control se puede ver a continuación.

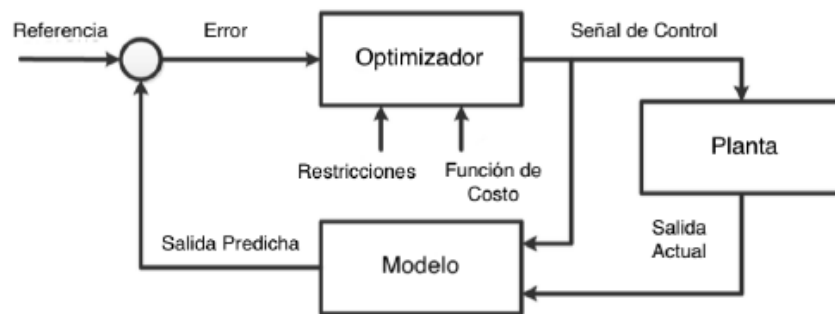


Figura 1.3: Diagrama de bloques MPC

En definitiva, y como se menciona con anterioridad, MPC optimiza la toma de decisiones de control prediciendo el estado futuro de la planta a lo largo del llamado “horizonte de predicción”, que se refiere al lapso de tiempo hacia el futuro en el que llega a predecirse el estado de la planta. Esto se logra gracias a que en cada instante de tiempo se calcula, a partir del modelo, como evolucionará el sistema en función de las acciones actuales y futuras, teniendo en cuenta las restricciones inherentes al problema y a las impuestas con tal de cumplir el objetivo de control. Esta capacidad de predecir el futuro convierte a la estrategia de MPC en una sumamente dinámica y adaptable a situaciones de incertidumbre.

La metodología de MPC se basa en la iteración de tres pasos en cada instante de tiempo sobre un horizonte de predicción “N”. Estos pasos son:

1. Predicción: A partir del modelo y las variables de entrada del instante “t” actual se predice las salidas futuras y el estado de la planta de todos los siguientes instantes de tiempo dentro del horizonte de predicción para un conjunto de entradas futuras.
2. Optimizar: El pronóstico anterior se utiliza para calcular las señales de control que optimizan la función de costo asociada al modelo para todo el horizonte de predicción.
3. Control: Por último, ya calculadas todas las señales de control que optimizan la función del modelo en todo el horizonte de predicción, se implementa solo la primera señal de control obtenida.

Este proceso se repite en cada instante de tiempo descartando el resto de las señales de control obtenidas (se aplica la primera en cada instante), ya que al ser solo un modelo y no la planta real, el estado de esta no será exactamente el esperado, además de estar sujeta a otro tipo de factores externos que no siempre son predecibles dentro de un modelo.

El desafío que encuentra MPC es el tiempo que toma calcular la variable de control óptima en cada instante de tiempo, ya que este debe ser lo suficientemente bajo para controlar correctamente la planta. Si una planta necesita ser controlada en intervalos de tiempo demasiado bajos, es probable que MPC no sea una buena estrategia, ya que el tiempo de cálculo puede llegar a ser mayor al intervalo de tiempo respectivo.

A pesar de lo anterior, es una herramienta poderosa cuando se busca controlar sistemas donde el tiempo de cálculo no sea un problema, por ejemplo en aplicaciones ligadas a la generación y venta de energías renovables.

1.1.5 Programación lineal entera mixta

La “programación lineal entera mixta”, o MILP por sus siglas en inglés para “Mixed Integer Linear Programming”, es una extensión de la “programación lineal” (LP), y como tal comparten el ser técnicas matemáticas utilizadas para encontrar la solución óptima de un problema en específico.

Para lograr su cometido, MILP (y LP) se basa en maximizar o minimizar una ganancia o costo respectivamente. Este objetivo es representado a partir de un modelo que permita cuantificar la ganancia o costo mediante una función matemática que es llamada “función de costo”, siendo esta la función que se busca optimizar (maximizar o minimizar).

La optimización de la función de costo se consigue a través del cálculo de las “variables de decisión”, las cuales representan el aspecto controlable del problema. Cuando estas variables toman los valores que consiguen la optimización del problema, al reemplazarlas en la función de costo se obtiene la solución óptima. Las variables del problema están sometidas a restricciones inherentes éste, ya sean las variables de decisión o variables que no son directamente controlables. Estas restricciones sirven para definir el espacio factible en el que puede existir la solución óptima.

MILP y LP comparten que la función de costo y las restricciones tienen la característica de ser lineales en sus variables, sin embargo MILP se diferencia en la presencia de variables enteras (o binarias). Estas variables son necesarias en momentos que un aspecto del problema no pueda ser decimal (por ejemplo el número de personas de algún lugar), además de dar más versatilidad al modelado.

1.1.6 MATLAB y Cplex

MATLab es un entorno de programación y una herramienta con muchas capacidades, diseñado para realizar cálculos matemáticos, análisis de datos y creación de gráficos de manera eficiente.

Cplex, por otro lado, es uno de los software de optimización que existen en el mercado, y es utilizado para resolver problemas de programación matemática, como LP, MILP, etc.

Lo interesante es que Cplex ofrece una interfaz de MATLAB, la que nos permite utilizar métodos con los que hacer uso de las potentes capacidades de su “solver”, llamado así porque es el algoritmo que utiliza al resolver problemas de optimización tales como problemas de programación lineal (LP), programación cuadrática (QP) y programación lineal entera y mixta (MILP). Por supuesto, existen condiciones para resolver problemas de optimización usando Cplex en MATLAB, entre ellas que solo admite restricciones del tipo “menor-igual”, forzando a hacer los ajustes necesarios a las restricciones del problema para poder utilizar esta herramienta. Se debe conseguir que el problema sea de la siguiente forma:

$$\begin{aligned} & \max_{x \in \mathcal{X}} f^T x, \\ \text{s.a.} \quad & Ax \leq a, \end{aligned}$$

Finalmente es evidente que la ventaja de utilizar estas dos herramientas juntas es que se pueden describir fácilmente problemas complejos de optimización, que utilizando otro tipo de entorno de programación podría resultar poco trivial.

Capítulo 2

Objetivos y Alcances

El principal objetivo de este proyecto es implementar un Control Predictivo Basado en Modelos (MPC) en plantas fotovoltaicas (PV) utilizando Programación Lineal Entera Mixta (MILP) para optimizar el sistema de almacenamiento de energía en baterías (BESS) utilizando variables binarias que contribuyan a minimizar el desgaste de la vida útil de las baterías. Por otro lado, se busca maximizar el rédito económico de la planta PV mediante el uso de un control eficiente en los momentos de carga y descarga de la batería, considerando las fluctuaciones en la generación de energía solar y las condiciones del mercado energético.

El proyecto abarca el modelado de la planta fotovoltaica y el sistema BESS, el diseño de un esquema de control predictivo mediante el uso de MATLAB y el solucionador Cplex para MILP. Se analizarán los beneficios económicos de esta estrategia de control, excluyendo la implementación en planta física, y se limitará el análisis a simulaciones basadas en datos reales del mercado de energía.

2.1 Objetivos específicos

En resumen, los objetivos del trabajo presentado en este proyecto se definen como:

- Implementar MPC para una planta PV utilizando MILP.
- Maximizar el rédito económico obtenido de la venta y almacenamiento de energía.

- El control debe tener la particularidad de disminuir el desgaste a la vida útil de la batería a largo plazo.
- El control debe ser viable en cada momento.

Capítulo 3

Modelo de control predictivo para una planta PV

3.1 Especificación del problema

En los mercados no regulados de energía se debe comprometer con anterioridad la producción de energía (unas cuantas horas antes del momento de entrega real), tal como se discutió en la sección 1.1.2. Buscando eliminar las penalizaciones económicas debido a una producción de energía menor a la esperada y comprometida es que se busca la utilización de BESS en las plantas PV. Sin embargo, en ocasiones incluso la energía almacenada en los sistemas BESS resulta insuficiente para cubrir completamente el desbalance causado por la baja producción de energía debido a la baja irradiación solar. Debido a esto es necesario tomar decisiones que gestionen adecuadamente la energía almacenada y producida.

Tomando como objetivo el maximizar el rédito económico, se busca que el algoritmo de control de nuestro BESS indique las acciones de inyectar energía a la red principal, almacenar la energía producida desde la planta PV o mantener la batería en reposo, teniendo en cuenta el deterioro que sufre la vida útil de las baterías por los constantes cambios entra carga y descarga [12]. Con este en mente, los autores [2] plantean el siguiente problema de interés, el que es representado por el diagrama de la fig. 3.1.

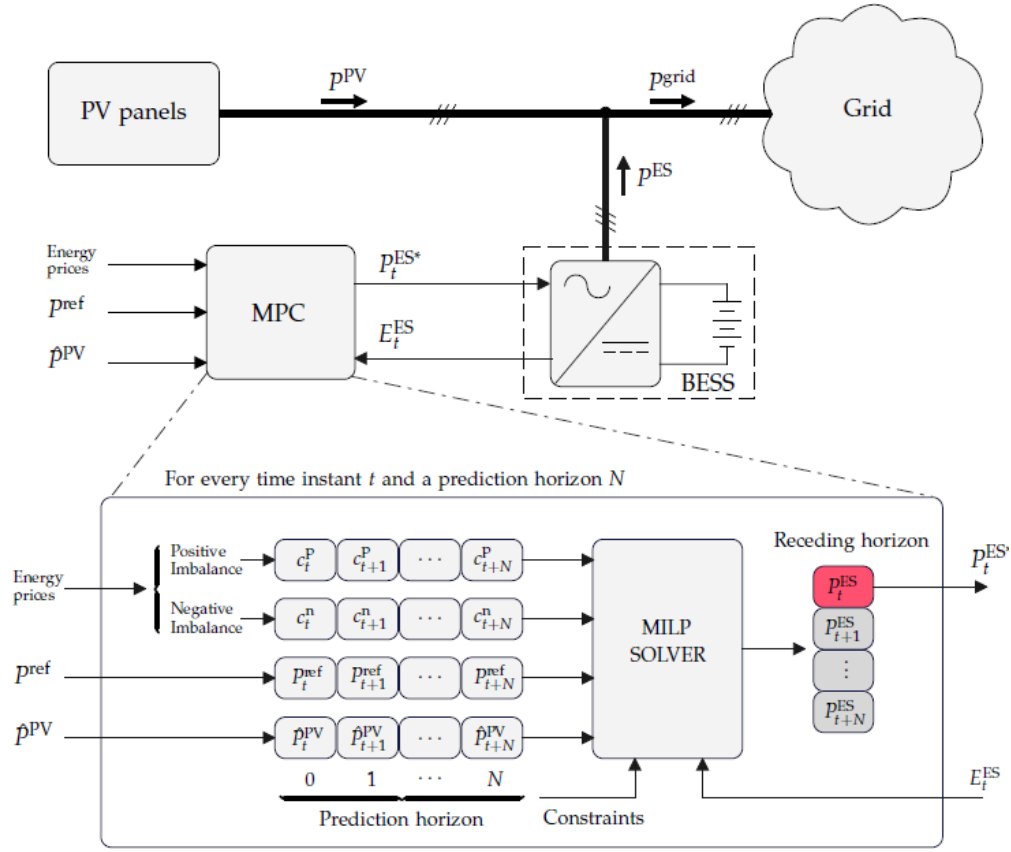


Figura 3.1: Diagrama del modelo de control

Imagen recuperada de [2].

Para facilitar el entendimiento del diagrama, a continuación se definen los valores que aparecen en él.

- P^{PV} : Potencia generada por los paneles PV.
- P^{grid} : Potencia enviada a la red.
- P^{ES} : Potencia inyectada a a red desde el BESS.
- Energy prices: Precios de venta de la energía.
- P^{ref} : Potencia comprometida a entregar a la red.
- \hat{P}^{PV} : Predicción de la producción de energia (en potencia) por parte de los paneles PV.

- P_t^{ES*} : Señal desde bloque de control (MPC) que determina el valor y dirección de P^{ES} en el instante t, esta es la variable de control. Por simplicidad se considera $P_t^{ES*} = P^{ES}$.
- E_t^{ES} : La energía almacenada en la batería en el instante t.

En el caso de estudio, es decir, el mercado español, se fijan dos costos que son c_t^p y c_t^n , los cuales se logran apreciar en el diagrama, estos son aquellos que representan el costo de la energía cuando se genera un desbalance en la potencia que se inyecta a la red (P^{grid}) y la potencia prometida a la red (P^{ref}). c_t^p es el costo de la energía cuando el desbalance es positivo, y c_t^n es el costo de penalización cuando el desbalance es negativo. Estos costos son impuestos por el operador del sistema al rededor del precio c^m , que fue anteriormente acordado por el mercado. Entonces el costo queda definido de la siguiente forma:

$$c = \begin{cases} c^p & \text{si } P^{grid} \geq P^{ref} \\ c^n & \text{si } P^{grid} < P^{ref} \end{cases} \quad (3.1)$$

Por otro lado, “t” representa el instante de tiempo actual en el que se esta buscando una optimización de las variables, y la notación de los subíndices “t+k”, con $k = 0,1,\dots,N-1,N$, simbolizan los valores que toman las variables en los N+1 instantes siguientes del horizonte de predicción, y para esto, posteriormente se definirá un período de muestreo “T” que separa cada uno de estos instantes en el tiempo.

En adición a lo anterior, los autores en [2] plantean el incluir la eficiencia de la carga y descarga de la batería representada por la variable “ η ”, con $0 < \eta \leq 1$. Sin embargo, la eficiencia de carga no es necesariamente igual a la eficiencia de la descarga de la batería, por lo tanto se define la eficiencia de carga como η^c , y la eficiencia de descarga como $1/\eta^d$, con $0 < \eta^c \leq 1$ y $0 < \eta^d \leq 1$. Entonces, considerando que el estado de carga y descarga de la batería es determinado por P^{ES} (ver fig. 1.3) tenemos:

$$\eta = \begin{cases} 1/\eta^d & \text{si } P^{ES} \geq 0 \\ \eta^c & \text{si } P^{ES} < 0 \end{cases} \quad (3.2)$$

3.2 Problema de optimización

A partir de lo visto en la sección anterior, ahora es posible generar la función de costo y restricciones que representan al problema de optimización. Para esto se establece primeramente el objetivo del problema, que es el de maximizar la ganancia económica generada por el suministro de energía a la red. Factores a tomar en cuenta para el desarrollo de la función de costo y sus restricciones son:

- Si la planta PV genera más que la energía comprometida ($P^{PV} \geq P^{ref}$), esta energía puede ser suministrada a la red o almacenada en el BESS.
- La energía almacenada en el BESS puede ser suministrada a la red en exceso cuando el precio de compra de la energía es alto, o puede ser usada para atenuar las penalizaciones económicas cuando la producción es baja.
- Si se predice una generación menor a la energía comprometida se deberá usar la energía almacenada en las BESS para cubrir este desbalance, o almacenarlas para cubrir un intervalo de tiempo en el que la penalización económica sea mayor que el actual.
- Existe un máximo de potencia de entrada y salida a la batería.
- Existe un máximo de carga y descarga de la batería.
- La modelado debe seguir la lógica del diagrama mostrado en la fig. 3.1.

Considerando los puntos anteriores, los autores en [2] especifican la siguiente función de costo, donde la variable de decisión es P^{ES} :

$$\max_{P_{t+k}^{ES}} \quad \mathcal{J}_1 = c_{t+N+1}^{soc} E_{t+N+1}^{ES} + \sum_{k=0}^N c_{t+k} T \left(P_{t+k}^{grid} - P_{t+k}^{ref} \right) - c_{t+N+1}^{soc} T \eta_{t+k} P_{t+k}^{ES}, \quad (3.3)$$

s.a.

$$c_{t+k} = \begin{cases} \alpha^{-k} c_{t+k}^p & \text{si } P_{t+k}^{grid} - P_{t+k}^{ref} \geq 0 \\ \alpha^{-k} c_{t+k}^n & \text{si } P_{t+k}^{grid} - P_{t+k}^{ref} < 0 \end{cases}, \quad (3.4)$$

$$\eta_{t+k} = \begin{cases} 1/\eta^d & \text{si } P_{t+k}^{ES} \geq 0 \\ \eta^c & \text{si } P_{t+k}^{ES} < 0 \end{cases}, \quad (3.5)$$

$$P_{t+k}^{grid} = P_{t+k}^{PV} + P_{t+k}^{ES}, \quad (3.6)$$

$$E_{t+k+1}^{ES} = E_{t+k}^{ES} + TP_{t+k}^{ES}, \quad (3.7)$$

$$P_{min}^{ES} \leq P_{t+k}^{ES} \leq P_{max}^{ES}, \quad (3.8)$$

$$E_{min}^{ES} \leq E_{t+k}^{ES} \leq E_{max}^{ES}, \quad (3.9)$$

Para comprender de mejor manera el significado de cada término, es que a continuación se explican aquellos que no fueron definidos con anterioridad en la sección 3.1:

- \mathcal{J}_1 : Representa la ganancia económica esperada por el suministro de energía a la red en cada intervalo de tiempo en el horizonte de predicción.
- c_{t+N+1}^{soc} : Precio dado para la energía almacenada al final del horizonte de predicción.
- E_{t+N+1}^{ES} : Energía almacenada en la batería al final del horizonte de predicción.
- T : Período de muestreo de MPC.
- α : Es un parámetro de peso para el costo de la energía, donde $0 < \alpha < 1$.

Entonces el primer término de la función de costo $c_{t+N+1}^{soc} E_{t+N+1}^{ES}$ representa el valor económico total de la energía almacenada al final del horizonte de predicción. Este valor puede ser considerado como “constante”, ya que solo toma un valor final, por lo tanto su escritura en las ecuaciones se obviará. El factor $c_{t+N+1}^{soc} T \eta_{t+k} P_{t+k}^{ES}$ que se observa en la función de costo (3.10) aparece a modo de resta, ya que este factor se utiliza para fomentar la reserva de energía en la batería al final del horizonte de predicción cuando el precio de ésta es elevado.

Si reemplazamos la restricción 3.6 en la función de costo 3.3 y en la restricción 3.4, además de reemplazar la restricción 3.7 en la restricción 3.9 (considerando su recursividad) obtenemos:

$$\max_{P_{t+k}^{ES}} \mathcal{J}_1 = \sum_{k=0}^N c_{t+k} T \left[P_{t+k}^{ES} - \left(P_{t+k}^{ref} - P_{t+k}^{PV} \right) \right] - c_{t+N+1}^{soc} T \eta_{t+k} P_{t+k}^{ES}, \quad (3.10)$$

s.a.

$$c_{t+k} = \begin{cases} \alpha^{-k} c_{t+k}^p & \text{si } P_{t+k}^{ES} - \left(P_{t+k}^{ref} - P_{t+k}^{PV} \right) \geq 0 \\ \alpha^{-k} c_{t+k}^n & \text{si } P_{t+k}^{ES} - \left(P_{t+k}^{ref} - P_{t+k}^{PV} \right) < 0 \end{cases}, \quad (3.11)$$

$$\eta_{t+k} = \begin{cases} 1/\eta^d & \text{si } P_{t+k}^{ES} \geq 0 \\ \eta^c & \text{si } P_{t+k}^{ES} < 0 \end{cases}, \quad (3.12)$$

$$P_{min}^{ES} \leq P_{t+k}^{ES} \leq P_{max}^{ES}, \quad (3.13)$$

$$E_{min}^{ES} \leq E_t^{ES} - T \sum_{s=0}^{k-1} \eta_{t+k} P_{t+s}^{ES} \leq E_{max}^{ES}, \quad (3.14)$$

Con este reemplazo se hace notar la presencia de la variable de decisión en el primer termino de la sumatoria en la función de costo, el cual fomenta que, si la energía generada por los paneles PV es insuficiente para llegar a suministrar la energía comprometida ($P^{ref} - P^{PV} \geq 0$), entonces se inyecte energía a la red desde la batería ($P^{ES} \geq 0$) para eliminar o minimizar la penalización económica (c^n). Por el contrario, si existe un exceso en la energía generada por los paneles PV en comparación a la energía comprometida ($P^{ref} - P^{PV} < 0$), ésta se almacene en la batería, o se inyecte en exceso a la red si el precio de la energía (c^p) maximiza las ganancias.

Finalmente describiendo el problema de interés, a la vez que se integran las igualdades dentro de las otras restricciones. Sin embargo, se puede ver la no linealidad en el problema, para esto se utilizan variables binarias que nos permitan conservar la linealidad.

3.3 Variables binarias

Como en la sección anterior existen restricciones que involucran reglas lógicas (3.4 y 3.5), se consideran variables binarias para las condiciones que causan la toma de diferentes valores por parte de c_{t+k} y η_{t+k} .

Para el parámetro c_{t+k} se asocia la variable binaria $\delta_{t+k} \in \{0, 1\}$ tal que:

$$\delta_{t+k} = \begin{cases} 1 & \Leftrightarrow P_{t+k}^{ES} - (P_{t+k}^{ref} - P_{t+k}^{PV}) \geq 0 \\ 0 & \Leftrightarrow P_{t+k}^{ES} - (P_{t+k}^{ref} - P_{t+k}^{PV}) < 0 \end{cases} \quad (3.15)$$

Y por otro lado, al parámetro η_{t+k} se asocia la variable binaria $\beta_{t+k} \in \{0, 1\}$ tal que:

$$\beta_{t+k} = \begin{cases} 1 & \Leftrightarrow P_{t+k}^{ES} \geq 0 \\ 0 & \Leftrightarrow P_{t+k}^{ES} < 0 \end{cases} \quad (3.16)$$

Sin embargo, el definir estas variables binarias no tiene utilidad si no podemos mantener la linealidad de la función de costo y sus restricciones. Por esto es que los autores en [6] proponen una transformación para este tipo de problema donde la variable de decisión esta acotada por limites definidos.

3.4 Equivalencia de variables binarias

Para explicar la transformación, a continuación se observa un ejemplo de problema de maximización:

$$\max g(x) = \rho f(x), \quad (3.17)$$

s.a.

$$m \leq f(x) \leq M, \quad (3.18)$$

$$\rho = \begin{cases} a & \text{si } f(x) \geq 0 \\ b & \text{si } f(x) < 0 \end{cases}, \quad (3.19)$$

Donde m y M están definidos y $f(x)$ es del tipo $f(x) = \mathcal{A}x + c$.

Asociando la condición de $f(x)$, que aparece en la restricción 3.19, a la variable binaria $\delta \in \{0, 1\}$, de modo que $f(x) \geq 0 \leftrightarrow \delta = 1$, y $f(x) < 0 \leftrightarrow \delta = 0$. Entonces se puede ver que se cumple:

$$f(x) \geq m(1 - \delta), \quad (3.20)$$

$$f(x) + \epsilon \leq (M + \epsilon)\delta, \quad (3.21)$$

Donde ϵ es un escalar cuyo valor es cercano a cero y representa a la tolerancia típicamente elegida por la maquina de cálculo.

En orden de continuar la transformación es que se debe redefinir la función $g(x)$ reescribiéndola de la siguiente manera:

$$g(x) = (a - b)\delta f(x) + bf(x), \quad (3.22)$$

De esta manera se cumple que cuando $\delta = 1$ entonces $g(x) = af(x)$, y que cuando $\delta = 0$ entonces $g(x) = bf(x)$, cumpliendo así con la restricción 3.19.

No obstante, en 3.22 hay una multiplicación entre la variable presente en $f(x)$ y la variable binaria δ , y por lo tanto rompiendo la linealidad buscada. Para corregir esto se necesita una

variable auxiliar $z(x) = \delta f(x)$, la cual se define:

$$z \leq M\delta, \tag{3.23}$$

$$z \geq m\delta, \tag{3.24}$$

$$z \leq f(x) - m(1 - \delta), \tag{3.25}$$

$$z \geq f(x) - M(1 - \delta), \tag{3.26}$$

De esta forma se cumple $\delta = 1 \Rightarrow (m \leq z \leq M; z = f(x))$ y $\delta = 0 \Rightarrow (m \leq f(x) \leq M; z = 0)$.

Reescribiendo nuevamente el problema queda:

$$\max g(x) = (a - b)\delta z + bf(x), \tag{3.27}$$

s.a.

$$z \leq M\delta, \tag{3.28}$$

$$z \geq m\delta, \tag{3.29}$$

$$z \leq f(x) - m(1 - \delta), \tag{3.30}$$

$$z \geq f(x) - M(1 - \delta), \tag{3.31}$$

$$f(x) \geq m(1 - \delta), \tag{3.32}$$

$$f(x) + \epsilon \leq (M + \epsilon)\delta, \tag{3.33}$$

3.5 Reformulación de problema de optimización

Se definen variables auxiliares “ z ” y “ w ” para los parámetros “ c_{t+k} ” y “ η_{t+k} ” respectivamente, y utilizando las variables binarias de 3.15 y 3.65.

Considerando que si

$$P_{min}^{ES} \leq P_{t+k}^{ES} \leq P_{max}^{ES}$$

Entonces

$$P_{min}^{ES} - (P_{t+k}^{ref} - P_{t+k}^{PV}) \leq P_{t+k}^{ES} - (P_{t+k}^{ref} - P_{t+k}^{PV}) \leq P_{max}^{ES} - (P_{t+k}^{ref} - P_{t+k}^{PV})$$

Para $w_{t+k} = P_{t+k}^{ES} \beta_{t+k}$

$$w_{t+k} \leq P_{max}^{ES} \beta_{t+k}, \quad (3.34)$$

$$w_{t+k} \geq P_{min}^{ES} \beta_{t+k}, \quad (3.35)$$

$$w_{t+k} \leq P_{t+k}^{ES} - P_{min}^{ES} (1 - \beta_{t+k}), \quad (3.36)$$

$$w_{t+k} \geq P_{t+k}^{ES} - P_{max}^{ES} (1 - \beta_{t+k}), \quad (3.37)$$

Para $z_{t+k} = [P_{t+k}^{ES} - (P_{t+k}^{ref} - P_{t+k}^{PV})] \delta_{t+k}$.

$$z_{t+k} \leq [P_{max}^{ES} - (P_{t+k}^{ref} - P_{t+k}^{PV})] \delta_{t+k}, \quad (3.38)$$

$$z_{t+k} \geq [P_{min}^{ES} - (P_{t+k}^{ref} - P_{t+k}^{PV})] \delta_{t+k}, \quad (3.39)$$

$$z_{t+k} \leq P_{t+k}^{ES} - (P_{t+k}^{ref} - P_{t+k}^{PV}) - [P_{min}^{ES} - (P_{t+k}^{ref} - P_{t+k}^{PV})] (1 - \delta_{t+k}), \quad (3.40)$$

$$z_{t+k} \geq P_{t+k}^{ES} - (P_{t+k}^{ref} - P_{t+k}^{PV}) - [P_{max}^{ES} - (P_{t+k}^{ref} - P_{t+k}^{PV})] (1 - \delta_{t+k}), \quad (3.41)$$

Se reescribe el problema de optimización como se ve a continuación:

$$\begin{aligned} \max_{P_{t+k}^{ES}} \quad \mathcal{J}_2 = & \sum_{k=0}^N \alpha^{-k} T (c_{t+k}^p - c_{t+k}^n) z_{t+k} + T (\alpha^{-k} c_{t+k}^n - c_{t+N+1}^{soc} \eta^c) P_{t+k}^{ES} \\ & - c_{t+N+1}^{soc} T (\eta^d - \eta^c) w_{t+k} \end{aligned} \quad (3.42)$$

s.a.

$$P_{t+k}^{ES} \leq \left[P_{max}^{ES} - (P_{t+k}^{ref} - P_{t+k}^{PV}) \right] \delta_{t+k} + (P_{t+k}^{ref} - P_{t+k}^{PV}), \quad (3.43)$$

$$P_{t+k}^{ES} \geq \left[P_{min}^{ES} - (P_{t+k}^{ref} - P_{t+k}^{PV}) \right] (1 - \delta_{t+k}) + (P_{t+k}^{ref} - P_{t+k}^{PV}), \quad (3.44)$$

$$z_{t+k} \leq \left[P_{max}^{ES} - (P_{t+k}^{ref} - P_{t+k}^{PV}) \right] \delta_{t+k}, \quad (3.45)$$

$$z_{t+k} \geq \left[P_{min}^{ES} - (P_{t+k}^{ref} - P_{t+k}^{PV}) \right] \delta_{t+k}, \quad (3.46)$$

$$z_{t+k} \leq P_{t+k}^{ES} - (P_{t+k}^{ref} - P_{t+k}^{PV}) - \left[P_{min}^{ES} - (P_{t+k}^{ref} - P_{t+k}^{PV}) \right] (1 - \delta_{t+k}), \quad (3.47)$$

$$z_{t+k} \geq P_{t+k}^{ES} - (P_{t+k}^{ref} - P_{t+k}^{PV}) - \left[P_{max}^{ES} - (P_{t+k}^{ref} - P_{t+k}^{PV}) \right] (1 - \delta_{t+k}), \quad (3.48)$$

$$P_{t+k}^{ES} \geq P_{min}^{ES} (1 - \beta_{t+k}), \quad (3.49)$$

$$P_{t+k}^{ES} \leq P_{max}^{ES} \beta_{t+k}, \quad (3.50)$$

$$w_{t+k} \leq P_{max}^{ES} \beta_{t+k}, \quad (3.51)$$

$$w_{t+k} \geq P_{min}^{ES} \beta_{t+k}, \quad (3.52)$$

$$w_{t+k} \leq P_{t+k}^{ES} - P_{min}^{ES} (1 - \beta_{t+k}), \quad (3.53)$$

$$w_{t+k} \geq P_{t+k}^{ES} - P_{max}^{ES} (1 - \beta_{t+k}), \quad (3.54)$$

$$\delta_{t+k} \in \{0, 1\}, \quad (3.55)$$

$$\beta_{t+k} \in \{0, 1\}, \quad (3.56)$$

$$(\delta_{t+k} - \beta_{t+k}) (P_{t+k}^{ref} - P_{t+k}^{PV}) \leq 0, \quad (3.57)$$

$$E_{min}^{ES} \leq E_t^{ES} - T \sum_{s=0}^{k-1} \eta_{t+k} P_{t+s}^{ES} \leq E_{max}^{ES}, \quad k = 1, 2, \dots, N-1, N, \quad (3.58)$$

Por lo tanto, en este problema de optimización las variables de decisión resultan ser P^{ES} , z , δ , w y β .

Con esto ya es posible utilizar el solver de MILP perteneciente Cplex en MATLAB para resolver el problema de optimización.

3.6 Validación del modelo con dos variables binarias

Para la validación del modelo de dos variables binarias se simulará una situación hipotética de control. Sin embargo no es posible solucionar directamente el problema de optimización con el solver de Cplex en MATLAB, ya que éste requiere que el problema sea escrito en la forma estándar de los problemas de programación lineal entera y mixta, como se ve en la sección 1.1.6.

El problema reescrito se ve a continuación.

$$\max_{y \in \mathcal{Y}} g^T y, \quad (3.59)$$

$$s.a. \quad By \leq b, \quad (3.60)$$

Donde resulta necesario definir:

- g : Vector que condensa los factores que acompañan a las variables de decisión en la función de costo, tanto costos de desbalance como los valores de la energía al final del horizonte de predicción.
- y : Vector que contiene las variables de decisión.
- B : Matriz que contiene los factores que acompañan a las variables de decisión al lado izquierdo de las restricciones.
- b : Vector de valores al lado derecho de las restricciones.
- \mathcal{Y} : Representa al conjunto de todas los valores posibles de las variables de decisión en y .

3.6.1 Representación matricial

Las matrices y vectores que resultan necesarias para la resolución del problema se muestran a continuación:

$$c_{soc} = \begin{bmatrix} C_{t+N+1}^{soc} \\ C_{t+N+1}^{soc} \\ \vdots \\ C_{t+N+1}^{soc} \end{bmatrix}, \quad c_n = \begin{bmatrix} C_{t+0}^n \\ C_{t+1}^n \\ \vdots \\ C_{t+N}^n \end{bmatrix}, \quad c_p = \begin{bmatrix} C_{t+0}^p \\ C_{t+1}^p \\ \vdots \\ C_{t+N}^p \end{bmatrix}, \quad (M1)$$

$$\alpha = \begin{bmatrix} \alpha^{-0} \\ \alpha^{-1} \\ \vdots \\ \alpha^{-N} \end{bmatrix}, \quad \eta_c = \begin{bmatrix} \eta^c \\ \eta^c \\ \vdots \\ \eta^c \end{bmatrix}, \quad \eta_d = \begin{bmatrix} \frac{1}{\eta^d} \\ \frac{1}{\eta^d} \\ \vdots \\ \frac{1}{\eta^d} \end{bmatrix}. \quad (M2)$$

$$P_{min} = \begin{bmatrix} P_{min}^{ES} \\ P_{min}^{ES} \\ \vdots \\ P_{min}^{ES} \end{bmatrix}, \quad P_{max} = \begin{bmatrix} P_{max}^{ES} \\ P_{max}^{ES} \\ \vdots \\ P_{max}^{ES} \end{bmatrix}, \quad P_{ref} = \begin{bmatrix} P_{t+0}^{ref} \\ P_{t+1}^{ref} \\ \vdots \\ P_{t+N}^{ref} \end{bmatrix}, \quad P_{PV} = \begin{bmatrix} P_{t+0}^{PV} \\ P_{t+1}^{PV} \\ \vdots \\ P_{t+N}^{PV} \end{bmatrix}, \quad (M3)$$

$$E_{min} = \frac{1}{T} \begin{bmatrix} E_t^{ES} - E_{max}^{ES} \\ E_t^{ES} - E_{max}^{ES} \\ \vdots \\ E_t^{ES} - E_{max}^{ES} \end{bmatrix}, \quad E_{max} = \frac{1}{T} \begin{bmatrix} E_t^{ES} - E_{min}^{ES} \\ E_t^{ES} - E_{min}^{ES} \\ \vdots \\ E_t^{ES} - E_{min}^{ES} \end{bmatrix}, \quad F = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 1 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{bmatrix}, \quad (M4)$$

Donde los vectores que aparecen en M1, M2, M3 y M4 pertenecen al conjunto de \mathbb{R}^{N+1} , y la matriz $F \in \mathbb{R}^{(N+1) \times (N+1)}$, matriz que nos permite lidiar con las restricciones con recursividad, como la que se ve en 3.58.

Entonces para construir la función de costo (3.59) para resolver el problema de optimización se utiliza:

$$g = \begin{bmatrix} g_1 \\ g_2 \\ 0 \\ g_3 \\ 0 \end{bmatrix}, \quad \begin{aligned} g_1 &= \alpha \odot c_n - c_{soc} \odot \eta_c, \\ g_2 &= \alpha \odot (c_p - c_n), \\ g_3 &= c_{soc} \odot (\eta_c - \eta_d). \end{aligned} \quad (\text{M5})$$

Y el vector “y”, el cual contiene las variables de decisión se construye:

$$y = \begin{bmatrix} P_{ES} \\ z \\ \delta \\ w \\ \beta \end{bmatrix}, \quad (\text{M5})$$

donde

$$P_{ES} = \begin{bmatrix} P_{t+0}^{ES} \\ P_{t+1}^{ES} \\ \vdots \\ P_{t+N}^{ES} \end{bmatrix}, \quad z = \begin{bmatrix} z_{t+0} \\ z_{t+1} \\ \vdots \\ z_{t+N} \end{bmatrix}, \quad \delta = \begin{bmatrix} \delta_{t+0} \\ \delta_{t+1} \\ \vdots \\ \delta_{t+N} \end{bmatrix}, \quad w = \begin{bmatrix} w_{t+0} \\ w_{t+1} \\ \vdots \\ w_{t+N} \end{bmatrix}, \quad \beta = \begin{bmatrix} \beta_{t+0} \\ \beta_{t+1} \\ \vdots \\ \beta_{t+N} \end{bmatrix}, \quad (\text{M7})$$

Considerando que los vectores de las variables de decisión que se observan en M7 pertenecen al conjunto \mathbb{R}^{N+1} .

Entonces la inecuación 3.60 es dada por las siguientes matrices:

$$\mathbf{B} = \begin{bmatrix} -\mathbf{I} & 0 & -\text{diag}\{\mathbf{P}_{\min} - \mathbf{R}\} & 0 & 0 \\ +\mathbf{I} & 0 & -\text{diag}\{\mathbf{P}_{\max} - \mathbf{R}\} & 0 & 0 \\ 0 & -\mathbf{I} & +\text{diag}\{\mathbf{P}_{\min} - \mathbf{R}\} & 0 & 0 \\ 0 & +\mathbf{I} & -\text{diag}\{\mathbf{P}_{\max} - \mathbf{R}\} & 0 & 0 \\ -\mathbf{I} & +\mathbf{I} & -\text{diag}\{\mathbf{P}_{\min} - \mathbf{R}\} & 0 & 0 \\ +\mathbf{I} & -\mathbf{I} & +\text{diag}\{\mathbf{P}_{\max} - \mathbf{R}\} & 0 & 0 \\ -\mathbf{I} & 0 & 0 & 0 & -\text{diag}\{\mathbf{P}_{\min}\} \\ +\mathbf{I} & 0 & 0 & 0 & -\text{diag}\{\mathbf{P}_{\max}\} \\ 0 & 0 & 0 & -\mathbf{I} & +\text{diag}\{\mathbf{P}_{\min}\} \\ 0 & 0 & 0 & +\mathbf{I} & -\text{diag}\{\mathbf{P}_{\max}\} \\ -\mathbf{I} & 0 & 0 & +\mathbf{I} & -\text{diag}\{\mathbf{P}_{\min}\} \\ +\mathbf{I} & 0 & 0 & -\mathbf{I} & +\text{diag}\{\mathbf{P}_{\max}\} \\ +\mathbf{V}_1 & 0 & 0 & +\mathbf{V}_2 & 0 \\ -\mathbf{V}_1 & 0 & 0 & -\mathbf{V}_2 & 0 \\ 0 & 0 & +\text{diag}\{\mathbf{R}\} & 0 & -\text{diag}\{\mathbf{R}\} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} -\mathbf{P}_{\min} \\ \mathbf{R} \\ 0 \\ 0 \\ -\mathbf{P}_{\min} \\ \mathbf{P}_{\max} \\ -\mathbf{P}_{\min} \\ 0 \\ 0 \\ 0 \\ -\mathbf{P}_{\min} \\ \mathbf{P}_{\max} \\ \mathbf{E}_{\max} \\ -\mathbf{E}_{\min} \\ 0 \end{bmatrix}, \quad (\text{M8})$$

Donde $\mathbf{V}_1 = \mathbf{F}\eta_c$, $\mathbf{V}_2 = \mathbf{F}(\eta_d - \eta_c)$ y $\mathbf{R} = \mathbf{P}_{ref} - \mathbf{P}_{PV}$. $\mathbf{I} \in \mathbb{R}^{N+1 \times N+1}$ corresponde a la matriz identidad, y “0” representa al vector cero $\in \mathbb{R}^{N+1}$, o a la matriz cero $\in \mathbb{R}^{N+1 \times N+1}$, dependiendo del contexto, en orden de que las dimensiones de las matrices \mathbf{B} y \mathbf{b} sean consistentes.

3.6.2 Sintaxis para simulación en MATLAB

Para simular el problema de optimización y su resolución utilizando MATLAB, se hará uso del solver de Cplex. La sintaxis que se utiliza para hacer uso del solver se presenta a continuación:

$$\mathbf{x} = \text{cplexmilp}(f, \mathbf{A}_{ineq}, \mathbf{b}_{ineq}, \mathbf{A}_{eq}, \mathbf{b}_{eq}, \text{sostype}, \text{sosind}, \text{soswt}, \mathbf{lb}, \mathbf{ub}, \text{ctype}, \mathbf{x}_0, \text{options})$$

Donde,

- f : Vector de coeficientes de función de costos (g).
- A_{ineq} : Matriz de coeficientes del lado izquierdo de las restricciones (B).
- b_{ineq} : Matriz de coeficientes del lado derecho de las restricciones (b).
- A_{eq} : Matriz de coeficientes del lado izquierdo de restricciones de igualdad ([]).
- b_{eq} : Matriz de coeficientes del lado derecho de restricciones de igualdad ([]).
- $sostype$: Tipo de restricciones SOS (Set Of Separable Constraints) ([]).
- $sosind$: Índices de las variables en cada conjunto SOS ([]).
- $soswt$: Pesos asociados a las variables en cada conjunto SOS ([]).
- lb : Vector de límites inferiores para las variables ([]).
- ub : Vector de límites superiores para las variables ([]).
- $ctype$: Vector con parámetros que determina los tipos de las variables, tal que “C” es para variables continuas, “I” para variables enteras y “B” para variables binarias (se definirá mas adelante).
- $x0$: Vector de valores iniciales para las variables ([]).
- $options$: Estructura de opciones para configurar el solver, por ejemplo tolerancias, limites de tiempo, etc. ([]).

Entre paréntesis al final de cada definición se observa el valor que tomará cada parámetro de `cplexmip` para el caso a simular, tal que “[]” se refiere a que no toma valor en la simulación. Para $ctype$ se debe observar el vector de variables de decisión, que en este caso corresponde a y (M5), entonces:

$$ctype = \left['C' \ 'C' \ 'B' \ 'C' \ 'B' \right],$$

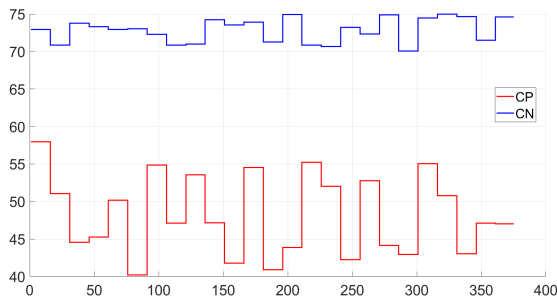
Donde cada uno de estos parámetros se repite $N+1$ veces, ya que los vectores de las variables de decisión tienen esta longitud (M7). Por otro lado, “x” almacena los valores óptimos ya calculados por el solver para las variables de decisión

3.6.3 Simulación

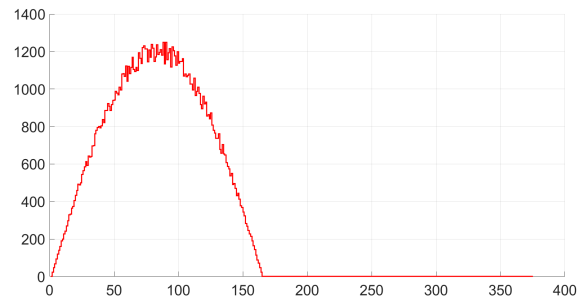
La simulación cuenta con los valores de la tabla 3.1 para los parámetros, y el código de MATLAB a utilizar se encuentra en la sección de “Anexos”.

N	T	c^P	c^n	η^c	η^d	α
60	4 min	ver fig. 3.2a	ver fig. 3.2a	0.89	1/0.89	1/0.999
c_{t+N+1}^{soc}	P_{min}^{ES}	P_{max}^{ES}	E_{min}^{ES}	E_{max}^{ES}	E_0^{ES}	Per.
$= c^n$	-500 kW	+500 kW	0	48 MWh	24 MWh	165

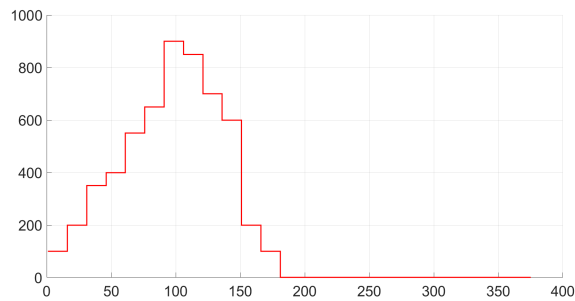
Tabla 3.1: Parámetros de simulación



(a) Precios de energía



(b) Potencia desde los paneles PV



(c) Potencia comprometida a la red

Figura 3.2: Datos de simulación

Utilizando el solver de Cplex se obtiene:

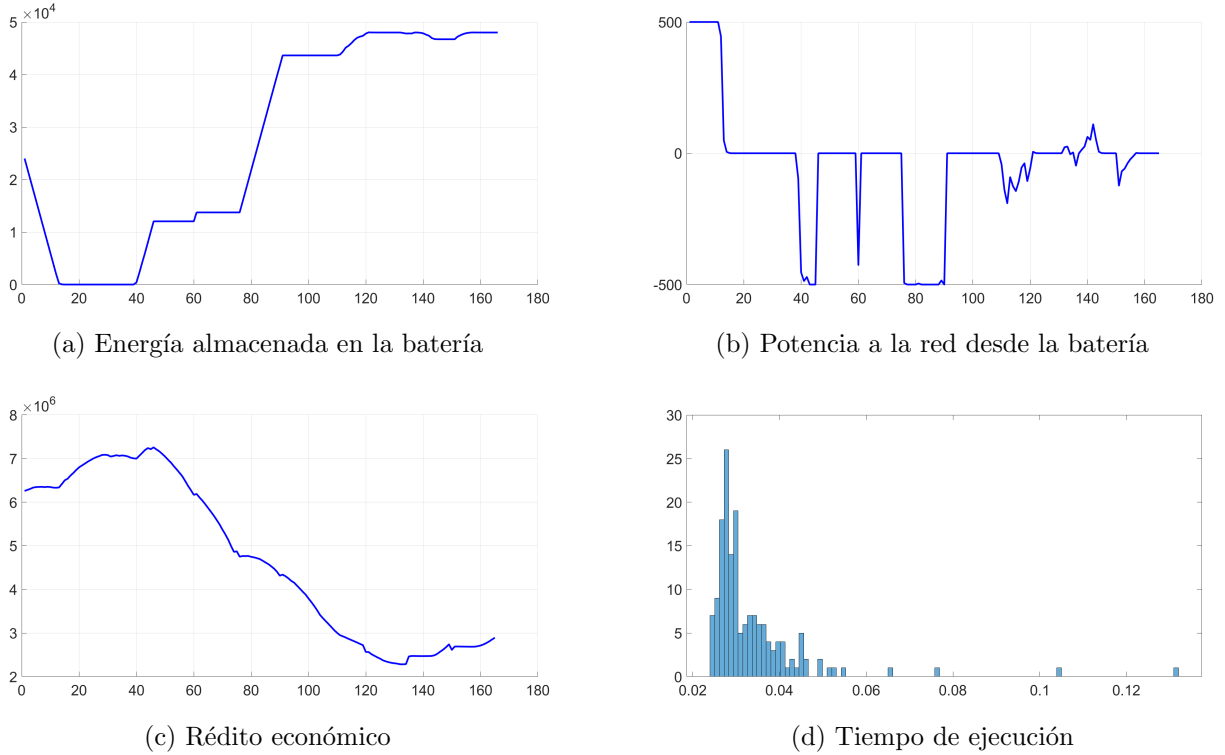


Figura 3.3: Resultados de simulación con 165 períodos

El parámetro “Per.” que se observa en la tabla 3.1, se refiere al número de veces que se aplica MPC, siendo aplicado 165 períodos, lo que equivale 11 horas debido a que $T = 4$ min. El eje x de los gráficos está en esta medida (cada unidad equivale a 4 minutos).

Donde, en la figura 3.3, el gráfico (a) describe el comportamiento de la energía almacenada en la batería en cada instante de tiempo (E_t^{ES}), (b) es la variable de control en el tiempo P_t^{ES} que determina la dirección de la energía entre la batería y la red (ver fig. 3.1), (c) representa la ganancia económica (\mathcal{J}_2), y (d) muestra el tiempo que tarda la ejecución del calculo realizado por el solver. En el gráfico se observa que en el momento en que $P^{ES} > 0$ entonces la energía almacenada en la batería (E^{ES}) decrece, lo que es consistente para validar el modelo. Por otro lado, el tiempo de ejecución es cercano a los 0.13 segundos en el peor caso, lo cual es aproximadamente 0.05% del período escogido de 4 minutos (240 segundos), es por esto que es posible decir que la ejecución del calculo es suficientemente rápida para una correcta aplicación del control predictivo.

La figura 3.4 muestra la diferencia entre la potencia generada por los paneles PV y la energía comprometida, es decir, $P^{PV} - P^{ref}$, figura donde se destaca que al existir un desbalance de potencia negativo el control predictivo busca suplir esta diferencia para no incurrir en penalizaciones

económicas con costo c^n (ver figura 3.2a).

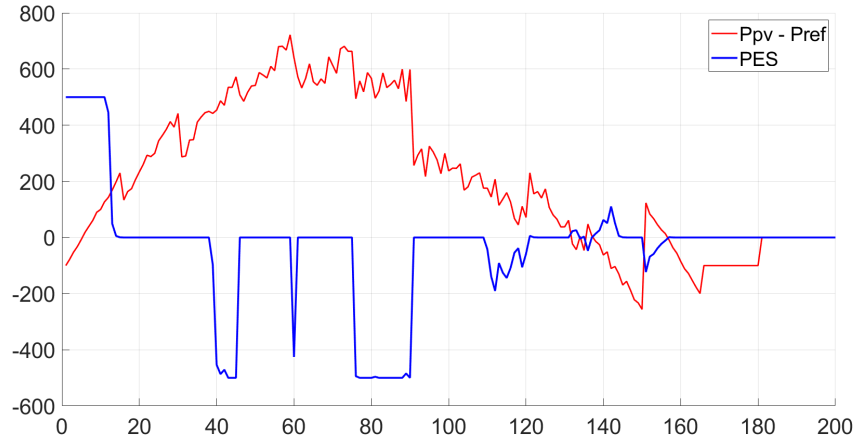


Figura 3.4: Desbalance de potencia

Por otro lado, la figura 3.4 muestra la tendencia a almacenar energía en la batería ($P^{ES} < 0$) debido a que la ganancia que se consigue con la energía almacenada al final del horizonte de predicción (c_{t+N+1}^{soc}) es considerablemente mas alta que la ganancia obtenida por la venta de la energía como exceso (c^p).

Hasta aquí, se ha demostrado la validez del control MPC para un modelo que no considera el deterioro a la vida útil de la batería. El problema que se tiene al no considerar el daño a la batería es que se puede incurrir en costos mayores al tener que reemplazar con mayor frecuencia las baterías que se utilizan en esta aplicación, ya que, como se observa en 3.3b, los cambios de estado entre carga y descarga son continuos y, en ocasiones, la potencia de los cambios es la máxima posible (P^{max} y P^{min}).

3.7 Desgaste de batería

Hasta este punto el problema de optimización describe las características del modelo propuesto en el la figura 1.3, considerando los precios del mercado y la eficiencia de carga y descarga de la batería de la BESS. Sin embargo, como la vida útil de las baterías es finita, resulta necesario añadir factores que nos permitan modelar el costo relacionado con este desgaste de las baterías.

Uno de los factores que aporta directamente al desgaste en la vida útil las baterías es el constante

cambio entre los estados de carga y descarga que pueden tener en este tipo de aplicaciones [12], ya que en cada intervalo de tiempo el sistema de control predictivo toma decisiones que cambian el sentido de la energía con valores de potencia bastante altos.

3.7.1 Propuesta

Para suplir esta necesidad se propone la integración de nuevas variables que permitan contar la cantidad de veces que el estado de la batería cambia de carga a descarga y viceversa. Esto con la intención de lograr disminuir, o directamente limitar, el número de veces que sucede este cambio de estado en la batería. Para esto se debe tener en conocimiento el momento en el que existe un cambio en el estado de la batería, lo que ocurre en el momento que P^{ES} pasa de ser mayor que cero a menor que cero y viceversa.

En el problema de optimización ya se ha definido la variable binaria “ β_{t+k} ”, la cual permite observar el estado de la batería en casa instante “ $t+k$ ”, de modo que cuando $\beta_{t+k} = 1$ la batería se carga, y cuando $\beta_{t+k} = 0$ la batería se descarga (3.65). Entonces, gracias a que en MPC observamos los estados futuros del modelo, se debe cumplir la siguiente tabla de verdad:

β_{t+k-1}	β_{t+k}	Cambio
0	0	0
0	1	1
1	0	1
1	1	0

Tabla 3.2: Tabla de verdad

En la tabla 3.2 se observa la lógica que se debe seguir para determinar cuando existe un cambio de estado en la batería, donde “ β_{t+k-1} ” representa el valor anterior al instante de tiempo “ $t+k$ ” con $k = 1, 2, \dots, N$. Como se puede observar, la tabla de verdad concuerda con el operador binario “xor” entre β_{t+k-1} y β_{t+k} , y su resultado lo podemos condensar en una variable binaria auxiliar “ λ_{t+k} ” que se define a continuación:

$$\lambda_{t+k} \geq \beta_{t+k-1} - \beta_{t+k} \quad (3.61)$$

$$\lambda_{t+k} \geq \beta_{t+k} - \beta_{t+k-1} \quad (3.62)$$

Considerando $\lambda \in \{0, 1\}$, se puede comprobar fácilmente que λ cumple con la columna “Cambio” de la tabla de verdad 3.2. Para el caso particular de λ_t se observa:

$$\lambda_t \geq \beta_0 - \beta_t \quad (3.63)$$

$$\lambda_t \geq \beta_t - \beta_0 \quad (3.64)$$

Donde “ β_0 ” es el valor de β inicial, el cual es equivalente al valor calculado en el instante de tiempo anterior al de la resolución del problema de optimización, en primera éste instancia toma el valor de “0”.

Entonces,

$$\lambda_{t+k} = \begin{cases} 1 & \leftrightarrow \beta_{t+k-1} \neq \beta_{t+k} \\ 0 & \leftrightarrow \beta_{t+k-1} = \beta_{t+k} \end{cases}, \quad (3.65)$$

Con esta variable binaria es posible contar el número de cambios en el estado de la batería a lo largo del horizonte de predicción utilizando la sumatoria que se ve a continuación:

$$\sum_{s=0}^{N-1} \lambda_{t+s} \quad (3.66)$$

Con este conteo resulta necesario reescribir el problema de optimización para limitar los cambios de estado de la batería.

3.8 Problema de optimización con tres variables binarias

3.8.1 Restricciones

Para reescribir el problema de optimización con el añadido de la nueva variable binaria λ como variable de decisión, es necesario definir una nueva restricción que se ve a continuación:

$$\text{Sum}_0 + \sum_{s=0}^{k-1} \lambda_{t+s} \leq \text{Lim} \quad (3.67)$$

Donde aparecen dos nuevos parámetros que se definen como:

- Sum_0 : Representa la suma acumulada de los instantes de tiempo anteriores al actual. En primera instancia su valor es cero.
- Lim : Representa el límite impuesto por el usuario.

Además se deben añadir como restricciones las inecuaciones vistas en 3.61 y 3.62.

3.8.2 Reformulación del problema de optimización

Finalmente, con las nuevas restricciones se puede reescribir el problema de optimización de la siguiente forma:

$$\begin{aligned} \max_{P_{t+k}^{ES}} \mathcal{J}_3 = & \sum_{k=0}^N \alpha^{-k} T (c_{t+k}^p - c_{t+k}^n) z_{t+k} + T (\alpha^{-k} c_{t+k}^n - c_{t+N+1}^{soc} \eta^c) P_{t+k}^{ES} \\ & - c_{t+N+1}^{soc} T (\eta^d - \eta^c) w_{t+k} \end{aligned} \quad (3.68)$$

s.a.

$$P_{t+k}^{ES} \leq \left[P_{max}^{ES} - \left(P_{t+k}^{ref} - P_{t+k}^{PV} \right) \right] \delta_{t+k} + \left(P_{t+k}^{ref} - P_{t+k}^{PV} \right), \quad (3.69)$$

$$P_{t+k}^{ES} \geq \left[P_{min}^{ES} - \left(P_{t+k}^{ref} - P_{t+k}^{PV} \right) \right] (1 - \delta_{t+k}) + \left(P_{t+k}^{ref} - P_{t+k}^{PV} \right), \quad (3.70)$$

$$z_{t+k} \leq \left[P_{max}^{ES} - \left(P_{t+k}^{ref} - P_{t+k}^{PV} \right) \right] \delta_{t+k}, \quad (3.71)$$

$$z_{t+k} \geq \left[P_{min}^{ES} - \left(P_{t+k}^{ref} - P_{t+k}^{PV} \right) \right] \delta_{t+k}, \quad (3.72)$$

$$z_{t+k} \leq P_{t+k}^{ES} - \left(P_{t+k}^{ref} - P_{t+k}^{PV} \right) - \left[P_{min}^{ES} - \left(P_{t+k}^{ref} - P_{t+k}^{PV} \right) \right] (1 - \delta_{t+k}), \quad (3.73)$$

$$z_{t+k} \geq P_{t+k}^{ES} - \left(P_{t+k}^{ref} - P_{t+k}^{PV} \right) - \left[P_{max}^{ES} - \left(P_{t+k}^{ref} - P_{t+k}^{PV} \right) \right] (1 - \delta_{t+k}), \quad (3.74)$$

$$P_{t+k}^{ES} \geq P_{min}^{ES} (1 - \beta_{t+k}), \quad (3.75)$$

$$P_{t+k}^{ES} \leq P_{max}^{ES} \beta_{t+k}, \quad (3.76)$$

$$w_{t+k} \leq P_{max}^{ES} \beta_{t+k}, \quad (3.77)$$

$$w_{t+k} \geq P_{min}^{ES} \beta_{t+k}, \quad (3.78)$$

$$w_{t+k} \leq P_{t+k}^{ES} - P_{min}^{ES} (1 - \beta_{t+k}), \quad (3.79)$$

$$w_{t+k} \geq P_{t+k}^{ES} - P_{max}^{ES} (1 - \beta_{t+k}), \quad (3.80)$$

$$\lambda_{t+k} \geq \beta_{t+k-1} - \beta_{t+k}, \quad (3.81)$$

$$\lambda_{t+k} \geq \beta_{t+k} - \beta_{t+k-1}, \quad (3.82)$$

$$\delta_{t+k} \in \{0, 1\}, \quad (3.83)$$

$$\beta_{t+k} \in \{0, 1\}, \quad (3.84)$$

$$\lambda_{t+k} \in \{0, 1\}, \quad (3.85)$$

$$(\delta_{t+k} - \beta_{t+k}) \left(P_{t+k}^{ref} - P_{t+k}^{PV} \right) \leq 0, \quad (3.86)$$

$$E_{min}^{ES} \leq E_t^{ES} - T \sum_{s=0}^{k-1} \eta_{t+k} P_{t+s}^{ES} \leq E_{max}^{ES}, \quad k = 1, 2, \dots, N-1, N, \quad (3.87)$$

$$\text{Sum}_0 + \sum_{s=0}^{k-1} \lambda_{t+s} \leq \text{Lim}, \quad (3.88)$$

Donde podemos ver las nuevas restricciones en 3.117, 3.118 y 3.88.

La forma estándar para la posterior resolución utilizando Cplex queda escrita:

$$\max_{x \in \mathcal{X}} f^T x, \quad (3.89)$$

$$s.a. \quad Dx \leq d, \quad (3.90)$$

Donde la definición de “x”, “ \mathcal{X} ”, “f”, “D” y “d”, es la misma que la de sus homólogos en 3.59 y 3.60.

3.8.3 Representación matricial

Las matrices y vectores que son modificadas o agregadas, respecto a las vistas en 3.6.1, se muestran a continuación:

El vector “x”, el cual contiene las variables de decisión se construye:

$$x = \begin{bmatrix} P_{ES} \\ z \\ \delta \\ w \\ \beta \\ \lambda \end{bmatrix}, \quad (M9)$$

donde

$$\lambda = \begin{bmatrix} \lambda_{t+0} \\ \lambda_{t+1} \\ \vdots \\ \lambda_{t+N} \end{bmatrix}, \quad (M10)$$

Como la función de costo \mathcal{J}_3 es igual a la función de costo \mathcal{J}_2 , entonces:

$$f = \begin{bmatrix} g \\ 0 \end{bmatrix} = \begin{bmatrix} g_1 \\ g_2 \\ 0 \\ g_3 \\ 0 \\ 0 \end{bmatrix}, \quad \begin{aligned} g_1 &= \alpha \odot c_n - c_{soc} \odot \eta_c, \\ g_2 &= \alpha \odot (c_p - c_n), \\ g_3 &= c_{soc} \odot (\eta_c - \eta_d). \end{aligned} \quad (\text{M11})$$

Los vectores y matrices observables en M1, M2, M3, M4 y M7 se mantienen para el problema de optimización con tres variables binarias.

Las matrices D y d presentes en 3.90 se muestran a continuación:

$$\mathbf{D} = \begin{bmatrix} -I & 0 & -\text{diag}\{P_{\min} - R\} & 0 & 0 & 0 \\ +I & 0 & -\text{diag}\{P_{\max} - R\} & 0 & 0 & 0 \\ 0 & -I & +\text{diag}\{P_{\min} - R\} & 0 & 0 & 0 \\ 0 & +I & -\text{diag}\{P_{\max} - R\} & 0 & 0 & 0 \\ -I & +I & -\text{diag}\{P_{\min} - R\} & 0 & 0 & 0 \\ +I & -I & +\text{diag}\{P_{\max} - R\} & 0 & 0 & 0 \\ -I & 0 & 0 & 0 & -\text{diag}\{P_{\min}\} & 0 \\ +I & 0 & 0 & 0 & -\text{diag}\{P_{\max}\} & 0 \\ 0 & 0 & 0 & -I & +\text{diag}\{P_{\min}\} & 0 \\ 0 & 0 & 0 & +I & -\text{diag}\{P_{\max}\} & 0 \\ -I & 0 & 0 & +I & -\text{diag}\{P_{\min}\} & 0 \\ +I & 0 & 0 & -I & +\text{diag}\{P_{\max}\} & 0 \\ +V_1 & 0 & 0 & +V_2 & 0 & 0 \\ -V_1 & 0 & 0 & -V_2 & 0 & 0 \\ 0 & 0 & +\text{diag}\{R\} & 0 & -\text{diag}\{R\} & 0 \\ 0 & 0 & 0 & 0 & 0 & F \\ 0 & 0 & 0 & 0 & Q & -I \\ 0 & 0 & 0 & 0 & -Q & -I \end{bmatrix}, \quad \mathbf{d} = \begin{bmatrix} -P_{\min} \\ R \\ 0 \\ 0 \\ -P_{\min} \\ P_{\max} \\ -P_{\min} \\ 0 \\ 0 \\ 0 \\ -P_{\min} \\ P_{\max} \\ E_{\max} \\ -E_{\min} \\ 0 \\ \text{st} \\ \text{zb} \\ -\text{zb} \end{bmatrix}, \quad (\text{M12})$$

Donde la matriz “Q” y los vectores “st” y “zb” se definen:

$$Q = \begin{bmatrix} -1 & 0 & 0 & \dots & 0 & 0 \\ 1 & -1 & 0 & \dots & 0 & 0 \\ 0 & 1 & -1 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & -1 \end{bmatrix}, \quad st = \begin{bmatrix} \text{Lim} - \text{Sum}_0 \\ \text{Lim} - \text{Sum}_0 \\ \text{Lim} - \text{Sum}_0 \\ \vdots \\ \text{Lim} - \text{Sum}_0 \end{bmatrix}, \quad zb = \begin{bmatrix} -\beta_0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad (\text{M13})$$

Donde $Q \in \mathbb{R}^{N+1 \times N+1}$, st y $zb \in \mathbb{R}^{N+1}$. $I \in \mathbb{R}^{N+1 \times N+1}$ corresponde a la matriz identidad, y “0” representa al vector cero $\in \mathbb{R}^{N+1}$, o a la matriz cero $\in \mathbb{R}^{N+1 \times N+1}$, dependiendo del contexto, en orden de que las dimensiones de las matrices D y d sean consistentes.

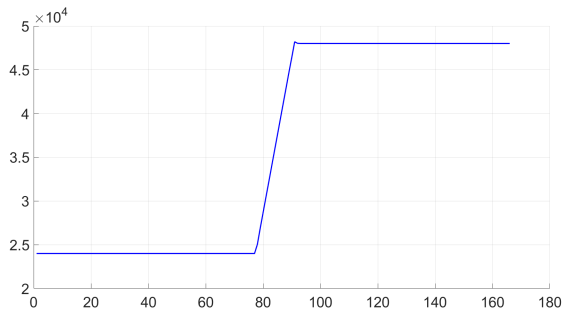
3.8.4 Simulación

La sintaxis de la función utilizada para solucionar el problema de optimización “cplexmilp()” se mantiene casi idéntica, con el único cambio en el parámetro “ctype”, el cual para este caso es:

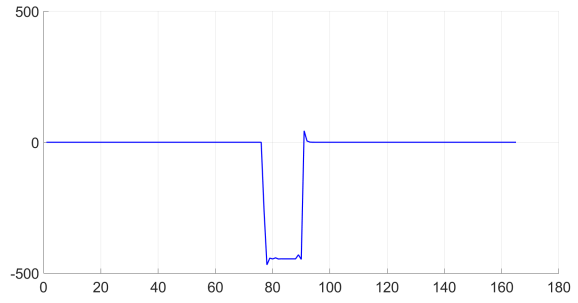
$$ctype = ['C' \ 'C' \ 'B' \ 'C' \ 'B' \ 'B'],$$

Donde la diferencia con respecto al parámetro “ctype” mostrado en la sección 3.6.2 radica en que en este caso se añade una nueva variable de decisión binaria, por lo tanto, también se añade ‘B’ al parámetro “ctype”, dando cuenta de la propiedad binaria de la nueva variable.

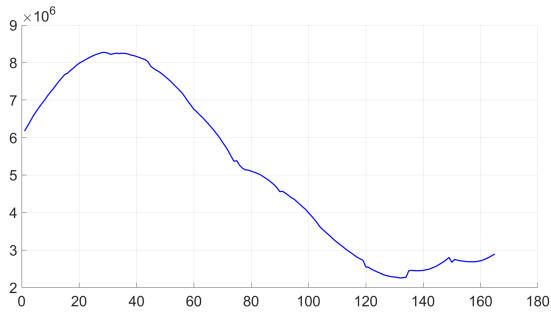
Utilizando los mismos parámetros de la tabla 3.1, los datos en 3.2 y el parámetro “Lim” = 1 se obtiene:



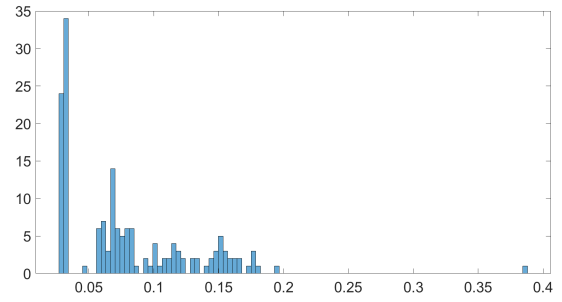
(a) Energía almacenada en la batería



(b) Potencia a la red desde la batería



(c) Rédito económico



(d) Tiempo de ejecución

Figura 3.5: Resultados de simulación con 165 periodos

Donde, en la figura 3.5, el gráfico (a) describe el comportamiento de la energía almacenada en la batería en cada instante de tiempo (E_t^{ES}), (b) es la variable de control en el tiempo P_t^{ES} que determina la dirección de la energía entre la batería y la red (ver fig. 3.1), (c) representa la ganancia económica (\mathcal{J}_3), y (d) muestra el tiempo que tarda la ejecución del calculo realizado por el solver. En el gráfico se observa que el tiempo de ejecución es cercano a los 0.4 segundos en el peor caso, lo cual es aproximadamente 0.17% del período escogido de 4 minutos (240 segundos), es por esto que es posible decir que la ejecución del calculo es suficientemente rápida para una correcta aplicación del control predictivo con tres variables binarias.

Si se observa el gráfico 3.5b puede observarse que la potencia P^{ES} solo cruza el eje x una vez, es decir, solo hay un cambio de carga y descarga en el estado de la batería, lo cual se condice con el limite de un cambio impuesto por el parámetro “Lim”, por lo que este caso ejemplifica perfectamente que se puede limitar los cambios de estado en la batería con el uso de la variable binaria λ sin afectar demasiado el tiempo de ejecución del solver.

3.9 Conflicto en modelo con tres variables y solución

En el caso de ejemplo anterior se demuestra que si es posible llegar al objetivo de limitar la cantidad de cambios en el estado de la batería mediante el modelo de tres variables, sin embargo este modelo presenta el inconveniente de que en ocasiones la solución al problema de optimización es infactible. Esto es debido a que las restricciones que limitan el cambio de estado en la batería podrían entrar en conflicto con la restricción que representa la capacidad máxima de energía que puede almacenar la batería. Por ejemplo, si en el instante actual la carga de la batería es máxima ($E_t^{ES} = E_{max}^{ES}$) y el flujo de batería va en favor a la carga de ésta ($P_t^{ES} < 0$), entonces el control debiese invertir o detener el flujo de energía hacia la batería ($P_t^{ES} \geq 0$), sin embargo, si el límite de cambios de estado en la batería ya ha sido alcanzado este no podrá realizar esta acción y el solver de Cplex determinara que la solución es infactible.

3.9.1 Propuesta

Para solucionar el conflicto presentado anteriormente, se propone que en vez de limitar directamente los cambios de estado de carga y descarga en la batería, se sugiera un número menor de cambios. Esto se consigue añadiendo una nueva variable entera que cuente los cambios de estado realizados, para luego utilizarla en la función de costo a modo de penalización. De esta forma el solver, buscando maximizar el rédito económico, naturalmente buscará disminuir el valor de esta variable, lo que implica disminuir el número de cambios en el estado de la batería.

Con esta finalidad se añade la nueva variable entera “ S_{t+k} ”, la cual acumula el conteo de los cambios de estado y se define como:

$$S_{t+k+1} = S_{t+k} + \lambda_{t+k+1}, \quad (3.91)$$

Además se define un factor de peso “ c_{sw} ”, el cual ayuda a desincentivar el cambio de estado cuando se añade restando en la función de costo \mathcal{J}_2 .

3.10 Problema de optimización con tres variables binarias y una entera

3.10.1 Restricciones

Agregando la definición 3.91 al problema de optimización en reemplazo de la restricción 3.88, se logra eliminar el conflicto que ésta última genera con la restricción 3.121), ya que se elimina el límite explícito a los cambios de estado de carga y descarga en la batería.

Sin embargo, para que posteriormente se pueda utilizar el solver de Cplex en MATLAB con el fin de resolver el problema de optimización es necesario que todas las restricciones sean de la forma “ $Rx \leq r$ ”, ya que la herramienta no admite restricciones de igualdad. Es por esto que resulta necesario ajustar la definición 3.91, lo que se hace a continuación:

Observe que,

$$S_{t+1} = S_t + \lambda_{t+1}, \quad (3.92)$$

$$S_{t+2} = S_{t+1} + \lambda_{t+2}, \quad (3.93)$$

$$S_{t+3} = S_{t+2} + \lambda_{t+3}, \quad (3.94)$$

$$\vdots \quad (3.95)$$

$$S_{t+k} = S_{t+k-1} + \lambda_{t+k}, \quad (3.96)$$

Reemplazando 3.93 y 3.94 en 3.95 se obtiene:

$$S_{t+3} = S_t + \lambda_{t+1} + \lambda_{t+2} + \lambda_{t+3}, \quad (3.97)$$

$$S_{t+3} = S_t + \sum_{i=1}^3 \lambda_{t+i}, \quad (3.98)$$

Por lo tanto, se puede comprobar que:

$$S_{t+k} = S_t + \sum_{i=1}^k \lambda_{t+i}, \quad (3.99)$$

Considerando el caso particular de S_t :

$$S_t = S_0 + \lambda_t, \quad (3.100)$$

Se define el parámetro S_0 como el conteo de cambios de estado acumulado hasta el instante de tiempo “t”. El primer valor que toma este parámetro es cero.

Entonces,

$$S_{t+k} = S_0 + \sum_{i=0}^k \lambda_{t+i}, \quad (3.101)$$

Por lo tanto, las restricciones resultan ser las siguientes:

$$S_{t+k} - \sum_{i=0}^k \lambda_{t+i} \leq S_0, \quad (3.102)$$

$$-S_{t+k} + \sum_{i=0}^k \lambda_{t+i} \leq -S_0, \quad (3.103)$$

De esta forma el objetivo de escribir las restricciones en su forma estándar queda completado.

3.10.2 Reformulación del problema de optimización

Con los cambios mencionados anteriormente, el problema de optimización se reescribe:

$$\max_{P_{t+k}^{ES}} \mathcal{J}_4 = \sum_{k=0}^N \alpha^{-k} T (c_{t+k}^p - c_{t+k}^n) z_{t+k} + T (\alpha^{-k} c_{t+k}^n - c_{t+N+1}^{soc} \eta^c) P_{t+k}^{ES} - c_{t+N+1}^{soc} T (\eta^d - \eta^c) w_{t+k} - c_{sw} S_{t+k} \quad (3.104)$$

s.a.

$$P_{t+k}^{ES} \leq \left[P_{max}^{ES} - (P_{t+k}^{ref} - P_{t+k}^{PV}) \right] \delta_{t+k} + (P_{t+k}^{ref} - P_{t+k}^{PV}), \quad (3.105)$$

$$P_{t+k}^{ES} \geq \left[P_{min}^{ES} - (P_{t+k}^{ref} - P_{t+k}^{PV}) \right] (1 - \delta_{t+k}) + (P_{t+k}^{ref} - P_{t+k}^{PV}), \quad (3.106)$$

$$z_{t+k} \leq \left[P_{max}^{ES} - (P_{t+k}^{ref} - P_{t+k}^{PV}) \right] \delta_{t+k}, \quad (3.107)$$

$$z_{t+k} \geq \left[P_{min}^{ES} - (P_{t+k}^{ref} - P_{t+k}^{PV}) \right] \delta_{t+k}, \quad (3.108)$$

$$z_{t+k} \leq P_{t+k}^{ES} - (P_{t+k}^{ref} - P_{t+k}^{PV}) - \left[P_{min}^{ES} - (P_{t+k}^{ref} - P_{t+k}^{PV}) \right] (1 - \delta_{t+k}), \quad (3.109)$$

$$z_{t+k} \geq P_{t+k}^{ES} - (P_{t+k}^{ref} - P_{t+k}^{PV}) - \left[P_{max}^{ES} - (P_{t+k}^{ref} - P_{t+k}^{PV}) \right] (1 - \delta_{t+k}), \quad (3.110)$$

$$P_{t+k}^{ES} \geq P_{min}^{ES} (1 - \beta_{t+k}), \quad (3.111)$$

$$P_{t+k}^{ES} \leq P_{max}^{ES} \beta_{t+k}, \quad (3.112)$$

$$w_{t+k} \leq P_{max}^{ES} \beta_{t+k}, \quad (3.113)$$

$$w_{t+k} \geq P_{min}^{ES} \beta_{t+k}, \quad (3.114)$$

$$w_{t+k} \leq P_{t+k}^{ES} - P_{min}^{ES} (1 - \beta_{t+k}), \quad (3.115)$$

$$w_{t+k} \geq P_{t+k}^{ES} - P_{max}^{ES} (1 - \beta_{t+k}), \quad (3.116)$$

$$\lambda_{t+k} \geq \beta_{t+k-1} - \beta_{t+k}, \quad (3.117)$$

$$\lambda_{t+k} \geq \beta_{t+k} - \beta_{t+k-1}, \quad (3.118)$$

$$\delta_{t+k} \in \{0, 1\}, \quad \beta_{t+k} \in \{0, 1\}, \quad \lambda_{t+k} \in \{0, 1\}, \quad (3.119)$$

$$(\delta_{t+k} - \beta_{t+k}) (P_{t+k}^{ref} - P_{t+k}^{PV}) \leq 0, \quad (3.120)$$

$$E_{min}^{ES} \leq E_t^{ES} - T \sum_{s=0}^{k-1} \eta_{t+k} P_{t+s}^{ES} \leq E_{max}^{ES}, \quad k = 1, 2, \dots, N-1, N, \quad (3.121)$$

$$S_{t+k} - \sum_{i=0}^k \lambda_{t+i} \leq S_0, \quad k = 0, 1, 2, \dots, N-1, N, \quad (3.122)$$

$$-S_{t+k} + \sum_{i=0}^k \lambda_{t+i} \leq -S_0, \quad k = 0, 1, 2, \dots, N-1, N, \quad (3.123)$$

Las restricciones nuevas se observan en 3.122 y 3.123.

La forma estándar para la posterior resolución utilizando Cplex queda escrita:

$$\max_{\phi \in \Phi} h^T \phi, \quad (3.124)$$

$$s.a. \quad E\phi \leq e, \quad (3.125)$$

Donde la definición de “ ϕ ”, “ Φ ”, “ h ”, “ E ” y “ e ”, es la misma que la de sus homólogos en 3.59 y 3.60.

3.10.3 Representación matricial

Las matrices y vectores que son modificadas o agregadas, respecto a las vistas en 3.8.3, se muestran a continuación:

El vector ϕ , el cual contiene las variables de decisión se construye:

$$\phi = \begin{bmatrix} P_{ES} \\ z \\ \delta \\ w \\ \beta \\ \lambda \\ S \end{bmatrix}, \quad (M14)$$

donde

$$S = \begin{bmatrix} S_{t+0} \\ S_{t+1} \\ \vdots \\ S_{t+N} \end{bmatrix}, \quad (M15)$$

Como la función de costo \mathcal{J}_3 es casi igual a la función de costo \mathcal{J}_4 , esta ultima se puede escribir:

$$h = \begin{bmatrix} f \\ h_4 \end{bmatrix} = \begin{bmatrix} g_1 \\ g_2 \\ 0 \\ g_3 \\ 0 \\ 0 \\ h_4 \end{bmatrix}, \quad \begin{aligned} g_1 &= \alpha \odot c_n - c_{soc} \odot \eta_c, \\ g_2 &= \alpha \odot (c_p - c_n), \\ g_3 &= c_{soc} \odot (\eta_c - \eta_d), \\ h_4 &= -c_{sw}, \end{aligned} \quad (\text{M16})$$

Los vectores y matrices observables en M1, M2, M3, M4 y M7 se mantienen para el problema de optimización con tres variables binarias y una entera.

Las matrices E y e presentes en 3.125 se muestran a continuación:

$$\mathbf{E} = \begin{bmatrix} -\text{I} & 0 & -\text{diag}\{\text{P}_{\min} - \text{R}\} & 0 & 0 & 0 & 0 \\ +\text{I} & 0 & -\text{diag}\{\text{P}_{\max} - \text{R}\} & 0 & 0 & 0 & 0 \\ 0 & -\text{I} & +\text{diag}\{\text{P}_{\min} - \text{R}\} & 0 & 0 & 0 & 0 \\ 0 & +\text{I} & -\text{diag}\{\text{P}_{\max} - \text{R}\} & 0 & 0 & 0 & 0 \\ -\text{I} & +\text{I} & -\text{diag}\{\text{P}_{\min} - \text{R}\} & 0 & 0 & 0 & 0 \\ +\text{I} & -\text{I} & +\text{diag}\{\text{P}_{\max} - \text{R}\} & 0 & 0 & 0 & 0 \\ -\text{I} & 0 & 0 & 0 & -\text{diag}\{\text{P}_{\min}\} & 0 & 0 \\ +\text{I} & 0 & 0 & 0 & -\text{diag}\{\text{P}_{\max}\} & 0 & 0 \\ 0 & 0 & 0 & -\text{I} & +\text{diag}\{\text{P}_{\min}\} & 0 & 0 \\ 0 & 0 & 0 & +\text{I} & -\text{diag}\{\text{P}_{\max}\} & 0 & 0 \\ -\text{I} & 0 & 0 & +\text{I} & -\text{diag}\{\text{P}_{\min}\} & 0 & 0 \\ +\text{I} & 0 & 0 & -\text{I} & +\text{diag}\{\text{P}_{\max}\} & 0 & 0 \\ +\text{V}_1 & 0 & 0 & +\text{V}_2 & 0 & 0 & 0 \\ -\text{V}_1 & 0 & 0 & -\text{V}_2 & 0 & 0 & 0 \\ 0 & 0 & +\text{diag}\{\text{R}\} & 0 & -\text{diag}\{\text{R}\} & 0 & 0 \\ 0 & 0 & 0 & 0 & \text{Q} & -\text{I} & 0 \\ 0 & 0 & 0 & 0 & -\text{Q} & -\text{I} & 0 \\ 0 & 0 & 0 & 0 & 0 & \text{F} & -\text{I} \\ 0 & 0 & 0 & 0 & 0 & -\text{F} & \text{I} \end{bmatrix}, \quad \mathbf{e} = \begin{bmatrix} -\text{P}_{\min} \\ \text{R} \\ 0 \\ 0 \\ -\text{P}_{\min} \\ \text{P}_{\max} \\ -\text{P}_{\min} \\ 0 \\ 0 \\ 0 \\ -\text{P}_{\min} \\ \text{P}_{\max} \\ \text{E}_{\max} \\ -\text{E}_{\min} \\ 0 \\ \text{zb} \\ -\text{zb} \\ -\text{So} \\ \text{So} \end{bmatrix}, \quad (\text{M17})$$

Donde el vector $S_0 \in \mathbb{R}^{N+1}$, y se define:

$$S_0 = \begin{bmatrix} S_0 \\ S_0 \\ S_0 \\ \vdots \\ S_0 \end{bmatrix}, \quad (M18)$$

$I \in \mathbb{R}^{N+1 \times N+1}$ corresponde a la matriz identidad, y “0” representa al vector cero $\in \mathbb{R}^{N+1}$, o a la matriz cero $\in \mathbb{R}^{N+1 \times N+1}$, dependiendo del contexto, en orden de que las dimensiones de las matrices E y e sean consistentes.

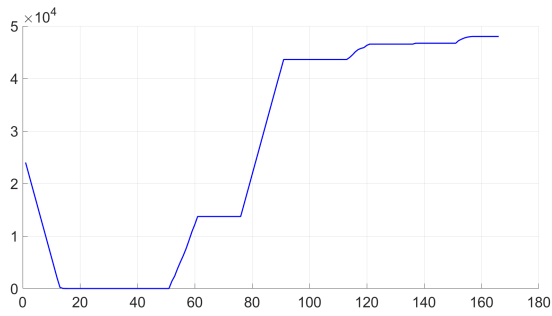
3.10.4 Simulación

La sintaxis de la función utilizada para solucionar el problema de optimización “cplexmilp()” se mantiene casi idéntica, con el único cambio en el parámetro “ctype”, el cual para este caso es:

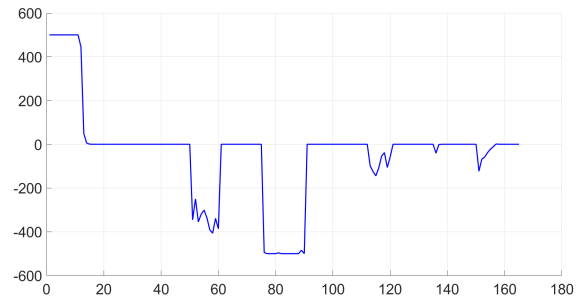
$$ctype = ['C' \ 'C' \ 'B' \ 'C' \ 'B' \ 'B' \ 'I'],$$

Donde la diferencia con respecto al parámetro “ctype” mostrado en la sección 3.8.4 radica en que en este caso se añade una nueva variable de decisión de tipo entera, por lo tanto, también se añade 'I' al parámetro “ctype” lo cual le indica a Cplex que esa variable de decisión es de tipo entera.

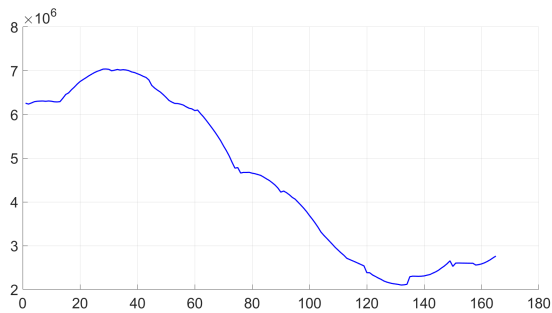
Utilizando los mismos parámetros de la tabla 3.1 y los datos en 3.2, se escoge arbitrariamente el parámetro “ c_{sw} ” = 700 para el ejemplo, ya que se necesita un valor suficientemente alto para que cause impacto en la función de costo a corto plazo, considerando que el objetivo es disminuir el desgaste existente de la batería a mediano y largo plazo. Con lo anterior se obtienen los gráficos:



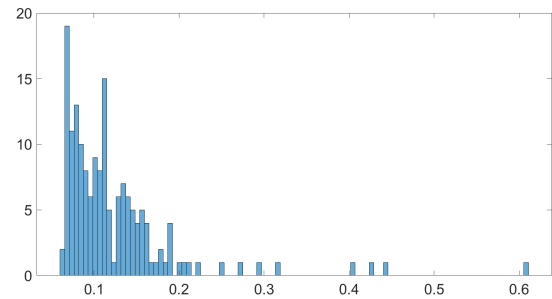
(a) Energía almacenada en la batería



(b) Potencia a la red desde la batería



(c) Rédito económico



(d) Tiempo de ejecución

Figura 3.6: Resultados de simulación con 165 períodos

Donde, en la figura 3.6, el gráfico (a) describe el comportamiento de la energía almacenada en la batería en cada instante de tiempo (E_t^{ES}), (b) es la variable de control en el tiempo P_t^{ES} que determina la dirección de la energía entre la batería y la red (ver fig. 3.1), (c) representa la ganancia económica (\mathcal{J}_4), y (d) muestra el tiempo que tarda la ejecución del calculo realizado por el solver. En el gráfico se observa que el tiempo de ejecución es cercano a los 0.62 segundos en el peor caso, lo cual es aproximadamente 0.26% del período escogido de 4 minutos (240 segundos), es por esto que es posible decir que la ejecución del cálculo es suficientemente rápida para una correcta aplicación del control predictivo con tres variables binarias y una entera.

En el gráfico 3.6b se puede observar que el estado de la batería cambia pocas veces, lo que indica que la implementación de la nueva variable “S” resulta efectiva para el objetivo de bajar indirectamente los cambios de estado.

3.11 Unificación de métodos

En las secciones 3.8 y 3.10 se observan dos formas de abordar el problema que nos lleva a disminuir los ciclos de carga y descarga de las baterías; esto se consigue con limitar directa o indirectamente los cambios entre los estados de carga y descarga de las baterías.

La ventaja que presenta el método de la sección 3.8 radica en que sabemos la cantidad máxima de veces que P^{ES} cambia entre carga y descarga desde la declaración del parámetro “Lim”. Sin embargo, esto presenta el inconveniente ya visto en 3.9. Esta desventaja se soluciona en 3.10, pero, al utilizar este método, se pierde la información concreta sobre las veces que existe un cambio en la dirección del flujo de la energía.

3.11.1 Propuesta

Se propone unificar los métodos de tal manera que se mantenga la obtención de información del flujo de batería, pero manteniendo la viabilidad del problema. La ventaja adicional de esto es que el enfoque radica en disminuir el flujo de energía, es decir, disminuir los ciclos de carga y descarga. Esto se consigue “castigando” la carga y descarga de la batería (cuando $P_{t+k}^{ES} \neq 0$).

Variable binaria:

Para lograr esto, es necesario conocer los instantes de tiempo donde $P_{t+k}^{ES} \neq 0$. Primeramente se crea la nueva variable binaria $v \in [0; 1]$ con el objetivo que se cumpla:

$$v_{t+k} = 1 \leftrightarrow P_{t+k}^{ES} > 0, \quad (3.126)$$

$$v_{t+k} = 0 \leftrightarrow P_{t+k}^{ES} \leq 0 \quad (3.127)$$

Para esto, y haciendo uso de lo visto en 3.4, se define:

- $f(t): P_t^{ES} - \epsilon$,
- $m: P_{min}^{ES} - \epsilon$
- $M: P_{max}^{ES} - \epsilon$

Con ϵ representando el error interno de la maquina (en este caso MATLAB) el cual es positivo y cercano a cero.

Definiendo la variable auxiliar:

$$u_{t+k} = v_{t+k}(P_{t+k}^{ES} - \epsilon) \quad (3.128)$$

donde:

$$u_{t+k} \leq (P_{max}^{ES} - \epsilon)v_{t+k}, \quad (3.129)$$

$$u_{t+k} \geq (P_{min}^{ES} - \epsilon)v_{t+k}, \quad (3.130)$$

$$u_{t+k} \leq (P_{t+k}^{ES} - \epsilon) - (P_{min}^{ES} - \epsilon)(1 - v_{t+k}), \quad (3.131)$$

$$u_{t+k} \geq (P_{t+k}^{ES} - \epsilon) - (P_{max}^{ES} - \epsilon)(1 - v_{t+k}), \quad (3.132)$$

$$P_{t+k}^{ES} - \epsilon \geq (P_{min}^{ES} - \epsilon)(1 - v_{t+k}), \quad (3.133)$$

$$P_{t+k}^{ES} - \epsilon + \epsilon \leq (P_{max}^{ES} - \epsilon + \epsilon)v_{t+k} \quad (3.134)$$

A partir de esto se observa que el objetivo de 3.126 y 3.127 no se cumple, sin embargo se obtiene que:

$$v_{t+k} = 1 \leftrightarrow P_{t+k}^{ES} \geq \epsilon, \quad (3.135)$$

$$v_{t+k} = 0 \leftrightarrow P_{t+k}^{ES} < \epsilon \quad (3.136)$$

Como ϵ es un escalar positivo y su valor es del orden de 10^{-16} , podemos decir que virtualmente se cumplen las condiciones y, por lo tanto, el objetivo para esta aplicación.

Con la variable binaria v_{t+k} y la variable binaria β_{t+k} se logra determinar los intervalos en que P_{t+k}^{ES} es estrictamente positivo, estrictamente negativo, y cero. Esto se consigue mediante la resta de ambas variables binarias antes mencionadas. Para mayor claridad se observan los valores que toma esta resta para los distintos valores que puede tomar P^{ES} (positivo, negativo o cero) en la tabla 3.3.

En la tabla se observa que con la resta $(\beta_{t+k} - v_{t+k})$ se conoce en que intervalos $P^{ES} = 0$, logrando uno de los objetivos mencionados en la sección 3.11, el cual consiste en conocer información concreta

P_{t+k}^{ES}	β_{t+k}	v_{t+k}	$(\beta_{t+k} - v_{t+k})$
> 0	1	1	0
< 0	0	0	0
$= 0$	1	0	1

Tabla 3.3: Valores variables binarias para distintos P^{ES}

de cada intervalo de tiempo en relación al estado de las baterías (carga, descarga o “idle”).

Como el objetivo es disminuir el tiempo (o la cantidad de intervalos de tiempo) en que $P^{ES} \neq 0$, y la resta $(\beta_{t+k} - v_{t+k})$ es 1 cuando $P^{ES} = 0$, entonces se ocupará el complemento:

$$1 - (\beta_{t+k} - v_{t+k}) \quad (3.137)$$

De esta forma se obtiene si en el intervalo de tiempo “k” $P^{ES} \neq 0$.

Variable auxiliar entera:

Usando la misma estrategia de la propuesta vista en 3.9.1 se hace uso de la variable auxiliar entera “ S_{t+k} ”, de modo que esta variable sea la que “cuente” los intervalos de tiempo en que $P^{ES} \neq 0$. Para lograrlo se suma cada intervalo en que $P^{ES} \neq 0$, es decir, cada intervalo que la resta $[1 - (\beta_{t+k} - v_{t+k})] = 1$.

$$S_{t+N} = 1 - (\beta_{t+0} - v_{t+0}) + 1 - (\beta_{t+1} - v_{t+1}) + 1 - (\beta_{t+2} - v_{t+2}) + \dots + 1 - (\beta_{t+N} - v_{t+N}), \quad (3.138)$$

A partir de lo anterior es fácil reducir de la siguiente forma:

$$S_{t+k} = (k + 1) - \sum_{i=0}^k (\beta_{t+i} - v_{t+i}), \quad k = 0, 1, 2, 3, \dots, N, \quad (3.139)$$

Por otro lado, se debe añadir en cada iteración del control predictivo la suma anterior de intervalos en que $P^{ES} \neq 0$. Para esto se añade S_0 , que representa lo anteriormente mencionado, ya que, al solo aplicar el primer valor optimo obtenido mediante control predictivo, la variable entera S_{t+k} va tomando el valor de 0 en cada iteración.

Entonces se obtiene

$$S_{t+k} = S_0 + (k + 1) - \sum_{i=0}^k (\beta_{t+i} - v_{t+i}), \quad k = 0, 1, 2, 3, \dots, N, \quad (3.140)$$

Además se define un factor de peso “ c_{bls} ”, que representa el costo asociado al deterioro de la vida útil de las baterías (“Batería Life-Span”), el cual permite desincentivar el uso activo de estas, o en otras palabras, reducir el número de intervalos en que $P^{ES} \neq 0$.

3.12 Problema de optimización con tres variables binarias y conteo de intervalos donde $P^{ES} \neq 0$

3.12.1 Restricciones

A partir del problema de optimización con dos variables binarias visto en 3.5 se añaden las ecuaciones 3.133 y 3.134, que reducidas quedan:

$$-P_{t+k}^{ES} - (P_{min} - \epsilon) \leq -P_{min}, \quad (3.141)$$

$$P_{t+k}^{ES} - P_{max} \cdot v_{t+k} \leq 0 \quad (3.142)$$

El resto de ecuaciones relacionadas con la variable continua auxiliar “ u_{t+k} ” no son necesarias en esta ocasión.

Por otro lado, se debe añadir las restricciones asociadas a la variable “ S_{t+k} ”, referentes a la ecuación 3.140.

$$S_{t+k} + \sum_{i=0}^k (\beta_{t+i} - v_{t+i}) \leq (k+1) + S_0, \quad (3.143)$$

$$-S_{t+k} - \sum_{i=0}^k (\beta_{t+i} - v_{t+i}) \leq -(k+1) - S_0 \quad (3.144)$$

3.12.2 Reformulación del problema de optimización

Finalmente, con las nuevas restricciones se puede escribir el problema de optimización de la siguiente forma:

$$\max_{P_{t+k}^{ES}} \mathcal{J}_5 = \sum_{k=0}^N \alpha^{-k} T (c_{t+k}^p - c_{t+k}^n) z_{t+k} + T \left(\alpha^{-k} c_{t+k}^n - c_{t+N+1}^{soc} \eta^c \right) P_{t+k}^{ES} - c_{t+N+1}^{soc} T \left(\eta^d - \eta^c \right) w_{t+k} - c_{bls} S_{t+k} \quad (3.145)$$

s.a.

$$P_{t+k}^{ES} \leq \left[P_{max}^{ES} - \left(P_{t+k}^{ref} - P_{t+k}^{PV} \right) \right] \delta_{t+k} + \left(P_{t+k}^{ref} - P_{t+k}^{PV} \right), \quad (3.146)$$

$$P_{t+k}^{ES} \geq \left[P_{min}^{ES} - \left(P_{t+k}^{ref} - P_{t+k}^{PV} \right) \right] (1 - \delta_{t+k}) + \left(P_{t+k}^{ref} - P_{t+k}^{PV} \right), \quad (3.147)$$

$$z_{t+k} \leq \left[P_{max}^{ES} - \left(P_{t+k}^{ref} - P_{t+k}^{PV} \right) \right] \delta_{t+k}, \quad (3.148)$$

$$z_{t+k} \geq \left[P_{min}^{ES} - \left(P_{t+k}^{ref} - P_{t+k}^{PV} \right) \right] \delta_{t+k}, \quad (3.149)$$

$$z_{t+k} \leq P_{t+k}^{ES} - \left(P_{t+k}^{ref} - P_{t+k}^{PV} \right) - \left[P_{min}^{ES} - \left(P_{t+k}^{ref} - P_{t+k}^{PV} \right) \right] (1 - \delta_{t+k}), \quad (3.150)$$

$$z_{t+k} \geq P_{t+k}^{ES} - \left(P_{t+k}^{ref} - P_{t+k}^{PV} \right) - \left[P_{max}^{ES} - \left(P_{t+k}^{ref} - P_{t+k}^{PV} \right) \right] (1 - \delta_{t+k}), \quad (3.151)$$

$$P_{t+k}^{ES} \geq P_{min}^{ES} (1 - \beta_{t+k}), \quad (3.152)$$

$$P_{t+k}^{ES} \leq P_{max}^{ES} \beta_{t+k}, \quad (3.153)$$

$$w_{t+k} \leq P_{max}^{ES} \beta_{t+k}, \quad (3.154)$$

$$w_{t+k} \geq P_{min}^{ES} \beta_{t+k}, \quad (3.155)$$

$$w_{t+k} \leq P_{t+k}^{ES} - P_{min}^{ES} (1 - \beta_{t+k}), \quad (3.156)$$

$$w_{t+k} \geq P_{t+k}^{ES} - P_{max}^{ES} (1 - \beta_{t+k}), \quad (3.157)$$

$$-P_{t+k}^{ES} - (P_{min} - \epsilon) \leq -P_{min}, \quad (3.158)$$

$$P_{t+k}^{ES} - P_{max} \cdot v_{t+k} \leq 0, \quad (3.159)$$

$$\delta_{t+k} \in \{0, 1\}, \quad (3.160)$$

$$\beta_{t+k} \in \{0, 1\}, \quad (3.161)$$

$$v_{t+k} \in \{0, 1\}, \quad (3.162)$$

$$(\delta_{t+k} - \beta_{t+k}) \left(P_{t+k}^{ref} - P_{t+k}^{PV} \right) \leq 0, \quad (3.163)$$

$$E_{min}^{ES} \leq E_t^{ES} - T \sum_{s=0}^{k-1} \eta_{t+k} P_{t+s}^{ES} \leq E_{max}^{ES}, \quad k = 1, 2, \dots, N-1, N, \quad (3.164)$$

$$S_{t+k} + \sum_{i=0}^k (\beta_{t+i} - v_{t+i}) \leq (k+1) + S_0, \quad k = 0, 1, 2, \dots, N-1, N, \quad (3.165)$$

$$-S_{t+k} - \sum_{i=0}^k (\beta_{t+i} - v_{t+i}) \leq -(k+1) - S_0, \quad k = 0, 1, 2, \dots, N-1, N. \quad (3.166)$$

La forma estándar para la posterior resolución utilizando Cplex queda escrita:

$$\max_{\theta \in \Theta} y^T \theta, \quad (3.167)$$

$$s.a. \quad V\theta \leq v, \quad (3.168)$$

Donde la definición de “ θ ”, “ Θ ”, “ y ”, “ V ” y “ v ”, es la misma que la de sus homólogos en 3.59 y 3.60.

3.12.3 Representación matricial

Las matrices y vectores que son modificadas o agregadas, respecto a las vistas en 3.8.3, se muestran a continuación:

El vector ϕ , el cual contiene las variables de decisión se construye:

$$\theta = \begin{bmatrix} P_{ES} \\ z \\ \delta \\ w \\ \beta \\ v \\ S \end{bmatrix}, \quad (M19)$$

donde la matriz “ S ” es la misma definida en M15.

Como la función de costo \mathcal{J}_3 es casi igual a la función de costo \mathcal{J}_5 , esta última se puede escribir:

$$y = \begin{bmatrix} f \\ y_4 \end{bmatrix} = \begin{bmatrix} g_1 \\ g_2 \\ 0 \\ g_3 \\ 0 \\ 0 \\ y_4 \end{bmatrix}, \quad \begin{aligned} g_1 &= \alpha \odot c_n - c_{soc} \odot \eta_c, \\ g_2 &= \alpha \odot (c_p - c_n), \\ g_3 &= c_{soc} \odot (\eta_c - \eta_d), \\ y_4 &= -c_{bls}, \end{aligned} \quad (M20)$$

Los vectores y matrices observables en M1, M2, M3, M4 y M7 se mantienen para el problema de optimización con tres variables binarias y contando cuando $P^{ES} \neq 0$.

Las matrices V y v presentes en 3.168 se muestran a continuación:

$$\mathbf{V} = \begin{bmatrix} -I & 0 & -\text{diag}\{P_{\min} - R\} & 0 & 0 & 0 & 0 \\ +I & 0 & -\text{diag}\{P_{\max} - R\} & 0 & 0 & 0 & 0 \\ 0 & -I & +\text{diag}\{P_{\min} - R\} & 0 & 0 & 0 & 0 \\ 0 & +I & -\text{diag}\{P_{\max} - R\} & 0 & 0 & 0 & 0 \\ -I & +I & -\text{diag}\{P_{\min} - R\} & 0 & 0 & 0 & 0 \\ +I & -I & +\text{diag}\{P_{\max} - R\} & 0 & 0 & 0 & 0 \\ -I & 0 & 0 & 0 & -\text{diag}\{P_{\min}\} & 0 & 0 \\ +I & 0 & 0 & 0 & -\text{diag}\{P_{\max}\} & 0 & 0 \\ 0 & 0 & 0 & -I & +\text{diag}\{P_{\min}\} & 0 & 0 \\ 0 & 0 & 0 & +I & -\text{diag}\{P_{\max}\} & 0 & 0 \\ -I & 0 & 0 & +I & -\text{diag}\{P_{\min}\} & 0 & 0 \\ +I & 0 & 0 & -I & +\text{diag}\{P_{\max}\} & 0 & 0 \\ +V_1 & 0 & 0 & +V_2 & 0 & 0 & 0 \\ -V_1 & 0 & 0 & -V_2 & 0 & 0 & 0 \\ 0 & 0 & +\text{diag}\{R\} & 0 & -\text{diag}\{R\} & 0 & 0 \\ -I & 0 & 0 & 0 & 0 & -\text{diag}\{P_{\min} - \epsilon\} & 0 \\ +I & 0 & 0 & 0 & 0 & -\text{diag}\{P_{\max}\} & 0 \\ 0 & 0 & 0 & 0 & F & -F & I \\ 0 & 0 & 0 & 0 & -F & F & -I \end{bmatrix}, \quad \mathbf{v} = \begin{bmatrix} -P_{\min} \\ R \\ 0 \\ 0 \\ -P_{\min} \\ P_{\max} \\ -P_{\min} \\ 0 \\ 0 \\ 0 \\ -P_{\min} \\ P_{\max} \\ E_{\max} \\ -E_{\min} \\ 0 \\ -P_{\min} \\ 0 \\ U + So \\ -(U + So) \end{bmatrix}, \quad (M21)$$

Donde el vector $S_0 \in \mathbb{R}^{N+1}$ se define igual a M18, y el vector “U” se define:

$$U = \begin{bmatrix} 1 \\ 2 \\ 3 \\ \vdots \\ N \\ N + 1 \end{bmatrix}, \quad (\text{M19})$$

Con $U \in \mathbb{R}^{N+1}$.

$I \in \mathbb{R}^{N+1 \times N+1}$ corresponde a la matriz identidad, y “0” representa al vector cero $\in \mathbb{R}^{N+1}$, o a la matriz cero $\in \mathbb{R}^{N+1 \times N+1}$, dependiendo del contexto, en orden de que las dimensiones de las matrices V y v sean consistentes.

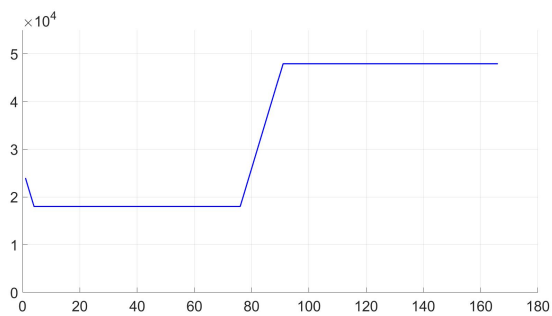
3.12.4 Simulación

La sintaxis de la función utilizada para solucionar el problema de optimización “cplexmilp()” se mantiene idéntica a la simulación en 3.10.4, donde el parámetro “ctype” es:

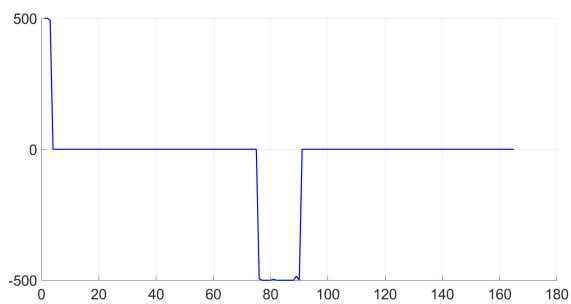
$$ctype = ['C' \ 'C' \ 'B' \ 'C' \ 'B' \ 'B' \ 'I'],$$

Utilizando los mismos parámetros de la tabla 3.1 y los datos en 3.2, se escoge arbitrariamente el parámetro “ c_{bls} ” = 100 para el ejemplo. Nótese que a diferencia del parámetro “ c_{sw} ”, este nuevo parámetro no necesita ser tan alto, ya que la variable “S” crece mas rápido en este caso que en el anterior debido a que esta vez cuenta todos los periodos o intervalos en que P^{ES} es distinto de cero, cuando anteriormente solo contaba los cambios de estado en la batería (carga o descarga).

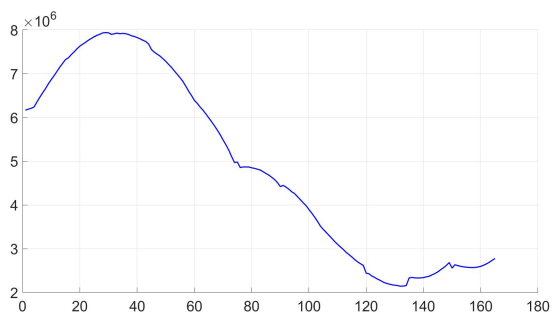
Con lo anterior se obtienen los gráficos:



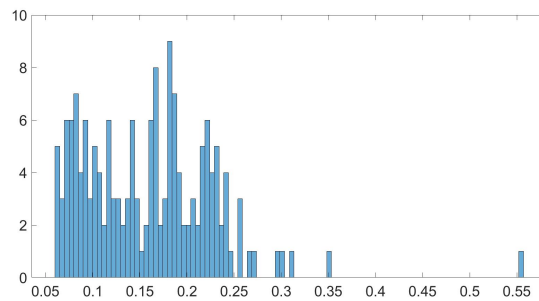
(a) Energía almacenada en la batería



(b) Potencia a la red desde la batería



(c) Rédito económico



(d) Tiempo de ejecución

Figura 3.7: Resultados de simulación con 165 períodos

Donde, en la figura 3.7, el gráfico (a) describe el comportamiento de la energía almacenada en la batería en cada instante de tiempo (E_t^{ES}), (b) es la variable de control en el tiempo P_t^{ES} que determina la dirección de la energía entre la batería y la red (ver fig. 3.1), (c) representa la ganancia económica (\mathcal{J}_4), y (d) muestra el tiempo que tarda la ejecución del calculo realizado por el solver. En el gráfico se observa que el tiempo de ejecución es cercano a los 0.55 segundos en el peor caso, lo cual es aproximadamente 0.23% del período escogido de 4 minutos (240 segundos), es por esto que es posible decir que la ejecución del cálculo es suficientemente rápida para una correcta aplicación del control predictivo.

En el gráfico 3.7b se puede observar que el estado de la batería cambia pocas veces, y además, el tiempo que es P^{ES} es distinto de cero es inferior a su contraparte vista en la propuesta anterior (3.10.4), lo que indica que la nueva propuesta resulta efectiva para el objetivo de bajar indirectamente el uso activo de las baterías, además de lograr conservar la información de la cantidad de intervalos en que existe un uso activo de las baterías a través del valor final de “S”, resultando este en 18 intervalos.

3.13 Análisis comparativo de resultados

A continuación se compararan los gráficos obtenidos en los tres casos: dos variables binarias, tres variables binarias contando los cambios de estado, y tres variables binarias contando $P^{ES} \neq 0$. Principalmente se muestran casos que resultan ser representativos de modelo. No se comparará el caso en que se limita directamente los cambios de estado de la batería, por su gran inconveniente de arrojar una solución infactible en ciertos casos.

3.13.1 Factores de peso $c_{sw} = c_{bls} = 0$

Este caso se refiere al ya visto en el resto del documento, con los parámetros descritos en 3.1, además de los parámetros $c_{sw} = c_{bls} = 0$.

Con la finalidad de observar de mejor forma las diferencias existentes entre los gráficos obtenidos al aplicar MILP con 2 y 4 variables se debe observar la figura 3.8, donde se aprecian los efectos causados por la variable “S” a la energía almacenada E^{ES} .

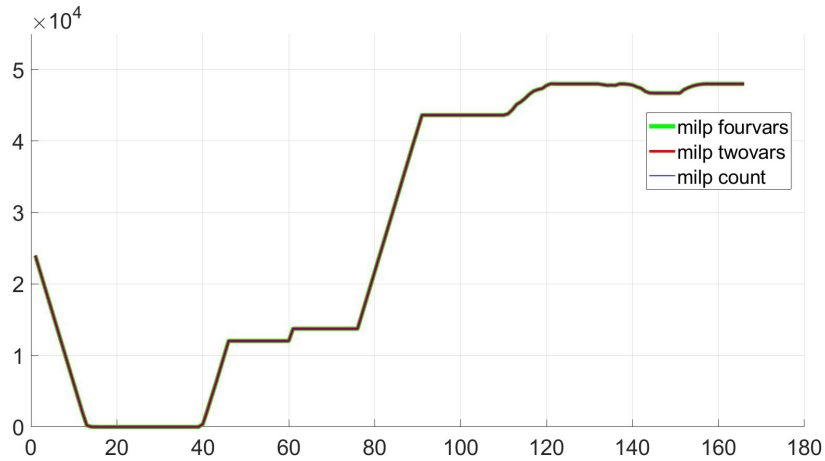
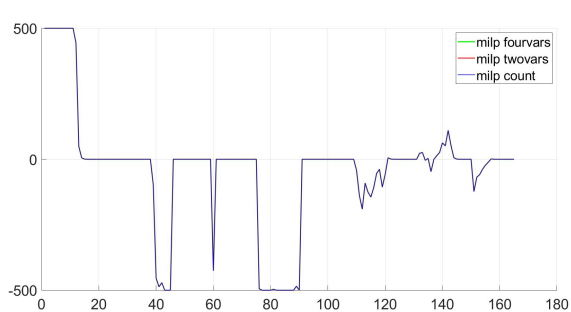
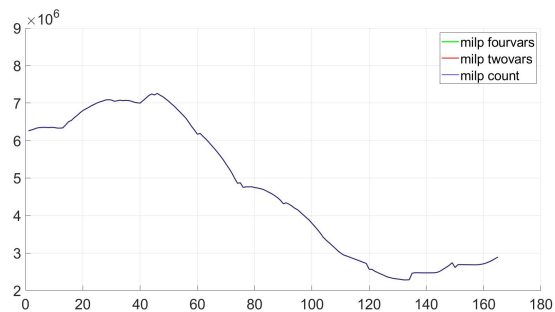


Figura 3.8: E^{ES} para 2 variables, 4 variables y conteo de $P^{ES} \neq 0$

En este caso, y como era de esperarse, los tres gráficos están superpuestos, ya que al hacer los parámetros c_{sw} y c_{bls} cero, la función de costo para MILP de cuatro variables y MILP con conteo de $P^{ES} \neq 0$ queda equivalente a la función de costo para MILP de dos variables binarias. Esto también se puede observar en la figura 3.9, donde las curvas se superponen.



(a) Potencia a la red desde la batería



(b) Rédito económico

Figura 3.9: Comparación simulación para 2 variables, 4 variables y conteo $P^{ES} \neq 0$

3.13.2 Factores de peso $c_{sw} = 700$, $c_{bls} = 50$

Se obtienen los gráficos para la comparación de la potencia desde la batería a la red:

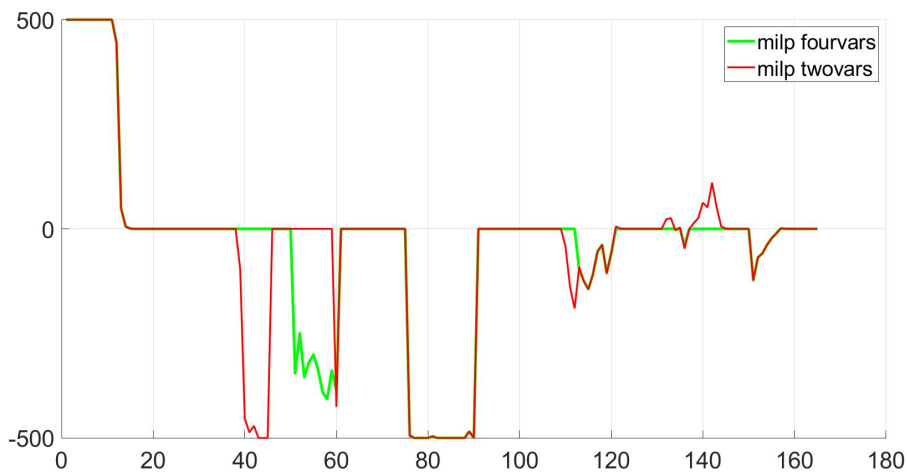


Figura 3.10: P^{ES} para 2 y 4 variables

En el gráfico 3.10 se observa en rojo la curva de P^{ES} para el caso de MILP con dos variables binarias, y en verde la curva de P^{ES} para el caso de MILP de cuatro variables (conteo de cambios de estado de las baterías). Se denota que el uso activo de las baterías (ya sea carga o descarga) ha disminuido en el caso de MILP con cuatro variables, esto en comparación al caso de MILP para dos variables binarias, ya que con un factor de peso c_{sw} de 700 resulta suficiente para bajar el número de cambios de estado en las baterías, e indirectamente baja levemente el uso activo de las baterías.

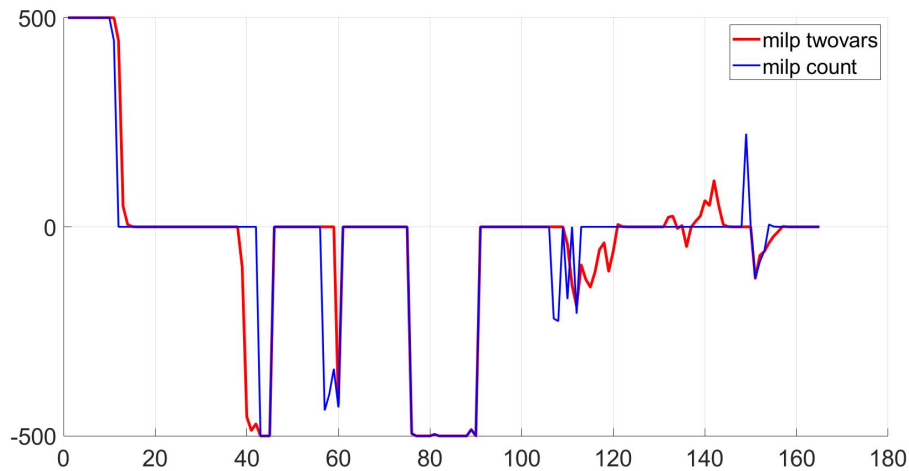


Figura 3.11: P^{ES} para 2 variables binarias, y conteo de $P^{ES} \neq 0$

En el gráfico 3.11 se observa en rojo la curva de P^{ES} para el caso de MILP con dos variables binarias, y en azul la curva de P^{ES} para el caso de MILP con conteo de intervalos en que $P^{ES} \neq 0$. Se denota que el uso activo de las baterías (ya sea carga o descarga) ha disminuido en el caso de MILP con conteo de intervalos en que $P^{ES} \neq 0$, esto en comparación al caso de MILP para dos variables binarias, ya que con un factor de peso c_{bls} de 50 resulta suficiente para bajar levemente el uso activo de las baterías, donde el número de intervalos en que $P^{ES} \neq 0$ asciende a 49.

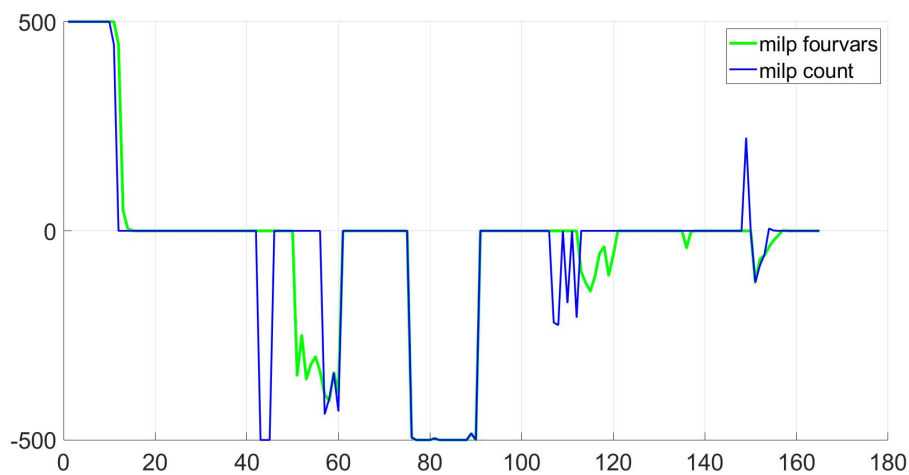


Figura 3.12: P^{ES} para 4 variables, y conteo de $P^{ES} \neq 0$

En la figura 3.12 se comparan ambas propuestas, es decir, la propuesta de MILP con cuatro variables y MILP con conteo de $P^{ES} \neq 0$. Se observa que no existe una diferencia sustancial en

el uso activo de las baterías, sin embargo se ve un peak positivo en el caso de MILP con conteo de $P^{ES} \neq 0$, el cual no se ve en el caso de MILP con cuatro variables, esto es porque el último de estos disminuye el número de cambios de estado (carga y descarga) de las baterías, en cambio en el primer caso esto no se limita, provocando así esta diferencia.

Por otro lado, se observa que el rédito económico obtenido en ambos casos no se ve afectado significativamente, como se puede observar a continuación:

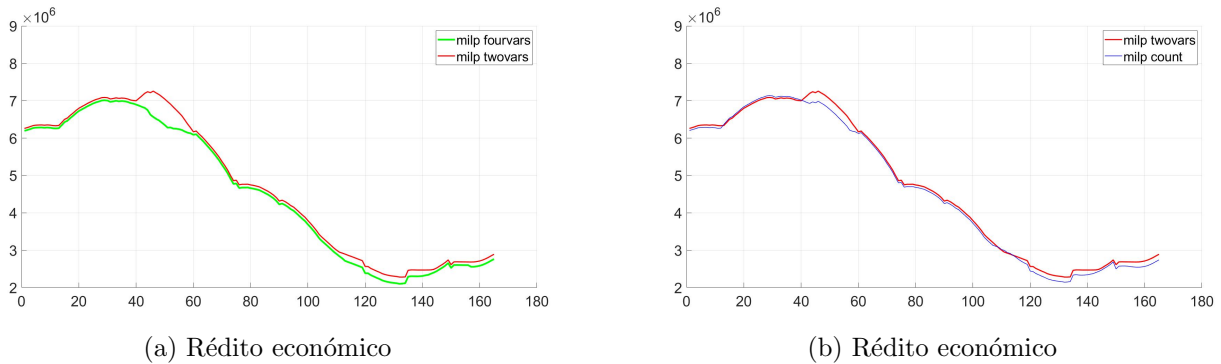


Figura 3.13: Comparación simulación para 2 variables, 4 variables y conteo $P^{ES} \neq 0$

3.13.3 Factores de peso $c_{sw} = 700$, $c_{bls} = 100$

Se obtienen los gráficos para la comparación de la potencia desde la batería a la red:

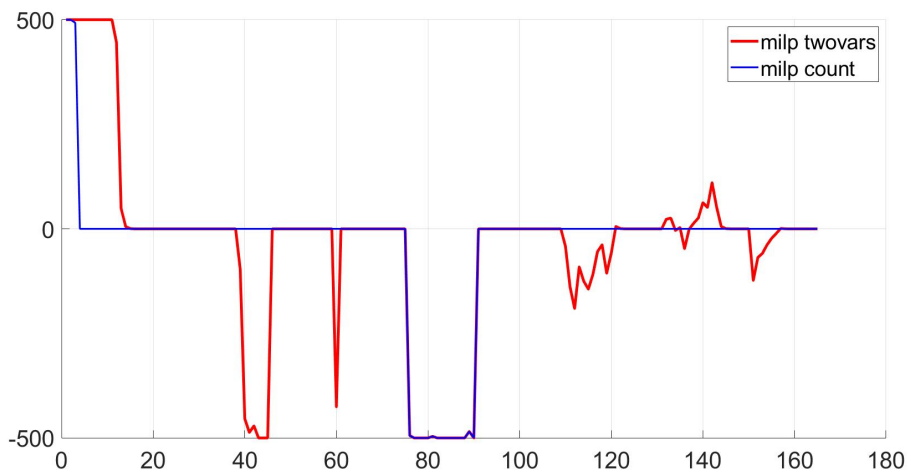


Figura 3.14: P^{ES} para 2 variables binarias, y conteo de $P^{ES} \neq 0$

En el gráfico ?? se observa en rojo la curva de P^{ES} para el caso de MILP con dos variables binarias, y en azul la curva de P^{ES} para el caso de MILP con conteo de intervalos en que $P^{ES} \neq 0$. Se denota que el uso activo de las baterías (ya sea carga o descarga) ha disminuido sustancialmente en el caso de MILP con conteo de intervalos en que $P^{ES} \neq 0$, esto en comparación al caso de MILP para dos variables binarias, ya que con un factor de peso c_{bls} de 100 resulta eficiente para bajar significativamente el uso activo de las baterías, donde el número de intervalos en que $P^{ES} \neq 0$ baja a 18, en comparación con los 49 intervalos del caso anterior.

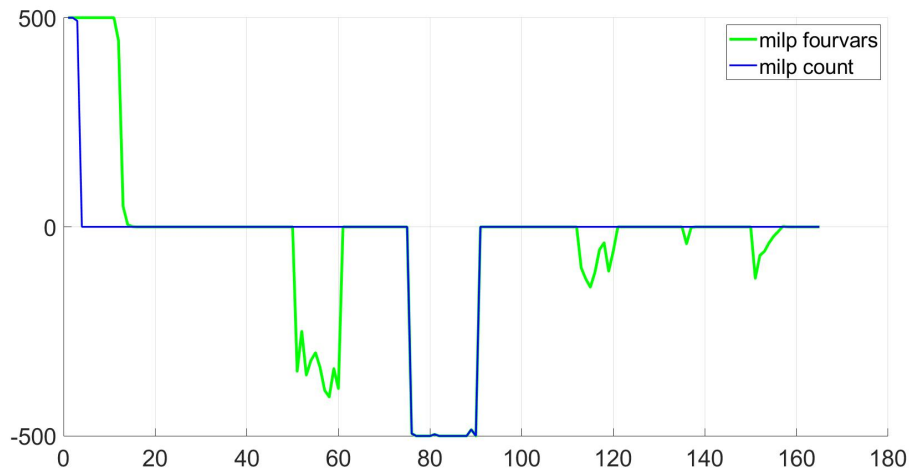


Figura 3.15: P^{ES} para 4 variables, y conteo de $P^{ES} \neq 0$

En la figura 3.15 se comparan ambas propuestas, es decir, la propuesta de MILP con cuatro variables y MILP con conteo de $P^{ES} \neq 0$. En este caso ya se observa que existe una diferencia sustancial en el uso activo de las baterías, donde en el caso de MILP con conteo de $P^{ES} \neq 0$ (curva azul) se observa uno mucho menor.

Por otro lado, se observa que el rédito económico para MILP con conteo de $P^{ES} \neq 0$ es, por algunos momentos, mayor al obtenido al utilizar MILP con dos variables binarias. Esto se debe al parámetro c_{soc} , el cual para la simulación tomo el valor de c_n , lo que favorece la acumulación de energía en las baterías.

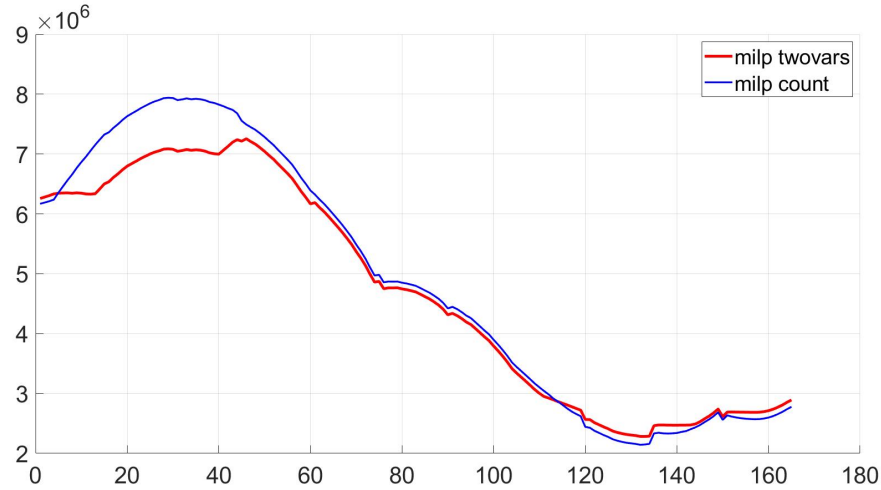


Figura 3.16: Comparación rédito económico para 2 variables binarias y conteo $P^{ES} \neq 0$

3.13.4 Factores de peso $c_{sw} = 1400$, $c_{bls} = 50$

Se obtienen los gráficos para la comparación de la potencia desde la batería a la red:

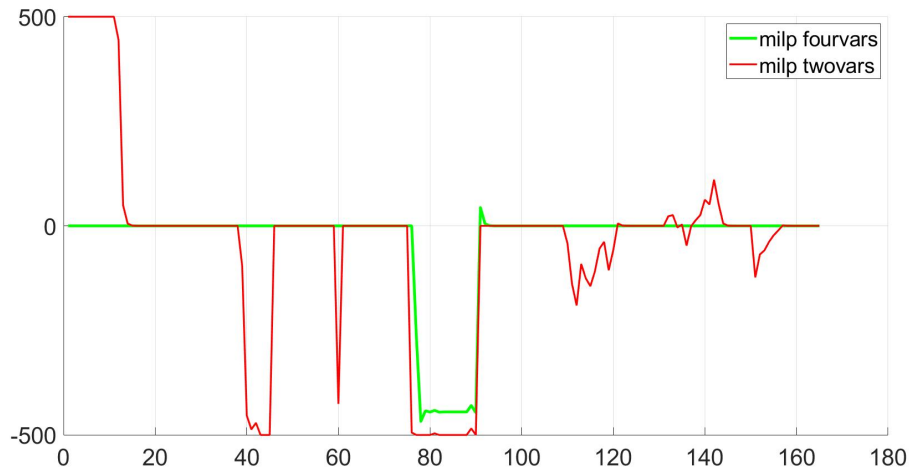


Figura 3.17: P^{ES} para 2 variables binarias y 4 variables

En el gráfico 3.17 se observa en rojo la curva de P^{ES} para el caso de MILP con dos variables binarias, y en verde la curva de P^{ES} para el caso de MILP de cuatro variables (conteo de cambios de estado de las baterías). Se denota que el uso activo de las baterías (ya sea carga o descarga) ha disminuido significativamente en el caso de MILP con cuatro variables, esto en comparación al

caso de MILP para dos variables binarias, ya que con un factor de peso c_{sw} de 1400 resulta eficiente para bajar el número de cambios de estado en las baterías, e indirectamente baja sustancialmente el uso activo de las baterías, donde no se superan los tres cambios de estado.

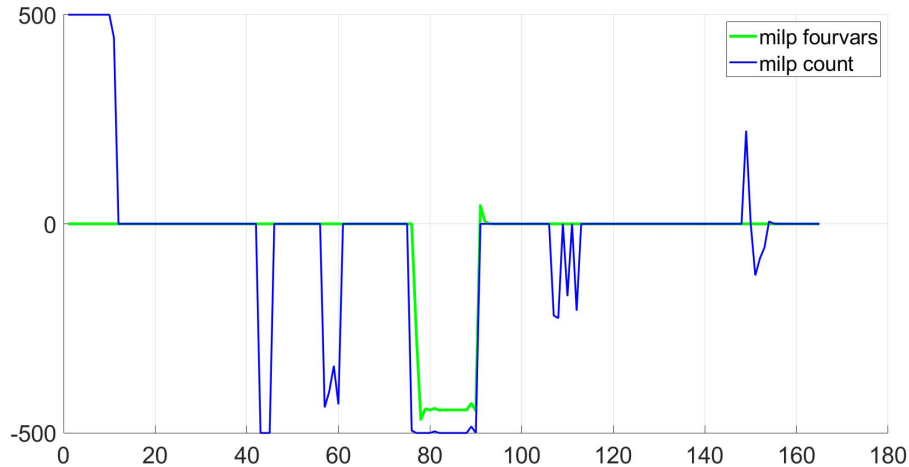


Figura 3.18: P^{ES} para 4 variables, y conteo de $P^{ES} \neq 0$

En la figura 3.21 se comparan ambas propuestas, es decir, la propuesta de MILP con cuatro variables y MILP con conteo de $P^{ES} \neq 0$. En este caso ya se observa que existe una diferencia sustancial en el uso activo de las baterías, donde en el caso de MILP con 2 variables binarias (curva verde) se observa uno mucho menor, ya que en este caso se aumento al doble el parámetro c_{sw} , en cambio el parámetro c_{bls} volvió a un valor menor de 50, esto para realizar la diferenciación entre ambos métodos.

Por otro lado, se observa que el rédito económico para MILP con cuatro variables es, por algunos momentos, mayor al obtenido al utilizar MILP con dos variables binarias. Esto se debe al parámetro c_{soc} , el cual para la simulación tomo el valor de c_n , lo que favorece la acumulación de energía en las baterías.

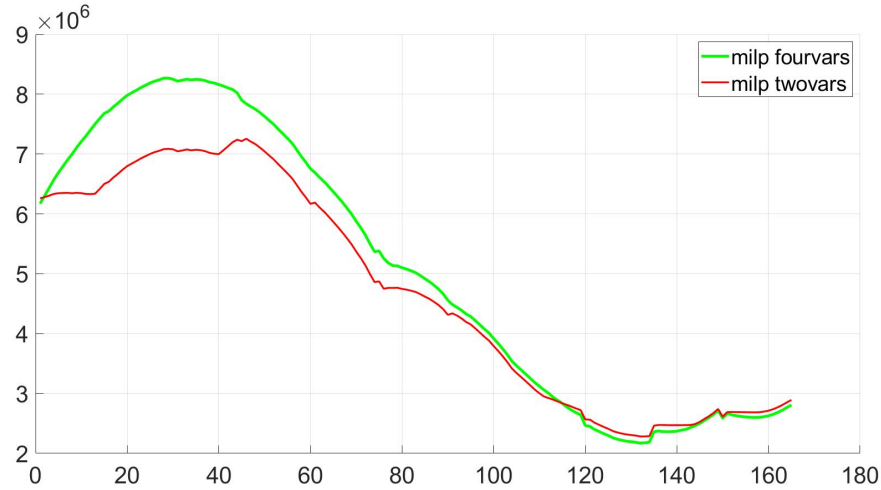


Figura 3.19: Comparación crédito económico para 2 variables binarias y 4 variables

3.13.5 Factores de peso $c_{sw} = 1400$, $c_{bls} = 100$

Se observa en los casos anteriores que se pueden obtener resultados parecidos para ambos métodos, es decir, para MILP con conteo de $P^{ES} \neq 0$ y MILP con cuatro variables. A continuación se comparan los casos donde $c_{sw} = 1400$ y $c_{bls} = 100$.

Se obtienen los gráficos para la comparación de la potencia desde la batería a la red:

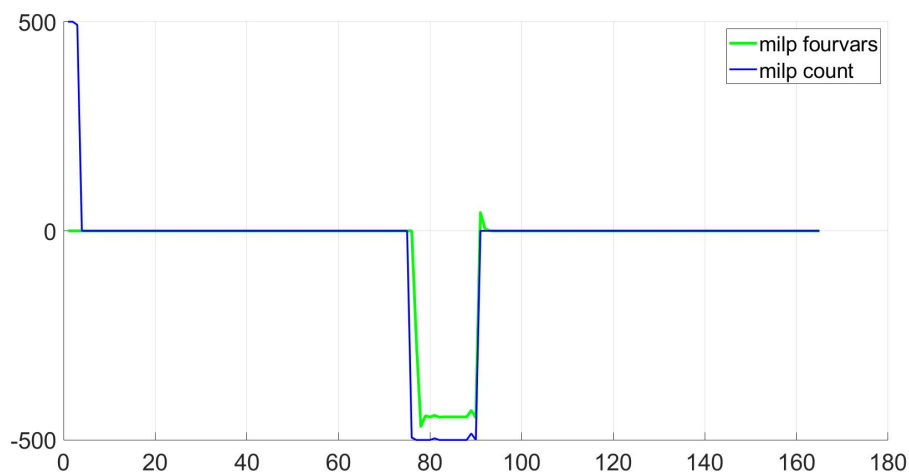


Figura 3.20: P^{ES} para 4 variables, y conteo de $P^{ES} \neq 0$

En la figura 3.20 se comparan ambas propuestas, es decir, la propuesta de MILP con cuatro variables y MILP con conteo de $P^{ES} \neq 0$. En este caso ya se observa que existe una leve diferencia en el uso activo de las baterías, donde en el caso de MILP con 2 variables binarias (curva verde) se observa uno ligeramente menor, esto es debido a la forma en que usan ambos métodos para disminuir el uso activo de las baterías, donde en MILP con conteo de $P^{ES} \neq 0$ le otorga mas libertad a los cambios de estado de las baterías, ya que el método de 4 variables busca limitar esto.

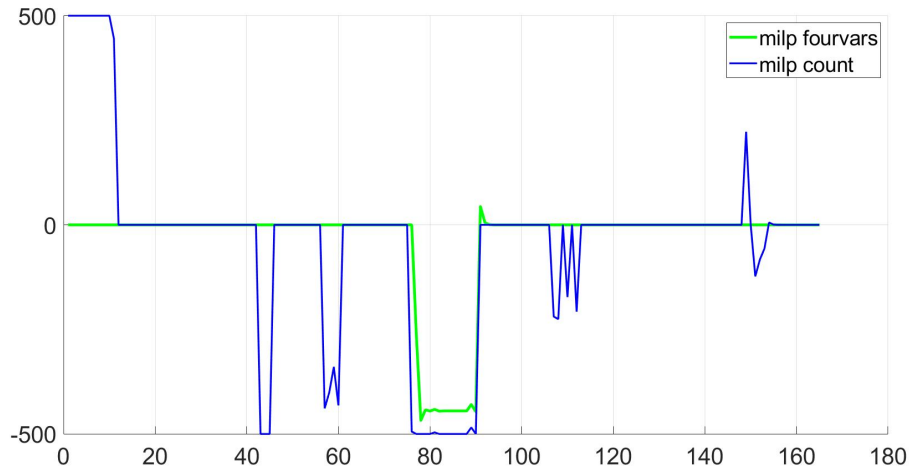


Figura 3.21: P^{ES} para 4 variables, y conteo de $P^{ES} \neq 0$

Por otro lado, se observa que el rédito económico para MILP con cuatro variables es, por algunos momentos, mayor al obtenido al utilizar MILP con conteo de $P^{ES} \neq 0$. Esto se debe al parámetro c_{soc} , el cual para la simulación tomo el valor de c_n , lo que favorece la acumulación de energía en las baterías y, como se observa en la figura 3.21, MILP con cuatro variables almacena mas energía en las baterías que MILP con conteo de $P^{ES} \neq 0$.

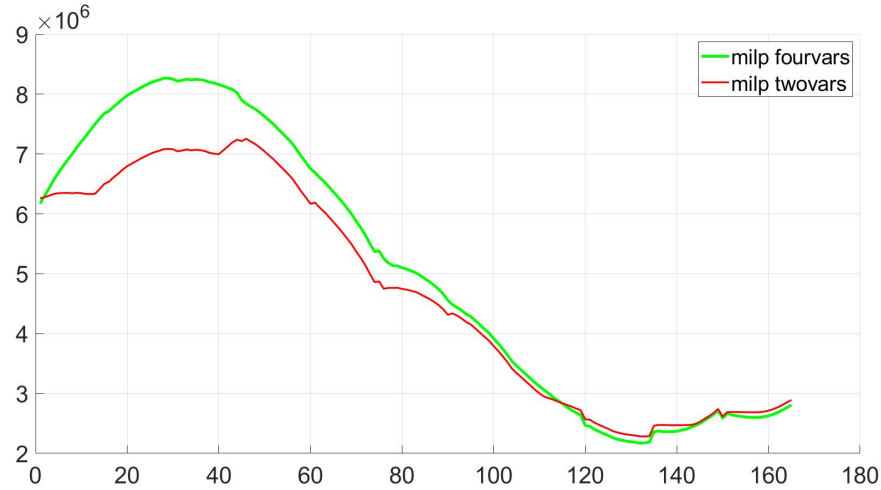


Figura 3.22: Comparación rédito económico para 4 variables, y conteo de $P^{ES} \neq 0$

Para denotar que esto se debe al parámetro c_{soc} , a continuación se observa el rédito económico obtenido cuando ahora c_{soc} se iguala a c_p , lo que debiese impulsar la inyección de energía desde las baterías a la red:

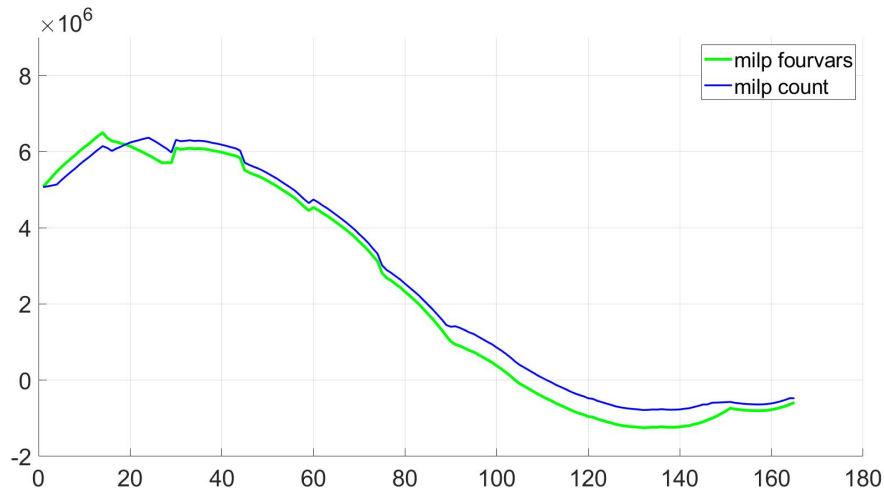


Figura 3.23: Comparación rédito económico para 4 variables, y conteo de $P^{ES} \neq 0$

En la figura 3.23 se observa que el rédito económico de MILP con conteo de $P^{ES} \neq 0$ (curva azul) es superior en la mayoría del tiempo a su contraparte MILP con cuatro variables. Agregado a lo anterior, se observa en la figura 3.24 que ahora en ambos casos existe una mayor inyección de energía a la red, solo debido al cambio en el valor de c_{soc} .

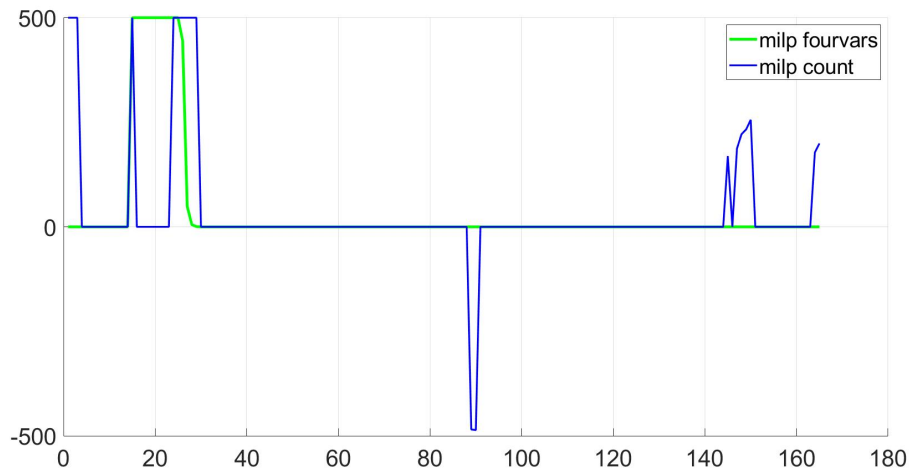


Figura 3.24: P^{ES} para 4 variables, y conteo de $P^{ES} \neq 0$

Por otro lado, se observa que el uso activo de las baterías, en el caso de MILP con conteo de $P^{ES} \neq 0$, es superior al uso activo en MILP con cuatro variables. Todo lo anterior es gracias a que el primer caso le otorga mayor libertad de acción al uso de las baterías, en cuando a cambios de estado en estas se refiere, provocando que se pueda obtener mayores ganancias (o menores pérdidas).

Estos casos representativos dan validez a la propuesta y demuestran que es posible considerar los efectos negativos que sufre la batería cuando se realiza control predictivo en plantas PV. Además se demuestra que haciendo uso de variables binarias y enteras se puede conseguir un efecto muy parecido al de limitar los cambios de estado en la batería, además de disminuir el uso activo de las baterías sin afectar en demasía el rédito económico ni afectar la viabilidad del control .

Capítulo 4

Conclusiones y Recomendaciones

En este capítulo se presentan las conclusiones que se desprenden del desarrollo de este documento, como también dejar posibles caminos para avanzar trabajos futuros en donde se pueda utilizar de base los métodos aplicados.

4.1 Conclusiones y resultados

A lo largo del presente documento se ha desarrollado una estrategia de control predictivo basado en modelos que permite controlar de forma óptima el sistema de baterías de una planta fotovoltaica. La estrategia propuesta maximiza el beneficio económico obtenido de la participación en los mercados de energía, esto mediante la programación de la potencia que se inyecta a la red, optimizando el uso del sistema de almacenamiento de baterías. Además, esta propuesta disminuye el desgaste de las baterías, aumentando el beneficio económico de manera indirecta, sobre todo a largo plazo. El trabajo se ha validado mediante simulación y comparación entre propuestas, donde en el modelo mas simple no se considera el factor degenerativo de las baterías. Además, se demuestra que al agregar este factor, el control propuesto es factible dentro de un período de muestro de cuatro minutos ($T = 4\text{min}$), lo que es suficiente para la aplicación del control en plantas PV. En consecuencia, se considera que la metodología de control propuesta permitirá reducir los costos de producción asumidos por los agentes del mercado, lo que finalmente podría traducirse en un menor costo de energías renovables para los consumidores finales.

4.2 Recomendaciones

A pesar de lo conseguido hasta el momento, resulta posible, y a veces necesario, que se siga avanzando por este camino. Por lo tanto, algunas recomendaciones para seguir la línea de trabajo se exponen:

- Realizar simulaciones con casos reales. Por ejemplo, para este trabajo se supuso que la potencia captada por los paneles solares era igual a la potencia predicha, lo cual en un caso real no es efectivo.
- Considerar desarrollo de algoritmos propios que resuelvan MILP, de modo que el control pueda ser integrado en cualquier dispositivo afín.
- Revisar la posibilidad de entrenar una inteligencia artificial con el modelo para así disminuir los tiempos de ejecución, lo que podría permitir el uso de MILP para MPC en otro tipo de aplicaciones que necesiten más velocidad.

Anexos

Anexo A

Código principal MATLAB

En este anexo se muestra el código principal en MATLAB utilizado para la simulación de MPC con el problema de optimización con 2 variables binarias, 3 variables binarias, 4 variables u conteo de $P^{ES} \neq 0$. Este código contiene todos los parámetros a utilizar y llama a las funciones que resuelven de manera específica cada uno de los problemas de optimización.

(1)

```
%% GENERATION OF IRRADIATION DATA
clc;clear all;
close all;

rng(3000000);

% MPC Tunning parameters
G.sim.N=60;
G.sim.L=165;
G.sim.T=4;
G.sim.alpha=1/0.999;
```

```

% Imbalance prices and power commitment Pref
cpmin=40;%10;%50;%40;%45;%60;          Con (65-75;65-80) no
funciona "perez's solution"
cpmax=60;%15;%60;%70;%65;
cp=(cpmin + (cpmax-cpmin).*rand(25,1));
cnmin=70;
cnmax=75;
cn=(cnmin + (cnmax-cnmin).*rand(25,1));
nzref=[100;200;350;400;550;650;900;850;700;600;200;100];
Pref=[nzref;zeros(13,1)];
G.sim.cp = repelem(cp,15);
G.sim.cn = repelem(cn,15);
G.sim.Pref = repelem(Pref,15);
G.sim.csoc = G.sim.cp;
G.sim.csw = 1400;%0;%1400;%700; % Costo asociado al switcheo del
banco de baterias.

G.sim.cb1s = 120;%100;%50;%0;

xx=0:1:164;
Ppv=1200*sin((1/52.21)*xx);
Ppvhat=[Ppv';zeros(210,1)];

% MPC simulations

%----constraints:----%
G.sim.Ees0=24000;
G.sim.Eesmin=0;
G.sim.Eesmax=48000;
G.sim.Pmin=-500;
G.sim.Pmax=500;

Ees1(1)=G.sim.Ees0;
Ees2(1)=G.sim.Ees0;
Ees3(1)=G.sim.Ees0;

```

```

a=0.95; b=1.05;
r = a + (b-a).*rand(375,1);
G.sim.Ppvhat=Ppvhat.*r;

%% MPC simulations - 2 binary vars -

G.sim.eta_d=1/0.89;
G.sim.eta_c=0.89;

Ees4(1)=G.sim.Ees0;
Ees5(1)=G.sim.Ees0;

T2milp=[]; T2slack=[];

%
=====

% Solving using twovars milp solution
Prob=[];

Prob.T=G.sim.T;
Prob.N=G.sim.N;
Prob.alpha=G.sim.alpha;
Prob.Emin=G.sim.Eesmin;
Prob.Emax=G.sim.Eesmax;
Prob.mk=G.sim.Pmin;
Prob.Mk=G.sim.Pmax;
Prob.eta_d=G.sim.eta_d;
Prob.eta_c=G.sim.eta_c;
expo=0:1:Prob.N;

for t=1:G.sim.L
    Prob.r=G.sim.Pref(t:t+Prob.N)-G.sim.Ppvhat(t:t+Prob.N);
    Prob.cp=G.sim.cp(t:t+Prob.N).*(Prob.alpha.^(-expo)')*Prob.T;
    Prob.cn=G.sim.cn(t:t+Prob.N).*(Prob.alpha.^(-expo)')*Prob.T;
    Prob.b=Prob.T*G.sim.csoc(t+Prob.N+1);

```

```

Prob.E0=Ees4(t);
Prob.cte=G.sim.csoc(t+Prob.N+1)*Ees4(t);

tic
[uvec4,J4(t)] = prob_twovars_milp_solution(Prob);
T2milp(t,1)=toc;

Pes4(t)=uvec4(1);
Ees4(t+1)=Ees4(t)-Prob.T*Pes4(t);
end
%% MPC simulations - 3 binary vars -

G.sim.eta_d=1/0.89;
G.sim.eta_c=0.89;
%%%
G.sim.beta0 = 0;
G.sim.suma0 = 0;
%%%

Ees6(1)=G.sim.Ees0;
Ees7(1)=G.sim.Ees0;

%%%
Beta0(1)=G.sim.beta0;
Suma(1)=G.sim.suma0;
%%%
T3milp=[]; T3slack=[];
%
=====

% Solving using threevars milp solution
Prob=[];
%%%
Prob.Lmax = 1;           % Determina la cantidad maxima de veces
    que cambia el ciclo de carga y descarga
Prob.uvec = zeros(366,1);

```

```

%%%

Prob.T=G.sim.T;
Prob.N=G.sim.N;
Prob.alpha=G.sim.alpha;
Prob.Emin=G.sim.Eesmin;
Prob.Emax=G.sim.Eesmax;
Prob.mk=G.sim.Pmin;
Prob.Mk=G.sim.Pmax;
Prob.eta_d=G.sim.eta_d;
Prob.eta_c=G.sim.eta_c;
expo=0:1:Prob.N;

for t=1:G.sim.L
    Prob.r=G.sim.Pref(t:t+Prob.N)-G.sim.Ppvhat(t:t+Prob.N);
    Prob.cp=G.sim.cp(t:t+Prob.N).*(Prob.alpha.^(-expo)')*Prob.T;
    Prob.cn=G.sim.cn(t:t+Prob.N).*(Prob.alpha.^(-expo)')*Prob.T;
    Prob.b=Prob.T*G.sim.csoc(t+Prob.N+1);
    Prob.E0=Ees6(t);
    Prob.cte=G.sim.csoc(t+Prob.N+1)*Ees6(t);
    %%%
    Prob.Beta0 = Beta0(t);
    Prob.Suma = Suma(t);
    %%%
    tic
    [uvec6,J6(t),betas,lambdas] = prob_threevars_milp_solution(Prob
        );
    T3milp(t,1)=toc;
    Pes6(t)=uvec6(1);
    Ees6(t+1)=Ees6(t)-Prob.T*Pes6(t);
    Beta0(t+1)=betas(1);
    Suma(t+1)=Suma(t) + lambdas(1);
end
Sum_threevars = Prob.Suma;
%% MPC simulations - 3 binary vars - "Suma" -

```

```

G.sim.eta_d=1/0.89;
G.sim.eta_c=0.89;
%%%
G.sim.beta0 = 0;
G.sim.suma0 = 0;
%%%

Ees8(1)=G.sim.Ees0;
Ees9(1)=G.sim.Ees0;

%%%
Beta0(1)=G.sim.beta0;
Suma0(1)=G.sim.suma0;
%%%
T4milp=[]; T4slack=[];
%
=====

% Solving using threevars and sum milp solution
Prob=[];
%%%
Prob.uvec = zeros(366,1);
%%%

Prob.T=G.sim.T;
Prob.N=G.sim.N;
Prob.alpha=G.sim.alpha;
Prob.Emin=G.sim.Eesmin;
Prob.Emax=G.sim.Eesmax;
Prob.mk=G.sim.Pmin;
Prob.Mk=G.sim.Pmax;
Prob.eta_d=G.sim.eta_d;
Prob.eta_c=G.sim.eta_c;
expo=0:1:Prob.N;

```

```

for t=1:G.sim.L
    Prob.r=G.sim.Pref(t:t+Prob.N)-G.sim.Ppvhat(t:t+Prob.N);
    Prob.cp=G.sim.cp(t:t+Prob.N).*(Prob.alpha.^(-expo)')*Prob.T;
    Prob.cn=G.sim.cn(t:t+Prob.N).*(Prob.alpha.^(-expo)')*Prob.T;
    Prob.csw=G.sim.csw;
    Prob.b=Prob.T*G.sim.csoc(t+Prob.N+1);

    Prob.E0=Ees8(t);
    Prob.cte=G.sim.csoc(t+Prob.N+1)*Ees8(t);
    %%%
    Prob.Beta0 = Beta0(t);
    Prob.Suma0 = Suma0(t);
    %%%
    tic
    [uvec8,J8(t),betas,lambdas] = prob_fourvars_milp_solution(Prob)
        ;
    T4milp(t,1)=toc;
    % Actualizacion de datos recursivos:
    Pes7(t)=uvec8(1);
    Ees8(t+1)=Ees8(t)-Prob.T*Pes7(t);
    Beta0(t+1)=betas(1);
    Suma0(t+1)=Suma0(t) + lambdas(1);
end
%% MPC simulations - 3 binary vars - PES changing -

G.sim.eta_d=1/0.89;
G.sim.eta_c=0.89;
%%
G.sim.beta0 = 0;
G.sim.suma0 = 0;
%%

Ees8(1)=G.sim.Ees0;

%%
Beta0(1)=G.sim.beta0;

```

```

Suma0(1)=G.sim.suma0;
%%%
T4milp=[]; T4slack=[];
%
=====

% Solving using threevars and sum milp solution
Prob=[];
%%%
Prob.uvec = zeros(366,1);
%%%

Prob.T=G.sim.T;
Prob.N=G.sim.N;
Prob.alpha=G.sim.alpha;
Prob.Emin=G.sim.Eesmin;
Prob.Emax=G.sim.Eesmax;
Prob.mk=G.sim.Pmin;
Prob.Mk=G.sim.Pmax;
Prob.eta_d=G.sim.eta_d;
Prob.eta_c=G.sim.eta_c;
expo=0:1:Prob.N;

for t=1:G.sim.L
    Prob.r=G.sim.Pref(t:t+Prob.N)-G.sim.Ppvhat(t:t+Prob.N);
    Prob.cp=G.sim.cp(t:t+Prob.N).*(Prob.alpha.^(-expo)')*Prob.T;
    Prob.cn=G.sim.cn(t:t+Prob.N).*(Prob.alpha.^(-expo)')*Prob.T;
    Prob.csw=G.sim.csw;
    Prob.b=Prob.T*G.sim.csoc(t+Prob.N+1);

    Prob.E0=Ees8(t);
    Prob.cte=G.sim.csoc(t+Prob.N+1)*Ees8(t);
    %%%
    Prob.Beta0 = Beta0(t);
    Prob.Suma0 = Suma0(t);
    %%%

```

```

tic
[uvec8,J8(t),betas,lambdas] = prob_fourvars_milp_solution(Prob)
;
T4milp(t,1)=toc;
% Actualizacion de datos recursivos:
Pes7(t)=uvec8(1);
Ees8(t+1)=Ees8(t)-Prob.T*Pes7(t);
Beta0(t+1)=betas(1);
Suma0(t+1)=Suma0(t) + lambdas(1);
end

```

```

%% MPC simulations - 3 binary vars - Counting TimeSteps PES!=0

```

```

G.sim.eta_d=1/0.89;

```

```

G.sim.eta_c=0.89;

```

```

%%

```

```

G.sim.suma0 = 0;

```

```

%%

```

```

Ees9(1)=G.sim.Ees0;

```

```

%%

```

```

Suma0(1)=G.sim.suma0;

```

```

%%

```

```

T5milp=[]; T5slack=[];

```

```

%

```

```

=====

```

```

% Solving using 3vars and sum counting milp solution

```

```

Prob=[];

```

```

%%

```

```

Prob.uvec = zeros(366,1);

```

```

%%

```

```

Prob.T=G.sim.T;
Prob.N=G.sim.N;
Prob.alpha=G.sim.alpha;
Prob.Emin=G.sim.Eesmin;
Prob.Emax=G.sim.Eesmax;
Prob.mk=G.sim.Pmin;
Prob.Mk=G.sim.Pmax;
Prob.eta_d=G.sim.eta_d;
Prob.eta_c=G.sim.eta_c;
expo=0:1:Prob.N;

for t=1:G.sim.L
    Prob.r=G.sim.Pref(t:t+Prob.N)-G.sim.Ppvhat(t:t+Prob.N);
    Prob.cp=G.sim.cp(t:t+Prob.N).*(Prob.alpha.^(-expo))*Prob.T;
    Prob.cn=G.sim.cn(t:t+Prob.N).*(Prob.alpha.^(-expo))*Prob.T;
    Prob.cb1s=G.sim.cb1s;
    Prob.b=Prob.T*G.sim.csoc(t+Prob.N+1);

    Prob.E0=Ees9(t);
    Prob.cte=G.sim.csoc(t+Prob.N+1)*Ees9(t);
    %%%
    Prob.Suma0 = Suma0(t);
    %%%
    tic
    [uvec9,J10(t),count] = prob_threevars_milp_solution_count(Prob)
        ;
    T5milp(t,1)=toc;
    % Actualizacion de datos recursivos:
    Pes9(t)=uvec9(1);
    Ees9(t+1)=Ees9(t)-Prob.T*Pes9(t);
    Suma0(t+1)=count(1);
end
changes=Suma0(length(Suma0));
-----Codigo principal-----

```


Anexo B

Código problema de 2 variables en MATLAB

En este anexo se muestran los códigos utilizados para la resolución del problema de optimización con dos variables binarias, el primero que define las matrices necesarias para la utilización del solver de Cplex, además de la utilización misma de esta herramienta mediante la función “cplexmilp()”, y el segundo que realiza el cálculo de la función de costo para 2 y 3 variables.

(3)

```
function [u,fun]=prob_twovars_milp_solution(Prob)

    IN=eye(Prob.N+1);
    ZE=zeros(Prob.N+1);
    ze=zeros(Prob.N+1,1);

    m=kron(ones(Prob.N+1,1),Prob.mk);
    M=kron(ones(Prob.N+1,1),Prob.Mk);

    amin=kron(ones(Prob.N+1,1),(Prob.E0-Prob.Emax)/Prob.T);
    amax=kron(ones(Prob.N+1,1),(Prob.E0-Prob.Emin)/Prob.T);

    % This product is because eta_d and eta_c are constants
    V1=(Prob.eta_c)*tril(ones(Prob.N+1));
    V2=(Prob.eta_d-Prob.eta_c)*tril(ones(Prob.N+1));

    r=Prob.r;

    eps=0;

    C=[-IN, ZE,-diag(m-r),ZE,ZE;...
        IN, ZE,-diag(M-r+eps),ZE,ZE;...
        ZE,-IN,+diag(m-r),ZE,ZE;...

```

```

    ZE, IN, -diag(M-r), ZE, ZE; ...
-IN, IN, -diag(m-r), ZE, ZE; ...
    IN, -IN, +diag(M-r), ZE, ZE; ...
-IN, ZE, ZE, ZE, -diag(m); ...
    IN, ZE, ZE, ZE, -diag(M+eps); ...
    ZE, ZE, ZE, -IN, +diag(m); ...
    ZE, ZE, ZE, IN, -diag(M); ...
-IN, ZE, ZE, IN, -diag(m); ...
    IN, ZE, ZE, -IN, +diag(M); ...
    V1, ZE, ZE, V2, ZE; ...
-V1, ZE, ZE, -V2, ZE; ...
    ZE, ZE, diag(r), ZE, -diag(r)];

```

```

d=[ -m; ...
    r-eps; ...
    ze; ...
    ze; ...
-m; ...
    M; ...
-m; ...
    ze-eps; ...
    ze; ...
    ze; ...
-m; ...
    M; ...
    amax;
-amin; ...
    ze];

```

```

g1=Prob.cn-Prob.b*Prob.eta_c;
g2=Prob.cp-Prob.cn;
g3=kron(ones(Prob.N+1,1),Prob.b*(Prob.eta_c-Prob.eta_d));
g = [ g1; ...
      g2; ...
      ze; ...

```

```

        g3;...
        ze];

% Setting the integer (binary) variables
ctype=[repmat('C',1,2*(Prob.N+1)),repmat('B',1,Prob.N+1),repmat
('C',1,(Prob.N+1)),repmat('B',1,Prob.N+1)];

% Solving the optimization problem
[uvec] = cplexmilp(-g,C,d,[],[],[],[],[],[],[],[],ctype);

% Obtaining the optimal u neglecting the others variables
u=uvec(1:Prob.N+1);

% Computing the optimal cost
[fun] = cost_twovars(u,Prob);
end
-----Solver para 2 variables binarias-----

```

(4)

(5)

```

function [J] = cost_twovars(u,Prob)
    J=0;
    for t=1:1:Prob.N+1
        if u(t)-Prob.r(t)>=0
            ct=Prob.cp(t);
        else
            ct=Prob.cn(t);
        end
        if u(t)>=0

```

```
        eta=Prob.eta_d;
else
        eta=Prob.eta_c;
end
J=J+ct*(u(t)-Prob.r(t))-Prob.b*eta*u(t);
end
J=Prob.cte+J;
end
-----Funcion de costo para 2 y 3 variables binarias-----
```

(6)

Anexo C

Código problema de 3 variables en MATLAB

En este anexo de muestra los código utilizado para la resolución del problema de optimización con tres variables binarias, donde se definen las matrices necesarias para la utilización del solver de Cplex, además de la utilización misma de esta herramienta mediante la función “cplexmilp()”. El código que realiza el calculo de la función de costo para 3 variables se encuentra en el anexo B (4.2).

(7)

```
function [u,fun,betas,lambdas]=prob_threevars_milp_solution(Prob)

    IN=eye(Prob.N+1);
    ZE=zeros(Prob.N+1);
    ze=zeros(Prob.N+1,1);

    m=kron(ones(Prob.N+1,1),Prob.mk);
    M=kron(ones(Prob.N+1,1),Prob.Mk);
    amin=kron(ones(Prob.N+1,1),(Prob.E0-Prob.Emax)/Prob.T);
    amax=kron(ones(Prob.N+1,1),(Prob.E0-Prob.Emin)/Prob.T);

    % This product is because eta_d and eta_c are constants
    V1=(Prob.eta_c)*tril(ones(Prob.N+1));
    V2=(Prob.eta_d-Prob.eta_c)*tril(ones(Prob.N+1));

    % Matriz para nueva variable binaria (tercera)
    st = kron(ones(Prob.N+1,1),(Prob.Lmax-Prob.Suma));
    zb = [-Prob.Beta0;zeros(Prob.N,1)];
    F=tril(ones(Prob.N+1));
    Q = [[-1,zeros(1,Prob.N)];[eye(Prob.N),zeros(Prob.N,1)] - [
        zeros(Prob.N,1),eye(Prob.N)]]];
```

```
r=Prob.r;
```

```
eps=0;
```

```
D=[-IN, ZE,-diag(m-r),ZE,ZE,ZE;...  
    IN, ZE,-diag(M-r+eps),ZE,ZE,ZE;...  
    ZE,-IN,+diag(m-r),ZE,ZE,ZE;...  
    ZE, IN,-diag(M-r),ZE,ZE,ZE;...  
    -IN, IN,-diag(m-r),ZE,ZE,ZE;...  
    IN,-IN,+diag(M-r),ZE,ZE,ZE;...  
    -IN, ZE, ZE, ZE,-diag(m),ZE;...  
    IN, ZE, ZE, ZE,-diag(M+eps),ZE;...  
    ZE, ZE, ZE,-IN,+diag(m),ZE;...  
    ZE, ZE, ZE, IN,-diag(M),ZE;...  
    -IN, ZE, ZE, IN,-diag(m),ZE;...  
    IN, ZE, ZE,-IN,+diag(M),ZE;...  
    V1, ZE, ZE, V2,ZE,ZE;...  
    -V1, ZE, ZE,-V2,ZE,ZE;...  
    ZE,ZE,diag(r),ZE,-diag(r),ZE;...%3 SIGUIENTES VAR Lambda  
    ZE,ZE,ZE,ZE,ZE,F;...  
    ZE,ZE,ZE,ZE,Q,-IN;...  
    ZE,ZE,ZE,ZE,-Q,-IN];
```

```
e=[ -m;...  
    r-eps;...  
    ze;...  
    ze;...  
    -m;...  
    M;...  
    -m;...  
    ze-eps;...  
    ze;...  
    ze;...  
    -m;...  
    M;...]
```

```

    amax;
-amin;...
    ze;...%3 SIGUIENTES VAR LAMBDA
    st;...
    zb;...
-zb];

k1=Prob.cn-Prob.b*Prob.eta_c;
k2=Prob.cp-Prob.cn;
k3=kron(ones(Prob.N+1,1),Prob.b*(Prob.eta_c-Prob.eta_d));
k = [ k1;...
      k2;...
      ze;...
      k3;...
      ze;...
      ze];

% Setting the integer (binary) variables
ctype=[repmat('C',1,2*(Prob.N+1)),repmat('B',1,Prob.N+1),repmat
('C',1,(Prob.N+1)),repmat('B',1,2*(Prob.N+1))];

% Solving the optimization problem
%[uvec] = cplexmilp(-k,D,e,[],[],[],[],[],[],[],[],ctype);
[uvec, fval, exitflag, output] = cplexmilp(-k, D, e, [], [],
[], [], [], [], [], ctype);
%
if isempty(uvec)
    disp('No se encontr una soluci n factible. ');
    switch exitflag
        case 1
            disp('Soluci n ptima encontrada. ');
        case 0
            disp('Se alcanz el l mite de tiempo. ');
        case -1
            disp('Problema no resuelto debido a limitaciones. ')
    ;

```

```

    case -2
        disp('El problema es infactible.');
```

otherwise

```

        disp(['Exit flag no reconocido: ', num2str(exitflag
            )]);
    end
end
%
% Obtaining the optimal u neglecting the others variables
Prob.uvec = uvec;
u=uvec(1:Prob.N+1);
betas=uvec(4*(Prob.N+1)+1:5*(Prob.N+1));
lambdas=uvec(5*(Prob.N+1)+1:6*(Prob.N+1));
% Computing the optimal cost
[fun] = cost_threevars(u,Prob);
end
-----Solver para 3 variables binarias-----
```

(8)

[Suspicious Content]

Anexo D

Código problema de 4 variables en MATLAB

En este anexo de muestran los códigos utilizados para la resolución del problema de optimización con tres variables binarias y una variable entera; el primero define las matrices necesarias para la utilización del solver de Cplex, además de la utilización misma de esta herramienta mediante la función “cplexmilp()”, y el segundo que realiza el calculo de la función de costo para 3 variables binarias y una entera.

(9)

```
function [u,fun,betas,lambdas]=prob_fourvars_milp_solution(Prob)

    IN=eye(Prob.N+1);
    ZE=zeros(Prob.N+1);
    ze=zeros(Prob.N+1,1);

    m=kron(ones(Prob.N+1,1),Prob.mk);
    M=kron(ones(Prob.N+1,1),Prob.Mk);
    amin=kron(ones(Prob.N+1,1),(Prob.E0-Prob.Emax)/Prob.T);
    amax=kron(ones(Prob.N+1,1),(Prob.E0-Prob.Emin)/Prob.T);

    % This product is because eta_d and eta_c are constants
    V1=(Prob.eta_c)*tril(ones(Prob.N+1));
    V2=(Prob.eta_d-Prob.eta_c)*tril(ones(Prob.N+1));

    % Matriz para nueva variable binaria (tercera)
    zb = [-Prob.Beta0;zeros(Prob.N,1)];
    F = tril(ones(Prob.N+1));
    Q = [[-1,zeros(1,Prob.N)];[eye(Prob.N),zeros(Prob.N,1)] - [
        zeros(Prob.N,1),eye(Prob.N)]];

    % Matriz Suma0
```

```

s0 = kron(ones(Prob.N+1,1),(Prob.Suma0));

r=Prob.r;

eps=0;

D=[-IN, ZE,-diag(m-r),ZE,ZE,ZE,      ZE;...
    IN, ZE,-diag(M-r+eps),ZE,ZE,ZE,    ZE;...
    ZE,-IN,+diag(m-r),ZE,ZE,ZE,      ZE;...
    ZE, IN,-diag(M-r),ZE,ZE,ZE,      ZE;...
    -IN, IN,-diag(m-r),ZE,ZE,ZE,      ZE;...
    IN,-IN,+diag(M-r),ZE,ZE,ZE,      ZE;...
    -IN, ZE, ZE, ZE,-diag(m),ZE,      ZE;...
    IN, ZE, ZE, ZE,-diag(M+eps),ZE,    ZE;...
    ZE, ZE, ZE,-IN,+diag(m),ZE,      ZE;...
    ZE, ZE, ZE, IN,-diag(M),ZE,      ZE;...
    -IN, ZE, ZE, IN,-diag(m),ZE,      ZE;...
    IN, ZE, ZE,-IN,+diag(M),ZE,      ZE;...
    V1, ZE, ZE, V2,ZE,ZE,      ZE;...
    -V1, ZE, ZE,-V2,ZE,ZE,      ZE;...
    ZE,ZE,diag(r),ZE,-diag(r),ZE,      ZE;...%2 SIGUIENTES VAR
    LAMBDA
    ZE,ZE,ZE,ZE,Q,-IN,      ZE;...
    ZE,ZE,ZE,ZE,-Q,-IN,      ZE;...%2 SIGUIENTES VAR Suma
    ZE,ZE,ZE,ZE,ZE,F,      -IN;...
    ZE,ZE,ZE,ZE,ZE,-F,      IN];

e=[ -m;...
    r-eps;...
    ze;...
    ze;...
    -m;...
    M;...
    -m;...
    ze-eps;...

```

```

ze;...
ze;...
-m;...
M;...
amax;
-amin;...
ze;...%2 SIGUIENTES VAR LAMBDA
zb;...
-zb;...%2 SIGUIENTES VAR Suma
-s0;...
s0];

k1=Prob.cn-Prob.b*Prob.eta_c;
k2=Prob.cp-Prob.cn;
k3=kron(ones(Prob.N+1,1),Prob.b*(Prob.eta_c-Prob.eta_d));
k4=kron(ones(Prob.N+1,1),-Prob.csw);
k = [ k1;...
      k2;...
      ze;...
      k3;...
      ze;...
      ze;...
      k4];

% Setting the integer (binary) variables
ctype=[ repmat('C',1,2*(Prob.N+1)), repmat('B',1,Prob.N+1), repmat
('C',1,(Prob.N+1)), repmat('B',1,2*(Prob.N+1)), repmat('I',1,(
Prob.N+1))];

% Solving the optimization problem
%[uvec] = cplexmilp(-k,D,e,[],[],[],[],[],[],[],[],ctype);
[uvec, fval, exitflag, output] = cplexmilp(-k, D, e, [], [],
[], [], [], [], [], ctype);
%
if isempty(uvec)
    disp('No se encontr una soluci n factible.');
```

```

switch exitflag
    case 1
        disp('Soluci n ptima encontrada.');
```

```

    case 0
        disp('Se alcanz el l mite de tiempo.');
```

```

    case -1
        disp('Problema no resuelto debido a limitaciones.')
```

```

        ;
```

```

    case -2
        disp('El problema es infactible.');
```

```

    otherwise
        disp(['Exit flag no reconocido: ', num2str(exitflag
            )]);
```

```

end
```

```

end
```

```

%
```

```

% Obtaining the optimal u neglecting the others variables
```

```

Prob.uevec = uevec;
```

```

u=uevec(1:Prob.N+1);
```

```

betas=uevec(4*(Prob.N+1)+1:5*(Prob.N+1));
```

```

lambdas=uevec(5*(Prob.N+1)+1:6*(Prob.N+1));
```

```

% Computing the optimal cost
```

```

[fun] = cost_fourvars(u,Prob);
```

```

end
```

```

-----Solver para 3 variables binarias y 1 entera
-----
```

(10)

```

function [J] = cost_fourvars(u,Prob)
    J=0;
    for t=1:1:Prob.N+1
        if u(t)-Prob.r(t)>=0
            ct=Prob.cp(t);
        else
            ct=Prob.cn(t);
        end
        if u(t)>=0
            eta=Prob.eta_d;
        else
            eta=Prob.eta_c;
        end
        J=J+ct*(u(t)-Prob.r(t))-Prob.b*eta*u(t)-(Prob.Suma0)*Prob.
            csw; %Recordar Prob.cp y Prob.cn ya tienen factor "T"
    end
    J=Prob.cte+J;
end
-----Funcion de costo para 3 variables binarias y 1 entera-----

```

Anexo E

Código problema de conteo $P^{ES} \neq 0$ en MATLAB

En este anexo se muestran los códigos utilizados para la resolución del problema de optimización con tres variables binarias y una variable entera donde se cuentan los intervalos en que $P^{ES} \neq 0$; el primero define las matrices necesarias para la utilización del solver de Cplex, además de la utilización misma de esta herramienta mediante la función “cplexmilp()”, y el segundo que realiza el cálculo de la función de costo correspondiente.

(13)

```
function [u,fun,count]=prob_threevars_milp_solution_count(Prob)

    IN=eye(Prob.N+1);
    ZE=zeros(Prob.N+1);
    ze=zeros(Prob.N+1,1);

    m=kron(ones(Prob.N+1,1),Prob.mk);
    M=kron(ones(Prob.N+1,1),Prob.Mk);
    amin=kron(ones(Prob.N+1,1),(Prob.E0-Prob.Emax)/Prob.T);
    amax=kron(ones(Prob.N+1,1),(Prob.E0-Prob.Emin)/Prob.T);

    % This product is because eta_d and eta_c are constants
    V1=(Prob.eta_c)*tril(ones(Prob.N+1));
    V2=(Prob.eta_d-Prob.eta_c)*tril(ones(Prob.N+1));

    % Matriz para nueva variable binaria (tercera)
    U = (1:Prob.N+1)';
    F = tril(ones(Prob.N+1));

    % Matriz Suma0
    s0 = kron(ones(Prob.N+1,1),(Prob.Suma0));
```

```

r=Prob.r;

% eps=0;

D=[-IN, ZE, -diag(m-r), ZE, ZE, ZE, ZE;...
    IN, ZE, -diag(M-r+eps), ZE, ZE, ZE, ZE;...
    ZE, -IN, +diag(m-r), ZE, ZE, ZE, ZE;...
    ZE, IN, -diag(M-r), ZE, ZE, ZE, ZE;...
    -IN, IN, -diag(m-r), ZE, ZE, ZE, ZE;...
    IN, -IN, +diag(M-r), ZE, ZE, ZE, ZE;...
    -IN, ZE, ZE, ZE, -diag(m), ZE, ZE;...
    IN, ZE, ZE, ZE, -diag(M+eps), ZE, ZE;...
    ZE, ZE, ZE, -IN, +diag(m), ZE, ZE;...
    ZE, ZE, ZE, IN, -diag(M), ZE, ZE;...
    -IN, ZE, ZE, IN, -diag(m), ZE, ZE;...
    IN, ZE, ZE, -IN, +diag(M), ZE, ZE;...
    V1, ZE, ZE, V2, ZE, ZE, ZE;...
    -V1, ZE, ZE, -V2, ZE, ZE, ZE;...

    ZE, ZE, diag(r), ZE, -diag(r), ZE, ZE;...
%2 SIGUIENTES VAR UPSILON
    -IN, ZE, ZE, ZE, ZE, -diag(m-eps), ZE;...
    +IN, ZE, ZE, ZE, ZE, -diag(M), ZE;...
%2 SIGUIENTES VAR Suma
    ZE, ZE, ZE, ZE, F, -F, IN;...
    ZE, ZE, ZE, ZE, -F, F, -IN];

e=[ -m;...
    r-eps;...
    ze;...
    ze;...
    -m;...
    M;...
    -m;...
    ze-eps;...

```

```

ze;...
ze;...
-m;...
M;...
amax;
-amin;...
ze;...

                %2 SIGUIENTES VAR UPSILON

-m;...
ze;...

                %2 SIGUIENTES VAR Suma

(U+s0);...
-(U+s0)];

k1=Prob.cn-Prob.b*Prob.eta_c;
k2=Prob.cp-Prob.cn;
k3=kron(ones(Prob.N+1,1),Prob.b*(Prob.eta_c-Prob.eta_d));
k4=kron(ones(Prob.N+1,1),-Prob.cb1s);
k = [ k1;...      %PES
      k2;...      %z
      ze;...      %delta
      k3;...      %w
      ze;...      %beta
      ze;...      %upsilon
      k4];        %Suma

% Setting the integer (binary) variables
ctype=[ repmat('C',1,2*(Prob.N+1)), repmat('B',1,Prob.N+1), repmat
('C',1,Prob.N+1), ...
       repmat('B',1,Prob.N+1), repmat('B',1,Prob.N+1), repmat('I',1,
       Prob.N+1)];

% Solving the optimization problem
%[uvec] = cplexmilp(-k,D,e,[],[],[],[],[],[],[],[],ctype);
[uvec, fval, exitflag, output] = cplexmilp(-k, D, e, [], [],
[], [], [], [], [], ctype);

```

```

%
if isempty(uvec)
    disp('No se encontr una soluci n factible. ');
    switch exitflag
        case 1
            disp('Soluci n ptima encontrada. ');
        case 0
            disp('Se alcanz el l mite de tiempo. ');
        case -1
            disp('Problema no resuelto debido a limitaciones. ');
            ;
        case -2
            disp('El problema es infactible. ');
        otherwise
            disp(['Exit flag no reconocido: ', num2str(exitflag
                )]);
    end
end
end
%

% Obtaining the optimal u neglecting the others variables
Prob.uvec = uvec;
u=uvec(1:Prob.N+1); %PES

count=uvec(6*(Prob.N+1)+1:7*(Prob.N+1)); %Suma

% Computing the optimal cost
[fun] = cost_threevars_count(u,Prob,count);
end
-----Solver para MILP con conteo de PES distinto a 0-----

```

(14)

```

function [J] = cost_threevars_count(u,Prob,S)
    J=0;
    for t=1:1:Prob.N+1
        if u(t)-Prob.r(t)>=0
            ct=Prob.cp(t);
        else
            ct=Prob.cn(t);
        end
        if u(t)>=0
            eta=Prob.eta_d;
        else
            eta=Prob.eta_c;
        end
        J=J+ct*(u(t)-Prob.r(t))-Prob.b*eta*u(t)-S(t)*Prob.cb1s; %
            Recordar Prob.cp y Prob.cn ya tienen factor "T"
    end
    J=Prob.cte+J;
end

```

-----Funcion de costo para MILP con conteo de PES distinto a 0-----

Referencias

- [1] B. Khan; O.P. Mahela; H.H. Alhelou; S. Padmanaban. *Deregulated Electricity Market: The Smart Grid Perspective*. Apple Academic Press: Palm Bay, FL, USA, 2022.
- [2] José Rojas Gonzalo Carvajal César Silva Angel L. Cedeño, Reinier López Ahuar and Juan C. Agüero. Model predictive control for photovoltaic plants with non-ideal energy storage using mixed integer linear programming. *MDPI, Energies*, 2022.
- [3] ESIOS. Red eléctrica de españa. sistema de información del operador del sistema. *Available online: <https://www.esios.ree.es>*, 2024.
- [4] Evangelos Rikos Alessandra Parisio and Luigi Glielmo. A model predictive control approach to microgrid operation optimization. *IEEE Transactions on control systems technology*, 22(5), 2014.
- [5] YASUHIRO HAYASHI RYU ANDO, HIDEO ISHII and GUIPING ZHU. A planned power generation for battery-assisted photovoltaic system using short-term forecast. *IEEE Access*, 9, 2021.
- [6] Manfred Morari Alberto Bemporad. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35:407–427, 1999.
- [7] M.A. Hannan; S.B. Wali; P.J. Ker; M.S. Abd Rahman; M. Mansor; V.K. Ramachandaramurthy; K.M. Muttaqi; T.M.I. Mahlia; Z.Y. Dong. Battery energy-storage system: A review of technologies, optimization objectives, constraints, approaches, and outstanding issues. *Journal of Energy Storage*, 42, 2021.
- [8] Chris Menictasb Merlinde Kaya Yuqing Yanga, Stephen Bremnera. Battery energy storage system size determination in renewable energy systems: A review. *Renewable and Sustainable Energy Reviews*, 91(109-125), 2018.
- [9] Timothy L. Skvarenina. *The Power Electronics Handbook*. CRC Press LC, 2002.

- [10] Xuesong Zhou Youjie Ma Zhiqiang Gao, Song Li. An overview of pv system. *IEEE International Conference on Mechatronics and Automation*, 2016, August, Harbin, China.
- [11] Ashkan Toopshekan Leyli Bahrami Amin Maghami Narges Ghorbani, Alibakhsh Kasaeian. Optimizing a hybrid wind-pv-battery system using ga-pso and mopso for reducing cost and increasing reliability. *Energy*, 154(1):581–591, 2018.
- [12] Lei Wu Qianli Ma, WeiWei and Shengwei Mei. Life-aware operation of battery energy storage in frequency regulation. *IEEE Transactions on sustainable energy*, 14(3), 2023.
- [13] Néstor Aparicio Emilio Pérez, Hector Beltran and Pedro Rodriguez. Predictive power control for pv plants with energy storage. *IEEE Transactions on sustainable energy*, 4, 2012.