Trabajo Fin de Máster
Máster en Ingeniería Industrial

Model Predictive Control Based on Deep Learning for Solar Parabolic-Trough Plants

Autor: Sara Ruiz Moreno

Tutores: Eduardo Fernández Camacho
José Ramón Domínguez Frejo

Dpto. Ingeniería de Sistemas y Automática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2021

Trabajo Fin de Máster

Máster en Ingeniería Industrial

Model Predictive Control Based on Deep Learning for Solar Parabolic-Trough Plants

Autor:

Sara Ruiz Moreno

Tutores:

Eduardo Fernández Camacho

Catedrático de Universidad

José Ramón Domínguez Frejo

Investigador Juan de la Cierva

Dpto. Ingeniería de Sistemas y Automática

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2021

Trabajo Fin de Máster: Model Predictive Control Based on Deep Learning for Solar Parabolic-Trough Plants

Autor: Sara Ruiz Moreno
Tutores: Eduardo Fernández Camacho, José Ramón Domínguez Frejo

El tribunal nombrado para juzgar el trabajo arriba indicado, compuesto por los siguientes profesores:

Presidente:

Vocal/es:

Secretario:

acuerdan otorgarle la calificación de:

El Secretario del Tribunal

Fecha:

Acknowledgment

De todo éxito, por pequeño que sea, surgirá un día un esfuerzo más grande que lo completará.

Walt Whitman

A mis padres, por enseñarme a querer y aprender

A mis hermanos, por acompañarme en mi crecimiento personal

A Raúl, por ser, además de un compañero, un ejemplo a seguir

A José Ramón, por apoyarme y motivarme desde el primer momento

A Eduardo, por ayudarme y darme la oportunidad de formar parte de este proyecto

A Antonio y Adolfo, por enseñarme y acompañarme en lo que me queda por recorrer

Resumen

En la actualidad, cada vez es mayor el interés por utilizar energías renovables, entre las que se encuentra la energía solar. Las plantas de colectores cilindro-parabólicos son un tipo de planta termosolar donde se hace incidir la radiación del Sol en unos tubos mediante el uso de unos espejos con forma de parábola. En el interior de estos tubos circula un fluido, generalmente aceite o agua, que se calienta para generar vapor y hacer girar una turbina, produciendo energía eléctrica.

Uno de los métodos más utilizados para manejar estas plantas es el control predictivo basado en modelo (*model predictive control*, MPC), cuyo funcionamiento consiste en obtener las señales de control óptimas que se enviarán a la planta basándose en el uso de un modelo de la misma. Este método permite predecir el estado que adoptará el sistema según la estrategia de control escogida a lo largo de un horizonte de tiempo.

El MPC tiene como desventaja un gran coste computacional asociado a la resolución de un problema de optimización en cada instante de muestreo. Esto dificulta su implementación en plantas comerciales y de gran tamaño, por lo que, actualmente, uno de los principales retos es la disminución de estos tiempos de cálculo, ya sea tecnológicamente o mediante el uso de técnicas subóptimas que simplifiquen el problema.

En este proyecto, se propone el uso de redes neuronales que aprendan *offline* de la salida proporcionada por un controlador predictivo para luego poder aproximarla. Se han entrenado diferentes redes neuronales utilizando un conjunto de datos de 30 días de simulación y modificando el número de entradas. Los resultados muestran que las redes neuronales son capaces de proporcionar prácticamente la misma potencia que el MPC con variaciones más suaves de la salida y muy bajas violaciones de las restricciones, incluso disminuyendo el número de entradas. El trabajo desarrollado se ha publicado en *Renewable Energy*, una revista del primer cuartil en *Green & sustainable science & technology* y *Energy and fuels* [83].

1 Introducción

Debido al creciente impacto del ser humano en el medio ambiente, hay cada vez un mayor interés en el uso de fuentes de energía renovables. Entre ellas, la más antigua y abundante es la energía solar [30], que posteriormente se transforma en energía térmica o eléctrica. Estas se pueden dividir en dos grandes grupos: celdas fotovoltaicas (*pthotovoltaic cells*, PV), que reciben la luz solar y la transforman directamente en electricidad, y las centrales térmicas solares (*concentrating solar power*, CSP), que utilizan espejos para concentrar la luz solar en unos receptores y calentar un fluido para producir vapor y accionar una turbina. Este segundo grupo posee la ventaja de poder incluir almacenamiento (*thermal energy storage*, TES) y se divide a su vez en colectores cilindro-parabólicos, colectores Fresnel, centrales de torre y colectores de plato. Este trabajo se centra en el uso de colectores cilindro parabólicos.

Una de las técnicas más importantes para controlar este tipo de plantas es el control predictivo debido a su capacidad de abordar comportamientos no lineales y restricciones teniendo en cuenta las salidas futuras de la planta. El problema de estos controladores es que necesitan resolver un problema de optimización cada cierto tiempo (segundos o minutos), lo que aumenta su coste computacional y dificulta su implementación en grandes plantas comerciales. Para resolverlo, en este trabajo se propone el uso de redes neuronales. El objetivo es aproximar la salida de un controlador MPC utilizando una ANN y realizar varios experimentos sobre un simulador de la planta.

Este trabajo forma parte del proyecto ERC H2020 OCONTSOLAR (*Optimal Control of Thermal Solar Energy Systems*) [No 789051], concedido al profesor Eduardo Fernández Camacho de la Universidad de Sevilla y cuyos objetivos son el control e integración de redes de sensores móviles en sistemas de control de plantas solares. Todo el trabajo realizado se ha aplicado en un modelo de la planta solar ACUREX [100], antiguamente situada en la Plataforma Solar de Almería (PSA), un centro de investigación perteneciente al Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas (CIEMAT).

2 Estado del Arte

En este capítulo se proporciona una visión general de las plantas termosolares, las ecuaciones utilizadas para describir el campo de colectores cilindro-parabólicos y los principales métodos de control empleados en la literatura, centrándose en el control predictivo y las redes neuronales. Además, se nombran algunos trabajos recientes existentes en la literatura.

2.1 Control Predictivo Basado en Modelo

Gran parte de los trabajos recientes en el control de colectores cilindro-parabólicos (*parabolic-trough collector*, PTC) se basan en MPC, una técnica consistente en la optimización de una función de coste (o función objetivo) que utiliza un modelo dinámico del proceso. A partir del modelo se realizan predicciones del estado futuro del sistema y se obtienen las señales de control óptimas. Las características del control predictivo le permiten hacer frente a perturbaciones y tiempos muertos. El optimizador proporciona las señales de control u a lo largo de un horizonte de control N_u utilizando generalmente el error entre una trayectoria de referencia y las salidas predichas en un horizonte de predicción N_p, pero se puede incluir cualquier objetivo en la función de coste. Teniendo en cuenta las predicciones $\hat{y}(t+k \mid t)$, obtenidas en el instante t para $k = 1,...,N_p$ y las salidas futuras $u(t+k \mid t)$, la siguiente ecuación muestra la forma de la función de coste típica, que considera el error de seguimiento de referencia y el esfuerzo de control Δu. Se añaden unos pesos $\delta(j)$ y $\lambda(j)$ para ajustar la penalización de cada término de la función objetivo.

$$\begin{aligned} J(\hat{y}(t+1 \mid t),...,\hat{y}(t+N \mid t),\Delta u(t),\Delta u(t+1),...,\Delta u(t+N_u-1)) = \\ = \sum_{j=1}^{N_p} \delta(j)[\hat{y}(t+j \mid t) - w(t+j)]^2 + \sum_{j=1}^{N_u} \lambda(j)[\Delta u(t+j-1)]^2 \end{aligned} \tag{0.1}$$

Además, se pueden incluir restricciones en el problema de optimización dadas generalmente por límites de operación y de seguridad.

Control Predictivo No Lineal

En muchas ocasiones, se utilizan modelos no lineales para representar los sistemas con mayor precisión, ya que esto mejora el comportamiento del controlador. Esto hace que se tengan que aplicar técnicas de optimización no lineal. Estos problemas no son necesariamente convexos, por lo que no garantizan que se alcance el óptimo global, y son mucho más difíciles de resolver. En estos casos, se realizan búsquedas de la solución de forma iterativa, lo que aumenta considerablemente los tiempos de cálculo y puede llevar al incumplimiento de las restricciones.

Los problemas lineales se pueden resolver mediante programación cuadrática. Frecuentemente, se aborda el problema de la no linealidad con técnicas de programación cuadrática secuencial (*sequential quadratic programming*, SQP) [17], un método iterativo que sustituye la función de coste y las restricciones por aproximaciones lineales en cada instante. Aun así, si el tiempo necesario para resolver el problema sigue siendo alto o se alcanza un óptimo local, se puede llegar a soluciones lejos del óptimo y que violen las restricciones. Para ello, el algoritmo utiliza varios puntos iniciales, haciendo que nuevamente aumente el coste computacional.

En este trabajo, se propone el uso de redes neuronales que aprendan la solución dada por un MPC no lineal calculado *offline* con alto coste computacional. Después, la red neuronal será capaz de aproximar su salida en tiempos mucho más cortos.

2.2 Redes neuronales

Una red neuronal es un algoritmo que trata de emular el funcionamiento del cerebro humano. De forma matemática, es una función $f_{\mathrm{NN}}(\cdot)$ formada por sumas ponderadas de funciones $g(\cdot)$. Como indica [79], el término inteligencia artificial (*artificial intelligence*, AI) se puede describir como la capacidad de un dispositivo de percibir el ambiente y realizar acciones para alcanzar un objetivo. Aprendizaje automático (*machile learning*, ML), por su parte, es la parte de la AI que dota a las máquinas de la capacidad de aprender, y aprendizaje profundo (*deep learning*, DL) es la parte del ML donde se utilizan grandes redes neuronales.

Regresión Lineal

Cada una de las neuronas (también llamadas nodos) de una red neuronal resuelve un problema de regresión lineal, que consiste en encontrar la recta que mejor se ajuste a un conjunto de datos. Este es un tipo de aprendizaje supervisado, es decir, la salida es previamente conocida y el objetivo es aproximarla lo mejor posible.

La técnica utilizada para obtener los parámetros de la curva es el descenso por gradiente. Se toma un punto y se busca la dirección de mayor gradiente. Los parámetros decrecen iterativamente de forma proporcional a la pendiente hasta que se encuentra un mínimo.

Perceptrón

La red neuronal más básica es el perceptrón, que consiste en una celda que recibe unas entradas, resuelve un problema de regresión lineal y proporciona una salida que posteriormente se binariza. Como esta celda por si sola solo es capaz de modelar relaciones lineales, se utiliza el perceptrón multicapa (*multilayer perceptron*, MLP), donde todas las neuronas de una capa están conectadas con todas las de la capa anterior. Las redes neuronales se distribuyen en tres tipos de capas: capa de entrada, capa de salida y capas ocultas (las que transforman la información entre la entrada y la salida). Cuando una red neuronal tiene dos o más capas ocultas, se dice que es profunda (*deep neural network*).

Funciones de Activación

A la salida de las neuronas se utilizan diferentes funciones para realizarles distintos tratamientos. Estas son las funciones de activación, que permiten añadir restricciones y límites y facilitan el proceso de entrenamiento. Las funciones de activación más típicas son las siguientes:

- Lineal: $g(z) = z$
- ReLU: $g(z) = \max(0,z)$
- Sigmoide: $g(z) = \frac{1}{1+e^{-z}}$
- Tangente hiperbólica: $g(z) = tanh(z) = \frac{e^{z}-e^{-z}}{e^{z}+e^{-z}}$

Propagación Hacia Atrás

Los parámetros de las redes neuronales necesitan ser determinados de forma iterativa. Esto se realiza mediante un proceso llamado propagación hacia atrás o retropropagación (más conocido por su nombre en inglés, *backpropagation*) en el que se calculan los errores de las neuronas a partir del error de salida mediante el siguiente algoritmo simplificado:

Algoritmo 1 Algoritmo de propagación hacia atrás [84].

```
Inicializar los pesos
para cada dato de entrenamiento hacer
    Leer el dato
    Obtener la salida
    Calcular el error de la salida
    Propagar los errores
    Obtener las derivadas parciales de una función objetivo respecto a los pesos
    Actualizar los parámetros
    Obtener el coste
fin para cada
```

Entrenamiento de una red neuronal

Para entrenar una red neuronal, se realiza un proceso de prueba y error, donde se preprocesan los datos, se dividen en subconjuntos de entrenamiento, validación y prueba, se selecciona la arquitectura y parámetros y se inicia el entrenamiento para calcular los pesos. Una vez finalizado éste, se evalúa la red neuronal y se vuelven a modificar los parámetros hasta obtener buenos resultados. Existen muchas métricas para evaluar la red neuronal. En este caso, se utiliza el coeficiente de correlación de Pearson R y el error cuadrático medio (*mean squared error*, MSE):

$$R = \frac{\sum_{i=1}^{N}(x_i - x_{media})(y_i - y_{media})}{\sqrt{\sum_{i=1}^{N}(x_i - x_{media})^2 \sum_{i=1}^{N}(y_i - y_{media})^2}} \tag{0.2}$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(x_i - y_i) \tag{0.3}$$

2.3 Planta Solar Térmica

El funcionamiento de las plantas de concentración solar CSP consiste en concentrar la energía solar en una zona para calentar un fluido y producir vapor que luego accionará una turbina. Existen diferentes tipos de plantas CSP:

- Colectores cilindro-parabólicos (*parabolic-trough collector*, PTC): un conjunto de espejos con forma parabólica concentran la radiación solar en una tubería que contiene un fluido (*heat transfer fluid*, HTF), generalmente agua o aceite sintético. Los colectores se agrupan en lazos que se organizan paralelamente.

- Colectores Fresnel (*linear Fresnel reflector*, LFR): un conjunto de espejos finos aproxima una parábola para concentrar los rayos solares en un receptor.

- Centrales de torre (*solar power tower*, SPT): un campo de reflectores concentra los rayos del sol en un receptor en lo alto de una torre.

- Colectores de plato (*parabolic dish system*, PDS): un colector con forma de plato parabólico concentra la radiación solar en un receptor donde se coloca un motor Stirling.

Componentes de una Planta Solar Térmica

Las plantas solares necesitan estar equipadas con sensores y actuadores que permitan controlar y monitorizar la temperatura, radiación y flujo, entre otras variables. La principal variable en este tipo de plantas es la luz solar, que actúa como una perturbación en el sistema de control debido a que no se puede manipular [16]. La irradiancia es la potencia incidente por unidad de superficie y se mide en W/m^2. La radiación es la integral de la irradiancia, medida en J/m^2. Cuando la luz solar llega a la atmósfera, sólo una parte alcanza directamente la superficie terreste, conocida como radiación solar directa, y el resto se dispersa, se refleja o es absorbida por el aire. La parte de esta radiación que finalmente llega a la tierra de forma indirecta se conoce como radiación difusa. Para medir la radiación se utilizan los piranómetros, que miden la suma de la irradiancia directa y difusa, y pirheliómetros, que miden la componente normal directa de la irradiancia.

Cuando los rayos solares inciden en los colectores, la luz se refleja en un receptor que contiene el HTF [49, 8]. Los fluidos utilizados se clasifican en aire y otros gases, agua o vapor, aceites térmicos, sales fundidas y metales líquidos. Ante la ausencia de luz solar, el HTF se puede almacenar en tanques haciendo frente al problema de intermitencia [49, 103, 16].

Modelo de la Planta

Las plantas de colectores cilindro-parabólicos se componen de lazos de colectores colocados en serie. Los lazos, a su vez, se disponen en paralelo para formar el campo solar. En este trabajo, se utiliza un lazo de colectores para realizar las simulaciones y aplicar las técnicas de control, aunque el sistema puede ser extendido a varios lazos. Cada lazo se divide en varios segmentos de tamaño $\Delta\mathrm{l}$, tal como semuestra en la figura 1.

La dinámica de la planta es gobernada por el HTF, que depende de las condiciones de operación. Existen dos tipos de modelos que se utilizan en la literatura: modelo de parámetros concentrados, que describe la variación de la energía interna mediante una ecuación diferencial sin considerar la distribución espacial de temperaturas ni la transmisión de calor metal-fluido, y el modelo de parámetros distribuidos [21, 34, 62], que contiene derivadas parciales y se utilizará para modelar y controlar la planta en este trabajo. Los balances de energía en cada segmento i entre los instantes $(k-1)\Delta\mathrm{T}$ y $k\Delta\mathrm{T}$ dan lugar a las siguientes ecuaciones, donde los subíndices f, m y a hacen referencia al fluido, el metal y el ambiente:

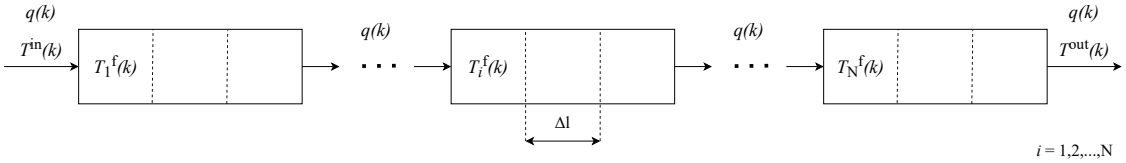

Figura 1 Esquema de un lazo del campo de colectores.

$$T_i^{\mathrm{m}}(k) = T_i^{\mathrm{m}}(k-1) + \frac{\Delta \mathrm{T}}{\rho^{\mathrm{m}} C^{\mathrm{m}} A^{\mathrm{m}}} \Big(\eta_i^{\mathrm{col}} G_i I_i(k) - \pi D^{\mathrm{m}} H_i^{\mathrm{l}}(k-1) \Big(T_i^{\mathrm{m}}(k-1) - T^{\mathrm{a}}(k) \Big) + \\ - \pi D^{\mathrm{f}} H_i^{\mathrm{t}}(k-1) \Big(T_i^{\mathrm{m}}(k-1) - T_i^{\mathrm{1f}}(k-1) \Big) \Big) \tag{0.4}$$

$$T_i^{\mathrm{f}}(k) = T_i^{\mathrm{f}}(k-1) + \frac{\pi D^{\mathrm{f}} H_i^{\mathrm{t}}(k-1) \Delta \mathrm{T}}{\rho_i^{\mathrm{f}}(k-1) C_i^{\mathrm{f}}(k-1) A^{\mathrm{f}}} \Big(T_i^{\mathrm{m}}(k-1) - T_i^{\mathrm{1f}}(k-1) \Big) \tag{0.5}$$

donde $T_i^{\mathrm{1f}}(k)$ es una temperatura auxiliar calculada como:

$$T_i^{\mathrm{1f}}(k) = T_i^{\mathrm{1f}}(k-1) - \frac{q(k) \Delta \mathrm{T}}{A^{\mathrm{f}}} \Big(T_i^{\mathrm{f}}(k) - T_{i-1}^{\mathrm{f}}(k) \Big) \tag{0.6}$$

La variable manipulada en este sistema es el caudal $q(k)$; y la irradiancia $I_i(k)$, la temperatura a la entrada $T^{\mathrm{in}}(k)$ y la temperatura ambiente $T^{\mathrm{a}}(k)$ son perturbaciones que deben ser medidas o estimadas

Principales Estrategias de Control

El objetivo de control es maximizar la potencia eléctrica obtenida en la planta manipulando el caudal, aunque en numerosas ocasiones se pretende realizar un seguimiento de temperatura comandada por un operario o maximizar la temperatura. Esto último, aunque aumenta la eficiencia del ciclo, también aumenta las pérdidas. La ventaja de optimizar la potencia radica en la necesidad de cumplir varios objetivos: mantener la temperatura entre unos límites adaptándose a los cambios en el ciclo solar y maximizar las ganancias. [16] recoge las principales arquitecturas de control de plantas de colectores cilindro-parabólicos:

- Control por prealimentación (más conocido como *feedforward*): Se añade un controlador en serie o en paralelo con el control principal para rechazar perturbaciones.
- PID: Se utiliza en la capa de bajo nivel.
- Control en cascada: Se utiliza un lazo de control interno para compensar perturbaciones y un lazo externo para controlar la entrada al sistema.
- Control adaptativo: Se modifican determinadas características del controlador en tiempo real para hacer frente a variaciones en el sistema.
- Planificación de ganancias (más conocido como *gain scheduling*): Se adapta a cambios en la dinámica del sistema modificando parámetros del controlador respecto a los valores de ciertas variables del proceso.
- Control por modelo interno: Se utiliza un modelo para estimar la salida y el resultado de una perturbación. Después, se añade una señal de realimentación del error entre la salida y el modelo.

- Compensación de tiempo de retardo: Se diseña un controlador que no tiene en cuenta los retardos puros y se realizan estimaciones con y sin retardo. El ejemplo más conocido es el predictor de Smith.
- Control óptimo: Se optimiza una función de coste.
- Control robusto: Se tiene en cuenta la incertidumbre del modelo, considerando errores de modelado y perturbaciones.
- Control no lineal: Se aplica a sistemas con ecuaciones no lineales, teniendo en cuenta las no linealidades.
- Control borroso: Se utilizan variables lingüísticas que representan distintas características, lo que permite controlar datos imprecisos e incompletos. Se realizan acciones de acuerdo a relaciones lógicas entre las variables.
- Control predictivo basado en modelo: Se minimiza una función de coste que tiene en cuenta predicciones de la salida del sistema mediante una estrategia de horizonte deslizante.
- Control neuronal: Se utilizan redes neuronales, normalmente para modelar el sistema, ajustar parámetros de controladores o directamente obtener la señal de control.

3 Formulación del problema

El uso de un controlador MPC en plantas solares permite introducir restricciones y tener en cuenta cómo afectaran las señales de control a la salida en instantes posteriores, pero tiene el problema de un alto coste computacional. Existen varias líneas de investigación para solucionar esta desventaja y en este caso se propone el uso de redes neuronales. Este capítulo pretende describir la estrategia de control implementada, la arquitectura de la red neuronal seleccionada y los datos de entrenamiento empleados.

3.1 Estrategia del control predictivo

El objetivo de control es maximizar la potencia térmica obtenida por la planta satisfaciendo una serie de restricciones y minimizando el esfuerzo de control siguiendo la estrategia implementada en [34] y [62]. Para ello, se minimiza la siguiente función de coste $J(k_c)$ en cada instante $t = k_c \Delta T_c$, donde ΔT_c es el tiempo de muestreo del controlador. Ésta contiene, además, restricciones blandas para limitar la temperatura del fluido. El problema de optimización se resuelve utilizando SQP, que consiste en resolver un subproblema local aproximando la función de coste en un punto. Se han seleccionado cuatro puntos iniciales: el perfil de caudales obtenidos en el instante anterior, el límite inferior, el límite superior y un punto aleatorio.

$$J(k_c) = \left(-W(k_c) + \psi \max \left(\frac{T_N^f(k_c) - T^{f,max}}{T^{f,max}}, \frac{T^{f,min} - T_N^f(k_c)}{T^{f,max}}, 0 \right)^2 + \varepsilon \left(q(k_c) - q(k_c - 1) \right)^2 \right) \tag{0.7}$$

donde $W(k_c)$ es la potencia térmica y ψ y ε son pesos de ponderación que permiten otorgar más o menos importancia a cada término.

La siguiente tabla recoge los parámetros de la planta ACUREX y del controlador, así como las restricciones de control.

3.2 Controlador Neuronal

En esta sección se describe la implementación de la red neuronal que sustituirá al MPC en línea para proporcionar un comportamiento cercano al óptimo. El objetivo es aproximar el caudal óptimo a enviar a

Tabla 1 Parámetros y restricciones.

Símbolo	Descripción	Valor	Unidad
ΔT	Tiempo de muestreo del modelo	0.5	s
ρ^m	Densidad del metal	7800	kg/m^3
C^m	Capacidad calorífica específica del metal	550	J/(kg °C)
D^m	Diámetro externo de la tubería	0.031	m
D^f	Diámetro interno de la tubería	0.0254	m
A^m	Área de sección cruzada del metal	$2.48 \cdot 10^{-4}$	m^2
A^f	Área de sección cruzada del fluido	$7.55 \cdot 10^{-4}$	m^2
ΔT_c	Tiempo de muestreo del controlador	60	s
$\Delta T_{m,c}$	Tiempo de muestreo del modelo usado en el controlador	3	s
Δl_c	Tamaño de los segmentos usados en el controlador	6	m
N_p	Horizonte de predicción	12	-
N_u	Horizonte de control	10	-
ψ	Peso de la función de coste	45	-
ε	Peso de la función de coste	3	-
q^{min}	Mínimo caudal	0.2	l/s
q^{max}	Máximo caudal	1.5	l/s
$T^{f,min}$	Mínima temperatura del fluido	220	°C
$T^{f,max}$	Máxima temperatura del fluido	300	°C

la planta poniéndolo como función del resto de variables que intervienen en su dinámica: el caudal en el instante anterior, temperaturas de entrada, salida, ambiente, fluido y metal e irradiancias en cada segmento del lazo de colectores.

$$q(k) = f\left(q(k-1), T^{in}(k), T^{out}(k), T^a(k), T_i^f(k), T_i^m(k), I_i(k), ..., I_i(k+N_p-1)\right) \tag{0.8}$$

Esto se realiza en dos etapas: en primer lugar, se llevan a cabo simulaciones de distintos días aplicando MPC para obtener un amplio conjunto de datos (en este caso, de 30 días) con los que entrenar la red neuronal *offline*. Después, se implementa la red neuronal y se utiliza para controlar la planta en tiempo real, sustituyendo al controlador MPC.

El conjunto de datos obtenido contiene 168867 y se ha dividido en conjuntos de entrenamiento (70%), validación (15%) y prueba (15%), seleccionados de forma aleatoria. Después, los datos se han escalado en el rango $[-1,1]$, se ha seleccionado un MSE de 10^{-9} como criterio para finalizar el entrenamiento y se han seleccionado funciones de activación tangente hiperbólica en todas las capas excepto la de salida, donde se ha empleado una función lineal. Para analizar la capacidad de interpolación de las redes neuronales, se han implementado redes neuronales de distintos tamaños, eliminando entradas y atendiendo a los siguientes casos:

- Caso 1: Temperaturas e irradiancia cada seis segmentos (410 entradas).
- Caso 2: Irradiancia cada seis segmentos y temperaturas en el centro de cada colector (360 entradas).
- Caso 3: Irradiancia cada seis segmentos, predicciones en los instantes 1,4,8 y 12, y temperaturas en el centro de cada colector (128 entradas).
- Caso 4: Irradiancia cada seis segmentos, predicciones en los instantes 1 y 12, y temperaturas en el centro de cada colector (70 entradas).
- Caso 5: Irradiancia cada seis segmentos, horizonte de predicción de 6, y temperaturas en el centro de cada colector (186 entradas).
- Caso 6: Irradiancia cada seis segmentos, horizonte de predicción de 3 y temperaturas en el centro de cada colector (99 entradas).

- Caso 7: Irradiancia cada seis segmentos, horizonte de predicción de 1 y temperaturas en el centro de cada colector (41 entradas).

- Caso 8: Temperatura e irradiancia en el primer colector, sin temperaturas de entrada y salida, y horizonte de predicción de 1 (5 entradas).

Para evaluar las redes neuronales, se han añadido dos métricas: el incremento de control absoluto acumulado (*accumulated absolute control increment*, AACI), que permite evaluar la suavidad de la señal de control, y la violación de restricciones media (*mean squared constraint violation*, MSCV), para medir el valor medio de la diferencia entre la temperatura de salida y su límite cuando se incumple una restricción.

$$\mathrm{AACI} = \sum_{k} |q(k) - q(k-1)| \tag{0.9}$$

$$\mathrm{MSCV} = \frac{1}{\mathrm{n_s}} \sum_{k\Delta\mathrm{T}=t_1}^{k\Delta\mathrm{T}=t_2} \left(\max \left(T^{\mathrm{f,min}} - T^{\mathrm{out}}(k), T^{\mathrm{out}}(k) - T^{\mathrm{f,max}}, 0 \right)^2 \right) \tag{0.10}$$

donde $\mathrm{n_s}$ es el número de muestras entre t_1 y t_2.

3.3 Datos de entrada

En todas las simulaciones se ha considerado una temperatura ambiente de 25 °C y se han usado diferentes perfiles de irradiancia. Para entrenar la red neuronal, se ha utilizado un conjunto de datos de 30 días y después se han utilizado cinco perfiles para validar los resultados. Dichos perfiles se muestran en esta sección.

4 Resultados de Simulación

En este capítulo se presentan los resultados de simulación obtenidos al aplicar las redes neuronales a la planta. Inicialmente, se realiza una prueba con diferentes redes neuronales en el caso 1 para seleccionar una arquitectura que se utilizará para el resto de casos y con los diferentes perfiles de irradiancia.

4.1 Selección de la Red Neuronal

En primer lugar, se entrenan diferentes redes neuronales variando el número de capas y neuronas. La siguiente tabla muestra los resultados obtenidos para el primer día del conjunto de datos utilizados para entrenar, indicando el número de neuronas de cada capa entre guiones. Así, por ejemplo, una red neuronal con tres capas intermedias que contiene 4 neuronas en la primera, 3 en la segunda y 2 en la tercera se indicaría como 4-3-2.

Tras analizar los resultados, se selecciona la arquitectura de la tercera red neuronal (15 neuronas en la primera capa y 10 en la segunda) debido a la baja violación de restricciones con una alta potencia obtenida. En la figura 2 se muestran los resultados obtenidos al aplicar la tercera red neuronal en el primer día de los datos utilizados para entrenar.

Tabla 2 Resultados de las diferentes redes neuronales probadas con el primer día del conjunto de datos para el caso 1.

Id	Neuronas	R (entr.)	R (valid.)	R (prueba)	Potencia media	AACI	MSCV
MPC	-	-	-	-	65.6607 kW	7.9309 l/s	0.0445
1	15	0.99720	0.99690	0.99651	65.6638 kW	2.8189 l/s	0.4706
2	15-5	0.99793	0.99678	0.99647	65.6515 kW	2.7983 l/s	0.1016
3	**15-10**	**0.99791**	**0.99636**	**0.99679**	**65.6580 kW**	**2.9247 l/s**	**0.0740**
4	20-10	0.99778	0.99664	0.99621	65.6508 kW	2.9479 l/s	0.1760
5	15-10-5	0.99771	0.99699	0.99686	65.6557 kW	2.8435 l/s	0.5388
6	15-10-10	0.99739	0.99685	0.99685	65.6619 kW	2.9479 l/s	0.5182

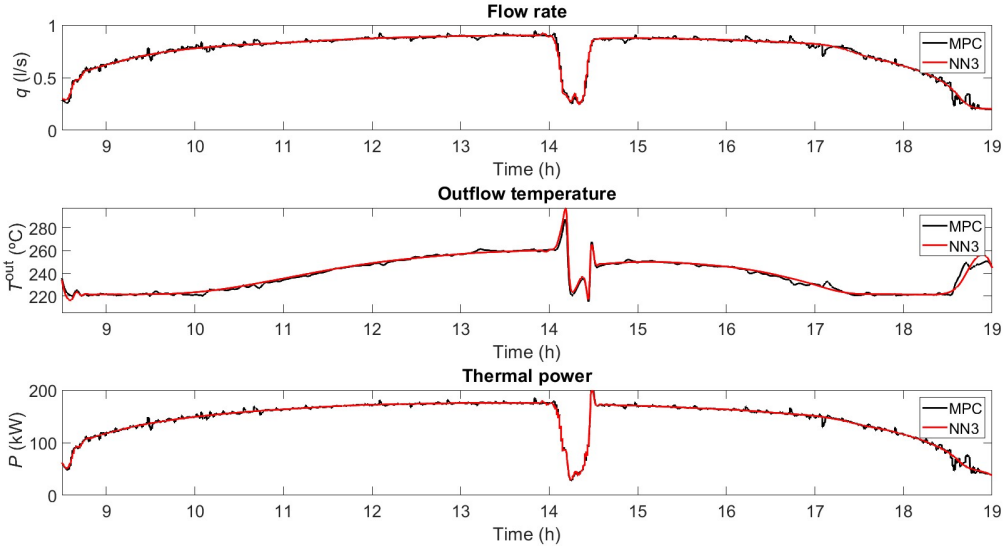

Figura 2 Evolución del caudal, la temperatura de salida y la potencia térmica obtenidos con el primer día del conjunto de datos para el caso 1.

La ventaja más importante de las redes neuronales es su velocidad de computación. En la figura 3 se muestra el tiempo necesitado por cada controlador para obtener la señal de control. El tiempo medio utilizado por el controlador MPC es 2.3929 s con una desviación típica de 1.1418 s, mientras que el tiempo medio utilizado por la ANN es 0.07 s con una desviación típica de 0.0087 s.

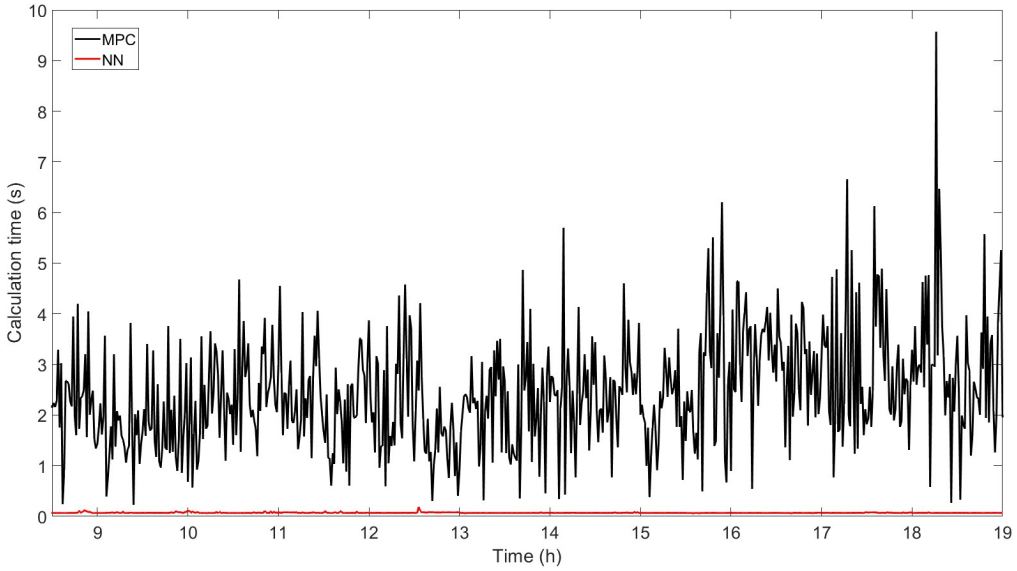

Figura 3 Tiempos de cálculo del MPC y la ANN con el primer día del conjunto de datos para el caso 1.

4.2 Diferente Número de Entradas

Tras seleccionar la arquitectura, se entrenan nuevas redes neuronales para el resto de los casos, modificando el tamaño de la capa de entrada. La tabla 3 recoge los resultados obtenidos para los casos 2 a 8 con el primer día de los datos utilizados para entrenar.

Tabla 3 Resultados de los controladores utilizados para los casos 2 a 8 con el primer día del conjunto de datos para el caso 1.

Caso	R (entr.)	R (calid.)	R (prueba)	Potencia media	AACI	T^{out} MSCV
MPC	-	-	-	65.6607 kW	7.9309 l/s	0.0445
1	0.99791	0.99636	0.99679	65.6580 kW	2.9247 l/s	0.0740
2	0.99760	0.99719	0.99671	65.6581 kW	2.9096 l/s	0.1561
3	0.99768	0.99708	0.99704	65.6535 kW	2.9483 l/s	0.2810
4	0.99748	0.99647	0.99692	65.6629 kW	2.7363 l/s	0.7317
5	0.99772	0.99668	0.99677	65.6548 kW	2.7612 l/s	0.2239
6	0.99694	0.99663	0.99670	65.6599 kW	2.8096 l/s	0.0767
7	0.99686	0.99575	0.99649	65.6707 kW	2.7097 l/s	1.3840
8	0.99547	0.99542	0.99426	65.5358 kW	2.7726 l/s	26.0222

4.3 Validación de Resultados

En este capítulo se muestran los resultados obtenidos con los cinco perfiles de irradiancia utilizados para validar los controladores, cada uno con diferentes nubes.

Primer Perfil

El primer perfil se corresponde con un día sin nubes. Los resultados obtenidos para todos los casos se recogen en la tabla 4 y la figura 4 compara los resultados del MPC y los casos 1 y 7. Los resultados son buenos en todos los casos, excepto el 8.

Tabla 4 Resultados de los controladores con el primer perfil de validación.

Neuronas	Potencia media	AACI	MSCV
MPC	64.2745 kW	7.1727 l/s	0.2684
Caso 1	64.2784 kW	1.3453 l/s	0.3114
Caso 2	64.2819 kW	1.3472 l/s	0.2694
Caso 3	64.2806 kW	1.3492 l/s	0.3180
Caso 4	64.2796 kW	1.3425 l/s	0.2996
Caso 5	64.2840 kW	1.3602 l/s	0.3392
Caso 6	64.2855 kW	1.3470 l/s	0.4184
Caso 7	64.2876 kW	1.3440 l/s	0.8203
Caso 8	64.1954 kW	2.2345 l/s	13.1414

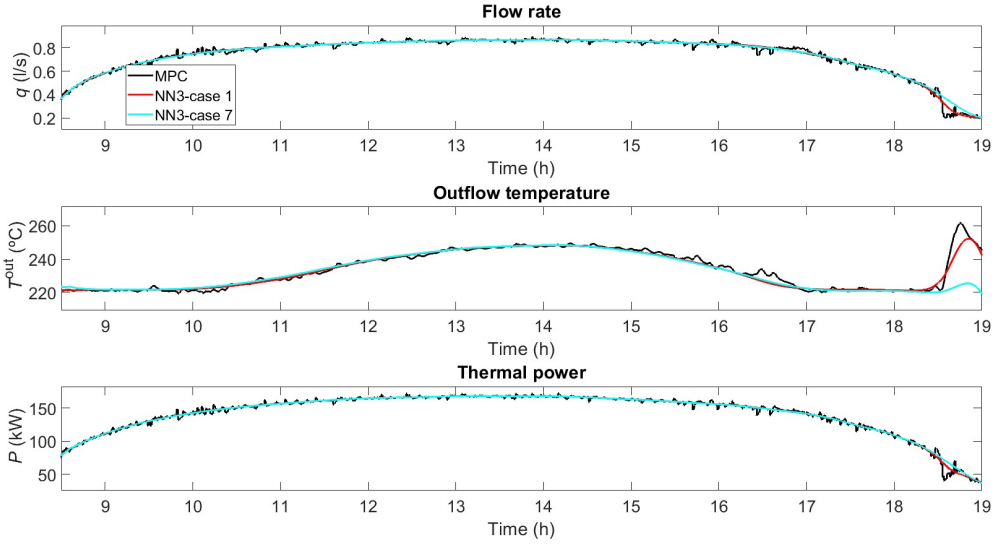

Figura 4 Evolución del caudal, la temperatura de salida y la potencia térmica obtenidos con el primer conjunto de validación para los casos 1 y 7.

Segundo Perfil

El segundo perfil tiene dos nubes de dimensiones moderadas. Los resultados se muestran en la tabla 5 y la figura 5. En este caso no se muestra el caso 8 por desestabilizar la planta. El caso 7 tampoco es aceptable, ya que proporciona temperaturas demasiado altas.

Tabla 5 Resultados de los controladores con el segundo perfil de validación.

Neuronas	Potencia media	AACI	MSCV
MPC	66.0270 kW	8.0466 l/s	$3.1234 \cdot 10^{-5}$
Caso 1	66.0303 kW	3.5807 l/s	0.5395
Caso 2	66.0271 kW	3.5370 l/s	0.1700
Caso 3	66.0248 kW	3.2665 l/s	0.2093
Caso 4	66.0292 kW	3.2773 l/s	0.2071
Caso 5	66.0265 kW	3.1172 l/s	0.0200
Caso 6	66.0270 kW	3.0346 l/s	0.8616
Caso 7	65.9811 kW	3.5878 l/s	4.0104
Caso 8	58.8387 kW	2.8028 l/s	$1.0843 \cdot 10^{4}$

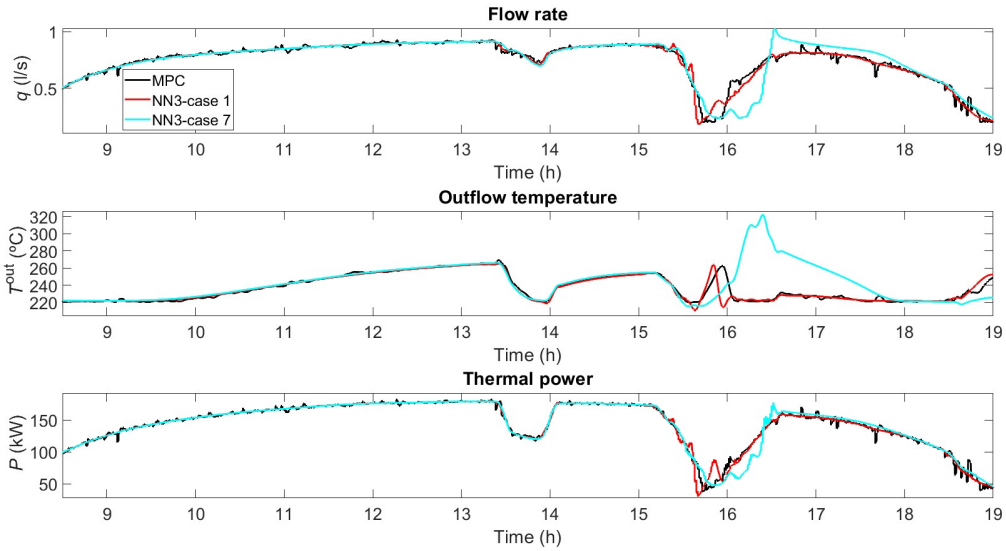

Figura 5 Evolución del caudal, la temperatura de salida y la potencia térmica obtenidos con el segundo conjunto de validación para los casos 1 y 7.

Tercer Perfil

El tercer perfil de irradiancia tiene nubes la mayor parte del tiempo. Los resultados se muestran en la tabla 6 y la figura 6.

Tabla 6 Resultados de los controladores con el tercer perfil de validación.

Neuronas	Potencia media	AACI	MSCV
MPC	61.8890 kW	8.9987 l/s	$3.3044 \cdot 10^{-5}$
Caso 1	61.8868 kW	1.9082 l/s	$4.4934 \cdot 10^{-4}$
Caso 2	61.8876 kW	1.8002 l/s	0.0047
Caso 3	61.8875 kW	1.8246 l/s	0
Caso 4	61.8892 kW	1.8058 l/s	0.0093
Caso 5	61.8912 kW	1.8234 l/s	0.0022
Caso 6	61.8893 kW	1.7970 l/s	0.0062
Caso 7	61.9003 kW	1.7914 l/s	$3.0552 \cdot 10^{-4}$
Caso 8	61.8239 kW	2.8947 l/s	70.9767

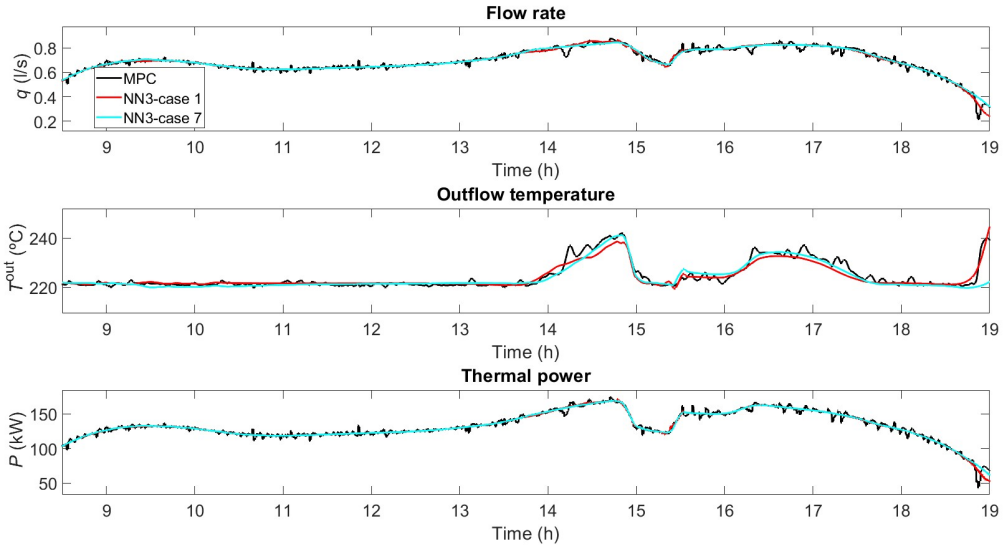

Figura 6 Evolución del caudal, la temperatura de salida y la potencia térmica obtenidos con el tercer conjunto de validación para los casos 1 y 7.

Cuarto Perfil

El cuarto perfil de irradiancia contiene una gran nube y los resultados se muestran en la tabla 7 y la figura 7. Nuevamente, el controlador del caso 8 desestabiliza la planta.

Tabla 7 Resultados de los controladores con el cuarto perfil de validación.

Neuronas	Potencia media	AACI	MSCV
MPC	58.0140 kW	9.1523 l/s	$4.9748 \cdot 10^{-5}$
Caso 1	58.0123 kW	3.4988 l/s	0.0256
Caso 2	58.0141 kW	3.2104 l/s	0.0439
Caso 3	58.0122 kW	3.0447 l/s	0.0658
Caso 4	58.0150 kW	3.0711 l/s	0.3500
Caso 5	58.0158 kW	2.9126 l/s	0.3882
Caso 6	58.0149 kW	3.2123 l/s	0.2948
Caso 7	58.0222 kW	3.1235 l/s	0.4339

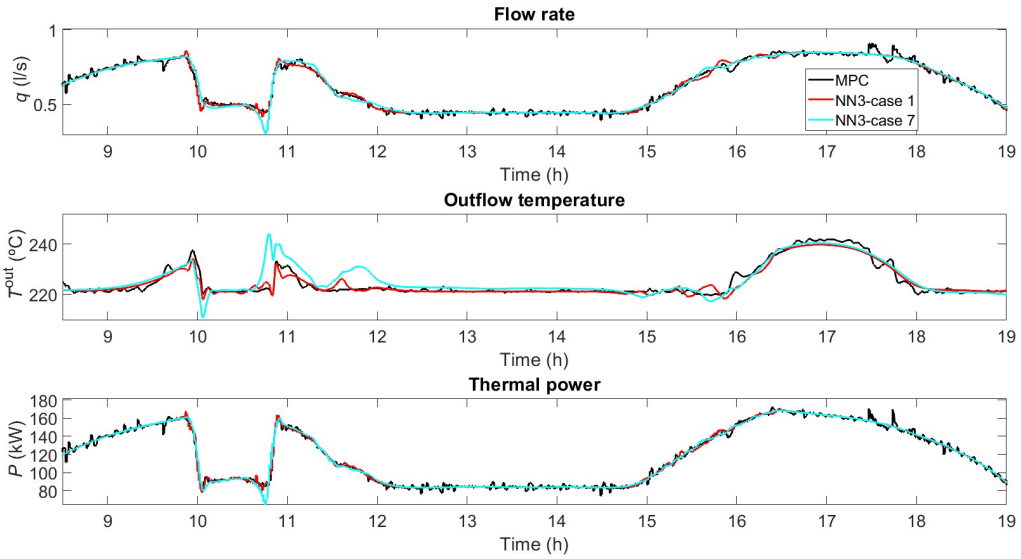

Figura 7 Evolución del caudal, la temperatura de salida y la potencia térmica obtenidos con el cuarto conjunto de validación para los casos 1 y 7.

Quinto Perfil

Finalmente, el quinto perfil de irradiancia cuenta con nubes de diferentes tipos. Los resultados se muestran en la tabla 8 y la figura 8.

Table 8 Resultados de los controladores con el quinto perfil de validación.

Neuronas	Potencia media	AACI	MSCV
MPC	64.0417 kW	9.9777 l/s	0.0033
Caso 1	64.0517 kW	3.8315 l/s	0.1490
Caso 2	64.0504 kW	3.5275 l/s	0.3651
Caso 3	64.0481 kW	3.2928 l/s	0.5636
Caso 4	64.0515 kW	3.0231 l/s	1.6368
Caso 5	64.0474 kW	3.4924 l/s	0.0393
Caso 6	64.0538 kW	3.2521 l/s	0.1317
Caso 7	64.0615 kW	2.9464 l/s	3.2281

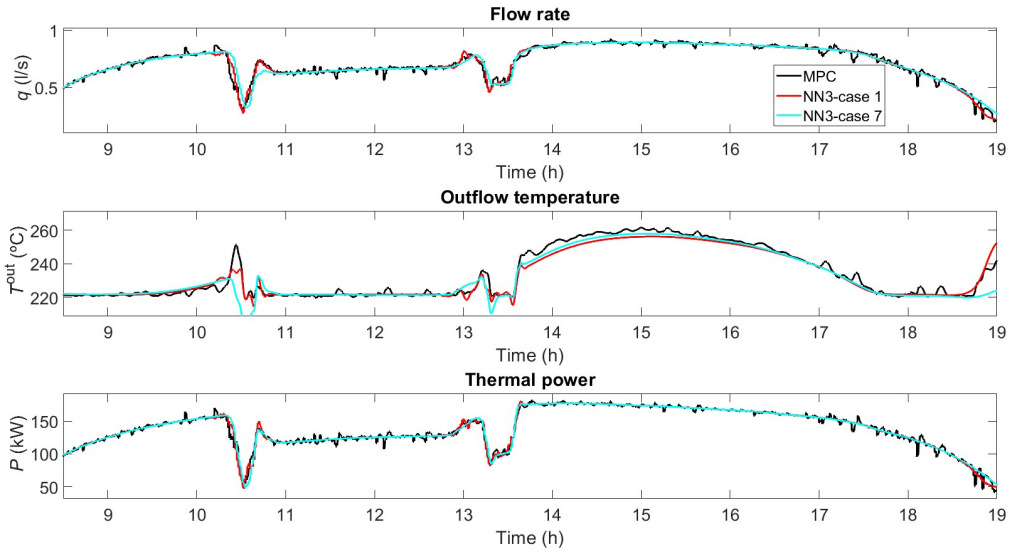

Figura 8 Evolución del caudal, la temperatura de salida y la potencia térmica obtenidos con el quinto conjunto de validación para los casos 1 y 7.

Análisis de Resultados

Esta sección proporciona un análisis de los resultados obtenidos en las secciones previas, remarcando el buen comportamiento de los controladores. Aunque el caso 1 es el que mejores resultados proporciona, los casos 2, 4 y 6 proporcionan valores de potencias y errores del mismo orden utilizando menos entradas, lo que disminuye el tiempo de entrenamiento y las necesidades computacionales.

5 Conclusiones, Contribuciones y Desarrollo Futuro

En este trabajo se han presentado las ventajas y desventajas de controlar una planta de colectores cilindro-parabólicos con MPC, muy utilizado en la literatura. Por una parte, permite maximizar la potencia térmica obtenida por la planta satisfaciendo una serie de restricciones y teniendo en cuenta el efecto de las posibles estrategias de control y futuros valores de irradiancia solar. Por otra parte, tiene un alto coste computacional debido a la necesidad de resolver un problema de optimización en cada instante de muestreo del controlador.

Para resolver este problema, este trabajo propone el uso de redes neuronales entrenadas *offline* a partir de los resultados de un controlador predictivo. El objetivo es utilizar las redes neuronales para sustituir al MPC, aprovechando que son mucho más rápidas. Las conclusiones generales son las siguientes:

- Las redes neuronales pueden proporcionar una potencia térmica similar al MPC.
- Es posible implementar redes neuronales con muy bajas violaciones en las restricciones de la temperatura de salida.
- El tiempo de computación es mucho más bajo con las redes neuronales que con el MPC.
- Las redes neuronales pueden aprender del MPC sin necesidad de utilizar un gran número de sensores y predicciones.
- El esfuerzo de control entre dos instantes consecutivos es mucho más bajo utilizando redes neuronales. Aunque esto podría conseguirse con el MPC, las redes neuronales evitan el esfuerzo asociado al reajuste de parámetros del controlador predictivo.

Las redes neuronales correspondientes a los casos 1 a 6 han proporcionado buenos resultados para todos los perfiles utilizados para validar, demostrando que es posible reducir el número de entradas de 410 a 70. Además, en días con nubes moderadas, la red neuronal del caso 7 también proporciona muy buenos resultados, necesitando sólo 41 entradas.

Los controladores se han aplicado en un lazo de colectores. En el caso del MPC, el tiempo de computación en cada instante se encuentra alrededor de los 3 segundos en la mayor parte de la simulación, alcanzando 7 y 9 segundos en algunas ocasiones. En el caso de ACUREX no sería posible obtener la solución óptima en tiempo real, ya que cuenta con 10 lazos. En cambio, los tiempos obtenidos con redes neuronales rondan los 0.1 segundos, lo que supone una reducción del tiempo de un 97%, permitiendo su implementación en tiempo real para plantas reales. Las plantas comerciales superan los 100 lazos, poniendo aún más en evidencia la necesidad de utilizar redes neuronales.

Un problema que surge al aplicar las redes neuronales es que las restricciones no se tienen en cuenta de manera directa, sino que las considera el controlador MPC del que están aprendiendo. Esto puede llevar a que la violación de restricciones sea alta en determinadas ocasiones, lo que ocurre en las redes neuronales con pocas entradas. Sin embargo, la mayoría de las redes neuronales responden bien a las perturbaciones sin sobrepasar en exceso los límites de temperatura. Aun así, estas violaciones se podrían reducir imponiendo restricciones más duras y rangos más pequeños en el controlador MPC utilizado para entrenar.

La investigación desarrollada en el presente Trabajo de Fin de Máster ha resultado en la publicación de un artículo en la revista *Renewable Energy*, del primer cuartil en las categorías de *Green & sustainable science & technology* y *Energy and fuels* [83].

Futuras líneas de desarrollo serán desarrollar el método a otros niveles:

- Aplicar estos controladores a plantas con mayor número de lazos.
- Combinar redes neuronales con otras estrategias como control distribuido o coalicional.
- Utilizar otros tipos de redes neuronales con, por ejemplo, aprendizaje no supervisado.

Abstract

Nowadays, there is an increasing interest in using renewable energy sources, including solar energy. Parabolic trough plants are a type of solar thermal power plant in which solar radiation is reflected onto tubes with parabolic mirrors. Inside these tubes circulates a fluid, usually oil or water, which is heated to generate steam and turn a turbine to produce electricity.

One of the most widely used methods to control these plants is model predictive control (MPC), which obtains the optimal control signals to send to the plant based on the use of a model. This method makes it possible to predict its future state according to the chosen control strategy over a time horizon.

The MPC has the disadvantage of a significant computational cost associated with resolving an optimization problem at each sampling time. This makes it challenging to implement in commercial and large plants, so currently, one of the main challenges is to reduce these computational times, either technologically or by using suboptimal techniques that simplify the problem.

This project proposes the use of neural networks that learn offline from the output provided by a predictive controller to then approximate it. Different neural networks have been trained using a 30-day simulation dataset and modifying the number of irradiance and temperature inputs. The results show that the neural networks can provide practically the same power as the MPC with smoother variations of the output and very low violations of the constraints, even when decreasing the number of inputs. The work has been published in *Renewable Energy*, a first quartile journal in Green & sustainable science & technology and Energy and fuels [83].

Short Contents

Contents

Notation

$x(t)$	Value of x at time t
$\boldsymbol{x}$	Vector
x	Scalar
x_i	i-th component of $\boldsymbol{x}$
$x(t+k \mid t)$	Value of x at time $t+k$ calculated at time t
$\hat{x}$	Predicted value of x
$\sum_{i=1}^{N}$	Sum from i equals 1 to N
$=$	Equal to
$<$	Less than
$>$	Greater than
$:=$	Assignment
Δx	Increment of x
$\sqrt{x}$	Square root of x
$f(x)$	Value of f evaluated in x
$\frac{\partial x}{\partial y}$	Partial derivative of x with respect to y
$\min_x f$	Minimum value of f varying x
$\max_x f$	Maximum value of f varying x
$\forall$	For all
s.t.	Subject to
$x_i^m(k)$	For physical variables, value of x at time k and segment i referring to metal
$x_i^f(k)$	For physical variables, value of x at time k and segment i referring to fluid
$x^a(k)$	For physical variables, ambient value of x at time k
$x^{\max}$	Maximum value of x
$x^{\min}$	Minimum value of x
J	Cost function

Parameters and Variables Description

ρ	Density (kg/m^3)
C	Specific heat capacity (J/kg $^\circ$C)
$\tilde{T}_{\text{out}}$	Outlet temperature after steam generator ($^\circ$C)
T	Temperature ($^\circ$C)
T^{1f}	Auxiliary temperature ($^\circ$C)
T_{in}	Inlet temperature ($^\circ$C)
T_{out}	Outlet temperature ($^\circ$C)

T_{mean}	Mean temperature between inlet and outlet (°C)
t	Time (s)
I	Direct solar irradiance (W/m^2)
K_{opt}	Optical efficiency
$n_o(t)$	Geometric efficiency
G	Collector aperture (m)
H^{l}	Thermal loss coefficient (W/m^2 °C)
H^{t}	Metal-fluid heat transmission coefficient (m^2 °C)
L	Length of the inner circumference of the pipe (m)
η^{col}	Collector efficiency
q	Flow rate (l/s)
L_{loop}	Loop length (m)
A^{f}	Transversal area of the interior pipe (m^2)
A^{m}	Cross-sectional area of the pipe (m^2)
D^{m}	Outside diameter of the pipe (m)
D^{f}	Inside diameter of the pipe (m)
N_{c}	Control horizon
$\Delta\text{T}_{\text{c}}$	Controller sample time (s)
ΔT	Sample time (s)
Δl	Segment length (m)
N	Number of segments
N_u	Control horizon
N_p	Prediction horizon
$\boldsymbol{w}$	Weight vector

1 Introduction

The growing impact of human activities on the environment is causing global warming, whose effects are disastrous for life on Earth. For this reason, there is interest in renewable energy sources. After the big oil crisis, there was an impulse to use renewable energy that ceased after the recovery of the economy and is on the rise again [13]. Specifically, the percentage of the Spanish energy production corresponding to renewables sources was 44.0% in 2020, and, in the first quarter of 2021, they have surpassed the non-renewable sources for the first time with a 52.2%, as shown in figure 1.1.

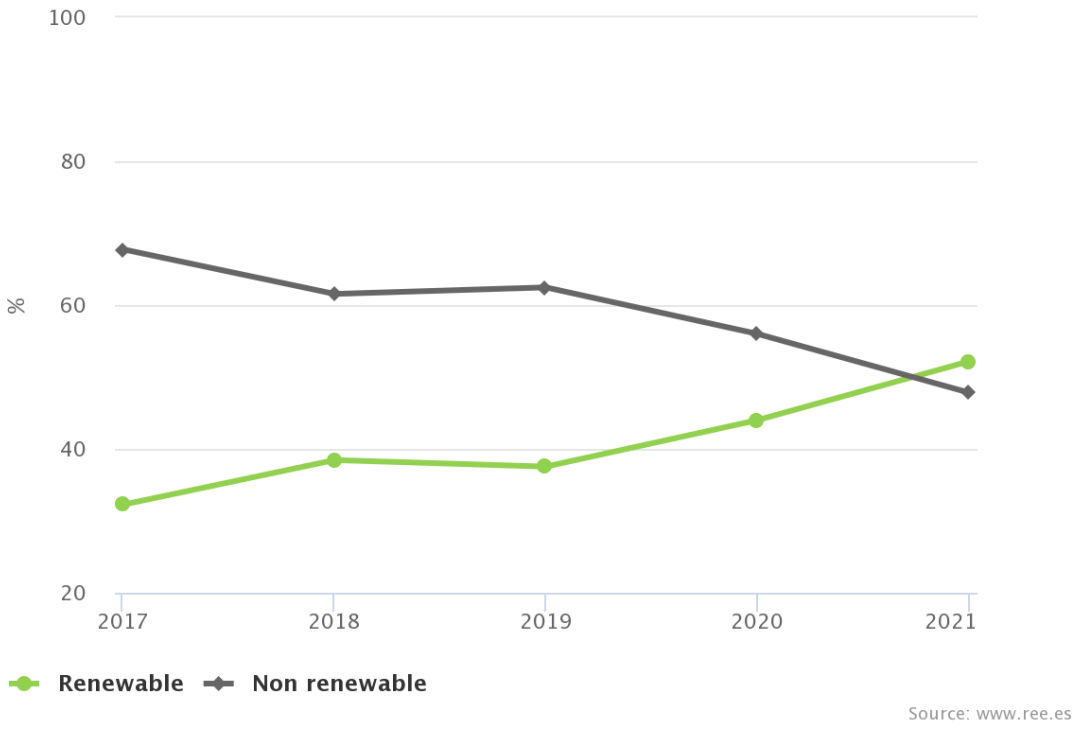

Figure 1.1 Distribution of the Spanish energy generation between 2017 and 2021. Extracted from [3].

The most ancient and abundant of these renewable sources is solar energy [30], which is harnessed and transformed into electric or thermal energy. There are different technologies of solar power generation that can be divided into two main groups:

- Photovoltaic cells (PV), which converts the sunlight into electricity using semiconducting materials that present the photovoltaic effect.

- Concentrating solar power (CSP), which uses mirrors to concentrate the sunlight into receivers filled with oil or water producing steam that drives turbine generators.

An advantage of the CSP technology over PV is the possibility of including thermal energy storage (TES). There are mainly four types within the CSP technology: parabolic-trough collectors, Fresnel collectors, tower plants, and dish collectors [103]. This work focuses on the control of parabolic-trough collectors, consisting of a set of mirrors with a cylindrical-parabolic shape and a tube on the focal axis through which the oil is circulated.

1.1 Objectives

Among the many different control algorithms used in the literature for these types of collectors, model predictive control (MPC) [17] is one of the most important ones due to its many advantages. Its ability to deal with nonlinear behaviors and constraints and the possibility of taking into account future outputs or not necessarily using a setpoint makes it widely used in the literature. On the other hand, an MPC controller needs to solve an optimization problem every few seconds or minutes, making it complicated to use in large plants in real time. This work proposes the use of an artificial neural network (ANN) to overcome this drawback.

The use of neural networks in the literature is mainly based on two approaches: using them to model the system or to calculate the control signal. To the best of our knowledge, neural networks have not yet been used in the literature to obtain the control signal in solar thermal plants. The main objective of this work is the application of neural networks to directly approximate the output of a nonlinear MPC controller in parabolic-trough collector fields. With this approach, the advantages of MPC are exploited by obtaining sub-optimal outputs and constraint compliance most of the time while achieving much shorter implementation times.

The following steps compose the methodology applied in this work:

- Use an MPC controller on the plant to generate a dataset.
- Train several artificial neural networks offline to learn the MPC controller outputs.
- Select a neural network controller and test it by simulation on the plant.

This work has resulted in the development of a journal paper [83]. This master thesis explains the methodology developed in the paper and shows different validation experiments.

1.2 Context

1.2.1 OCONTSOLAR

This work is part of the OCONTSOLAR (Optimal Control of Thermal Solar Energy Systems) project, which has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 789051) [2], granted to Dr. Eduardo Fernández Camacho, head of the group *Automatica y Robotica Industrial* of the University of Seville. The main objectives of the H2020 OCONTSOLAR are to control and integrate mobile sensor networks into solar plant control systems, use sensors mounted on UGVs (unmanned ground vehicles) and UAVs (unmanned aerial vehicles) to implement spatially distributed solar irradiance estimation methods and apply new MPC algorithms. These challenges are summarised in the following points:

- "Methods to control mobile sensor fleets and integrate them as an essential part of the overall control systems".

- "Spatially distributed solar irradiance estimation methods using a variable fleet of sensors mounted on drones and UGVs".

- "New model predictive control (MPC) algorithms that use mobile solar sensor estimations and predictions to yield safer and more efficient operation of the plants allowing the effective integration of solar energy in systems delivering energy to grids or other systems while satisfying production commitments".

Figure 1.2 OCONTSOLAR (Optimal Control of Thermal Solar Energy Systems) logo.

1.2.2 Solar Plant

This project has been carried out on a model of the ACUREX plant [100], which was located at the Plataforma Solar de Almería (PSA), a research center belonging to the Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas (CIEMAT). The modelled field is composed of East-West aligned single axis parabolic-trough collectors arranged in 10 loops of 174 m in parallel. Each loop is composed of 12 collectors connected in series. The length of the active part (the one that receives solar irradiance) is 144 m, and the length of the passive part is 30 m. The heat transfer fluid (HTF) is Therminol VP-1, which absorbs the thermal energy concentrated into its pipe to produce steam. The whole system was an experimental prototype of a commercial solar power plant scaled to 1 MW and was formed by the parabolic-trough collector field, a thermocline oil storage tank, an oil pump and an on–off valve.

1.2.3 Developing

MATLAB® (*MATrix LABoratory*) is a high-level development environment owned by MathWorks® [1], which allows many multidimensional operations to be performed easily. It has a large number of functions, including optimization functions, and has a convenient and comfortable visualization environment, so it is commonly used to simulate the behavior of different physical events before testing with other faster running languages. The development of this project has been carried out entirely in MATLAB®.

1.3 Structure of the Document

This document organizes as follows. After this first chapter providing an overview of the project, its contribution and the context in which it is framed, the solar plant is presented in chapter two. The types and principal components of solar thermal plants are described and the model of the parabolic-trough collector field is presented. Then, the chapter presents the state of the art and gathers a brief description of the most used algorithms to control these types of plants, with more detail on model predictive control and artificial neural networks. These methods are described and some applications to control systems are presented. The problem formulation is given in chapter three, describing the MPC strategy and the neural network implementation. Some simulation results are given in chapter four and, finally, some conclusions and future work are detailed in the fifth chapter.

2 State of the Art

This chapter aims to give an overview of solar thermal power plants, the equations used to describe the PTC field and the principal methods to control them used in the literature and commercial plants. The main control architectures will be briefly described, highlighting model predictive control and artificial neural networks, as they will be the applied control architectures in this work and MPC is one of the most used techniques in the literature.

2.1 Model Predictive Control

As aforementioned, many of the recent works on controlling PTCs are based on MPC. It will be described how the control methodology has evolved in recent years, highlighting the strengths and weaknesses that remain to be polished, opening the way to a path of research. In particular, one of the points being worked on is the reduction of computation times to allow the implementation of these controllers on large commercial plants in real-time.

Model predictive control [17], also known as receding horizon control, is a technique based on the optimization of a cost function –also called objective function–, calculated using a dynamic model of the process. This function represents the cost of the control action, given a specific criterium. By using a model of the system, a prediction of the state in the following timesteps is made and used in the cost function to obtain the optimal control signals. Figure 2.1 gives the general structure of the technique: the model is obtained to make the prediction that will be used to solve the optimization problem.

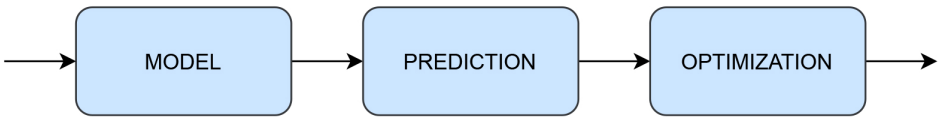

Figure 2.1 Structure of model predictive control.

MPC has been widely used in the literature for controlling parabolic-trough collector fields due to its many advantages. It can solve an optimization problem dealing with constraints and non-linear behaviors of the plant and uses a receding horizon to take account of feature outputs. With this strategy, the output is not the best immediate response but the one that considers how the system will be in the future depending on the actions taken at any given moment. Its characteristics make it able to cope with disturbances and dead times.

A model predictive controller is composed of an optimizer that gives the optimal control signals u by generally using the errors between the predicted future outputs and a reference trajectory. One other advantage

of MPC is that there is no need for a reference, so any control objective can be used, as long as it can be included in the cost function. The predictions $\hat{y}$ are calculated along a prediction horizon and are obtained from past inputs and outputs and from future outputs (figure 2.2). This is done by obtaining a temporal sequence of length N_p and sliding it at every timestep. With this strategy, it is possible to correct the outputs and the difference between the desired and the actual behavior of the plant.

On the other hand, the future outputs are calculated by the optimizer along the control horizon, of length N_u. At each timestep, this horizon is moved, and only its first control signal is used. The rest control signals of the horizon are discarded.

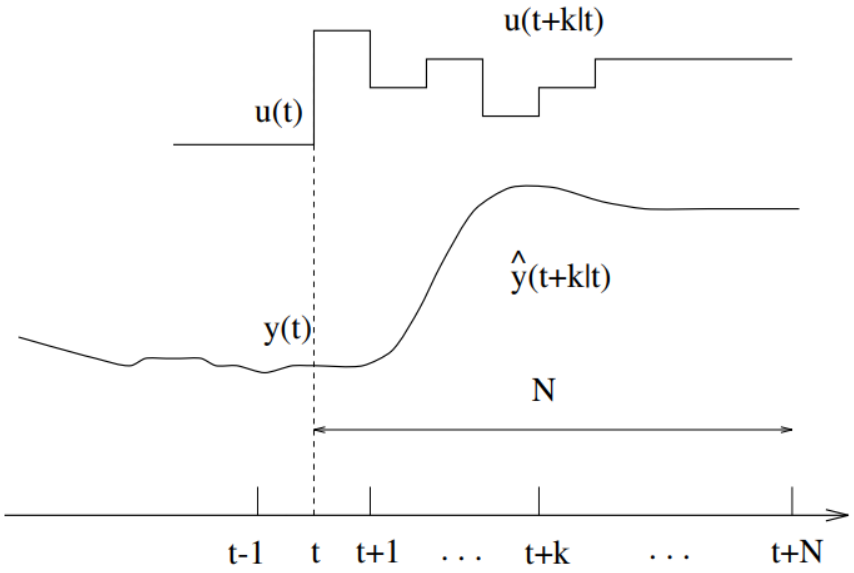

Figure 2.2 Model predictive control strategy. Extracted from [17].

Taking into account the predictions $\hat{y}(t+k \mid t)$, obtained at time t and with $k = 1,...,N_p$, and the future outputs $u(t+k \mid t)$, obtained at time t and with $k = 0,...,N_p - 1$, the model of the system is as follows:

$$\hat{y}(t+k \mid t) = f\left(y(t), u(t \mid t), u(t+1 \mid t), ..., u(t+N_u \mid t)\right), \; N_u < N_p \tag{2.1}$$

The cost function is usually in the form of equation 2.2, where the objective is to minimize the quadratic error between the predicted output and the reference and minimize the control effort. Depending on the application, the control signal may appear directly in the cost function or in the form of an increment. A weight ($\delta(j)$ and $\lambda(j)$) is added to each component of the cost function to adjust their penalization. Higher values of these weights relate to elements that are more important to minimize.

$$\begin{aligned} J(\hat{y}(t+1 \mid t), ..., \hat{y}(t+N \mid t), \Delta u(t), \Delta u(t+1), ..., \Delta u(t+N_u-1)) = \\ = \sum_{j=1}^{N_p} \delta(j)[\hat{y}(t+j \mid t) - w(t+j)]^2 + \sum_{j=1}^{N_u} \lambda(j)[\Delta u(t+j-1)]^2 \end{aligned} \tag{2.2}$$

The constraints included in the optimization problem are usually bounds for the system output, the control signal and the control increment. It is also included the model of the system, as the control signals must ensure that the future outputs obey the physics of the system.

2.1.1 Nonlinear Model Predictive Control

Linear MPC is characterized by using a linear model of the system, output and control signal constraints described by polyhedra and cost functions that are generally quadratic, or sometimes 1-norm or ∞-norm. Problems with these properties have a relatively simple solution. In the case of no constraints, the solution is implicit. If there are constraints, it can be obtained by solving quadratic or linear optimization problems. These are some of the main solving methods used:

- Linear programming (LP): characterized by a linear cost function with linear constraints [60].
- Mixed integer linear programming (MILP): similar to LP, but using discontinuous variables [74].
- Quadratic programming (QP): characterized by a quadratic cost function with linear constraints [7].
- Mixed integer quadratic programming (MIQP): similar to QP, but using discontinuous variables [65].
- Nonlinear programming (NLP): characterized by nonlinear restrictions [60].
- Mixed integer nonlinear programming (MINLP): similar to NLP, but using discontinuous variables [46].
- Stochastic programming (SP): the problem is modelled by stochastic variables [86].
- Dynamic programming (DP): it solves a complex problem by dividing it into subproblems [10].

When any of the premises for applying linear MPC are not met, non-linear model predictive control (NMPC) must be applied. Most of the processes are nonlinear. They may have startups and shutdown and important nonlinearities or be governed by operations that are never in steady-state, such as solar plants, where the operation is carried out in transient mode and many nonlinearities exist. Distributed solar collector fields are governed by highly nonlinear dynamics, so it is very convenient to use nonlinear models [19] and control strategies such as NMPC.

What makes linear MPC advantageous is that it is generally solved with the well-known QP problem, which is fast to solve. Nonetheless, when applied to processes with high nonlinearities and very changing dynamics, the results are not acceptable and NMPC has to be used. The drawback is that optimization problems in linear systems are not necessarily convex and it is much more difficult to solve them because there is no guarantee of reaching the optimal solution. Moreover, it is difficult to ensure convergence in suitable times for the system [16] and the computational times are exceedingly high. Furthermore, on many occasions, time is so important that it is not possible to reach the optimal solution without violating time constraints.

The problem is often solved with sequential quadratic programming (SQP) techniques, which are extensions of Newton-type methods [17]. They are an iterative method that solves an approximation of the nonlinear problem at each time by replacing the cost function and the constraints with linear approximations. Using this technique should help to deal with ill-conditioning and extreme nonlinearities. This method still has problems in real-time operation, since if the time constraints are shorter than the time needed to solve the problem, the control signal obtained in the last iteration will be sent, which will be non-optimal and may violate the constraints of the original problem. Furthermore, since the problem is non-convex, it must be taken into account that local minima can be reached, so the algorithm must be run several times using different starting points. Therefore, the computational cost is too high and makes it unimplementable in many real-time applications.

Different algorithms seek to improve these problems to the detriment of optimality and constraint respect. In this work, neural networks are used to learn from the solution provided by a NMPC calculated offline and using high computational times. Then, these neural networks will be able to approximate the solution obtained in much shorter times.

2.2 Overview of Recent Work

As mentioned, different control architectures have been used for controlling parabolic trough collectors and, more specifically, applied to ACUREX, an experimental plant that was located at the Plataforma Solar de Almería. Many of these control techniques are based on the use of MPC. This section aims to give an overview of some of the recent works present in the literature.

In [36], an adaptive state-space MPC is applied to ACUREX, using a unscented Kalman filter (UKF) for estimating radiation and temperatures. The control objective is to maintain the outlet temperature close to a set point. They use MPC that computes the forced response (relative to the variations in the control signal) with a linear model and the free response (the one that would be obtained if the control signal remains constant) with a simplification of the distributed parameter model. [37] apply an observer-based MPC with a parallel FF to ACUREX, adding a Luemberger observer to estimate temperatures, and compare the results with a gain scheduling generalized predictive control (GS-GPC) and PID. In [89] the different loops are taken into account. They add valves to each loop to homogenize temperatures and overcome the problems derived from dirt collectors producing different loop efficiencies. They apply a GS-GPC with a series FF and conduct temperature reference tracking. Then, [91] propose a strategy to manipulate the flow on the loops and reduce the imbalance of the temperature of the loops. For this purpose, they open the valves of the most efficient loops. In [88, 90], an MPC controller is implemented for controlling the defocus angle of the fourth and third collectors. [78] implements an adaptive control method by solving two consecutive quadratic programming (QP) problems: the first one establishes a momentaneous model, and the second one is dedicated to computing the MPC optimizer.

Non-linear control is used in [55], where a non-linear continuous-time MPC is applied to ACUREX for reference tracking without the use of FF controller. They avoid online optimization, shortening the computation times, at the expense of only solving a tracking error minimization problem. On the other hand, [67] apply MPC to ACUREX maximizing the electrical power obtained and compare the results when controlling the total oil flow and the flow of each loop –using the valves added by [89]. In this case, the problem of computational cost arises since 24 variables need to be controlled. This makes it not implementable in real-time. Along these lines, the work in [34] compare the results obtained applying different MPCs with power maximization, temperature maximization, temperature minimization and no-valves. They also propose a logic-based distributed approach for controlling the valves of each loop, reducing the computation times needed with the centralized control. Another approach to reducing the computational cost is applying coalitional MPC, as proposed in [62], where coalitions of two loops are made to solve the optimization problem.

Different approaches take into account other types of criteria based on market and economic gains. [18] apply non-linear MPC to ACUREX to optimize the temperature set-point depending on the environmental conditions based on economic profit. A three-layer hierarchical structure is applied to calculate the optimal solar field temperature reducing the operating costs. In [20], the aim is to maximize the return of investment (ROI). [94] proposes an MPC strategy that, combined with short-term direct normal irradiance forecasting, performs optimal scheduling by tracking the generation schedule and generating a schedule for the next day. The work in [27] includes a term to reduce generation variation and increase the cycle lifetime.

2.3 Artificial Neural Networks

The objective of this work is to apply MPC to the ACUREX plant and obtain an artificial neural network (ANN) that reproduces its behaviors as best as possible. For this purpose, an MPC will be used for simulation, and then, a multilayer perceptron (MLP) will be trained from its outputs.

This section gives a description of artificial neural networks, explaining their principle of operation, structure, components and training process. Then, a small overview of different ANNs to control systems will be given, dividing them into neural networks used for identification and for control.

2.3.1 Description

An artificial neural network is an algorithm that tries to emulate the functioning of the human brain, composed of several interconnected neurons performing different tasks. Mathematically, an ANN is a function $f_{\text{NN}}(\cdot)$ formed by weighted sums of functions $g(\cdot)$. Its origin takes back to 1943 when [63] modeled a neural network using electrical circuits. They were not very successful at the time of their birth, as the emergence of training algorithms came gradually. Even so, the capabilities of computers at the time made it difficult to use them in most applications. In recent years, advances in technology have made it possible to extend their use and to train large networks in reasonable times.

As pointed out by [79], the term artificial intelligence (AI) has many different definitions but could be described as *the ability of a device to perceive the environment and take the necessary actions to maximize the chances of satisfying an objective*. Machine Learning (ML) is a subset of AI that involves giving machines the ability to learn from pattern recognition. The difference between machine learning and deep learning (DL) is much more subtle. Deep learning (DL) is a subtype of machine learning where large neural networks are used, which are more complex algorithms that can make decisions.

ANNs have been widely used for modeling relationships between data and finding patterns. There are many types of applications of neural networks and one possible categorization is regression, classification and data processing [95]. Their functioning bases on how humans learn. Human brains have around 86 billion neurons, which are structures in the nervous system that receive a series of stimuli and transform and transmit them to the rest of the body. They are made up of the following elements (figure 2.3):

- Dendrites: they are branches that receive input from other cells.
- Axon: it is a long, thin structure located at the output that transmits the signal.
- Cell body or soma: it contains the nucleus with the genetic information and provides energy to the rest of the neuron.

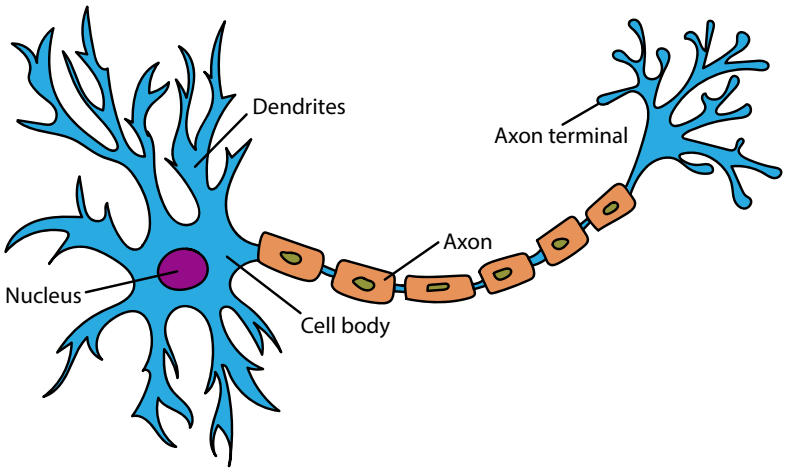

Figure 2.3 Main parts of a neuron.

When processing information, neurons connect with each other in what are known as synapses, interacting to process information. This allows them to parallelize processing and handle large amounts of information very quickly. ANNs try to emulate this process by layered structures containing an input layer (where information goes in) and an output layer (where information comes out). These layers are formed by what are called neurons or nodes.

Linear Regression

In an ANN, each neuron computes a linear regression problem, the activation function translates it to an active or non-active state and the use of several nodes gives rise to more complex functions. The problem of linear regression is one of the most common machine learning problems and consists of finding the curve that better fits a set of data. It is a type of supervised learning, which means that the output of the training data is beforehand labeled, so the output is known. The algorithm will try to fit a curve that makes predictions or classifications from new data.

For a regression problem, given an input vector x and an output vector y, a hypothesis function of the form $h_w(x) = w_0 + w_1 x_1 + \cdots + w_n x_n$ is needed, where w is the parametric vector that needs to be chosen. For this purpose, an optimization problem is solved, where the cost function is half of the mean squared error [66]:

$$\min_w J(w), \;\; J(w) = \frac{1}{2m} \sum_{i=1}^{m} (h_w(x_i) - y_i)^2 \tag{2.3}$$

To solve this minimization problem, the gradient descent technique is used. A point is taken, the direction with the steepest slope is sought repeatedly and w is decreased proportionally to the slope until a minimum is reached (which will be a global minimum because the function is convex).

$$w_j := w_j - \alpha \frac{\partial J(w)}{\partial w_j} = w_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_w(x_i) - y_i) x_i j \tag{2.4}$$

where α is the learning rate, which sets the size of the jumps at each iteration and, in consequence, the speed of learning.

Perceptron

The most basic form of ANN is the perceptron, which consists of a cell that receives a series of inputs and implements a linear regression problem so that $\sum_{i=1}^{n} w_i x_i$ is obtained. If this value is above a certain threshold, the output value will be 1. Otherwise, the neuron will output the value of 0. This structure alone is only capable of reproducing linear relationships with binary outputs. This is where the multilayer perceptron (MLP) comes in.

The MLP is one of the most useful neural networks in function approximation. It is a feedforward neural network, which means that all neurons in a layer are connected to the neurons in the layer before and there are not cycles between them [33]. Each neuron of the network computes equation 2.5, giving $z_i^{(l)}$. The superscript (l) indicates the layer, which is composed of $n^{(l)}$ neurons. $w_{ji}^{(l-1)}$ is the kernel between neurons j and i of layer $l-1$ and $b_i^{(l-1)}$ is the bias unit of neuron i in layer $l-1$. At the output of a neuron, there is an activation function $g^{(l)}$ giving the output $a_i^{(l)}$. The activation functions (explained later in this document) are usually the same for the entire layer.

$$a_i^{(l)} = g^{(l)}(z_i^{(l)}), \; z_i^{(l)} = \left(\sum_{j=1}^{n^{(l-1)}} w_{ji}^{(l-1)} a_j^{(l-1)} + b_i^{(l-1)} \right) \tag{2.5}$$

Neural networks are distributed in three types of layers:

- Input layer: the layer that receives the input, formed by as many nodes as inputs. It corresponds to the dendrites.

- Hidden layers: layers that transform the information, whose information is not directly related to the known data.

- Output layer: layer with the same number of nodes as outputs that corresponds to the axons.

According to the number of layers, neural networks can be shallow –if they have only one hidden layer– or deep –if they have two or more hidden layers. The general scheme of an MLP is shown in figure 2.4, where there is only one hidden layer (colored in yellow). The input layer is colored in red, and the output layer is colored in green.

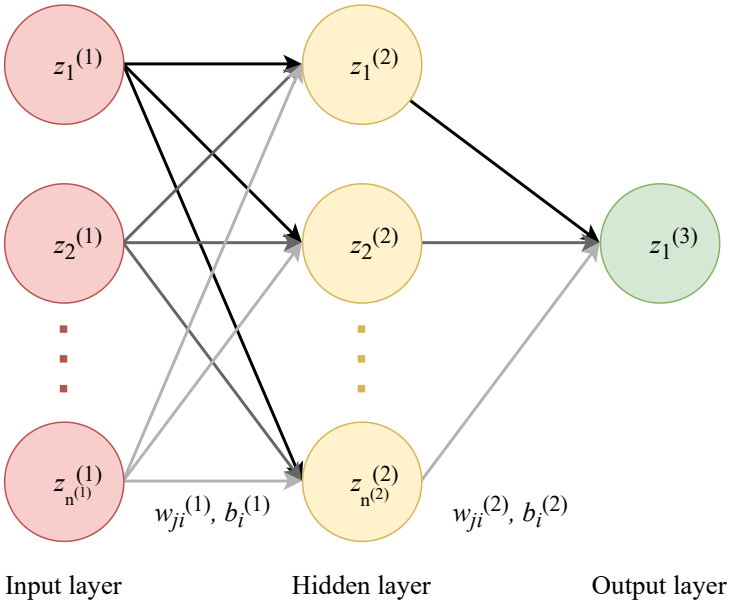

Figure 2.4 Diagram of a multilayer perceptron with one hidden layer, where $z_i^{(l)}$ is the output of node i and layer l, $w_{ji}^{(l)}$ is the weight vector between neurons i and j, and $b_i^{(l)}$ is the bias unit.

Activation Functions

As previously mentioned, the output of a neuron needs to be treated. For example, in the case of the simple perceptron, the output is binarized using a threshold. ANNs use different types of functions at the output of each neuron to satisfy particular demands. These functions are called activation functions [81], which give their name to the neuron with which they are associated. A sigmoid unit, for example, is a neuron that has a sigmoid function at its output.

Activation functions define the output of a neuron, allow the addition of constraints and limits in a certain range and can facilitate the training process of the neural network. As mentioned above, the original activation function was simply a relay, but over time other functions have emerged that are preferred. These functions have smooth gradients, which prevents the occurrence of jumps at the output. Different types of activation functions allow multi-output and facilitate the learning process. Figure 2.5 shows the most common activation functions, being described by the following equations:

- Linear: $g(z) = z$

- Rectified linear unit (ReLU): $g(z) = \max(0,z)$

- Sigmoid: $g(z) = \frac{1}{1+e^{-z}}$

- Hyperbolic tangent: $g(z) = tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$

Non-linear activation functions allow solving non-linear regression problems using multiple layers, while linear activation functions obtain simple linear combinations of linear functions, producing a linear regression. These functions are typically used in the output layer, as it does not need to be saturated. Saturating the output in intermediate layers allows homogenization and facilitates training. In addition, the hyperbolic tangent function is zero-centered and has a steeper derivative than the sigmoid function, making it more efficient. Sigmoid and hyperbolic tangent functions saturate their output when the values are too high or low. This is helpful to impose restrictions and to constraint the output between the values of the real one used for training. By changing the scaling of the input, it is also possible to change those limits.

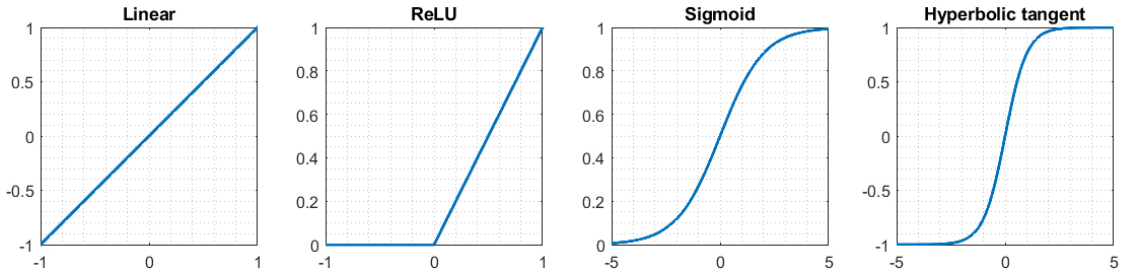

Figure 2.5 Most known activation functions.

Backpropagation

As with the aforementioned linear regressors, the parameters of the neural networks need to be determined by an iterative process. First, the weights are initialized and then the output error is calculated so that the weights can be adjusted. This is done until a particular criterion is satisfied. In the case of ANNs with hidden layers, the errors are not so easy to calculate and have to be obtained indirectly from the output error. This is accomplished with an algorithm called backpropagation [84].

This algorithm consists in finding the gradients of the weights to actualize them according to:

$$w_{ji}^{(l)} := w_{ji}^{(l)} - \eta \frac{\partial J}{\partial w_{ji}^{(l)}} \tag{2.6}$$

$$b_j^{(l)} := b_j^{(l)} - \eta \frac{\partial J}{\partial b_j} \tag{2.7}$$

Before applying the algorithm, the error of each neuron should be defined as:

$$\delta_j^{(l)} = \frac{\partial J}{\partial z_j^{(l)}} \tag{2.8}$$

Then, the algorithm is based on the following four equations [71], that can be obtained from the definition of the error applying the chain rule:

- Error in the output layer:

$$\delta_j^{(L)} = \frac{\partial J}{\partial a_j^{(L)}} g'\left(z_j^{(L)}\right) \tag{2.9}$$

- Error in layer l, obtained from error in layer $l+1$:

$$\delta_j^{(l)} = \sum_k \delta_k^{(l+1)} w_{kj}^{(l+1)} g\prime\left(z_j^{(l)}\right) \tag{2.10}$$

- Partial derivative of the cost with respect to the bias term:

$$\frac{\partial J}{\partial b_j} = \delta_j^{(l)} \tag{2.11}$$

- Partial derivative of the cost with respect to the weights:

$$\frac{\partial J}{\partial w_{ji}^{(l)}} = a_k^{(l-1)} \delta_j^{(l)} \tag{2.12}$$

Once described the equations needed, it is possible to apply the training algorithm:

Algorithm 2 Backpropagation algorithm [84].

```
Initialize weights
for each training sample do
    Read training data
    Obtain the output
    Calculate the output error using 2.9
    Propagate the rest of the errors using 2.10
    Obtain the partial derivatives of J with respect to the weights using 2.11 and 2.12
    Actualize the parameters using 2.6
    Obtain the cost
end for
```

Note that the cost function is not defined here. From the backpropagation method, different learning rules can be used. One of the most efficient backpropagation algorithms for medium-size neural networks is the Levenberg-Marquardt [56, 61], which uses a sum of squared errors as a loss function.

Training an ANN

By doing the whole process above, what the neural network equations do is solve a fitting problem. In this way, any MLP with a suitable number of nodes and layers will be able to approximate the output of any continuous function. To select the number of neurons, a trial-and-error process is performed: first, a small number of layers and neurons is used, which can be based on the experience and the number of inputs. Then, the parameters are selected and the ANN is trained to calculate the fitting error. More neurons are added and retrained until good results are obtained. The same is done to adjust other neural network parameters such as learning rate or activation functions.

Therefore, when training a neural network, it is necessary to follow a procedure:

- Pre-processing the data: To obtain a fast training, it is good to normalize the input and output variables beforehand. Typically, data are scaled between 0 and 1 or between -1 and +1.

- Defining the subsets: The data is usually divided into three subsets to train and then test the neural network using different data and avoid overfitting – i.e. when the output fits too well on the training data, but it does not adapt well to new data.

  - Training set: used for adjusting the parameters.
  - Validation set: used to validate the behavior of the network and adjust some parameters.
  - Test set: used for estimating the functioning of the network when fed with new data.

- Specifying the architecture and parameters of the network: This step must be repeated until finding a network that performs well.
- Learning: The input data is received and used for calculating the weights. This is the most time-consuming step.
- Evaluating the network: The Pearson correlation coefficient R and the mean squared error (MSE) are calculated for each subset. If these values are not good enough, the process must be repeated from the third step onwards. Other metrics can be used.

  The following equations describe the two metrics mentioned:

$$R = \frac{\sum_{i=1}^{N}(x_i - x_{mean})(y_i - y_{mean})}{\sqrt{\sum_{i=1}^{N}(x_i - x_{mean})^2 \sum_{i=1}^{N}(y_i - y_{mean})^2}} \tag{2.13}$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(x_i - y_i) \tag{2.14}$$

2.3.2 Application to Control Systems

Neural networks are now on the rise, which means that they are being applied to many different fields, from predicting defaulting customers to image recognition and control systems. The latter application is the one that occupies this project, with two main approaches: identification and obtaining the control signal, of which there have been numerous references since the late 1980s.

The most commonly used approach consists of using the ANN to model the behavior of the plant and using it as a black-box model. Precisely in the field of parabolic trough collectors, a neural network has been applied to obtain a nonlinear model of the ACUREX plant and then apply MPC and adaptive MPC in [9], [41] and [25]. In [40], another adaptive constrained MPC is implemented and the ANN is trained online. In [23], the ACUREX plant is controlled using PID control with a fuzzy logic switching supervisor. Those PIDs are tuned offline using an Elman neural network. [42] uses a recurrent ANN aimed at obtaining a pseudo-inverse of the ACUREX plant to apply FLC techniques. Another strategy implemented in [44] consists of using a neural network for scheduling between a set of PID controllers, a priori tuned in different operating points by means of Takahashi rules. [101] make three models of a PTC plant to estimate the hourly electric production. One of them is ANN and they conclude that this is the model that performs best

Regarding the use of neural networks to control other fields of industry, there are many applications. [102] apply MPC to sulphitation in sugar refining industry using an ANN as a prediction model of the plant. In [77], a recurrent neural network is used to model the dynamics of a steam generator and tune the parameters of a PI controller. In [45] a recurrent neural network is used to identify delay-free nonlinear dynamic systems. Another recurrent neural network is applied in [59]. They use the neural network to make the predictions for a predictive controller in an oil-cooling process. [53] use a neural network to estimate the optimal operating point in a real-time optimization scheme for photovoltaic panels and microbial fuel cells. In [38], an ANN is used for forecasting the next day's photovoltaic generation and load consumption in a building.

The second approach is to calculate the control signals directly using neural networks. No applications in solar thermal plants have been found in the literature, although they exist in other fields. [80] apply a neural

network controller to a plant that converts polar coordinates into cartesian coordinates. They use ANNs that learn the inputs by calculating the error. [47] apply a self-learning neural network to simple systems using evaluations as teachers. [76] control the trajectory of a robot by approximating the output of an MPC controller using multilayer feedforward neural networks. A hybrid control system combining a neural network with a compensated controller is proposed in [57]. [54] use a neural network to solve the optimization problem in MPC. Instead of using a previous MPC to train the network –as this document proposes–, it is trained to minimize the cost. A similar approach is proposed in [24], where the optimization problem is solved with dual neural networks that give the optimal solution and not approximation. [93] control the movement of a pendulum using a neural network with a denoising autoencoder that emulates the behavior of MPC. [28] make control strategies based on MPC to approximate its behavior to control the temperature in a building. [52] learn to approximate the output of an MPC controller applied to an energy management system in a smart building and [99] use neural networks to iteratively solve the quadratic problem in MPC. [98] implement distributed model predictive control using local neurodynamic optimization. ANNs are trained to optimize the output.

Some examples combine both approaches. [69, 70] control the steering of a trailer and use two neural networks: one to emulate the system and another one to control the emulator by self-learning calculating the difference between outputs of the system given by different control signals and the reference. Two networks are also used in [96] to control a turbogenerator: one for modeling and the other one for adaptive control. [58] model and control a pH process using neural networks. They use two methods for control: neural network with PID and model reference adaptive neural net control. [43] apply a combination of two neural networks for both modeling and control applied to signal processing in sound applications.

2.4 Solar Thermal Power Plant

In this work, a controller based on model predictive control and artificial neural network is developed to control the oil flow in a solar thermal power plant. For this purpose, a simulator of the ACUREX plant will be used, with which the necessary simulations will be carried out to train the neural networks and validate the method.

This section aims to describe the parabolic-trough collector (PTC) field. First, a brief introduction to solar thermal power plants will be given, together with a description of their principal components. Then, the models and parameters of the PTC are introduced.

2.4.1 Introduction

The use of solar collectors remounts to prehistoric times [51], when solar energy was harnessed to dry and preserve food or obtain salt from seawater. Since then, their use has fluctuated for different scientific, social and economic reasons. During the second half of the 1970s, following the big oil crisis that hit most developed countries, renewable energy sources were strongly promoted as an alternative to oil. Eventually, when oil prices fell, this energy source returned to the forefront, leaving renewables on the sidelines. Nowadays, they have been established for environmental causes so that renewable energies, in general, are expected to overthrow fossil fuels. We are currently in the process of transition to renewable energies.

Solar energy production consists of taking advantage of the solar radiation reaching the earth and transforming it into a form of energy useful for human beings. This can be done in two ways: directly by photovoltaic cells (PV) or indirectly by concentrating solar power (CSP). In the case of PV, the generation of electricity makes use of the photovoltaic effect. In materials that manifest this effect, the incoming photons hit the electrons producing electric current. On the other hand, CSP consists of concentrating the incoming solar power on a small area to heat a fluid and produce steam, which is used to drive a turbine. There are four types of CSP technologies: parabolic-trough collector (PTC), linear Fresnel reflector (LFR), solar power tower (SPT) and parabolic dish system (PDS) [103, 16].

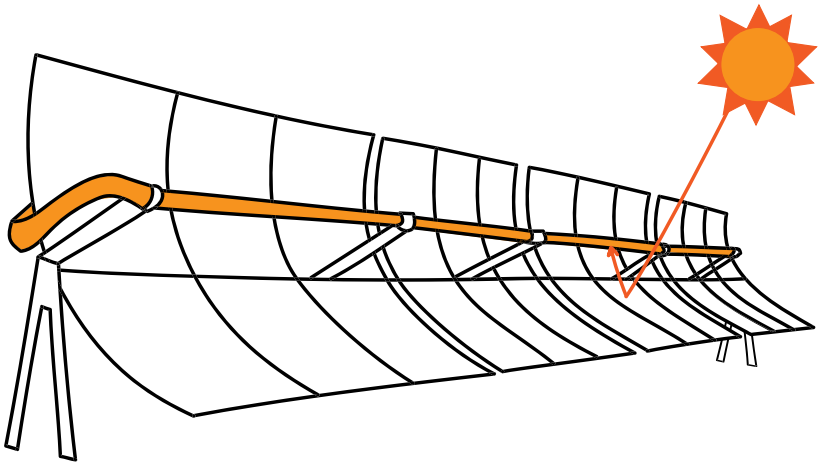

Figure 2.6 Parabolic-trough collectors.

A parabolic-trough collector (PTC) (figure 2.6) consists of a set of mirrors that concentrate the direct normal irradiance (DNI) onto a tube (the receiver) that contains the heat transfer fluid (HTF) –typically synthetic oil or water. These mirrors are curved in a parabolic shape, and the tube is placed in the focal line of the parabola. A group of collectors disposed in line is called a loop, and a set of parallel loops compose the solar field.

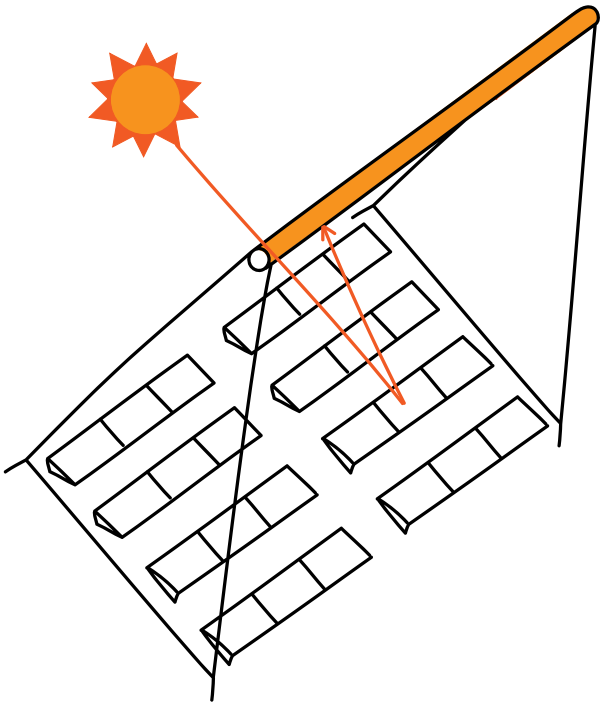

Figure 2.7 Linear Fresnel collector.

A linear Fresnel reflector (LFR) (figure 2.7) is composed of thin mirror strips approximating the parabolic shape and concentrating the sunlight onto a receiver placed at the focal point. They are cheaper, and the design is more accessible than parabolic-trough collectors, but they need a more complex tracking mechanism.

A solar power tower (SPT) (figure 2.8) consists of a field of heliostats –i.e., sun-tracking reflectors– that concentrate the sunlight onto a central receiver situated at the top of a tower. Heliostats can be flat or slightly concave mirrors that follow the sun along two axes.

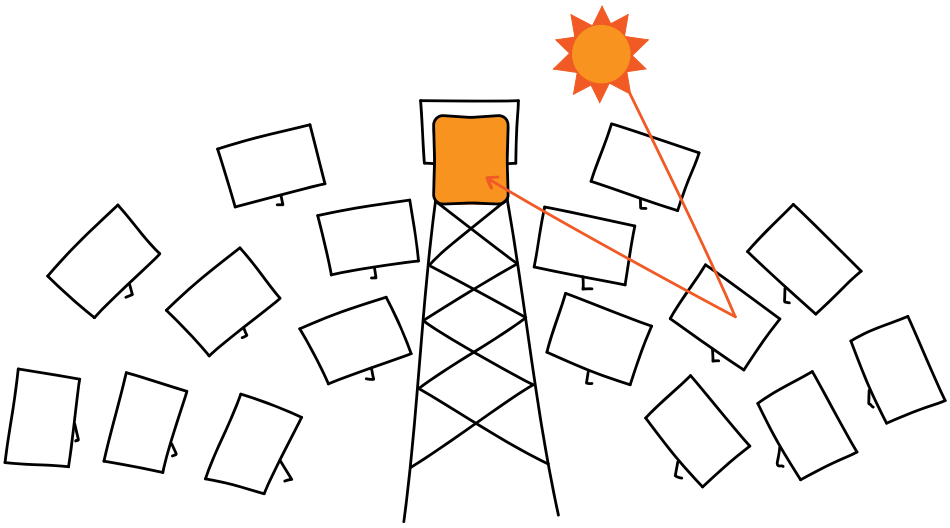

Figure 2.8 Solar power tower.

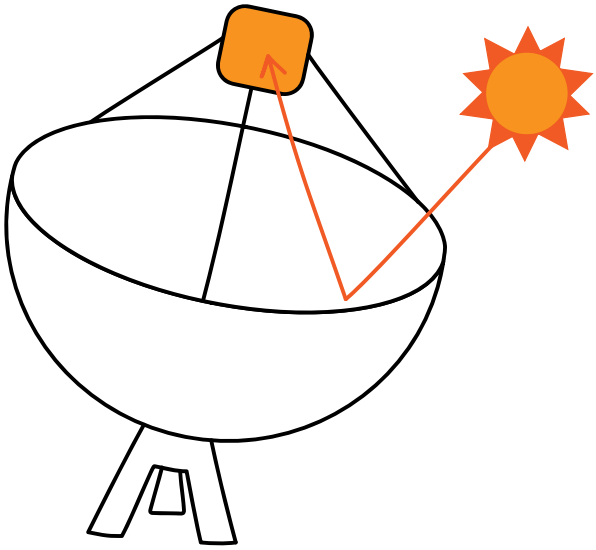

Figure 2.9 Parabolic dish system.

A parabolic dish system (PDS) (figure 2.10) is a parabolic collector dish that concentrates the sunlight onto a receiver at the focal point of the collector, where a Stirling engine is positioned. Both the dish and the receiver move and track the sun along two axes.

There is an additional technology called concentrated solar thermo-electric (CST) that can convert solar energy directly into electricity. They consist of a solar thermal collector conducting the heat to a thermo-electric generator with a thermal resistor.

2.4.2 Components of a Solar Thermal Power Plant

In a solar thermal power plant, several components are needed in addition to those described above. Some sensors and actuators used to control and monitor these types of plants are temperature, sunlight and flow-rate sensors, pumps, valves or sun trackers. Also, some additional elements are helpful to produce and store energy, which gives an advantage to CSP over PV.

The principal element in every solar power plant is the sunlight, which acts as a disturbance to the control system, given the fact that it is not possible to manipulate it [16]. Irradiance or insolation is the power that

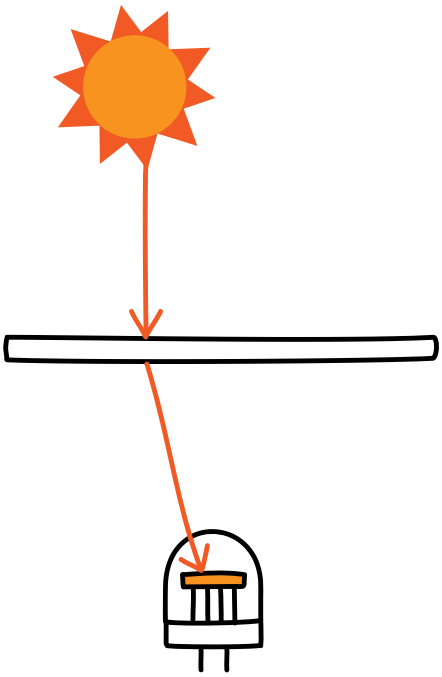

Figure 2.10 Concentrated solar thermoelectric system.

reaches a unit area on the earth and is measured in W/m^2. The radiation or irradiation is the integral of the irradiance, measured in J/m^2. It is worth mentioning that, although these two concepts are not the same, the solar irradiance is usually denominated as solar radiation [16]. When the solar radiation reaches the atmosphere, some of it is scattered, absorbed by the air or reflected. The part of this radiation that reaches the ground is the diffuse radiation, and the radiation that reaches the Earth's surface directly is the direct solar radiation (figure 2.11).

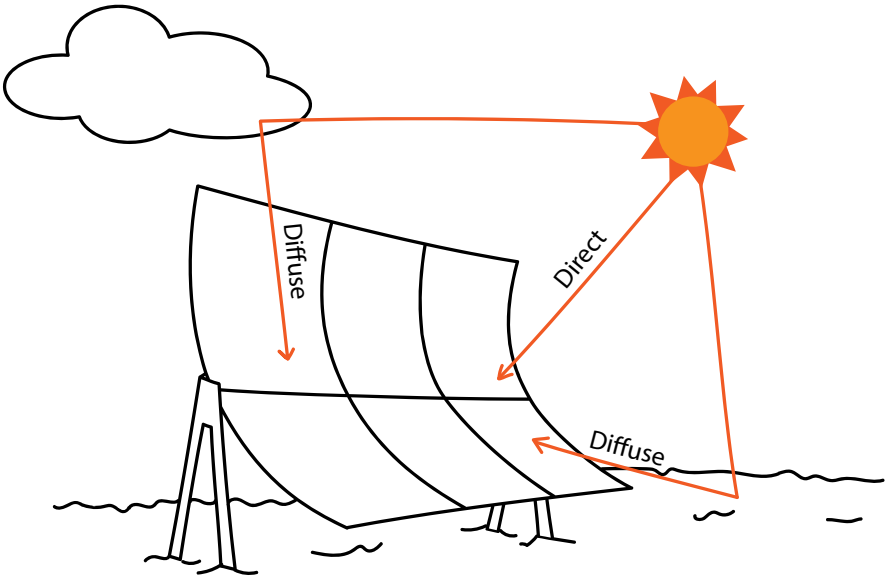

Figure 2.11 Types of radiation on a reflector.

The instruments used for measuring the solar irradiance are pyranometers and pyrheliometers. The pyranometers are used to measure the sum of direct and diffuse solar irradiance, while pyrheliometers measure the direct normal component of the solar irradiance. In a pyrheliometer, there is a small window

trough which the sunlight enters, and a thermopile converts it to an electric signal.

Once the sun reaches the mirrors, its light is reflected into a receiver containing the heat transfer fluid (HTF) [49, 8]. This fluid should have a low melting point and a high boiling point, thermal stability, low vapor pressure at high temperature, low viscosity, high heat capacity and high thermal conductivity, and it must also be low corrosive. The HTFs are classified into air and other gases, water or steam, thermal oils, molten salts, and liquid metals.

When there is no sunlight available, the HTF can be stored in a tank to overcome the problem of intermittency [49, 103, 16]. The thermal energy storage (TES) (figure 2.12) system can be active or passive, two-tank or single-tank and direct or indirect. The main objective is to use the excess heat of the field and store it through an energy transfer mechanism. The most used materials are synthetic oil and molten salt.

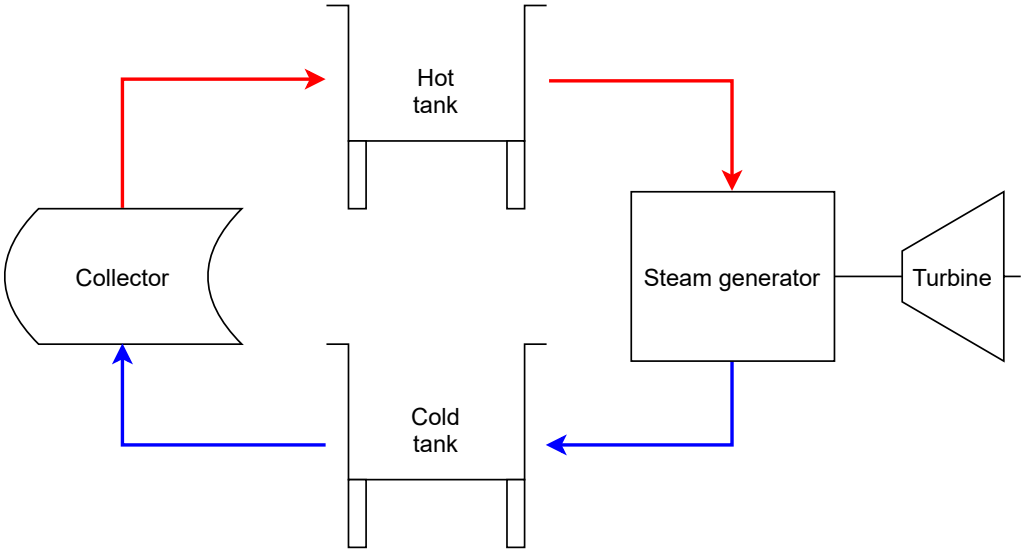

Figure 2.12 Two-tank thermal storage system.

The plants that do not have TES need some power reduction to process the solar energy on too sunny days. On those occasions, some plants have a safety mechanism based on defocusing some collectors, which consists of modifying the angle of the collector to take it out of focus and decrease its efficiency [88].

2.4.3 Model of the Plant

Parabolic trough plants are composed of a certain number of parallel loops formed by collectors. In this work, one loop of collectors is used to make the control and simulations and can be extended to more loops. Each loop is divided into segments of size Δl, as shown in figure 2.13.

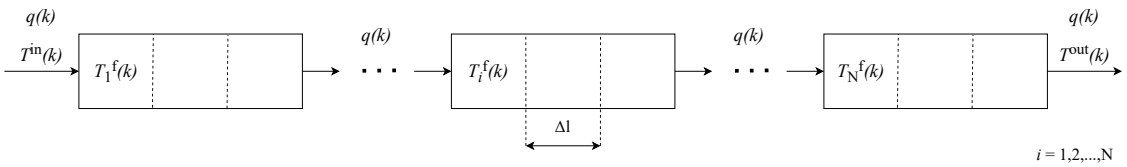

Figure 2.13 Schematic of one loop of the collector field.

The dynamics are mainly dictated by the HTF, which depends on the operating conditions: radiation, reflectance, inlet temperature, ambient temperature and outlet temperature [11]. Changes in the flow rate

lead to changes in the outlet temperature of the field. To model PTC plants, there are two well-known models mainly used in the literature: the lumped-parameter model and the distributed-parameter model.

The lumped (also concentrated-parameter model) is an approximation of the dynamics of the system that does not take into account the metal-fluid heat transmission and the spatial distribution of temperatures. It describes the variation of the internal energy of the fluid and is given by the following equation:

$$C_{\text{loop}} \frac{\partial T_{\text{out}}}{\partial t} = n_o K_{opt} SI - qP_{\text{cp}}(T_{\text{out}} - T_{\text{in}}) - H^l A^{\text{m}}(T_{\text{mean}} - T^{\text{a}}) \tag{2.15}$$

where $C_{\text{loop}} = \rho^{\text{m}} C^{\text{m}} A^{\text{f}} L_{\text{loop}}$ is the thermal capacity of the loop and $P_{\text{cp}} = \rho^{\text{m}} C^{\text{m}}$. The subscripts *in*, *out* and *mean* refer to inlet, outlet and the mean value between them.

The model employed in this work is the discretized distributed-parameter model [21, 34, 62], which not only uses differential equations –as the concentrated-parameter model– but it also takes into account partial derivatives. The subindexes f, m and a indicate fluid, metal and ambient.

The first equation represents the energy balance for the metal tube on each segment i between times $(k-1)\Delta\text{T}$ and $k\Delta\text{T}$:

$$\begin{aligned} T_i^{\text{m}}(k) = T_i^{\text{m}}(k-1) + \frac{\Delta\text{T}}{\rho^{\text{m}} C^{\text{m}} A^{\text{m}}} \Big(& \eta_i^{\text{col}} G_i I_i(k) - \pi D^{\text{m}} H_i^{\text{l}}(k-1)\Big(T_i^{\text{m}}(k-1) - T^{\text{a}}(k)\Big) + \\ & - \pi D^{\text{f}} H_i^{\text{t}}(k-1)\Big(T_i^{\text{m}}(k-1) - T_i^{\text{1f}}(k-1)\Big)\Big) \end{aligned} \tag{2.16}$$

The second equation is the energy balance on the fluid for each segment i:

$$T_i^{\text{f}}(k) = T_i^{\text{f}}(k-1) + \frac{\pi D^{\text{f}} H_i^{\text{t}}(k-1)\Delta\text{T}}{\rho_i^{\text{f}}(k-1) C_i^{\text{f}}(k-1) A^{\text{f}}} \Big(T_i^{\text{m}}(k-1) - T_i^{\text{1f}}(k-1)\Big) \tag{2.17}$$

where $T_i^{\text{1f}}(k)$ is an auxiliary temperature. It helps to take into account the temperature of the adjacent segment in the energy balance and is obtained from the following equation:

$$T_i^{\text{1f}}(k) = T_i^{\text{1f}}(k-1) - \frac{q(k)\Delta\text{T}}{A^{\text{f}}} \Big(T_i^{\text{f}}(k) - T_{i-1}^{\text{f}}(k)\Big) \tag{2.18}$$

In this system, the manipulated variable is the flow rate $q(k)$; and the DNI $I_i(k)$, the inlet temperature $T^{\text{in}}(k)$ and the ambient temperature $T^{\text{a}}(k)$ are acting as disturbances that must be measured or estimated. As will be explained later in this document, one loop of the plant will be controlled by MPC, which selects the control signals based on predictions along a prediction horizon. For this purpose, estimations of the irradiance will be necessary for this horizon.

To model the relationship between the inlet and outlet temperatures, the inlet temperature is expressed as a first-order system with a time constant of 10 min:

$$\frac{T^{\text{in}}(s)}{\tilde{T}^{\text{out}}(s)} = \frac{1}{600s+1} \tag{2.19}$$

where $\tilde{T}^{\text{out}}(s) = T^{\text{out}}(s) - 90$ allows to consider the temperature fall in the steam generator.

Then, equation 2.19 is discretized with a discretization time of 0.5 s:

$$T^{\text{in}}(k) = 0.999167 T^{\text{in}}(k-1) + 0.000833\,\big(T^{\text{out}}(k-1) - 90\big) \tag{2.20}$$

2.4.4 Main Control Strategies

The control subjects in solar systems are gathered in figure 2.14 [16]. In this work, the aim is to control the flow rate of the HTF to maximize the obtained electric power, but many different algorithms and criteria have been used as, for example, temperature tracking or maximizing the temperature.

In parabolic-trough collector plants, the control objective is usually to maintain the outlet temperature around a specific value selected by the operator. To optimize the operation, this temperature is often selected by a high-level control layer to maximize the obtained electric power. Due to the solar cycle characteristics, the reference temperature has to change considerably, which leads to variations in the response rate and dead times. As pointed out by [16], the ideal controller should be high-order and non-linear. In solar thermal plants, the characteristics of the system make it necessary to reach a compromise between different objectives in order to reach the best possible solution. In many cases, it has been chosen to maximize the HTF temperature since it increases the power cycle efficiency but also increases the losses. This is a clear example of where an optimization-based controller can be used.

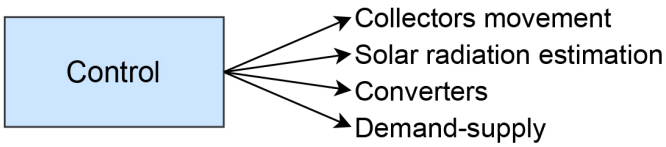

Figure 2.14 Main issues in solar systems control.

[16] summarizes the principal control architectures used for controlling parabolic trough collectors, dividing them into basic and advanced control algorithms:

Feedforward control

Feedforward (FF) control is widely used in the industry for its ability to reject solar radiation perturbations [22, 15], as the model used relates changes in outlet temperature only to changes in fluid flow. It consists of adding a controller in series (figure 2.15) or in parallel with the main controller to help avoid perturbations.

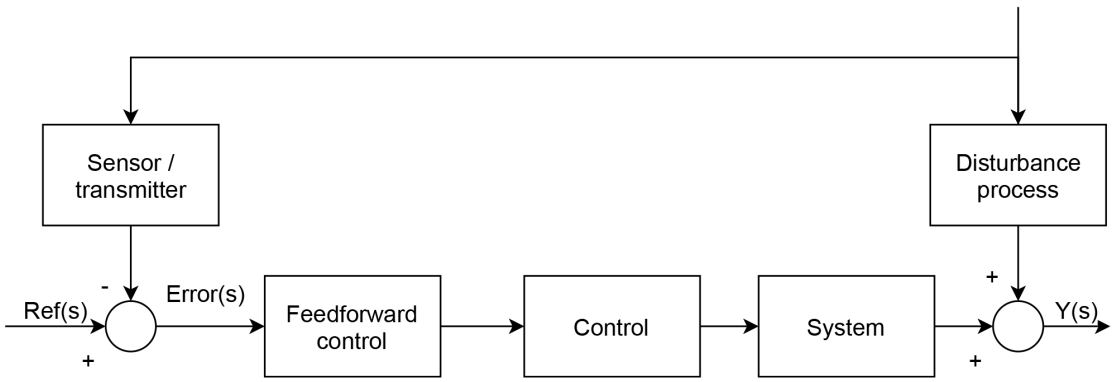

Figure 2.15 Series feedforward.

Because of their ability to cope with disturbances, FF controllers have been used on numerous occasions to improve the control of PTCs and are still in use today. [73] use a PI and a feedback controller combined with a FF loop that provides the required pump pressure difference. [36] apply adaptive MPC to ACUREX adding a parallel FF for disturbance rejection and, in [90], GS-GPC is used with a series FF controller.

PID

Due to the aforementioned dynamic characteristics, the use of proportional, integral, derivative (PID) controllers has been restricted to safe operation conditions as they need additional compensators in the control loop. Also, the performance is restricted by the excitation of resonance modes. They are used in the low-level layer.

It is a well-known method that consists of adding a proportional term, a integral term to eliminate the steady-state error and a derivative term to increase system stability and control oscillations. These three terms are represented in the following equation in the frequency domain.

$$C(s) = K_p(1 + \frac{1}{T_i s} + T_d s) \tag{2.21}$$

Examples of PID applications to PTCs are [23] and [39], where, depending on the operating point, different PID controllers are used. In [73], a PI (a PID without derivative term) is combined with a feedback and a feedforward controller. In [87] and [26], PIDs are combined with cascade control.

Cascade control

Cascade control (CC) is used to cancel the effect of disturbances (figure 2.16). An inner control loop (slave) is dedicated to compensating the disturbances and the outer control loop (master) controls the process input. The secondary loop must be faster than the primary to compensate for disturbances before they affect the output and avoid destabilizing the system. In [87] and [26], a cascade controller is applied to ACUREX using predictive adaptive control for disturbances in the inner loop and PID in the outer loop.

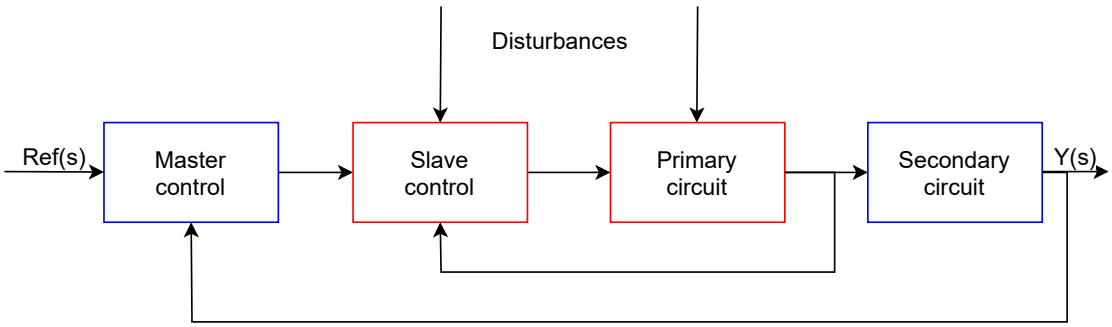

Figure 2.16 Cascade control. Inner control loop in red and outer control loop in blue.

Adaptive control

Adaptive control (AC) allows coping with variations by modifying the controller when the process dynamics change. The state variables are divided into two groups: the fastest ones correspond to the process variables, and the slowest ones correspond to the estimated process or controller parameters. Figure 2.17 shows a scheme of adaptive control that adapts the parameters by using a reference model. Examples of adaptive MPC application are [12], [40], [36], [90] and [78], as well as the aforementioned works in [87] and [26].

Gain scheduling

Gain scheduling (GS) is able to adapt to changes in the process dynamics, but it is not properly an adaptive controller, although it can be considered as such. The process dynamics can be associated with the value of some measurable process variables used to select the parameters of the controller. The block diagram of a GS is represented in figure 2.18.

In [14], a GS-GPC controller is proposed to control the ACUREX plant. In [85], a GS-GPC controller is used with a solar radiation estimator based on the concentrated parameter model. Other examples of application of GS-GPC to ACUREX are [37], [88] and [90].

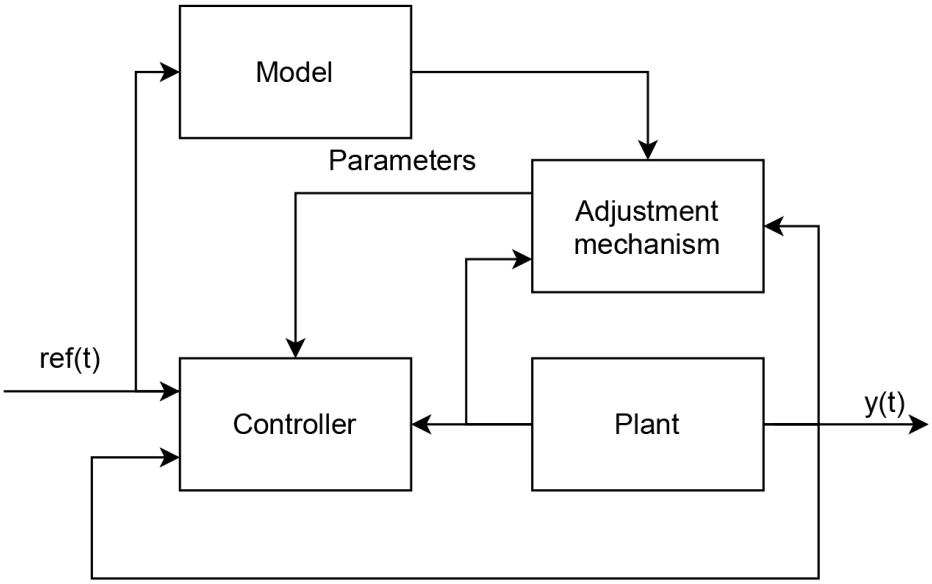

Figure 2.17 Model reference adaptive control.

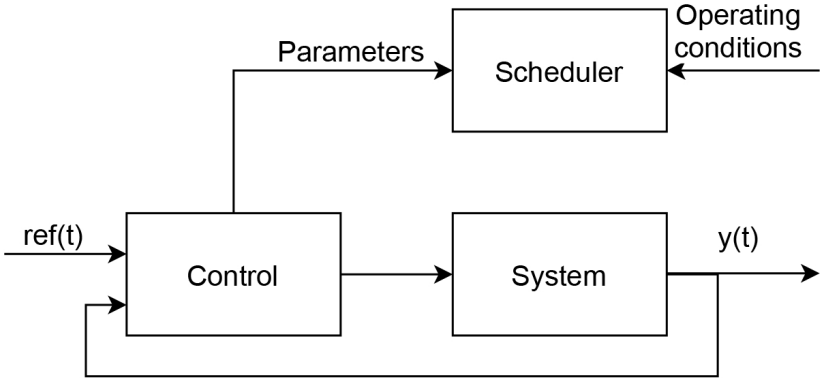

Figure 2.18 Gain-scheduling control.

Internal model control

Internal model control (IMC) uses a model to estimate the output and the result of a perturbation to the system and includes a feedback signal with the error between the outputs of the system and the model (figure 2.19). It is done to take into account uncertainties and disturbances, preserve the validity of linear models in their operating range and improve the response of linear controllers [35]. This signal is fed to the controller, which should compensate for the disturbances and uncertainties by cancelling the error. In [31] and [32], a model based on nonlinear partial differential equations is used as a part of the control design.

Time delay compensation

Time delay compensation (TDC) consists of designing a controller that does not take the pure delay into account by estimating it using a plant model without delay and a model of the pure delay. A block diagram of this kind of controller is shown in figure 2.20. The most known time delay compensator is the Smith Predictor. In [64], a control scheme using a transfer model with resonance characteristics is applied and in [72] a PI controller with dead time compensation is used which can be applied to plants with variable dead time.

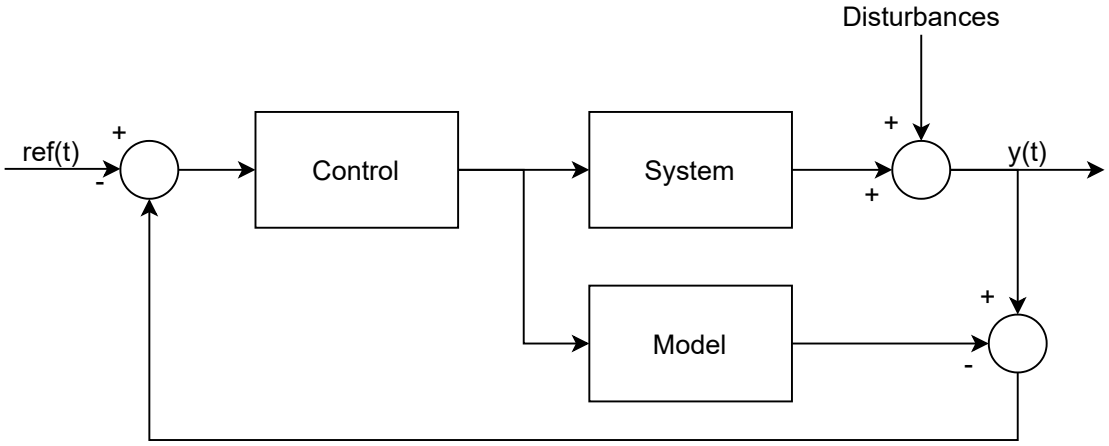

Figure 2.19 Internal model control.

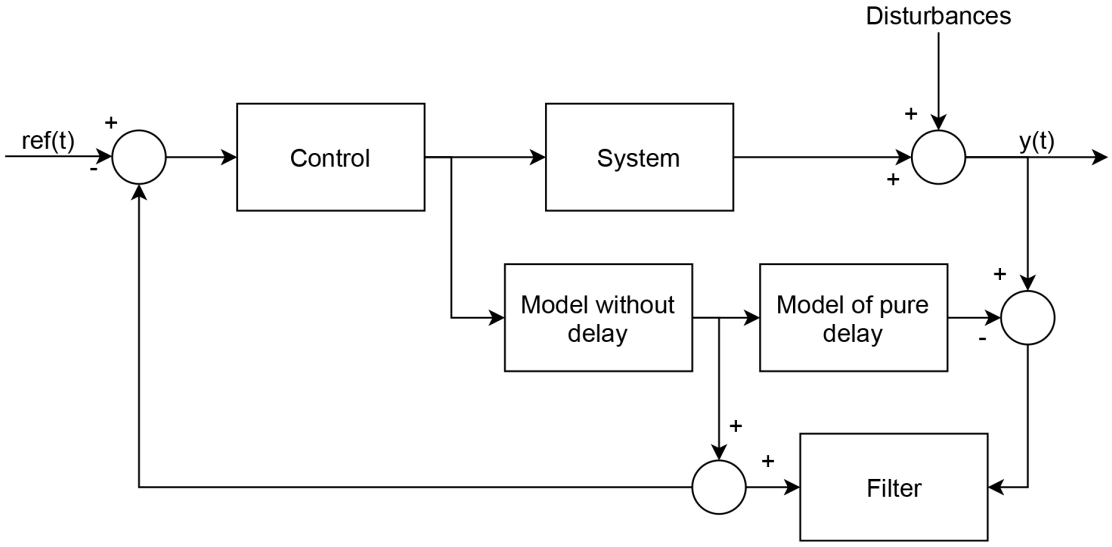

Figure 2.20 Time delay compensator.

Optimal control

The objective of optimal control is to optimize a cost function, generally to maximize the produced electrical power. Linear quadratic gaussian LQG control is one of the principal problems in optimal control, which combines the Kalman filter as an estimator with a linear quadratic regulator. The use of an optimal control formulation is proposed in [82] to maximize the power produced taking into account the pumping power. In [97], a two-level optimization problem is solved for reference tracking using a nonlinear hyperbolic partial differential equation based on the energy balance and, in [29], an algorithm is proposed for implementing a numerical solution of the optimal control.

Robust control

Robust control (RC) comprises different tools that allow considering the discrepancies between the model and the actual process, i.e., it takes into account the possible uncertainty caused by the existence of disturbances or the adjustment of some parameters. The best-known example of robust control is the H_∞ theory, which aims to tune controllers by minimizing the sensitivity of the system over its frequency spectrum. In [75] an H_∞ controller is developed together with a feedforward controller and applied to ACUREX ensuring high stability and [6] propose control strategies using Takagi–Sugeno fuzzy models combined with a parametric uncertainty robust control approach.

Non-linear control

Non-linear control (NC) takes into account non-linearities of the plant using strategies with nonlinear transformations of variables, augmenting the performance and robustness of the controller to the expense of the complexity. It receives its name because it is applied to systems governed by nonlinear equations where the superposition principle cannot be applied. [5] proposes the use of feedback linearization with Lyapunov-based adaptation and a simplified plant model, adding a nonlinear transformation on some variables to cope with plant nonlinearities. In [68], a multivariable nonlinear model-based adaptive predictive controller is proposed to increase the robustness

Fuzzy logic control

Fuzzy logic control (FLC) allows handling imprecise and incomplete data by quantifying certain linguistic variables that represent some characteristics. These variables can take different values provided by experts and are then controlled using heuristic rules. First, the input data go through a fuzzification, blurring or merging process, in which qualitative characteristics are given a numerical value. Subsequently, decisions are made according to logical relationships known as fuzzy rules. Finally, in the defuzzification stage, concrete data are obtained, required by controllable systems. The process is represented in figure 2.21.

Figure 2.21 Process of fuzzy logic.

[23] use different models for the different operating points and combine PID control with a fuzzy logic switching supervisor to switch between different PIDs. [48] apply fault-tolerant MPC using multiple takagi-Sugeno fuzzy models to detect abrupt and gradual faults. They have a nominal model and various fuzzy Takagi-Sugeno models used for residual generation. In [4], a neuro-fuzzy approach is presented for location optimization of solar plants.

Model predictive control

Model predictive control (MPC) uses a model to predict the output of the system and then select the optimal control signal by minimizing an objective function with constraints (figure 2.22). Both the calculation of the control signals and the prediction of the outputs are obtained over a period of time –the control horizon and the prediction horizon. This gives it a predictive character since the best control signal is chosen, taking into account how the system will be affected by them and by disturbances in the future. More details of this technique and recent references can be found in section 2.1.

Neural network control

Neural network control (NNC) is based on the use of an artificial neural network (ANN) to control the system, as it can solve regression problems to make predictions. Usually, the neural network is used for identification purposes and then combined with other techniques. In these cases, the ANN is working as a black-box model that approximates the outputs of complex systems given some inputs.

Apart from modeling, it can be used for implementing different parts of the controller, as governing a strategy from a higher layer, like switching between different previous controllers or approximating the results of an optimizer. In this work, a neural network is used to approximate the outputs of an MPC controller, avoiding solving the optimization problem at each sample time of the controller. A more detailed explanation of neural networks and their applications is given in section 2.3.

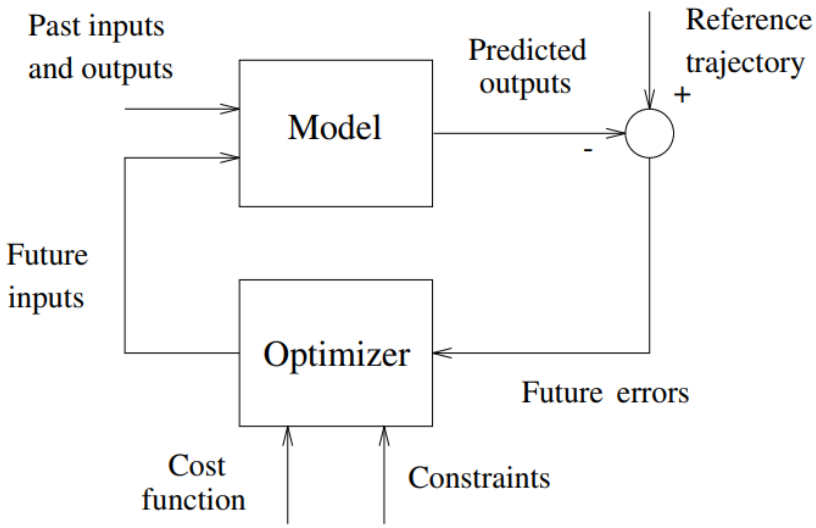

Figure 2.22 Scheme control of MPC. Extracted from [17].

3 Problem Formulation

As mentioned in previous chapters, model predictive control has been successfully applied to this plant on several occasions, but its computational cost is high for application in large real plants. Recent contributions made in [15, 34, 62] could be highlighted. MPC can be used to control non-linear systems (such as parabolic trough collectors) according to different criteria, and not only for reference tracking. In addition, it allows the introduction of constraints and, for example, to limit the temperature of the HTF. All this is done while maintaining a predictive character. This means that predictions of outputs in the system are taken into account for a given time and that at each instant, the immediate control signal is not calculated, but a strategy is established over a time horizon. This avoids too abrupt changes in the behaviour of the system since the optimal solution at one instant could make control impossible at later instants.

Despite all the advantages of MPC, there are currently lines of research trying to overcome its drawback, which is the computation time. In systems with a certain complexity, such as PTC fields, solving the optimization problem can be very costly, making it impossible to be implemented in large commercial plants. This work proposes using previously trained neural networks to relieve the computational burden while obtaining a suboptimal solution to the problem of maximizing the electrical power produced.

This chapter aims to describe the problem formulation for both the MPC and the ANN controllers. The parameters of the system are particularized for the ACUREX plant and the required constraints and boundary equations are described since it is essential to consider them when controlling the plant.

3.1 Model Predictive Control Strategy

Model predictive control (MPC) is a control technique in which a dynamic model of the system is used to predict its future outputs and consider them when calculating the control signal. An optimization problem is solved from these predictions to obtain the best control signals to send to the plant. This is done by minimizing a cost function in which different criteria can be taken into account.

In this work, an MPC controller has been implemented to maximize the electrical power produced in a parabolic trough collector plant. The applied strategy is similar to those used in [34] and [62], but characterized for a single collector loop. This controller has been used to teach optimal outputs to a neural network, which will be able to mimic its behavior while shortening computation times.

The control variable used is the flow rate through the loop piping. This variable is also the input to the distributed parameter model. Variations in the flow rate will allow increasing or decreasing the temperature of the fluid to produce steam then and turn a turbine. A first control objective could be to operate at the maximum temperature allowed by the characteristics of the materials and thus produce more steam increasing the efficiency of the power cycle. In fact, many commercial plants follow this strategy, but an analysis of

other aspects of the system shows that an increase in temperature also leads to an increase in the thermal losses of the field, which affects the overall efficiency, as indicated by [18]. This leads to the need to use other optimization criteria that allow a compromise between increasing thermal output and decreasing losses. Therefore, in this work, the objective is to maximize the net thermal power provided by the plant while fulfilling the constraints and minimizing the control effort.

The solution to the optimisation problem is given by the set of optimal flow rates over the control horizon $\mathrm{N_u}$. At each instant $t = k_\mathrm{c}\Delta\mathrm{T_c}$ the cost function J calculated at k_c is minimized for the whole prediction horizon, where $\Delta\mathrm{T_c}$ is the controller time step.

$$\begin{aligned} &\min_{\boldsymbol{q}_\mathrm{k}(k_\mathrm{c})} \sum_{k=k_\mathrm{c}}^{k_\mathrm{c}+\mathrm{N_p}} J(k \mid k_\mathrm{c}) \\ &\text{s.t. } q^{\min} < q(k \mid k_\mathrm{c}) < q^{\max} \;\forall\; k = k_\mathrm{c},\ldots,k_\mathrm{c}+\mathrm{N_p} \end{aligned} \tag{3.1}$$

where vector $\boldsymbol{q}_\mathrm{t}(k_\mathrm{c}) = [q(k_\mathrm{c} \mid k_\mathrm{c}), q(k_\mathrm{c}+1 \mid k_\mathrm{c}),\ldots, q(k_\mathrm{c}+N_\mathrm{u}-1 \mid k_\mathrm{c})]$ contains the flow-rate values throughout the control horizon calculated at $t = k_\mathrm{c}\Delta\mathrm{T_c}$, $\mathrm{N_p}$ is the prediction horizon, and N_u is the control horizon.

As mentioned above, the objective function includes a term to minimize the net thermal power $W(k_\mathrm{c})$. In addition, soft constraints are added, limiting the temperature. Instead of directly imposing that the temperature does not exceed some limits, slight violations are allowed to facilitate the resolution of the problem. For this purpose, the differences between each limit and the actual temperature are used. These terms are normalized with respect to each limit since, for example, a deviation of 2°C will not be as important if the limit was 5°C (40 % deviation) as if it was 50°C (4 % deviation). The selection of the maximum between the deviations with respect to the two limits and 0 allows to consider only the temperatures above the upper limit and those below the lower limit. The third term of the cost function allows penalizing the increments of the control variable to avoid abrupt changes in the flow rate from one instant to another. ψ and ε are tuning parameters.

$$J(k_\mathrm{c}) = \left(-W(k_\mathrm{c}) + \psi \max\left(\frac{T_\mathrm{N}^\mathrm{f}(k_\mathrm{c}) - T^{\mathrm{f,max}}}{T^{\mathrm{f,max}}}, \frac{T^{\mathrm{f,min}} - T_\mathrm{N}^\mathrm{f}(k_\mathrm{c})}{T^{\mathrm{f,max}}}, 0 \right)^2 + \varepsilon \left(q(k_\mathrm{c}) - q(k_\mathrm{c}-1) \right)^2 \right) \tag{3.2}$$

The net thermal power is computed by the difference between the output and input thermal powers for the collector field, $W^{\mathrm{out}}(k_\mathrm{c})$ and $W^{\mathrm{in}}(k_\mathrm{c})$. Neglecting the losses at the output, they can be approximated by the net power of the loop. If the controller was applied to the whole field, it could be approximated by the sum of the powers of all the loops and only the flow rates and restrictions of the other loops would have to be added to the function.

$$W(k_\mathrm{c}) = W^{\mathrm{out}}(k_\mathrm{c}) - W^{\mathrm{in}}(k_\mathrm{c}) \approx W_{\mathrm{loop}}^{\mathrm{out}}(k_\mathrm{c}) - W_{\mathrm{loop}}^{\mathrm{in}}(k_\mathrm{c}) \tag{3.3}$$

where $W^{\mathrm{out}}(k_\mathrm{c})$ and $W^{\mathrm{in}}(k_\mathrm{c})$ are the output and input thermal powers for the field and $W_{\mathrm{loop}}^{\mathrm{out}}(k_\mathrm{c})$ and $W_{\mathrm{loop}}^{\mathrm{in}}(k_\mathrm{c})$ are the output and input thermal powers of the control loop.

The output and input powers are computed by the following equations:

$$W_{\mathrm{loop}}^{\mathrm{out}}(k_\mathrm{c}) = \rho_\mathrm{N}^\mathrm{f}(k_\mathrm{c}) C_\mathrm{N}^\mathrm{f}(k_\mathrm{c}) q(k_\mathrm{c}) T_\mathrm{N}^\mathrm{f}(k_\mathrm{c}) \tag{3.4}$$

$$W_{\mathrm{loop}}^{\mathrm{in}}(k_\mathrm{c}) = \rho_1^\mathrm{f}(k_\mathrm{c}) C_1^\mathrm{f}(k_\mathrm{c}) q(k_\mathrm{c}) T_1^\mathrm{f}(k_\mathrm{c}) \tag{3.5}$$

The optimization problem is solved using SQP, an iterative nonlinear optimization method. It consists of solving a local subproblem by fitting a quadratic function to the objective function at the point considered,

replicating Newton's method for constrained optimization. A new point is obtained and a local QP is solved again until the optimal solution is reached. It is used in problems where the objective function is twice continuously derivable. Since this problem is highly nonconvex, it is quite possible to reach local minima. Therefore, it is necessary to run the optimization algorithm several times using different initial points. As in [34], four initial points have been taken:

- The profile obtained in the previous time step shifted.
- The lower bound.
- The upper bound.
- A random point.

As explained above, the obstacle with MPC is its computational requirements. It needs a long time to compute the optimal solution, which complicates its real-time application in large solar plants and with long control horizons. To solve this problem, this work proposes to use artificial neural networks that learn offline the controller output and apply it in real-time.

The last measured value of the outlet temperature $T^{\text{out}}(k_{\text{c}})$ is used for the entire prediction horizon and the cost function parameters ψ and ε are set to 45 and 3, respectively. The prediction and control horizons are 12 min and 10 min ($\text{N}_{\text{p}} = 12$, $N_{\text{u}} = 10$). The control step time is $\Delta T_{\text{c}} = 1$ min. The integration time of the model is $\Delta T_{\text{m,c}} = 3$ s and the controller uses $\Delta\text{l,c} = 6$ m instead of the original length $\Delta\text{l} = 1$ m in order to fasten the time needed for the simulation. In addition, losses are pre-calculated for a set of temperature and flow rate values.

3.1.1 Parameters

The considered collector field is ACUREX, which was located at the PSA, Spain. This solar plant has been used on numerous occasions in the literature. This work has been carried out based on the simulation of a loop of this plant.

The plant is formed by ten loops of single-axis parabolic-trough collectors aligned East-West. Each loop has a length of 174 m and has been divided into segments of $\Delta\text{l} = 1$ m, so it has $\text{N} = 174$ segments (although the MPC controller considers $\Delta\text{l}_{\text{c}} = 6$ m). Each loop has twelve collectors placed in series. The collectors are made up of an active part –the part that receives solar radiation– and an passive part –the part that does not receive solar radiation. The passive part corresponds to the segments (37,42), (79,96) and (133,138) and is 30 m long, while the length of the active part is 144 m, as shown in figure 3.1.

Therminol VP-1 has been used as HTF, which has a density of $\rho_i^{\text{f}}(k)$ and specific heat capacity of $C_i^{\text{f}}(k)$. The following equations allow us to obtain these values for each segment i and instant k:

$$\rho_i^{\text{f}}(k) = 903 - 0.672 T_i^{\text{f}}(k) \tag{3.6}$$

$$C_i^{\text{f}}(k) = 1820 - 3.478 T_i^{\text{f}}(k) \tag{3.7}$$

In addition, it is necessary to take into account the coefficient of transmission metal-fluid $H_i^t(k)$ and the coefficient of thermal losses $H_i^{\text{l}}(k)$, which are calculated by the following two equations:

$$H_i^t(k) = q_k^{0.8}\left(2.17\cdot 10^6 - 5.01\cdot 10^4 T_i^{\text{f}}(k) + 4.53\cdot 10^2 T_i^{\text{f}}(k)^2 - 1.64 T_i^{\text{f}}(k)^3 + 2.1\cdot 10^{-3} T_i^{\text{f}}(k)^4\right) \tag{3.8}$$

$$H_i^{\text{l}}(k) = 0.00249\left(T_i^{\text{f}}(k) - T^{\text{a}}(k)\right) - 0.06133 \tag{3.9}$$

Table 3.1 gathers the values of the parameters, both calculated or obtained from the plant and selected for simulation.

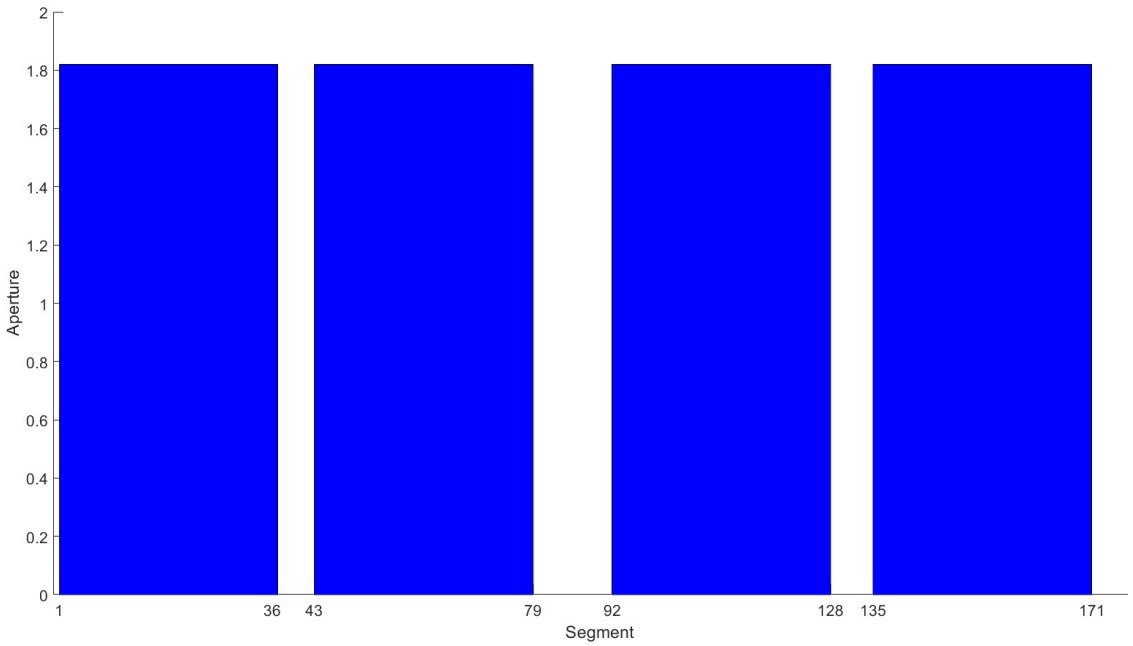

Figure 3.1 Arrangement of the segments in the loop.

Table 3.1 Parameters of ACUREX.

Symbol	Value	Unit
ΔT	0.5	s
ρ^{m}	7800	kg/m^3
C^{m}	550	J/(kg $^\circ$C)
D^{m}	0.031	m
D^{f}	0.0254	m
A^{m}	$2.48 \cdot 10^{-4}$	m^2
A^{f}	$7.55 \cdot 10^{-4}$	m^2

3.1.2 Constraints and Boundaries

Some constraints are required to be imposed for the correct functioning of the plant. A boundary equation is defined for the temperature of the first segment of the loop, assuming that it is equal to the inlet temperature (which acts as a disturbance):

$$T_1^{\mathrm{f}}(k) = T^{\mathrm{in}}(k) \tag{3.10}$$

Also, it is necessary to impose limits to the flow rate based on the minimum Reynolds number required to guarantee turbulent flow in the pipes and the maximum pressure drop, with $q^{\min}$ and $q^{\max}$ meaning the minimum and maximum flow rate, respectively:

$$q^{\min} \leq q(k) \leq q^{\max} \tag{3.11}$$

The outlet temperature of the HTF must be constrained too within its operational limits:

$$T^{\mathrm{f,min}} \leq T_N^{\mathrm{f}}(k) \leq T^{\mathrm{f,max}} \tag{3.12}$$

The values of the constraints for the ACUREX plant are gathered in table 3.2.

Table 3.2 Constraints of ACUREX..

Symbol	Value	Unit
$q^{\min}$	0.2	l/s
$q^{\max}$	1.5	l/s
$T^{\mathrm{f,min}}$	220	°C
$T^{\mathrm{f,max}}$	300	°C

3.2 Neural Network Controller

This section describes the implementation of the neural network controller based on MPC, which consists of training a neural network with the inputs and outputs of an MPC controller. The ultimate goal is to replace the original controller with a real-time usable ANN controller that provides near-optimal behavior, with low constraint violations and similar power obtained.

Based on the plant model, at each time-step, the MPC controller calculates the optimal value of the flow rate given a series of inputs: the flow rate applied at the previous instant, the inlet temperature, the outlet temperature, the ambient temperature, and the fluid temperature, metal temperature and irradiance in each segment predicted for the entire prediction horizon. The flow rate is, therefore, a function $f(\cdot)$ of all these inputs. Neural networks are good function approximators. The aim of this work is to find a function that approximates the output of the MPC controller without the need to solve a real-time optimization problem.

$$q(k) = f\left(q(k-1), T^{\mathrm{in}}(k), T^{\mathrm{out}}(k), T^{\mathrm{a}}(k), T_i^{\mathrm{f}}(k), T_i^{\mathrm{m}}(k), I_i(k), \ldots, I_i(k+\mathrm{N_p}-1)\right) \tag{3.13}$$

The inputs to the above equation can be collected in a vector to be used as inputs to an ANN. Following the notation of section 4, such a vector is $\boldsymbol{z}_1^{(1)}(k) = [q(k-1), T^{\mathrm{in}}(k), T^{\mathrm{out}}(k), T^{\mathrm{a}}(k), T_i^{\mathrm{f}}(k), T_i^{\mathrm{m}}(k), I_i(k), \ldots, I_i(k+\mathrm{N_p}-1)]$. The neural network will only be used to obtain the control signal to be applied at each instant, since the future flow rates obtained with the MPC controller were discarded.

$$\hat{q}(k) = f_{\mathrm{NN}}\left(\boldsymbol{z}_1^{(1)}(k)\right) \tag{3.14}$$

This work is composed of two stages. First, an offline training of the ANN (figure 3.2) is performed using the outputs from the MPC controller applied to a simulator of the plant with a given profile of irradiance. Once trained, the ANN is implemented online to substitute the MPC controller in real-time (figure 3.3).

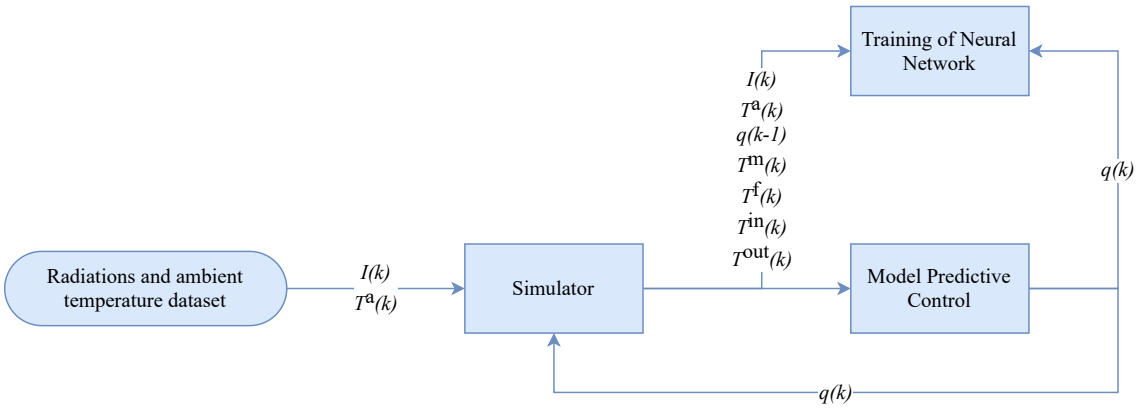

Figure 3.2 Offline training of the control algorithm.

The MPC controller is applied to the ACUREX plant simulator to obtain the dataset that will be used to train the neural networks. Specifically, a dataset from 30 days of simulation has been used, where the

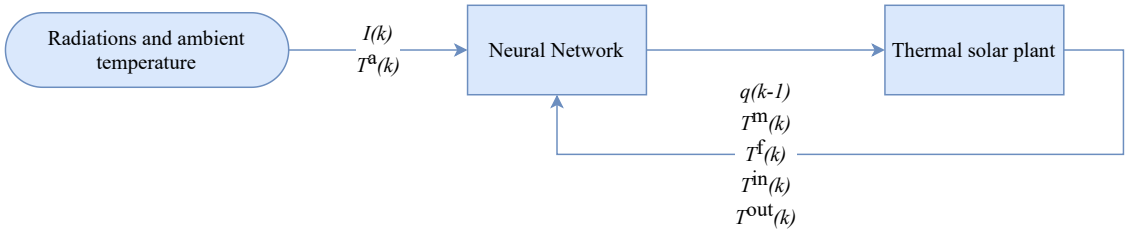

Figure 3.3 Online implementation of the control algorithm.

perturbations to the system have been modified. This dataset contains a total of 16867 samples and has been divided into a training set (70%), a validation set (15%) and a test set (15%), shuffled and randomly selected.

Once the three subsets are obtained, the data are scaled in the range $[-1,1]$. This is done to avoid inappropriate saturation of the data due to the use of activation functions. Also, if very high values were used in the input, the weight vectors would have to be very small. This would make the gradients too small and the training would be too slow [92, 50].

As explained above, the neural network selection process is performed by trial and error. Several neural networks with different sizes and number of neurons have been trained. The criterion for stopping the training was that a mean squared error under 10^{-9} was reached. In addition, two different activation functions were used: a linear function was used in the output layer to avoid modifying the results and hyperbolic tangent functions were used in the rest of the layers, restricting the outputs of intermediate neurons.

Finally, it is worth noting the predictive and interpolation capability of neural networks from a dataset, which essentially eliminates the need for external online estimation, as was done in the MPC. Therefore, different neural networks have been implemented by changing the number of irradiance predictions and the number of temperature sensors (or estimations of their values). According to this, eight different cases have been investigated:

- Case 1: Temperatures and irradiance every six segments (410 inputs).
- Case 2: Irradiance every six segments and temperatures at the center of each collector (360 inputs).
- Case 3: Irradiance every six segments, predictions at instants 1,4,8 and 12, and temperatures at the center of each collector (128 inputs).
- Case 4: Irradiance every six segments, predictions at instants 1 and 12, and temperatures at the center of each collector (70 inputs).
- Case 5: Irradiance every six segments, prediction horizon of 6, and temperatures at the center of each collector (186 inputs).
- Case 6: Irradiance every six segments, prediction horizon of 3, and temperatures at the center of each collector (99 inputs).
- Case 7: Irradiance every six segments, prediction horizon of 1, and temperatures at the center of each collector (41 inputs).
- Case 8: Temperature and irradiance in the first collector, without inlet and outlet temperatures, and prediction horizon of 1 (5 inputs).

One thing that characterizes neural networks and is an advantage over MPC is that a much smoother output is obtained, although this is at the expense of getting slight constraints violations. In order to compare this characteristic between each ANN and the MPC controller, a new metric is added, which we define as

accumulated absolute control increment (AACI). It is calculated as the accumulated absolute difference between the value of the flow rate in an instant and the next one.

$$\text{AACI} = \sum_{k} |q(k) - q(k-1)| \tag{3.15}$$

Regarding constraint violations, it is also necessary to introduce another metric which we define as mean squared constraint violation (MSCV). It measures the difference between the outlet temperature T^{out} and its limit when constraint violations occur.

$$\text{MSCV} = \frac{1}{\text{n}_\text{s}} \sum_{k\Delta\text{T}=t_1}^{k\Delta\text{T}=t_2} \left(\max \left(T^{\text{f,min}} - T^{\text{out}}(k), T^{\text{out}}(k) - T^{\text{f,max}}, 0 \right)^2 \right) \tag{3.16}$$

where n_s is the number of samples between instants t_1 and t_2.

3.3 Input data

In solar collectors, the disturbances are the ambient temperature and the effective DNI. The ambient temperature is taken to be constant and equal to 25 °C. A 30-days DNI profile has been used, in which different synthetic clouds have been included to obtain a heterogeneous dataset. Figure 3.4 shows the first day of the DNI used for creating the dataset, and the following figures show the profiles used for validating the results.

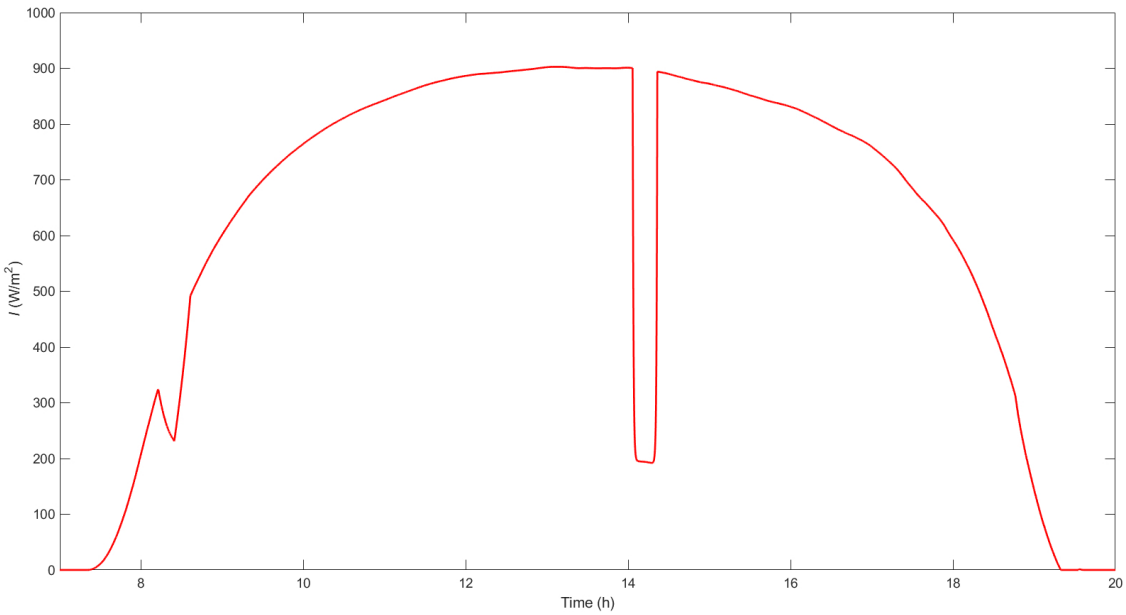

Figure 3.4 First day of the DNI profile used for obtaining the dataset.

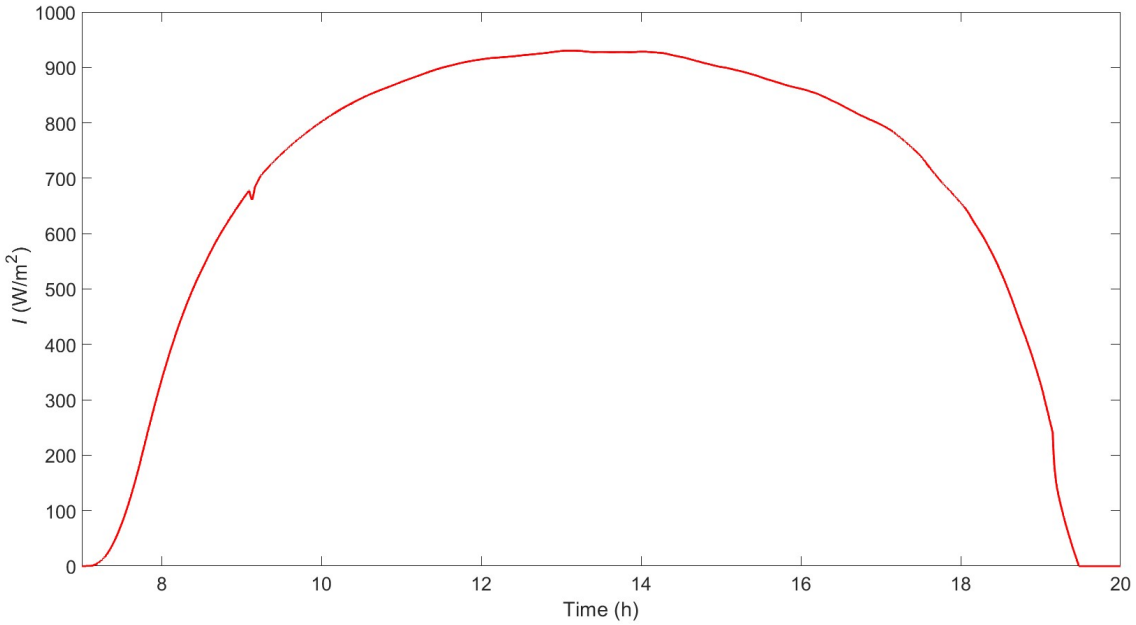

Figure 3.5 DNI profile used for validation 1.

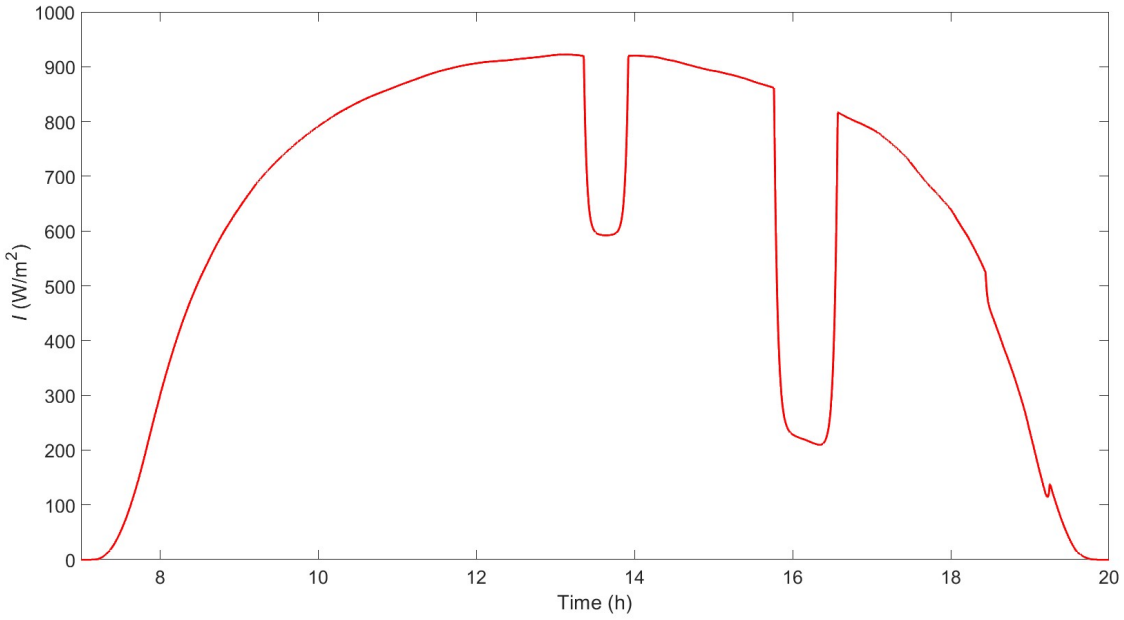

Figure 3.6 DNI profile used for validation 2.

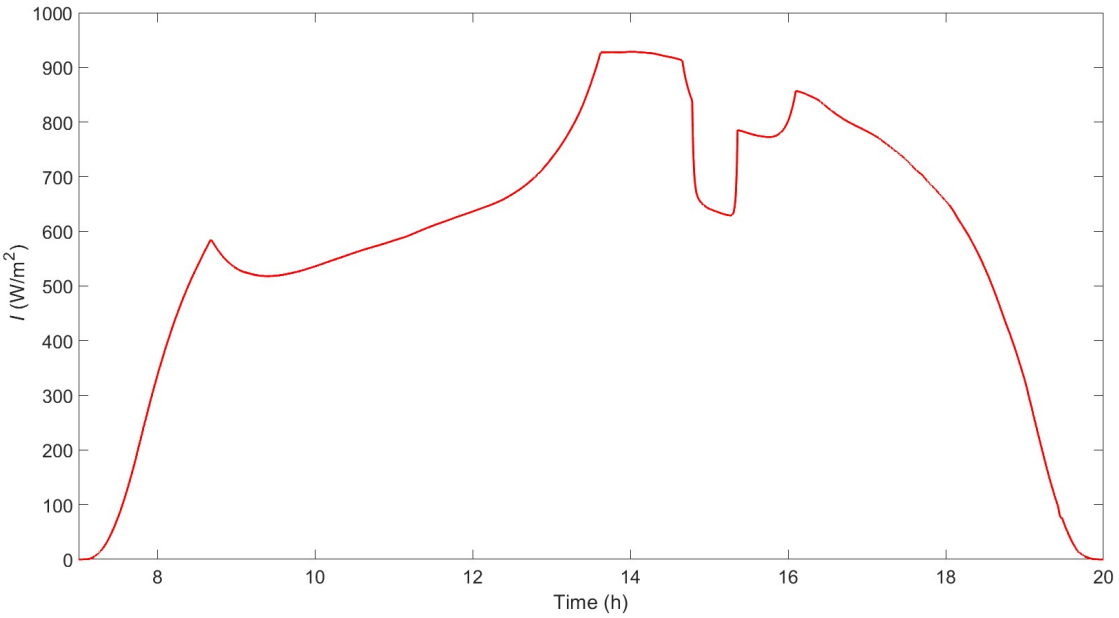

Figure 3.7 DNI profile used for validation 3.

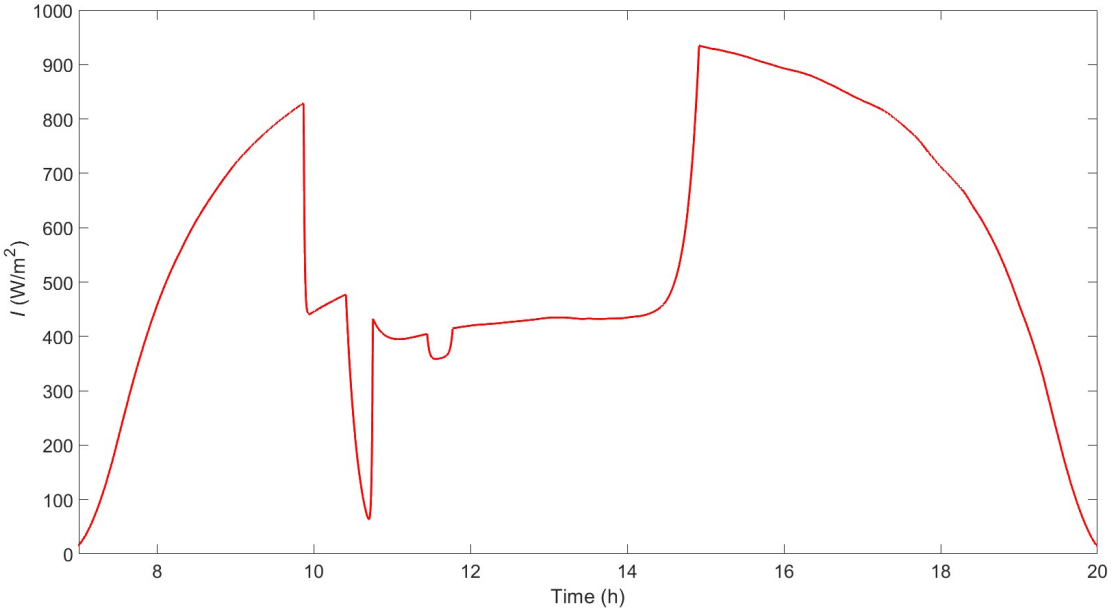

Figure 3.8 DNI profile used for validation 4.

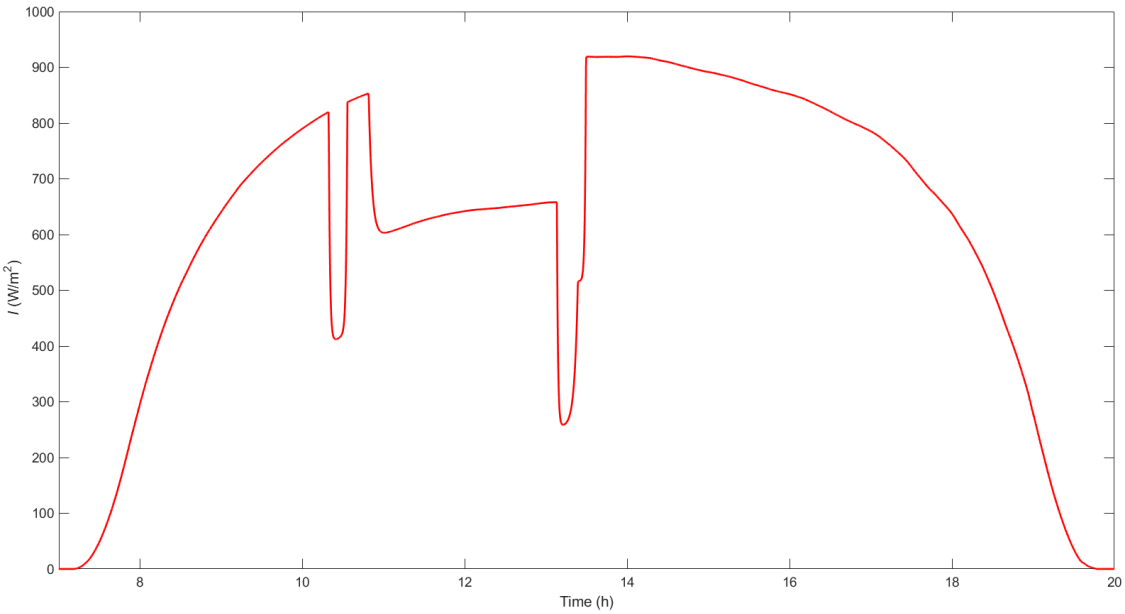

Figure 3.9 DNI profile used for validation 5.

4 Simulation Results

This chapter presents the simulation results obtained by applying to the plant the different neural networks for case 1–temperatures and irradiance every six segments, see section 3.2–and one network architecture is selected. Using this architecture, new neural networks are applied to the rest of the cases and trained, taking different points to measure temperatures and irradiances. Finally, the neural networks are tested with new validation profiles.

4.1 Neural Network Selection

First, a neural network is selected using the variables from case 1. Temperatures and irradiances are taken every six segments, and the entire prediction horizon (12 minutes) is used. Different neural networks have been trained with the dataset described in section 3.2 (corresponding to the irradiance profile of 30 days, of which the first day was shown in a graph).

After obtaining the neural networks, the Pearson correlation coefficient R was calculated for the three subsets (training, validation and test). This was done in an open-loop, comparing the flow rate obtained with MPC and the flow rate that the neural network approximates at each instant, using the MPC as the previous flow rate input. Then, the neural networks have been applied to control the plant in a closed-loop, disconnecting the MPC controller and using as feedback the flow rate provided by the ANN itself in the previous instant. From there, the AACI and the MSCV (defined in section 3.2), as well as the average power obtained, were calculated for the first day of the dataset. All this is represented in the table 4.1. Note that during startup and shutdown of the plant, at night, the dynamics of the process change a lot and the controller is not applied. For this reason, the MSCV is calculated between the hours 8:20 and 19:00.

Table 4.1 Results of the controllers used for case 1 with the first day of the dataset.

Id	Neurons	R (train)	R (validation)	R (test)	Mean power	AACI	MSCV
MPC	-	-	-	-	65.6607 kW	7.9309 l/s	0.0445
1	15	0.99720	0.99690	0.99651	65.6638 kW	2.8189 l/s	0.4706
2	15-5	0.99793	0.99678	0.99647	65.6515 kW	2.7983 l/s	0.1016
3	**15-10**	**0.99791**	**0.99636**	**0.99679**	**65.6580 kW**	**2.9247 l/s**	**0.0740**
4	20-10	0.99778	0.99664	0.99621	65.6508 kW	2.9479 l/s	0.1760
5	15-10-5	0.99771	0.99699	0.99686	65.6557 kW	2.8435 l/s	0.5388
6	15-10-10	0.99739	0.99685	0.99685	65.6619 kW	2.9479 l/s	0.5182

Considerating the AACI index, all neural networks (whether shallow networks or deep networks) provide much smoother outputs than the MPC controller. Moreover, the powers obtained are all of the same order.

Although the smallest neural network provides the highest power–even surpassing that of the MPC controller– it is because the constraint violation is higher. Analyzing the results, the third ANN (two hidden layers, with 15 neurons in the first one and 10 neurons in the second one), in which the constraint violation is low, is selected. A detail of the results during a passing cloud is shown in figure 4.1. It should be noted that the curves corresponding to the artificial neural networks are much smoother than those obtained with MPC.

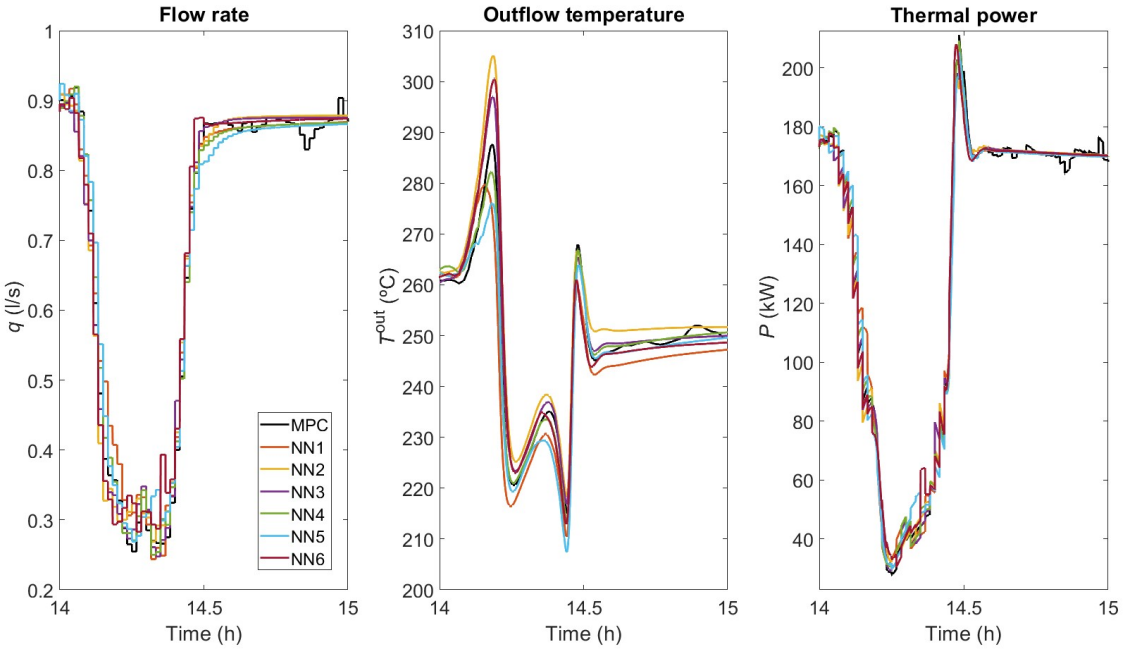

Figure 4.1 Evolution of the flow rate, the outlet temperature and the thermal power during a passing cloud for the first day of the dataset. Case 1.

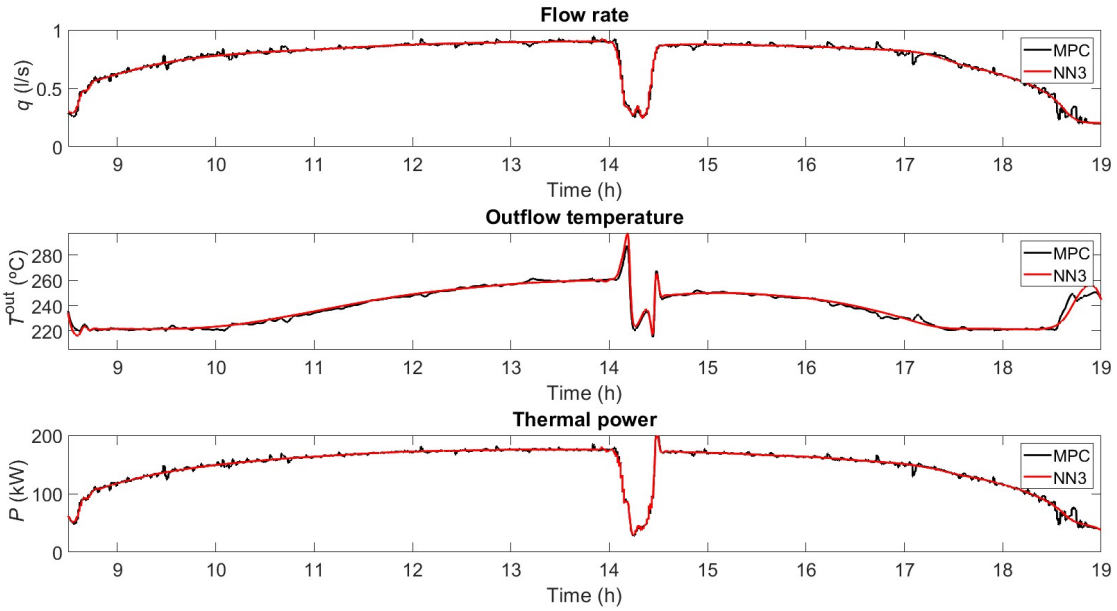

Figure 4.2 Evolution of the flow rate, the outlet temperature and the thermal power for the first day of the dataset. Case 1.

Figure 4.2 provides a clearer visualization of the results using the third ANN. It shows the evolution of the flow rate, the outlet temperature and the thermal power during the selected day of the dataset. Figure 4.3 shows the architecture of the selected neural network.

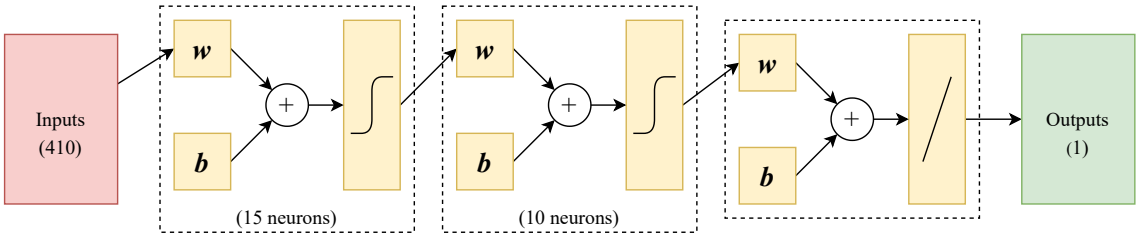

Figure 4.3 Diagram of the selected artificial neural network for case 1.

The main characteristic of neural networks of interest in this work is the speed of computation. The times that each controller needed to calculate the flow rate to be sent to the plant have been compared, obtaining the results shown in figure 4.4. It can be seen that the MPC controller is much slower than the ANN controller at all times. The mean time for the MPC controller is 2.3929 s with a standard deviation of 1.1418 s. In the case of the ANN, the mean time is 0.07 s and the standard deviation is 0.0087 s.

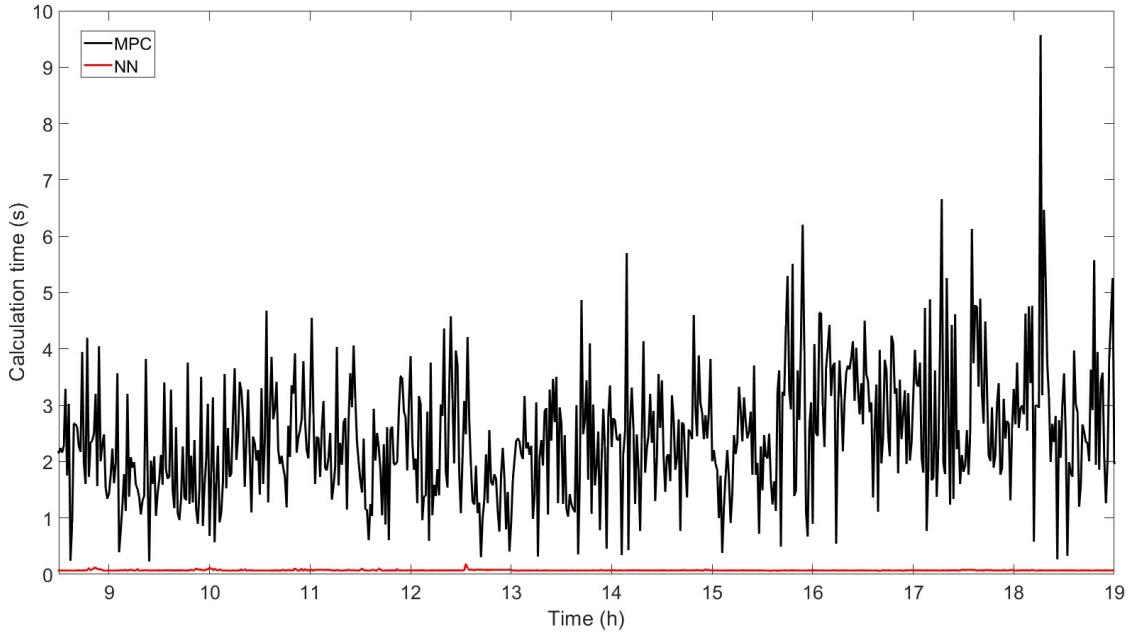

Figure 4.4 Computation times of MPC and the ANN with the first day of the dataset. Case 1.

4.2 Different Number of Inputs

After selecting the internal architecture of the neural network, ANNs are trained for the rest of the cases (cases 2 to 8), using a different number of inputs for each one, and keeping the rest of the parameters the same. Therefore, all neural networks will have 2 hidden layers, with 15 neurons in the first one and 10 in the second one. The only difference between the neural networks corresponding to each case is the input layer.

Table 4.2 shows the results obtained for the neural networks trained for cases 2 to 8 using the third neural network. It also shows the Pearson correlation coefficient for the training, validation and test subsets, the

mean power obtained in a one-day simulation, the accumulated absolute difference between the value of the flow rate in two consecutive instants and the MSCV.

Table 4.2 Results of the controllers used for cases 2 to 8 with the first day of the training data (15-10 neurons).

Case	R (train)	R (validation)	R (test)	Mean power	AACI	T^{out} MSCV
MPC	-	-	-	65.6607 kW	7.9309 l/s	0.0445
1	0.99791	0.99636	0.99679	65.6580 kW	2.9247 l/s	0.0740
2	0.99760	0.99719	0.99671	65.6581 kW	2.9096 l/s	0.1561
3	0.99768	0.99708	0.99704	65.6535 kW	2.9483 l/s	0.2810
4	0.99748	0.99647	0.99692	65.6629 kW	2.7363 l/s	0.7317
5	0.99772	0.99668	0.99677	65.6548 kW	2.7612 l/s	0.2239
6	0.99694	0.99663	0.99670	65.6599 kW	2.8096 l/s	0.0767
7	0.99686	0.99575	0.99649	65.6707 kW	2.7097 l/s	1.3840
8	0.99547	0.99542	0.99426	65.5358 kW	2.7726 l/s	26.0222

It can be extracted that, even eliminating some sensors and predictions, the artificial neural networks are able to learn by themselves thanks to the excellent behavior of the MPC controller used for training, which takes into account the predictions and future estimations of the whole loop. They are internally learning those predictions. The last two cases, where no predictions are used, are the worst performers, with large constraint violations. However, the average power is close to that obtained with the MPC controller and the flow rate is still smoother.

Figure 4.5 shows the flow rate, outlet temperature and thermal power obtained with the neural networks of the 8 cases with a passing cloud, compared with MPC. The last case, in which only 5 inputs were used, obtains much more constraints violations and a more retarded response. The rest of the cases are more similar between them. Although the constraint violations in case 7 (with an input size of 10% the size of case 1) are more significant than in the rest, they are still relatively low, and the number of inputs is quite slow, so this case will be selected for future tests. Figure 4.6 shows the results for the whole day with case 7.

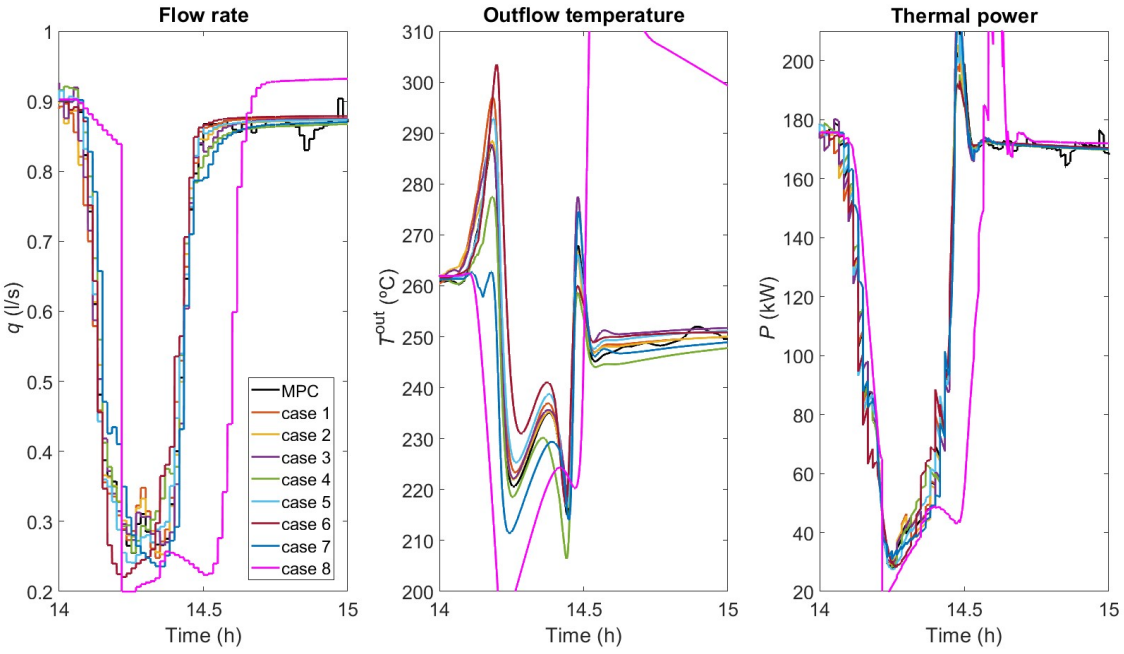

Figure 4.5 Evolution of the flow rate, the outlet temperature and the thermal power during a passing cloud for one day of the dataset. Cases 1 to 8.

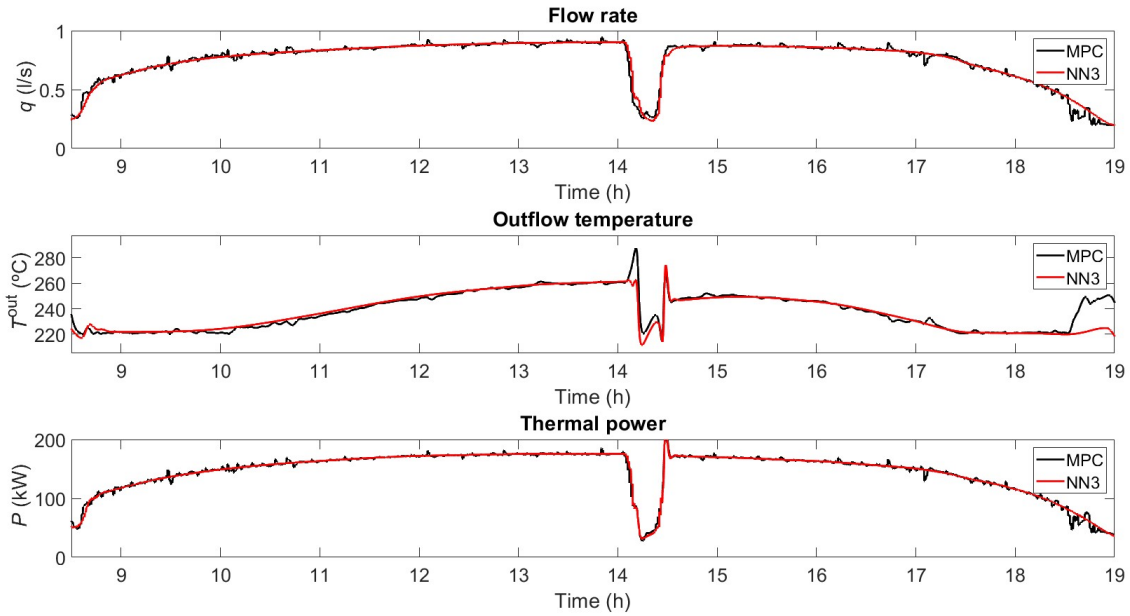

Figure 4.6 Evolution of the flow rate, the outlet temperature and the thermal power for one day of the dataset. Case 7.

Other neural networks and tests have been made for analytical purposes to visualize how the neural network responds to different prediction horizons. Figure 4.7 shows the MSCV and the AACI between two instants and their evolution when the prediction horizon decreases. It is worth noting that the neural networks are capable of replicating the behavior of MPC despite the decrease in the number of inputs, except for the case with only one prediction step. A slight decrease in the AACI is also visible as the prediction horizon decreases.

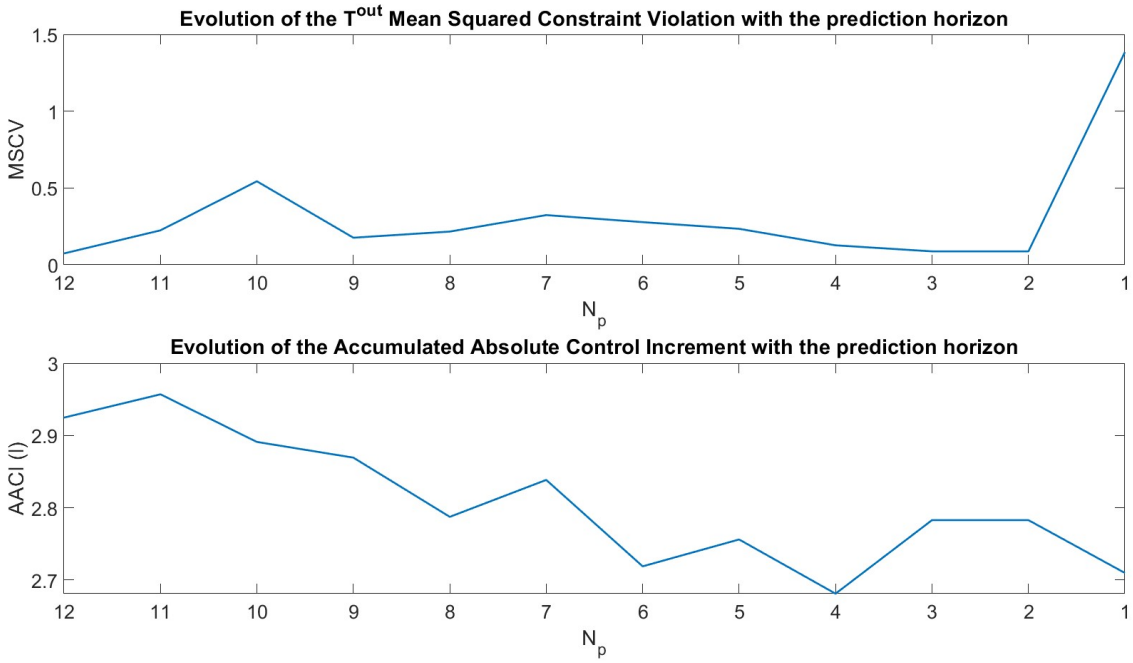

Figure 4.7 Evolution of the MSCV and AACI with different prediction horizons for the first day of the profile used for training.

4.3 Results Validation

To validate the results, five different DNI profiles have been created (represented in section 3.3), each one corresponding to one day with different passing clouds. Then, several simulations have been performed using those profiles and applying the selected controllers: the MPC controller and the different neural networks of cases 1 to 8, with the same parameters but changing the number of inputs related to irradiance and temperature.

4.3.1 First Profile

The first irradiance profile corresponds to a cloudless day, where the curves are very smooth and the perturbations are more predictable. Figure 4.8 shows the results obtained with the MPC controller and the neural networks for all cases (1 to 8). Again, the temperatures obtained with case 8 exceed the limits unacceptably, with a maximum temperature of 317.3240 °C and a minimum of 199.6973 °C between hours 8:20 and 19:00.

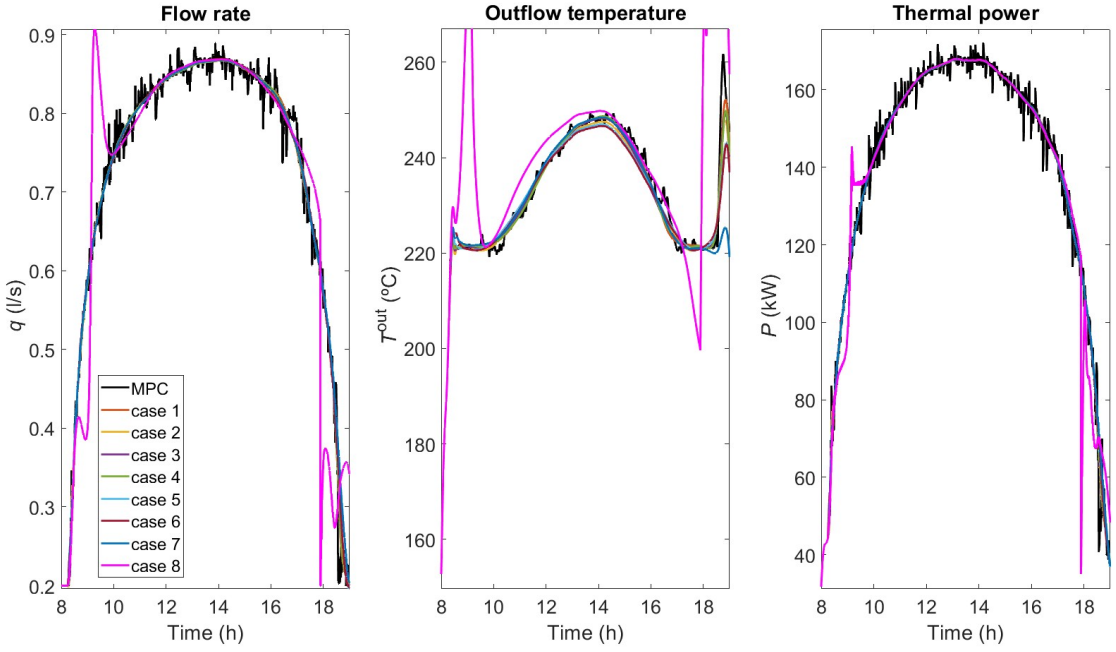

Figure 4.8 Evolution of the flow rate, the outlet temperature and the thermal power during a passing cloud for the first validation profile. Cases 1 to 8.

The mean power, AACI and MSCV are gathered in table 4.3. In all cases, the constraint violations (MSCV) increase, providing a slight increment in the mean power, except for case 8. Note that this neural network has only five inputs and no predictions are made. The overall performance for all cases is good and the AACI reduction is maintained.

Table 4.3 Results of the controllers used with the first validation profile.

Neurons	Mean power	AACI	MSCV
MPC	64.2745 kW	7.1727 l/s	0.2684
Case 1	64.2784 kW	1.3453 l/s	0.3114
Case 2	64.2819 kW	1.3472 l/s	0.2694
Case 3	64.2806 kW	1.3492 l/s	0.3180
Case 4	64.2796 kW	1.3425 l/s	0.2996
Case 5	64.2840 kW	1.3602 l/s	0.3392
Case 6	64.2855 kW	1.3470 l/s	0.4184
Case 7	64.2876 kW	1.3440 l/s	0.8203
Case 8	64.1954 kW	2.2345 l/s	13.1414

Figure 4.9 represents the results for MPC and the ANNs of cases 1 and 7. These two cases have been selected, as they are the ones with more and less prediction horizon, not taking into account case 8. Both neural networks show similar results and approximate the outputs closely with MPC until 18:30, where case 7 uses a slightly greater flow rate, giving a lower temperature. This difference is not significant because it corresponds to the evening, where irradiance is low and the plant is powering off.

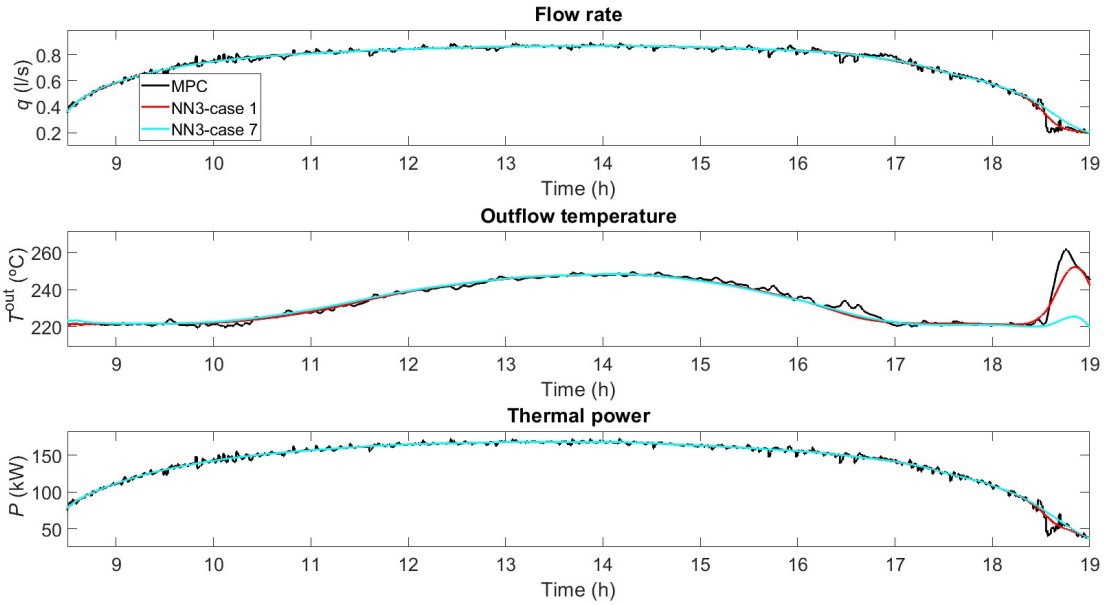

Figure 4.9 Evolution of the flow rate, the outlet temperature and the thermal power for the first validation profile. Cases 1 and 7.

4.3.2 Second Profile

The second validation profile corresponds to a day with two passing clouds of moderate dimensions. The results for neural networks 1 to 7 are represented in figure 4.10. Case 8 is not represented as it destabilizes from hour 16:30 and obtains a maximum temperature of 570.4656 °C and a minimum of 198.2294 °C. This verifies that the neural network of case 8 must be discarded. With this profile, the neural network of case 7 gives a maximum temperature of 322.1846 °C after the second cloud –of great dimension and duration–, making its use also unviable for big clouds, although it is noticeable that the final obtained power does not change exorbitantly.

These results are also gathered in table 4.4, where the mean power is approximately the same for the MPC controller and the neural networks of cases 1 to 6. The AACI and MSCV show small ups and downs, but all

non-discarded neural networks (1 to 6) agree on a large improvement in smoothness at the cost of a slight constraint violation.

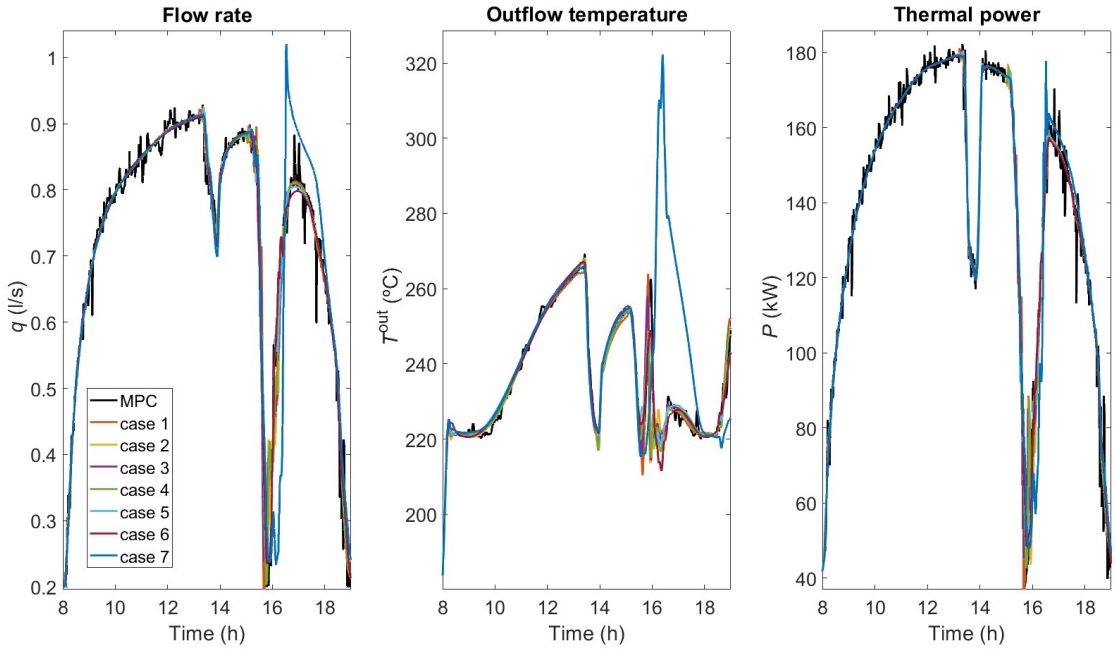

Figure 4.10 Evolution of the flow rate, the outlet temperature and the thermal power during a passing cloud for the second validation profile. Cases 1 to 7.

For better visualization of the results obtained, figure 4.11 shows the flow rate, outlet temperature and thermal power obtained with MPC and neural networks 1 and 7, where the temperature increase in case 7, due to a worse approximation of the flow rate, can be seen in more detail. Although the neural network of case 1 presents some difficulties to cope with the second cloud, this does not prevent a good approximation of the power without the need to increase the temperature deviation considerably.

Table 4.4 Results of the controllers used with the second validation profile.

Neurons	Mean power	AACI	MSCV
MPC	66.0270 kW	8.0466 l/s	$3.1234 \cdot 10^{-5}$
Case 1	66.0303 kW	3.5807 l/s	0.5395
Case 2	66.0271 kW	3.5370 l/s	0.1700
Case 3	66.0248 kW	3.2665 l/s	0.2093
Case 4	66.0292 kW	3.2773 l/s	0.2071
Case 5	66.0265 kW	3.1172 l/s	0.0200
Case 6	66.0270 kW	3.0346 l/s	0.8616
Case 7	65.9811 kW	3.5878 l/s	4.0104
Case 8	58.8387 kW	2.8028 l/s	$1.0843 \cdot 10^{4}$

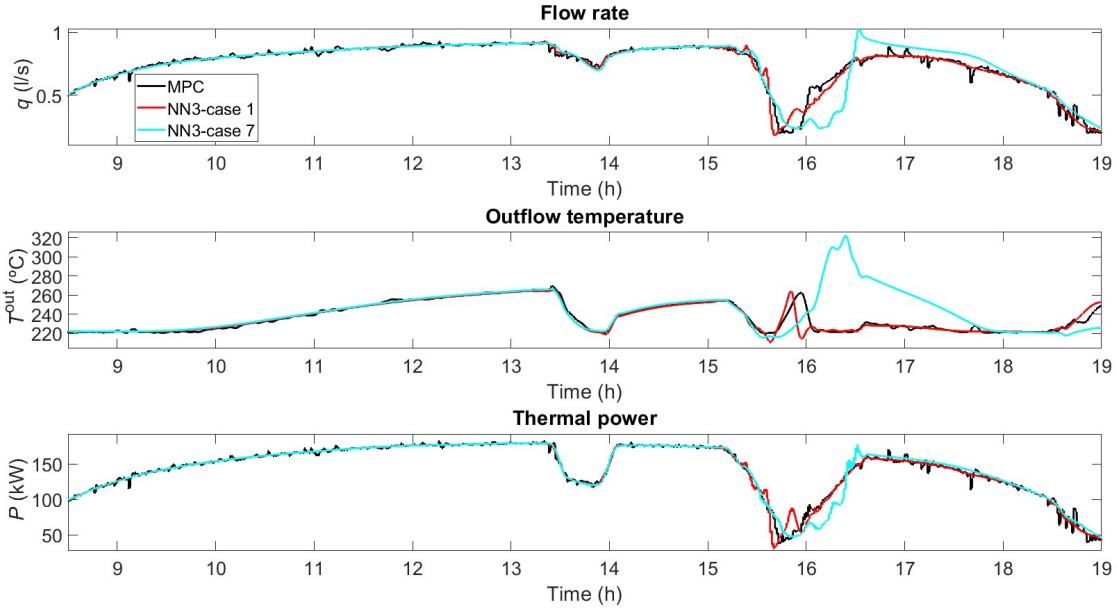

Figure 4.11 Evolution of the flow rate, the outlet temperature and the thermal power for the second validation profile. Cases 1 and 7.

4.3.3 Third Profile

The third DNI profile corresponds to a cloudy day, with light but steady clouds between 8:30 and 14:00 and smaller clouds between 15:00 and 16:00. Once again, neural networks of cases 1 to 7 are able to follow the MPC controller (figure 4.12) and the neural network of case 8 gives temperatures between 199.6174 °C and 297.7527 °C, which is not as large a violation as on other occasions.

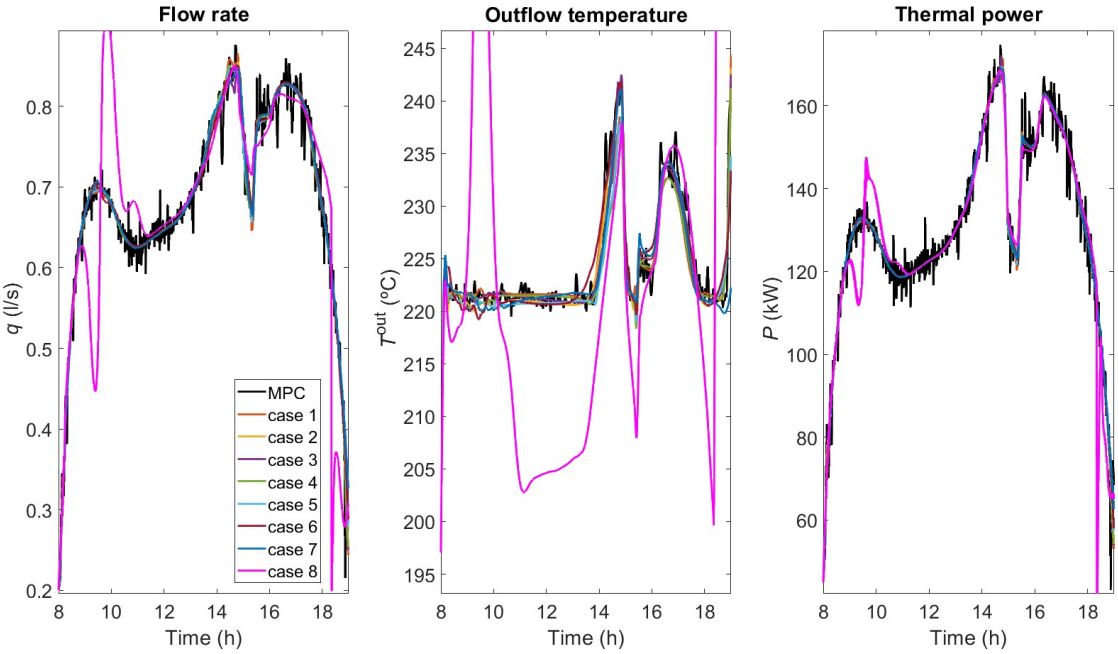

Figure 4.12 Evolution of the flow rate, the outlet temperature and the thermal power during a passing cloud for the third validation profile. Cases 1 to 8.

Moreover, as can be seen in table 4.5, case 3 obtains no constraint violations. This surpasses the results obtained with MPC at the detriment of only a loss of 1.5 W on average, a negligible value considering the size of the plant and the randomness associated with each controller run. In fact, all cases show good values of MSCV, being of the order of 10^{-4} with cases 1 and 7. Although the minimum constraint violation with case 8 was not as high as with other profiles, the total MSCV is 70.9767, which is not acceptable.

Table 4.5 Results of the controllers used with the third validation profile.

Neurons	Mean power	AACI	MSCV
MPC	61.8890 kW	8.9987 l/s	$3.3044 \cdot 10^{-5}$
Case 1	61.8868 kW	1.9082 l/s	$4.4934 \cdot 10^{-4}$
Case 2	61.8876 kW	1.8002 l/s	0.0047
Case 3	61.8875 kW	1.8246 l/s	0
Case 4	61.8892 kW	1.8058 l/s	0.0093
Case 5	61.8912 kW	1.8234 l/s	0.0022
Case 6	61.8893 kW	1.7970 l/s	0.0062
Case 7	61.9003 kW	1.7914 l/s	$3.0552 \cdot 10^{-4}$
Case 8	61.8239 kW	2.8947 l/s	70.9767

Figure 4.13 shows the good approximations from cases 1 and 7, compared to the MPC controller. Both neural networks are similar except for the last minutes of the day, where case 7 even surpasses the thermal power obtained with MPC. During the day, the three controllers perform similarly, with not very big differences in the moments when the cloud passes.

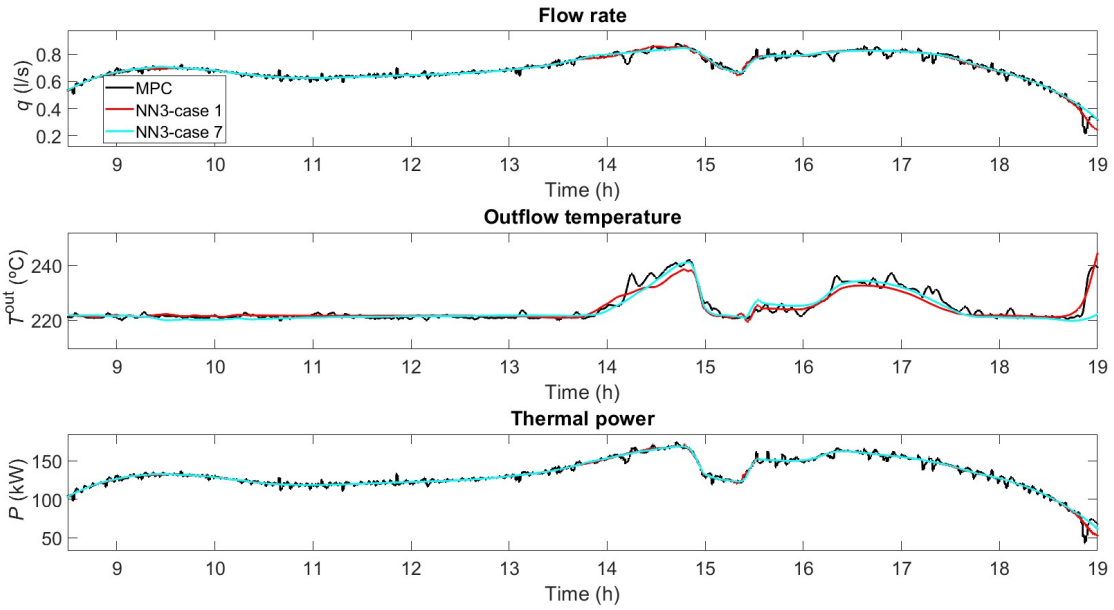

Figure 4.13 Evolution of the flow rate, the outlet temperature and the thermal power for the third validation profile. Cases 1 and 7.

4.3.4 Fourth Profile

The fourth DNI profile corresponds to a great cloud between 10:00 and 15:00, with greater intensity at 10:30. This is the most challenging day for the controllers, as shown in figure 4.14, where the MPC controller presents an abrupt output and each neural network behave differently during the beginning and the end of the cloud. The neural network of case 8 destabilizes the plant from hour 10:00, so it has not been represented.

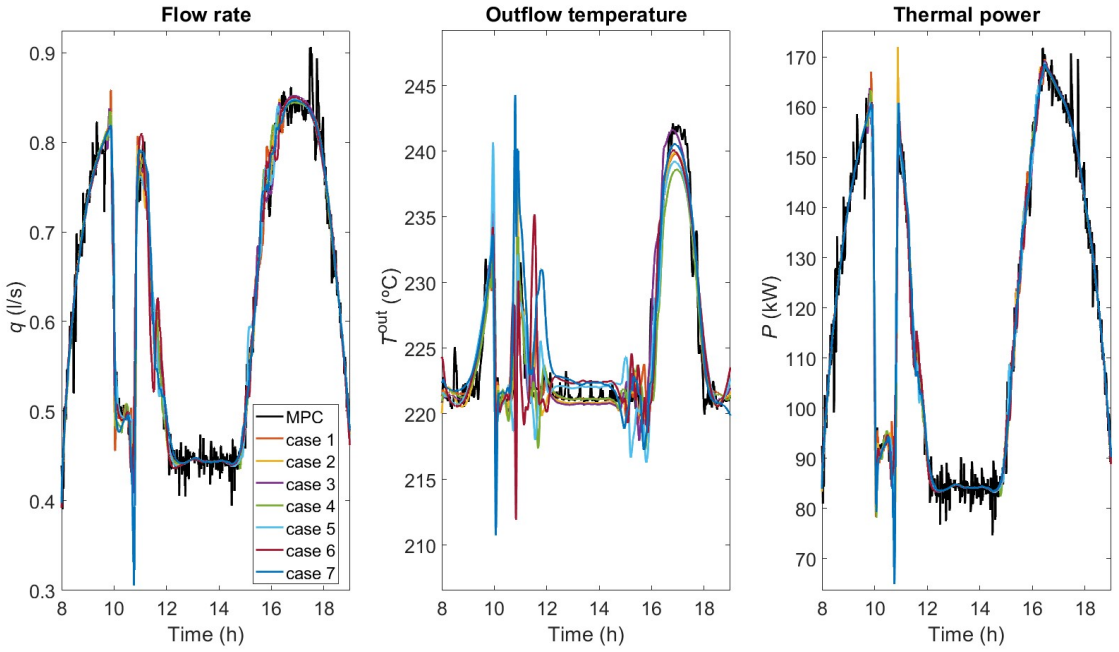

Figure 4.14 Evolution of the flow rate, the outlet temperature and the thermal power during a passing cloud for the fourth validation profile. Cases 1 to 7.

These results are accompanied by table 4.6, where it can be seen that all neural networks obtain an MSCV under 1 –greater for the neural networks with smaller input layers– and a significant diminution of the AACI with respect to the MPC controller. The obtained mean power is similar for all the controllers in the table.

Table 4.6 Results of the controllers used with the fourth validation profile.

Neurons	Mean power	AACI	MSCV
MPC	58.0140 kW	9.1523 l/s	$4.9748 \cdot 10^{-5}$
Case 1	58.0123 kW	3.4988 l/s	0.0256
Case 2	58.0141 kW	3.2104 l/s	0.0439
Case 3	58.0122 kW	3.0447 l/s	0.0658
Case 4	58.0150 kW	3.0711 l/s	0.3500
Case 5	58.0158 kW	2.9126 l/s	0.3882
Case 6	58.0149 kW	3.2123 l/s	0.2948
Case 7	58.0222 kW	3.1235 l/s	0.4339

Figure 4.15 represents the results obtained with the MPC controller and the neural networks of cases 1 and 7. With this profile, there is a slight difference between MPC and the ANNs that increases with case 7, without being too large.

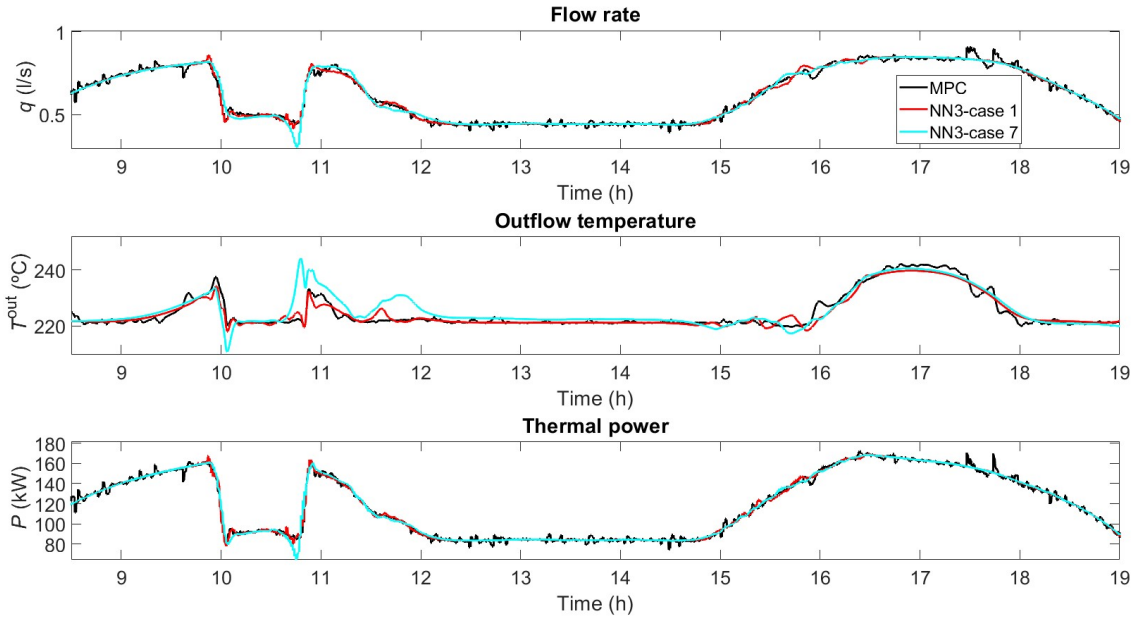

Figure 4.15 Evolution of the flow rate, the outlet temperature and the thermal power for the fourth validation profile. Cases 1 and 7.

4.3.5 Fifth Profile

The last validation profile corresponds to a day with clouds of various types. There is a medium cloud of short duration, a small cloud of long duration and a larger but fast cloud. The results are represented in figure 4.16 for cases 1 to 7, obtaining similar outputs for all of them. For case 8, the neural network is unable to control the system, which is destabilized from hour 11:00.

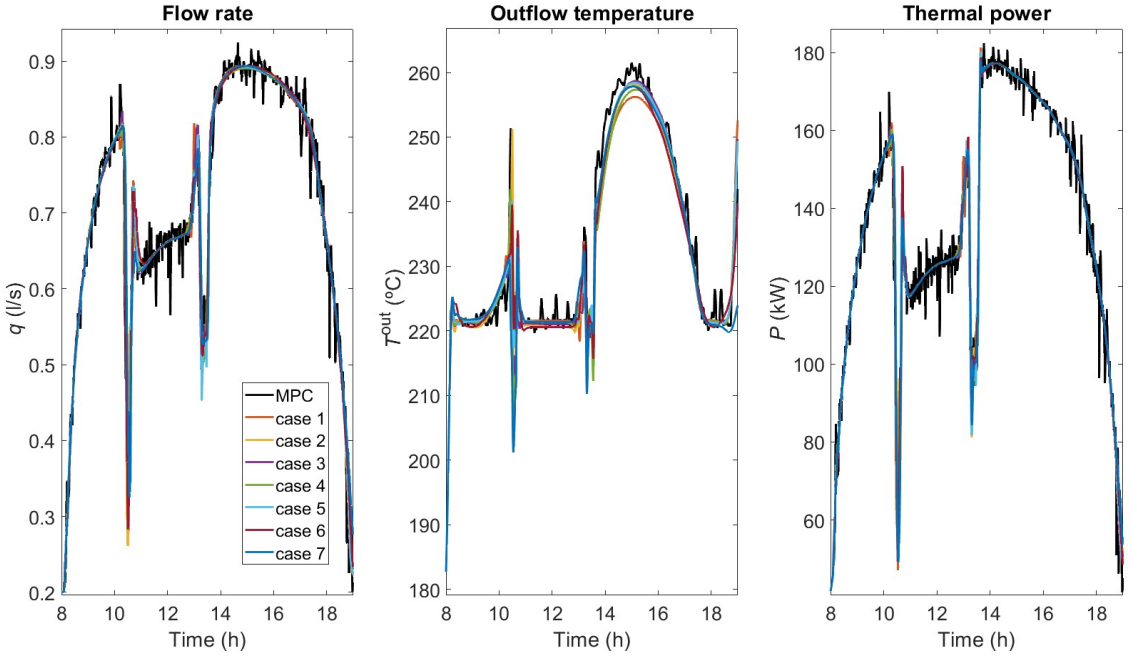

Figure 4.16 Evolution of the flow rate, the outlet temperature and the thermal power during a passing cloud for the fifth validation profile. Cases 1 to 7.

In table 4.7, it can be observed that, as with all other profiles, the AACI of all neural networks is lower than the one obtained with the MPC controller. In all cases, the MSCV is acceptable with the consideration of cases 4 and 7, where it is a little higher, without being too high. Also remarkable is case 5, which achieves a low MSCV, the lowest AACI and also the power is higher than with MPC.

Table 4.7 Results of the controllers used with the fifth validation profile.

Neurons	Mean power	AACI	MSCV
MPC	64.0417 kW	9.9777 l/s	0.0033
Case 1	64.0517 kW	3.8315 l/s	0.1490
Case 2	64.0504 kW	3.5275 l/s	0.3651
Case 3	64.0481 kW	3.2928 l/s	0.5636
Case 4	64.0515 kW	3.0231 l/s	1.6368
Case 5	64.0474 kW	3.4924 l/s	0.0393
Case 6	64.0538 kW	3.2521 l/s	0.1317
Case 7	64.0615 kW	2.9464 l/s	3.2281

The results obtained for cases 1 and 7 and the MPC controller are shown in figure 4.17. There is a slight decrease in the neural networks with respect to MPC that does not significantly affect the obtained thermal power. Once again, at the end of the day there is a slightly better performance of the neural network corresponding to case 7.

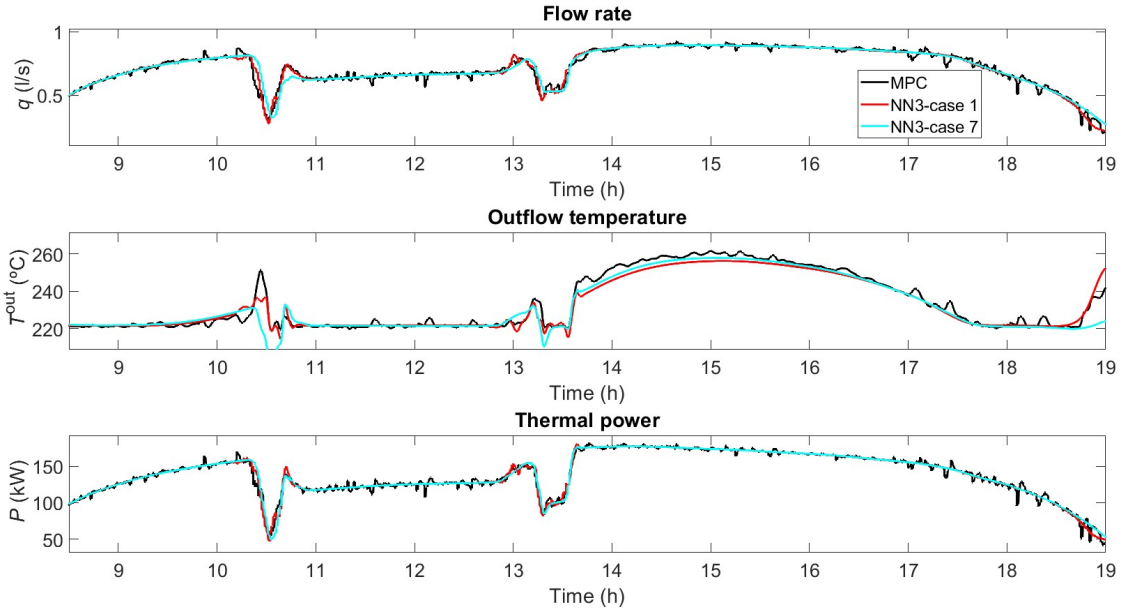

Figure 4.17 Evolution of the flow rate, the outlet temperature and the thermal power for the fourth validation profile. Cases 1 and 7.

4.4 Results Analysis

Different neural networks have been trained on 30 days of data and a single architecture has been selected. Then, the possibility of decreasing the number of inputs to the neural network has been analyzed and five validation experiments have been carried out. This section analyzes the results obtained in these experiments, from which the good performance of the neural network controllers is extracted.

First, neural networks of different sizes were trained, with between 1 and 3 hidden layers and between 5 and 20 neurons per layer. The neural network that gave the best results during training was a medium-sized

one, with two hidden layers of 15 and 10 neurons. It is worth mentioning that the rest of the ANNs also approximated the results of the MPC with high accuracy. In addition, it has been shown how the use of neural networks decreases computational times by 97%.

Then, 8 different cases were set up in which the inputs to the neural network were varied to show the performance capability even with fewer inputs and less prediction data. Analyzing the power obtained and the violation of restrictions, as expected, the neural network that has given the best results is the one corresponding to case 1, since it had more input information. On the other hand, the next best performing ones are those of cases 2, 4 and 6, with 360, 70 and 99 inputs, respectively. In situations where computational capacity is very low and low training times are required, the neural network corresponding to case 4 provides very good results using predictions only at instants 1 and 12 of the prediction horizon. For days with few clouds, the neural network of case 7 provides good results, but constraint violations increase on days with highly variable cloudiness. Case 8, on the other hand, destabilizes the plant on numerous occasions and is not suitable for controlling the plant.

5 Conclusions, Contributions and Future Work

This chapter aims to draw the conclusions obtained in the course of this project, both at a general and more specific level. In addition, the contributions made are highlighted and avenues for possible future development are detailed.

In this work, the advantages and disadvantages of controlling a parabolic trough plant by means of MPC, widely used in the literature, have been presented. On the one hand, it allows maximizing the thermal power obtained in the plant while obeying a series of operating restrictions. In addition, it takes into account the effect of possible control strategies and future values of solar radiation. On the other hand, they have the drawback of computational cost, since it is necessary to solve an optimization problem in real time at each sampling instant of the controller.

To solve this problem, this work proposes the use of neural networks trained offline from the results of a predictive controller to approximate its output in real time, since they are much faster than solving an optimization problem in nonlinear systems. The general conclusions drawn are the following:

- A similar thermal power is obtained using neural networks or MPC.
- It is possible to implement neural networks with very low violations in T^{out} constraints.
- The computation time is much lower using ANNs than with an MPC controller.
- Neural networks can learn from MPC without the need of many sensors and predictions.
- The control effort between consecutive instants is much slower for neural networks than for the MPC controller. This could be achieved with the MPC controller, but it would require a parameter resetting effort that is avoided with neural networks.

The neural networks corresponding to cases 1 to 6 have given good results for all the profiles used for validation, demonstrating that it is possible to decrease the number of inputs from 410 to 70. In addition, on days with moderate clouds, the neural network of case 7 also provides very good results, so that only 41 inputs would be necessary on these occasions. This neural network, however, is not valid for cases with more complex clouds.

The controllers have been applied in a collector loop. In the case of the MPC, the computation time at each instant has been around 3 seconds in most of the simulation. These times do not have a constant magnitude, reaching up to 7 or 9 seconds in some occasions. On the other hand, the computation times of the neural

network are around 0.1 seconds and the variations are smaller. This reduction of the computation times to a 3% allows its implementation in real time, since they are much lower than the sampling time of the controller. This allows the application of the controllers in real plants too. In the case of ACUREX, the plant has 10 loops, so it would not be possible to obtain the optimal solution at all times using MPC. In the case of commercial plants, which have more than 100 loops, the implementation of the MPC would be unfeasible, since it would be impossible to obtain the solution in real time, whereas the neural network controllers are still suitable.

One problem that can arise when applying neural networks is that the constraints are not taken into account directly, but are only considered by the MPC controller from which they learn. This leads to the fact that at different times the constraint violation may be too high, which occurs in the cases that had fewer inputs. In all other cases, the neural networks responded well to perturbations without too high or too low temperatures. However, constraint violations could be reduced by imposing harder constraints and smaller ranges on the MPC controller used for training.

The work presented in this master thesis has resulted in the submission of a paper accepted for the journal *Renewable Energy* of the first quartile in Green & sustainable science & technology and Energy and fuels [83].

Future research is to develop this method at other levels:

- Apply these controllers in plants with a larger number of loops.
- Combine neural networks with other control strategies such as distributed or coalitional control.
- Use other types of neural networks with, for example, non-supervised learning.

List of Figures

List of Tables

Bibliography

[1] *MathWorks*, *https://www.mathworks.com/*, May 2021, https://www.mathworks.com/.

[2] *OCONTSOLAR (optimal control of thermal solar energy systems)*, *https://cordis.europa.eu/project/id/789051/es*, May 2021.

[3] *Red eléctrica española*, *https://www.ree.es/en*, May 2021.

[4] A. Azadeh, M. Sheikhalishahi, and S. M. Asadzadeh, *A flexible neural network-fuzzy data envelopment analysis approach for location optimization of solar plants with uncertainty and complexity*, Renewable Energy **36** (2011), no. 12, 3394–3401.

[5] M. Barão, J. M. Lemos, and R. N. Silva, *Reduced complexity adaptive nonlinear control of a distributed collector solar field*, Journal of Process Control **12** (2002), no. 1, 131–141.

[6] A. Bayas, I. Škrjanc, and D. Sáez, *Design of fuzzy robust control strategies for a distributed solar collector field*, Applied Soft Computing **71** (2018), 1009–1019.

[7] A. Bemporad and P. Patrinos, *Simple and certifiable quadratic programming algorithms for embedded linear model predictive control*, IFAC Proceedings Volumes **45** (2012), no. 17, 14–20.

[8] H. Benoit, L. Spreafico, D.l Gauthier, and G. Flamant, *Review of heat transfer fluids in tube-receivers used in concentrating solar thermal systems: Properties and heat transfer coefficients*, Renewable and Sustainable Energy Reviews **55** (2016), 298–315.

[9] M. Berenguel, M. R. Arahal, and E. F. Camacho, *Modelling the free response of a solar plant for predictive control*, Control Engineering Practice **6** (1998), no. 10, 1257 – 1266.

[10] D. P. Bertsekas, *Dynamic programming and optimal control 3rd edition*, Belmont, MA: Athena Scientific **2** (2011).

[11] E. F. Camacho and M. Berenguel, *Application of generalized predictive control to a solar power plant*, 1994 Proceedings of IEEE International Conference on Control and Applications, vol. 3, 1994, pp. 1657–1662.

[12] E. F. Camacho, M. Berenguel, and C. Bordons, *Adaptive generalized predictive control of a distributed collector field*, IEEE Transactions on Control Systems Technology **2** (1994), no. 4, 462–467.

[13] E. F. Camacho, M. Berenguel, and A. J. Gallego, *Control of thermal solar energy plants*, Journal of Process Control **24** (2014), no. 2, 332 – 340, ADCHEM 2012 Special Issue.

[14] E. F. Camacho, M. Berenguel, and F. R. Rubio, *Application of a gain scheduling generalized predictive controller to a solar power plant*, Control Engineering Practice **2** (1994), no. 2, 227 – 238.

[15] ———, *Advanced control of solar plants*, Springer, Dec 1997.

[16] E. F. Camacho, M. Berenguel, F. R. Rubio, and D. M. Plaza, *Control of solar energy systems*, Springer, Jan 2012.

[17] E. F. Camacho and C. Bordons, *Model predictive control*, Springer Science & Business Media, 2013.

[18] E. F. Camacho and A. J. Gallego, *Optimal operation in solar trough plants: A case study*, Solar Energy **95** (2013), 106 – 117.

[19] ———, *Model predictive control in solar trough plants: A review*, IFAC-PapersOnLine **48** (2015), no. 23, 278 – 285, 5th IFAC Conference on Nonlinear Model Predictive Control NMPC 2015.

[20] ———, *Advanced control strategies to maximize roi and the value of the concentrating solar thermal (cst) plant to the grid*, Advances in Concentrating Solar Thermal Research and Technology (Manuel J. Blanco and Lourdes Ramirez Santigosa, eds.), Woodhead Publishing Series in Energy, Woodhead Publishing, 2017, pp. 311 – 336.

[21] E. F. Camacho, F. R. Rubio, M. Berenguel, and L. Valenzuela, *A survey on control schemes for distributed solar collector fields. part i: Modeling and basic control approaches*, Solar Energy **81** (2007), 1240–1251.

[22] E. F. Camacho, F. R. Rubio, and F. M. Hughes, *Self-tuning control of a solar power plant with a distributed collector field*, IEEE Control Systems Magazine **12** (1992), no. 2, 72–78.

[23] A. Cardoso, J. Henriques, and A. Dourado, *Fuzzy supervisor and feedforward control of a solar power plant using accessible disturbances*, 1999 European Control Conference (ECC), 1999, pp. 1711–1716.

[24] L. Cheng, Z. G. Hou, and M. Tan, *Constrained multi-variable generalized predictive control using a dual neural network*, Neural Computing and Applications **16** (2007), no. 6, 505–512.

[25] S. Chidrawar and B. Patre, *Generalized predictive control and neural generalized predictive control*, Leonardo journal of sciences **7** (2008), no. 13, 133–152.

[26] F. Coito, J. M. Lemos, R. N. Silva, and E. Mosca, *Adaptive control of a solar energy plant: Exploiting accessible disturbances*, International Journal of Adaptive Control and Signal Processing **11** (1997), no. 4, 327–342.

[27] E. G. Cojocaru, J. M. Bravo, M. J. Vasallo, and D. M. Santos, *Optimal scheduling in concentrating solar power plants oriented to low generation cycling*, Renewable Energy **135** (2019), 789 – 799.

[28] J. Drgoňa, D. Picard, M. Kvasnica, and L. Helsen, *Approximate model predictive building control via machine learning*, Applied Energy **218** (2018), 199 – 216.

[29] N. El Boukhari et al., *Optimal control of a parabolic trough receiver distributed model*, Recent Advances in Modeling, Analysis and Systems Control: Theoretical Aspects and Applications, Springer, 2020, pp. 205–219.

[30] Z. Şen, *Solar energy in progress and future research trends*, Progress in Energy and Combustion Science **30** (2004), no. 4, 367 – 416.

[31] I. Farkas and I. Vajk, *Experiments with internal model-based controller for acurex field*, Jul 2002.

[32] _____, *Internal model-based controller for a solar plant*, IFAC Proceedings Volumes **35** (2002), no. 1, 49–54, 15th IFAC World Congress.

[33] T. L. Fine, *Feedforward neural network methodology*, Springer Science & Business Media, 2006.

[34] J. R. D. Frejo and E. F. Camacho, *Centralized and distributed model predictive control for the maximization of the thermal power of solar parabolic-trough plants*, Solar Energy **204** (2020), 190–199.

[35] A. J. Gallego, *Control predictivo de sistemas de energía solar distribuidos*, Ph.D. thesis, Universidad de Sevilla, 2014.

[36] A. J. Gallego and E. F. Camacho, *Adaptative state-space model predictive control of a parabolic-trough field*, Control Engineering Practice **20** (2012), no. 9, 904 – 911.

[37] A. J. Gallego, F. Fele, E. F. Camacho, and L. Yebra, *Observer-based model predictive control of a parabolic-trough field*, Solar Energy **97** (2013), 426 – 435.

[38] G. S. Georgiou, P. Nikolaidis, S. A. Kalogirou, and P. Christodoulides, *A hybrid optimization approach for autonomy enhancement of nearly-zero-energy buildings based on battery performance and artificial neural networks*, Energies **13** (2020), no. 14.

[39] J. Gil, P.and Henriques, A. Cardoso, and A. Dourado, *Neural network in scheduling linear controllers with application to a solar power plant*, Proceedings of 5th IASTED International Conference on Control and Applications, Cancun, Mexico, 2002.

[40] P. Gil, A. Cardoso, J. Henriques, P. Carvalho, H. Duarte-Ramos, and A. Dourado, *Experiments with an adaptive neural model-based predictive controller applied to a distributed solar collector field: Performance and fault tolerance assessment*, IHP Programme. Research Results at PSA within the Year (2002).

[41] P. Gil, J. Henriques, P. Carvalho, H. Duarte-Ramos, and A. Dourado, *Adaptive neural model-based predictive control of a solar power plant*, Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No.02CH37290), vol. 3, 2002, pp. 2098–2103.

[42] P. Gil, J. Henriques, and A. Dourado, *Recurrent neural networks and feedback linearization for a solar power plant control*, at EUNIT01 (2001).

[43] M. T. Hagan, O. De Jesús, and R. Schultz, *Training recurrent networks for filtering and control*, Recurrent neural networks: Design and applications, CRC press, 1999, pp. 311–340.

[44] J. Henriques, P. Gil, and A. Dourado, *Neural output regulation for a solar power plant*, IFAC Proceedings Volumes **35** (2002), no. 1, 307 – 312, 15th IFAC World Congress.

[45] J. Q. Huang and F. L. Lewis, *Neural-network predictive control for nonlinear dynamic systems with time-delay*, IEEE Transactions on Neural Networks **14** (2003), no. 2, 377–389.

[46] L. Huynh, J. Kececioglu, M. Köppe, and I. Tagkopoulos, *Automatic design of synthetic gene circuits through mixed integer non-linear programming*, PloS one **7** (2012), no. 4, e35529.

[47] Y. Ichikawa and Sawa T., *Neural network application for direct feedback controllers*, IEEE Transactions on Neural Networks **3** (1992), no. 2, 224–231.

[48] A. Ichtev, J. Hellendoom, R. Babuska, and S. Mollov, *Fault-tolerant model-based predictive control using multiple takagi-sugeno fuzzy models*, 2002 IEEE World Congress on Computational Intelligence. 2002 IEEE International Conference on Fuzzy Systems. FUZZ-IEEE'02. Proceedings (Cat. No. 02CH37291), vol. 1, IEEE, 2002, pp. 346–351.

[49] F. Jalili Jamshidian, S. Gorjian, and M. Shafiee Far, *An overview of solar thermal power generation systems; components and applications*, 08 2018.

[50] S. A. Kalogirou, *Artificial neural networks in renewable energy systems applications: a review*, Renewable and Sustainable Energy Reviews **5** (2001), no. 4, 373 – 401.

[51] _____, *Solar thermal collectors and applications*, Progress in Energy and Combustion Science **30** (2004), no. 3, 231 – 295.

[52] B. Karg and S. Lucia, *Deep learning-based embedded mixed-integer model predictive control*, 2018 European Control Conference (ECC), IEEE, 2018, pp. 2075–2080.

[53] A. Kebir, L. Woodward, and O. Akhrif, *Real-time optimization of renewable energy sources power using neural network-based anticipative extremum-seeking control*, Renewable Energy **134** (2019), 914–926.

[54] B. M. Åkesson and H. T. Toivonen, *A neural network model predictive controller*, Journal of Process Control **16** (2006), no. 9, 937 – 946.

[55] B. Khoukhi, M. Tadjine, and M. S. Boucherit, *Nonlinear continuous-time generalized predictive control of solar power plant*, International Journal for Simulation and Multidisciplinary Design Optimization **6** (2015), A3.

[56] K. Levenberg, *A method for the solution of certain non-linear problems in least squares*, Quarterly of applied mathematics **2** (1944), no. 2, 164–168.

[57] F. J. Lin and R. J. Wai, *Hybrid control using recurrent fuzzy neural network for linear induction motor servo drive*, IEEE Transactions on Fuzzy Systems **9** (2001), no. 1, 102–115.

[58] A. P. Loh, K. O. Looi, and K. F. Fong, *Neural network modelling and control strategies for a pH process*, Journal of Process Control **5** (1995), no. 6, 355 – 362.

[59] C. H. Lu and C. C. Tsai, *Predictive control using recurrent neural networks for industrial processes*, Journal of the Chinese institute of engineers **32** (2009), no. 2, 277–283.

[60] D. G. Luenberger, Y. Ye, et al., *Linear and nonlinear programming*, vol. 2, Springer, 1984.

[61] D. W. Marquardt, *An algorithm for least-squares estimation of nonlinear parameters*, Journal of the society for Industrial and Applied Mathematics **11** (1963), no. 2, 431–441.

[62] E. Masero, J. R. D. Frejo, J. M. Maestre, and E. F. Camacho, *A light clustering model predictive control approach to maximize thermal power in solar parabolic-trough plants*, Solar Energy **214** (2021), 531 – 541.

[63] W. S. McCulloch and W. Pitts, *A logical calculus of the ideas immanent in nervous activity*, The bulletin of mathematical biophysics **5** (1943), no. 4, 115–133.

[64] A. Meaburn and F. M. Hughes, *A simple predictive controller for use on large scale arrays of parabolic trough collectors*, Solar Energy **56** (1996), no. 6, 583–595.

[65] P. R. C. Mendes, J. M. Maestre, C. Bordons, and J. E. Normey-Rico, *A practical approach for hybrid distributed mpc*, Journal of process Control **55** (2017), 30–41.

[66] T. M. Mitchell, *Machine learning*, (1997).

[67] S. J. Navas, F. R. Rubio, P. Ollero, and J. M. Lemos, *Optimal control applied to distributed solar collector fields with partial radiation*, Solar Energy **159** (2018), 811 – 819.

[68] A. Nevado Reviriego, F. Hernández-del Olmo, and L. Álvarez Barcia, *Nonlinear adaptive control of heat transfer fluid temperature in a parabolic trough solar power plant*, Energies **10** (2017), no. 8.

[69] D. Nguyen and B. Widrow, *Neural network for self-learning control systems*, Control Systems Magazine, IEEE **10** (1990), 18 – 23.

[70] ———, *The truck backer-upper: An example of self-learning in neural networks*, Advanced Neural Computers (Rolf ECKMILLER, ed.), North-Holland, Amsterdam, 1990, pp. 11 – 19.

[71] M. A. Nielsen, *Neural networks and deep learning*, vol. 25, Determination press San Francisco, CA, 2015.

[72] J. E. Normey-Rico, C. Bordons, M. Berenguel, and E. F. Camacho, *A robust adaptive dead-time compensator with application to a solar collector field1*, IFAC Proceedings Volumes **31** (1998), no. 19, 93–98, IFAC Workshop on Linear Time Delay Systems (LTDS '98), Grenoble, France, 6-7 July.

[73] K. Noureldin, T. Hirsch, P. Kuhn, B. Nouri, Z. Yasser, and R. Pitz-Paal, *Modelling an automatic controller for parabolic trough solar fields under realistic weather conditions*, AIP Conference Proceedings, vol. 2033, AIP Publishing LLC, 2018, p. 210009.

[74] A. Omu, R. Choudhary, and A. Boies, *Distributed energy resource system optimisation using mixed integer linear programming*, Energy Policy **61** (2013), 249–266.

[75] M. G. Ortega, F. R. Rubio, and M. Berenguel, *An h-infinity controller for a solar power plant*, Proc. of the IASTED Int. Conf. on Control, p. 122–125.

[76] T. Parisini and R. Zoppoli, *A receding-horizon regulator for nonlinear systems and a neural approximation*, Automatica **31** (1995), no. 10, 1443 – 1451.

[77] A. Parlos, S. Parthasarathy, and A. Atiya, *Neuro-predictive process control using online controller adaptation*, Control Systems Technology, IEEE Transactions on **9** (2001), 741 – 755.

[78] H. A. Pipino, M. M. Morato, E. Bernardi, E. J. Adam, and J. E.. Normey-Rico, *Nonlinear temperature regulation of solar collectors with a fast adaptive polytopic lpv mpc formulation*, Solar Energy **209** (2020), 214 – 225.

[79] D. L. Poole and A. K. Mackworth, *Artificial intelligence: foundations of computational agents*, Cambridge University Press, 2010.

[80] D. Psaltis, A. Sideris, and A. A. Yamamura, *A multilayered neural network controller*, IEEE Control Systems Magazine **8** (1988), no. 2, 17–21.

[81] P. Ramachandran, B. Zoph, and Q. V. Le, *Searching for activation functions*, arXiv preprint arXiv:1710.05941 (2017).

[82] C. Rorres, A. Orbach, and R. Fischl, *Optimal and suboptimal control policies for a solar collector system*, IEEE Transactions on Automatic Control **25** (1980), no. 6, 1085–1091.

[83] S. Ruiz-Moreno, J. R. D. Frejo, and E. F. Camacho, *Model predictive control based on deep learning for solar parabolic-trough plants*, Renewable Energy **180** (2021), 193–202.

[84] D. E. Rumelhart, G.y E. Hinton, and R. J. Williams, *Learning representations by back-propagating errors*, nature **323** (1986), no. 6088, 533–536.

[85] A. J. Sanchez, J. M. Escaño, N. Canty, A.J. Gallego, and E. F. Camacho, *Solar radiation estimator and fault tolerant model predictive control of a parabolic-trough field*, 2015 26th Irish Signals and Systems Conference (ISSC), 2015, pp. 1–7.

[86] A. Shapiro, D. Dentcheva, and A. Ruszczyński, *Lectures on stochastic programming: modeling and theory*, SIAM, 2009.

[87] R. N. Silva, L. M. Rato, J. M. Lemos, and F. Coito, *Cascade control of a distributed collector solar field*, Journal of Process Control **7** (1997), no. 2, 111 – 117.

[88] A. J. Sánchez, A. J. Gallego, J. M. Escaño, and E. F. Camacho, *Event-based mpc for defocusing and power production of a parabolic trough plant under power limitation*, Solar Energy **174** (2018), 570 – 581.

[89] _____, *Temperature homogenization of a solar trough field for performance improvement*, Solar Energy **165** (2018), 1 – 9.

[90] _____, *Adaptive incremental state space mpc for collector defocusing of a parabolic trough plant*, Solar Energy **184** (2019), 105 – 114.

[91] _____, *Thermal balance of large scale parabolic trough plants: A case study*, Solar Energy **190** (2019), 69 – 81.

[92] J. Sola and J. Sevilla, *Importance of input data normalization for the application of neural networks to complex industrial problems*, IEEE Transactions on Nuclear Science **44** (1997), no. 3, 1464–1468.

[93] D. Ushida and E. Konaka, *Model predictive control implementation on neural networks using denoising autoencoder*, 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2016, pp. 000149–000154.

[94] M. J. Vasallo and J. M. Bravo, *A mpc approach for optimal generation scheduling in csp plants*, Applied Energy **165** (2016), 357 – 370.

[95] B. Widrow, D. E. Rumelhart, and M. A. Lehr, *Neural networks: applications in industry, business and science*, Communications of the ACM **37** (1994), no. 3, 93–106.

[96] Q. H. Wu, B. W. Hogg, and G. W. Irwin, *A neural network regulator for turbogenerators*, IEEE Transactions on Neural Networks **3** (1992), no. 1, 95–100.

[97] X. Xu, Yuan Y., and S. Dubljevic, *Optimal control of a distributed solar collector field*, 2017 American Control Conference (ACC), 2017, pp. 1821–1826.

[98] Z. Yan, J. Lu, and G. Zhang, *Distributed model predictive control of linear systems with coupled constraints based on collective neurodynamic optimization*, Australasian Joint Conference on Artificial Intelligence, Springer, 2018, pp. 318–328.

[99] Z. Yan and J. Wang, *Model predictive control of nonlinear systems with unmodeled dynamics based on feedforward and recurrent neural networks*, IEEE Transactions on Industrial Informatics **8** (2012), no. 4, 746–756.

[100] L. Yebra, M. Berenguel, J. Bonilla, L. Roca, S. Dormido, and E. Zarza, *Object-oriented modelling and simulation of acurex solar thermal power plant*, Mathematical and Computer Modelling of Dynamical Systems **16** (2010), no. 3, 211–224.

[101] A. Zaaoumi, A. Bah, M. Ciocan, P. Sebastian, M.C. Balan, A. Mechaqrane, and M. Alaoui, *Estimation of the energy production of a parabolic trough solar thermal power plant using analytical and artificial neural networks models*, Renewable Energy **170** (2021), 620–638.

[102] J. M. Zamarreño and P. Vega, *Neural predictive control. application to a highly non-linear system*, Engineering Applications of Artificial Intelligence **12** (1999), no. 2, 149 – 158.

[103] H. L. Zhang, J. Baeyens, J. Degrève, and G. Cacères, *Concentrated solar power plants: Review and design methodology*, Renewable and Sustainable Energy Reviews **22** (2013), 466 – 481.

Glossary

AACI accumulated absolute control increment. 33, 37, 41–43, 47, 49, 54

AC adaptive control. 22

AI artificial intelligence. 9

ANN artificial neural network. 2, 3, 5, 8–11, 13–15, 25, 27, 31, 32, 37–39, 43, 47, 50, 51

CC cascade control. 22

CIEMAT Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas. IV, 3

CSP concentrating solar power. 2, 15, 17

CST concentrated solar thermo-electric. 17

DL deep learning. 9

DNI direct normal irradiance. 16, 20, 33, 42, 45, 47

DP dynamic programming. 7

FF feedforward. 8, 21

FLC fuzzy logic control. 14, 25

GS gain scheduling. 22

GS-GPC gain scheduling generalized predictive control. 8, 21, 22

HTF heat transfer fluid. 3, 16, 19, 21, 27, 29, 30

IMC internal model control. 23

LFR linear Fresnel reflector. 15, 16

LP linear programming. 7

LQG linear quadratic gaussian. 24

MILP mixed integer linear programming. 7

MINLP mixed integer nonlinear programming. 7

MIQP mixed integer quadratic programming. 7

ML machine learning. 9

MLP multilayer perceptron. 8, 10, 11, 13, 53

MPC model predictive control. 2, 3, 5–8, 14, 15, 20–22, 25–27, 29, 31, 32, 37–40, 42–47, 49–54

MSCV mean squared constraint violation. 33, 37, 40–43, 46, 47, 49, 54

MSE mean squared error. 14

NC nonlinear control. 25

NLP nonlinear programming. 7

NMPC non-linear model predictive control. 7

NNC neural network controller. 25

PDS parabolic dish system. 15, 17

PID proportional, integral, derivative. 8, 14, 15, 22, 25

PSA Plataforma Solar de Almería. IV, 3, 8, 29

PTC parabolic-trough collector. 3, 5, 14–16, 20–22

PV photovoltaic cells. 1, 2, 15, 17

QP quadratic programming. 7, 8, 29

RC robust control. 24

ReLU rectified linear unit. VI, 11

ROI return of investment. 8

SP stochastic programming. 7

SPT solar power tower. 15, 16

SQP sequential quadratic programming. 7, 28

TDC time delay compensation. 23

TES thermal energy storage. 2, 19

UKF unscented Kalman filter. 8