



GRADO EN INGENIERÍA ELECTROMECAÁNICA

TRABAJO FIN DE GRADO
DISEÑO E INTEGRACIÓN DE UN CONTROL
PREDICTIVO DE TEMPERATURA EN UN SISTEMA
DOMÓTICO

Autor: Sara Puente Iglesias
Director: Juan Luis Zamora Macho

Madrid
Diciembre de 2019

AUTORIZACIÓN PARA LA DIGITALIZACIÓN, DEPÓSITO Y DIVULGACIÓN EN RED DE PROYECTOS FIN DE GRADO, FIN DE MÁSTER, TESIS O MEMORIAS DE BACHILLERATO

1º. Declaración de la autoría y acreditación de la misma.

El autor D^a Sara Puente Iglesias como estudiante de la UNIVERSIDAD PONTIFICIA COMILLAS (ICAI, E.T.S.I. de Ingeniera Industrial)

DECLARA ser el titular de los derechos de propiedad intelectual de la obra: "DISEÑO E INTEGRACION DE UN CONTROL PREDICTIVO DE TEMPERATURA EN UN SISTEMA DOMOTICO", que ésta es una obra original, y que ostenta la condición de autor en el sentido que otorga la Ley de Propiedad Intelectual.

2º. Objeto y fines de la cesión.

Con el fin de dar la máxima difusión a la obra citada a través del Repositorio institucional de la Universidad, el autor **CEDE** a la Universidad Pontificia Comillas, de forma gratuita y no exclusiva, por el máximo plazo legal y con ámbito universal, los derechos de digitalización, de archivo, de reproducción, de distribución y de comunicación pública, incluido el derecho de puesta a disposición electrónica, tal y como se describen en la Ley de Propiedad Intelectual. El derecho de transformación se cede a los únicos efectos de lo dispuesto en la letra a) del apartado siguiente.

3º. Condiciones de la cesión y acceso

Sin perjuicio de la titularidad de la obra, que sigue correspondiendo a su autor, la cesión de derechos contemplada en esta licencia habilita para:

- a) Transformarla con el fin de adaptarla a cualquier tecnología que permita incorporarla a internet y hacerla accesible; incorporar metadatos para realizar el registro de la obra e incorporar "marcas de agua" o cualquier otro sistema de seguridad o de protección.
- b) Reproducirla en un soporte digital para su incorporación a una base de datos electrónica, incluyendo el derecho de reproducir y almacenar la obra en servidores, a los efectos de garantizar su seguridad, conservación y preservar el formato.
- c) Comunicarla, por defecto, a través de un archivo institucional abierto, accesible de modo libre y gratuito a través de internet.
- d) Cualquier otra forma de acceso (restringido, embargado, cerrado) deberá solicitarse expresamente y obedecer a causas justificadas.
- e) Asignar por defecto a estos trabajos una licencia Creative Commons.
- f) Asignar por defecto a estos trabajos un HANDLE (URL *persistente*).

4º. Derechos del autor.

El autor, en tanto que titular de una obra tiene derecho a:

- a) Que la Universidad identifique claramente su nombre como autor de la misma
- b) Comunicar y dar publicidad a la obra en la versión que ceda y en otras posteriores a través de cualquier medio.
- c) Solicitar la retirada de la obra del repositorio por causa justificada.
- d) Recibir notificación fehaciente de cualquier reclamación que puedan formular terceras personas en relación con la obra y, en particular, de reclamaciones relativas a los derechos de propiedad intelectual sobre ella.

5º. Deberes del autor.

El autor se compromete a:

- a) Garantizar que el compromiso que adquiere mediante el presente escrito no infringe ningún derecho de terceros, ya sean de propiedad industrial, intelectual o cualquier otro.
- b) Garantizar que el contenido de las obras no atenta contra los derechos al honor, a la intimidad y a la imagen de terceros.
- c) Asumir toda reclamación o responsabilidad, incluyendo las indemnizaciones por daños, que

podieran ejercitarse contra la Universidad por terceros que vieran infringidos sus derechos e intereses a causa de la cesión.

- d) Asumir la responsabilidad en el caso de que las instituciones fueran condenadas por infracción de derechos derivada de las obras objeto de la cesión.

6º. Fines y funcionamiento del Repositorio Institucional.

La obra se pondrá a disposición de los usuarios para que hagan de ella un uso justo y respetuoso con los derechos del autor, según lo permitido por la legislación aplicable, y con fines de estudio, investigación, o cualquier otro fin lícito. Con dicha finalidad, la Universidad asume los siguientes deberes y se reserva las siguientes facultades:

- La Universidad informará a los usuarios del archivo sobre los usos permitidos, y no garantiza ni asume responsabilidad alguna por otras formas en que los usuarios hagan un uso posterior de las obras no conforme con la legislación vigente. El uso posterior, más allá de la copia privada, requerirá que se cite la fuente y se reconozca la autoría, que no se obtenga beneficio comercial, y que no se realicen obras derivadas.
- La Universidad no revisará el contenido de las obras, que en todo caso permanecerá bajo la responsabilidad exclusiva del autor y no estará obligada a ejercitar acciones legales en nombre del autor en el supuesto de infracciones a derechos de propiedad intelectual derivados del depósito y archivo de las obras. El autor renuncia a cualquier reclamación frente a la Universidad por las formas no ajustadas a la legislación vigente en que los usuarios hagan uso de las obras.
- La Universidad adoptará las medidas necesarias para la preservación de la obra en un futuro.
- La Universidad se reserva la facultad de retirar la obra, previa notificación al autor, en supuestos suficientemente justificados, o en caso de reclamaciones de terceros.

Madrid, a 20 de diciembre de 2019

ACEPTA

Fdo.....



Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título
**"DISEÑO E INTEGRACION DE UN CONTROL PREDICTIVO DE
TEMPERATURA EN UN SISTEMA DOMOTICO"**

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el
curso académico 2019 es de mi autoría, original e inédito y

no ha sido presentado con anterioridad a otros efectos. El Proyecto no es
plagio de otro, ni total ni parcialmente y la información que ha sido tomada
de otros documentos está debidamente referenciada.



Fdo.: Sara Puentes Iglesias

Fecha: 20/12/2019

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.: Juan Luis Zamora Maacho

Fecha: 20/12/2019



DISEÑO E INTEGRACIÓN DE UN CONTROL PREDICTIVO DE TEMPERATURA EN UN SISTEMA DOMÓTICO

Alumno: Puente Iglesias, Sara

Director: Zamora Macho, Juan Luis

RESUMEN DEL PROYECTO

Las tecnologías de la información y la comunicación (TIC) han experimentado un gran auge durante éstos últimos años. Están presentes en prácticamente todos los ámbitos de nuestra vida incluyendo nuestro propio hogar.

Siempre ha existido la necesidad de dotar a la vivienda de un alto grado de control, seguridad, confort, versatilidad y ahorro. La Domótica, es el conjunto de tecnologías aplicadas al control y la automatización inteligente de la vivienda y tiene como objetivo ofrecer una mejor calidad de vida en el lugar habitual de residencia.

Un aspecto fundamental en una vivienda domótica es el confort y unos de los principales objetivos del confort es el correcto control de la temperatura. Los principales problemas de los termostatos actuales son: su excesivo precio y su funcionalidad limitada.

Objetivo:

En este proyecto se propone combatir los principales problemas de los termostatos actuales. Utilizando la tecnología Z-Wave y con una raspberry pi 3B+ (precio de mercado 33€) se pretende instalar un control predictivo de temperatura en un sistema de radiadores de agua, dando un valor añadido a la instalación domótica sin encarecerla.

Además, este control contará con una puesta en servicio, es capaz de sacar un modelo de la planta a partir de ensayos, de manera que el control puede adaptarse a las necesidades concretas de cada vivienda.

La ventaja principal de utilizar un control predictivo es, que permite optimizar la actuación a aplicar teniendo en cuenta la predicción de la evolución futura de temperatura. Además, puede tener en cuenta el aspecto económico, si se incluye en la función objetivo que se optimiza.

comandos http. Por ello, se ha tenido que diseñar un servidor en NODE JS e implantarlo en la raspberry. Este servidor recibe los comandos http del controlador y los transforma en comandos TCP/IP de modo que el control pueda entenderlo. Al revés hace lo mismo, transforma los comandos TCP/IP del control a comandos http que pueda entender el controlador.

A continuación, hemos realizado 3 máquinas de estado para las comunicaciones: la máquina de estados general, la del sensor y la del actuador:

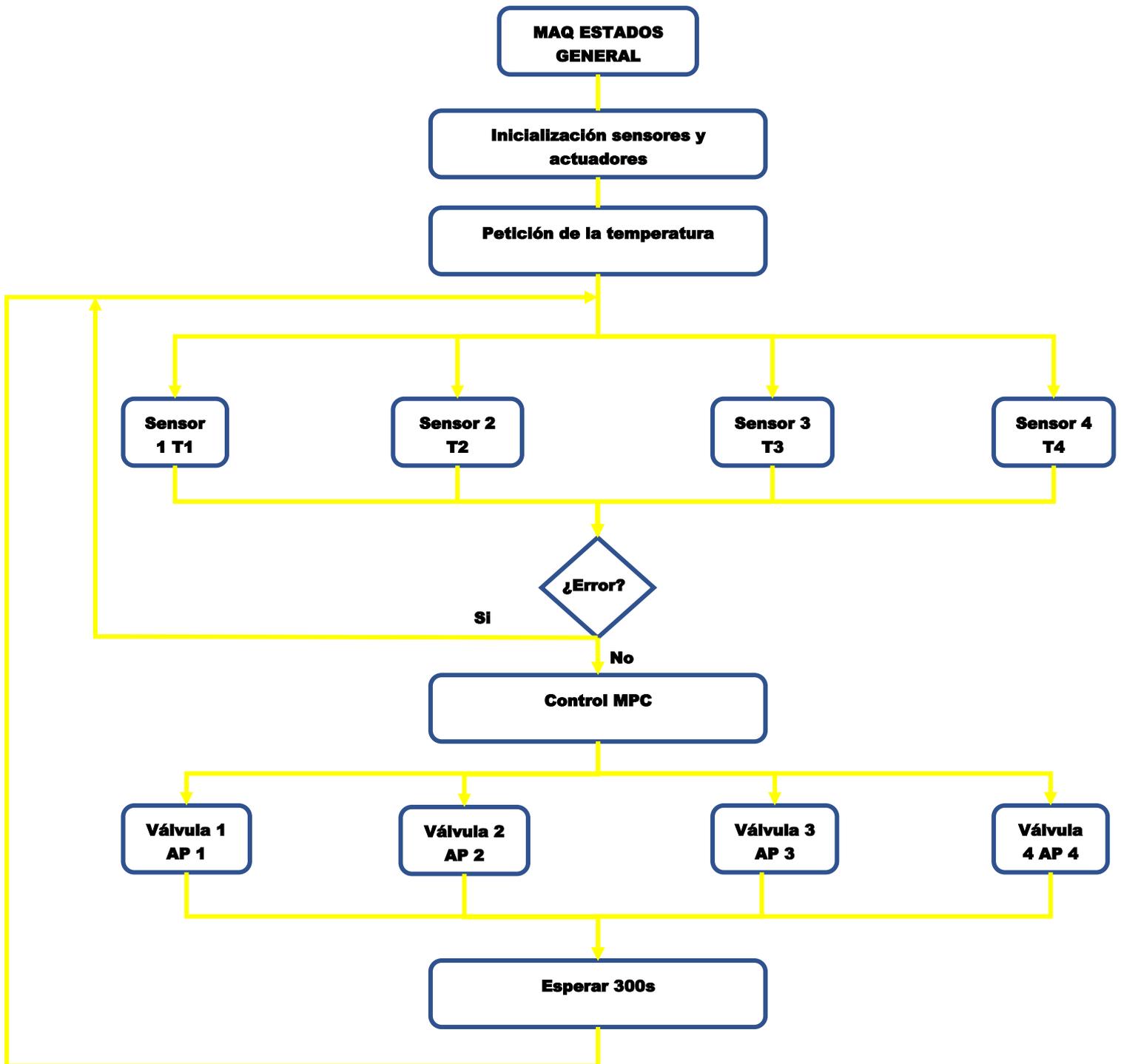


Figura 2. Máquina de estados general

Máquina de estados general:

Como podemos observar en la figura, primero inicializaremos los sensores y actuadores. A continuación, pediremos a los cuatro sensores la temperatura, si no ha habido ningún error las cuatro temperaturas se enviarán al control. El control calculará el porcentaje de apertura de cada habitación y se lo enviará a los actuadores correspondientes. Por último, un timer esperará el período de muestreo del control (5 min).

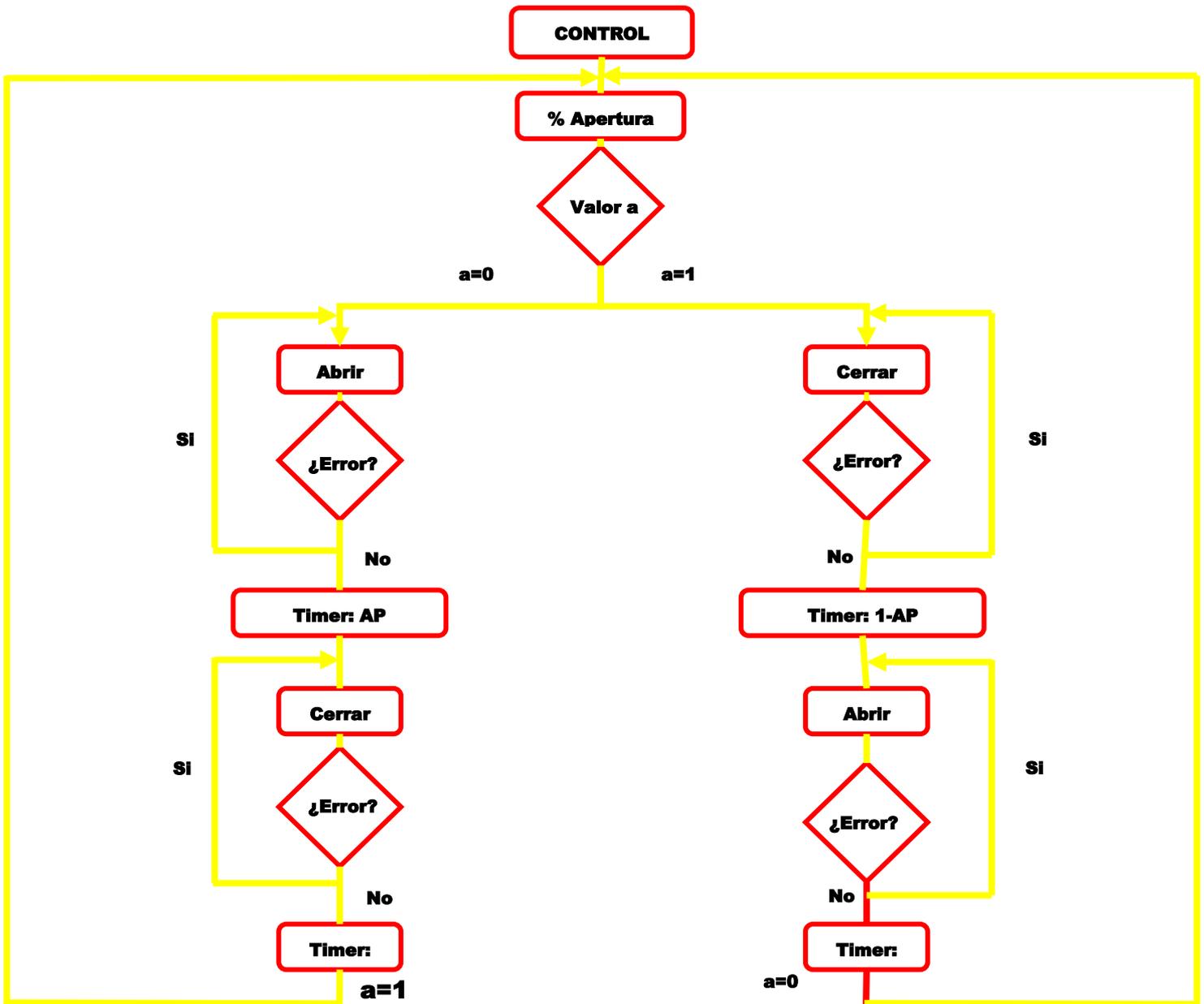


Figura 3. Máquina de estados de la válvula

Para que la válvula consiga abrir el porcentaje de apertura solicitado por el control, hemos tenido que diseñar un módulo PWM que abra el porcentaje indicado de los 5 min que dura el período de muestro y cierre el resto de los 5 min.

Para evitar un excesivo desgaste en los actuadores y reducir a la mitad las aperturas y cierres de la válvula, hemos tenido en cuenta la posición en la que ha acabado la válvula en la acción anterior del control. Si la válvula ha acabado abierta, en la siguiente acción, comenzará abriendo y si ha acabado cerrada, comenzará cerrando. Esto es precisamente lo que indica el valor a de la figura 3. Si $a=1$, significa que la válvula acabo su última acción abierta, por tanto, primero abrirá y después cerrará, si en cambio $a =0$, significa que la válvula en su última acción terminó cerrada, por tanto, primero comenzará cerrando y luego abriendo.

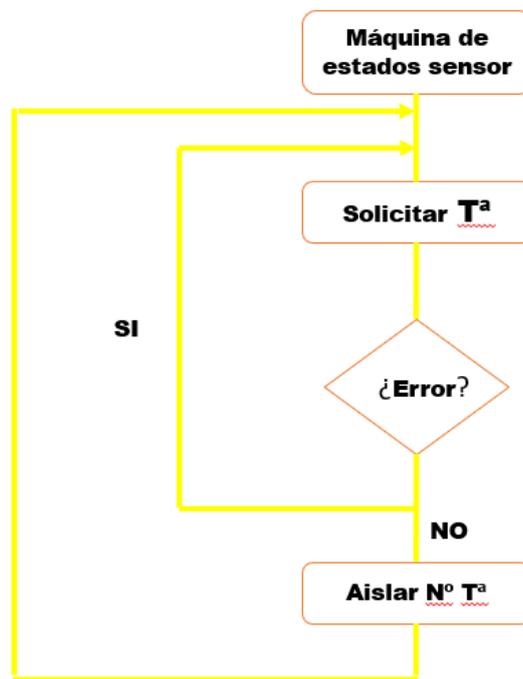


Figura 4. Máquina de estados del sensor.

Máquina de estados del sensor:

Comienza enviando al controlador un comando http solicitando la medida de temperatura al sensor. Por último, cuando el controlador envía la respuesta, la máquina de estados aísla el valor de temperatura para enviárselo al control.

La siguiente tarea a realizar, ha sido obtener un modelo para el control MPC a partir de los parámetros de la planta y diseñar el control predictivo a través de la herramienta “MPC Tool “de Matlab. Para este diseño hemos realizado un estudio de sensibilidad en esta herramienta, variando el peso tanto de los mandos como de las salidas, hasta conseguir un control que se adaptara a nuestras necesidades.

Por último, hemos realizado la comprobación del control en simulación, haciendo uso del modelo matemático de la planta y del control diseñado.

Resultados y conclusiones

La comunicación del controlador con el sensor y actuador en la red Z-Wave ha resultado exitosa y sin mayores complicaciones.

En cuanto a la comunicación entre el microcontrolador y el controlador hemos tenido más complicaciones ya que el controlador se comunica a través de comandos http, mientras que el control (instalado en el microcontrolador), lo hace a través de comandos TCP/IP. Por ello hemos tenido que instalar el servidor diseñado en NODE JS anteriormente mencionado. Una vez instalado el servidor, no se ha encontrado mayor complicación.

Por otro lado, se han realizado varias simulaciones para comprobar la efectividad del control. Se ha simulado el seguimiento en referencia en una habitación, el rechazo de un cambio brusco de temperatura exterior brusco, una temperatura exterior senoidal simulando un día de invierno y por último se ha considerado la presencia de personas que aportan calor en una de las habitaciones. Tras estas simulaciones hemos comprobado que el control predictivo diseñado cumple con las expectativas, presentando robustez, tiempos de alcance y establecimiento adecuados, teniendo en cuenta las constantes de tiempo del sistema.

Futuros desarrollos

De cara a futuros desarrollos y proyectos, lo primero que habría que hacer sería implantar el control en la raspberry y realizar los ensayos pertinentes en la planta para obtener los parámetros exactos de la misma. Para que, de esta manera, poder rediseñar y precisar el control. Y finalmente implantarlo y dejarlo en funcionamiento probando así su efectividad.

Por otro lado, también se ha pensado realizar este mismo proyecto, pero con un control adaptativo e implantarlo en instalaciones domóticas con diferentes protocolos de comunicación inalámbricos como ISTEON o Zigbee.

Por último, investigar la probabilidad de instalar este control en otros sistemas de calefacción como la de suelo radiante.

DESIGN AND INTEGRATION OF A PREDICTIVE CONTROL OF TEMPERATURE IN A DOMOTIC SYSTEM

Student: Puente Iglesias, Sara

Director: Zamora Macho, Juan Luis

PROJECT SUMMARY

Abstract

The Information and communication technologies have experienced a boom in recent years. They are present in virtually all areas of our life including our own home.

There has always been a need to provide the home with a high degree of control, safety, comfort, versatility and savings. Home automation is the set of technologies applied to intelligent control and automation of housing and aims to provide a better quality of life in the usual place of residence.

A fundamental aspect in a home automation home is comfort and one of the main objectives of comfort is the correct temperature control. The main problems of current thermostats are: their excessive price and limited functionality.

Objectives:

This project aims to combat the main problems of current thermostats. Using Z-Wave technology and with a raspberry pi 3B + (market price € 33) it is intended to install a predictive temperature control in a water radiator system, giving added value to the home automation installation without making it more expensive.

In addition, this control is able to, from trials, draw a model of the plant, so that the control can be adapted to the specific needs of each home.

The main advantage of using a predictive control is, which allows the optimization of the execution to be applied taking into account the prediction of the future evolution of the temperature. In addition, you can take into account the economic aspect, if it is included in the objective function that is optimized.

System structure

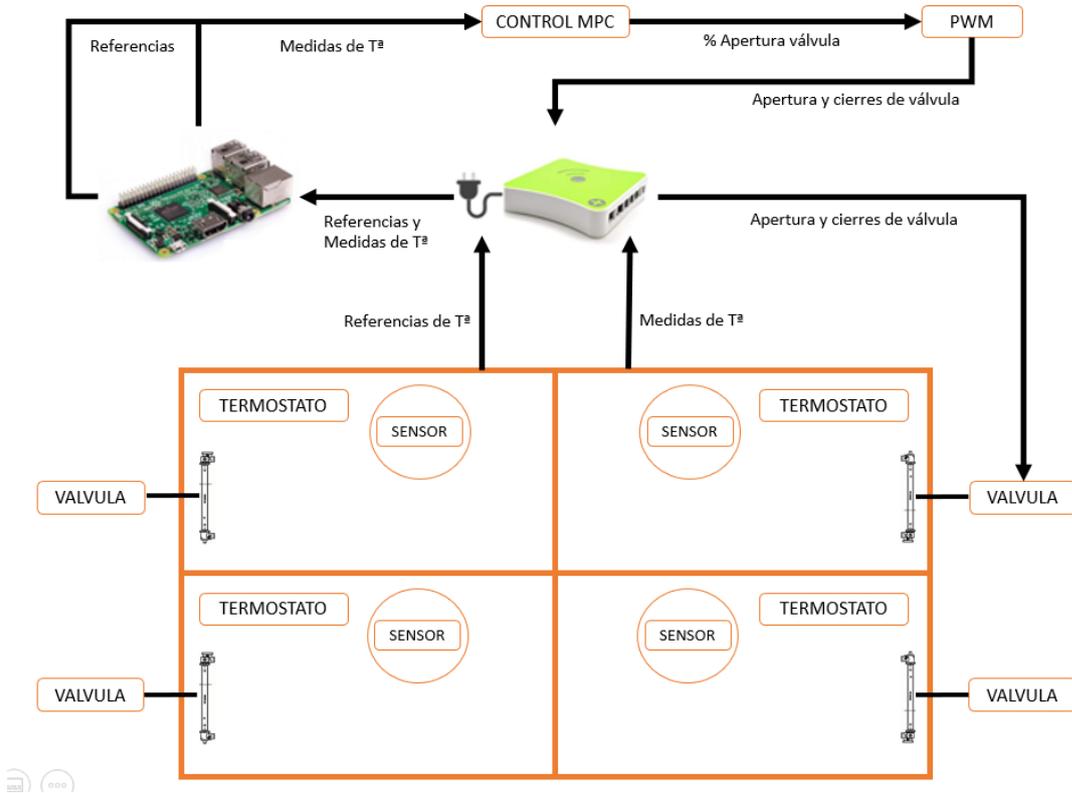


Figure 1. Project structure

In figure one, the scheme of the system that will follow the project is shown. The house has four rooms with radiator, a sensor will be placed in each of the rooms and an electrovalve in each radiator.

The control, designed and implanted in the raspberry, will request through the Eedomus controller the temperature measurement to the sensor, the sensor will send the temperature to the controller through a Z-Wave network which in turn will send it to the control.

The control will calculate the opening percentages of each valve and a pulse width modulation (PWM) of said percentages will be performed.

Finally, the control will indicate the controller when it must open and close each valve and the controller sending the order to each actuator.

Task done

The first thing that has been done in the project has been communications.

We have started with the communication of the controller with the sensor and the actuator. Simply, the sensor and the actuator have been included in the Z-Wave network created by the controller through the eedomus platform.

The second task that has been done has been to establish the communication of the controller with the microcontroller. The main problem of this communication has been

that the control communicates through TCP / IP commands, while the controller uses http commands. Therefore, we had to design a server in NODE JS and implement it in the raspberry. This server receives the http commands from the controller and transforms them into TCP / IP commands so that the control can understand. Upside down is the same, it transforms the control's TCP / IP commands to http commands that the controller can understand.

Next, we have made three state machines for communication: the general state machine, the sensor machine and the actuator machine.

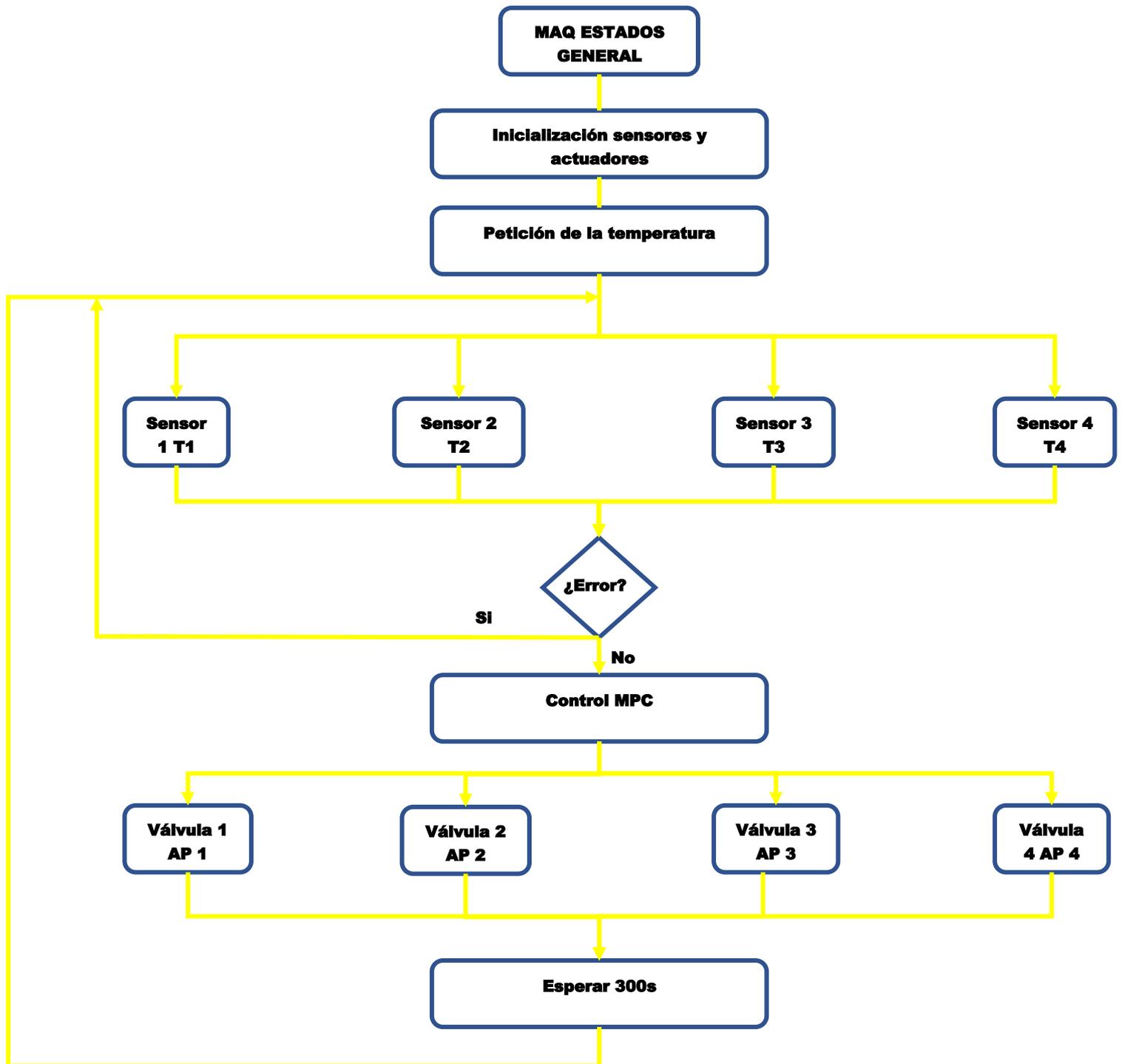


Figure 2. General state machine

General state machine:

As we can see in Figure 2, we will first initialize the sensors and actuators and then ask the four sensors for the temperature measurement, if there has been no error, all four temperatures will be sent to the control. The control will calculate the opening percentage of each room and send it to the corresponding actuators. Finally, a timer will wait for the control sampling period (5 min).

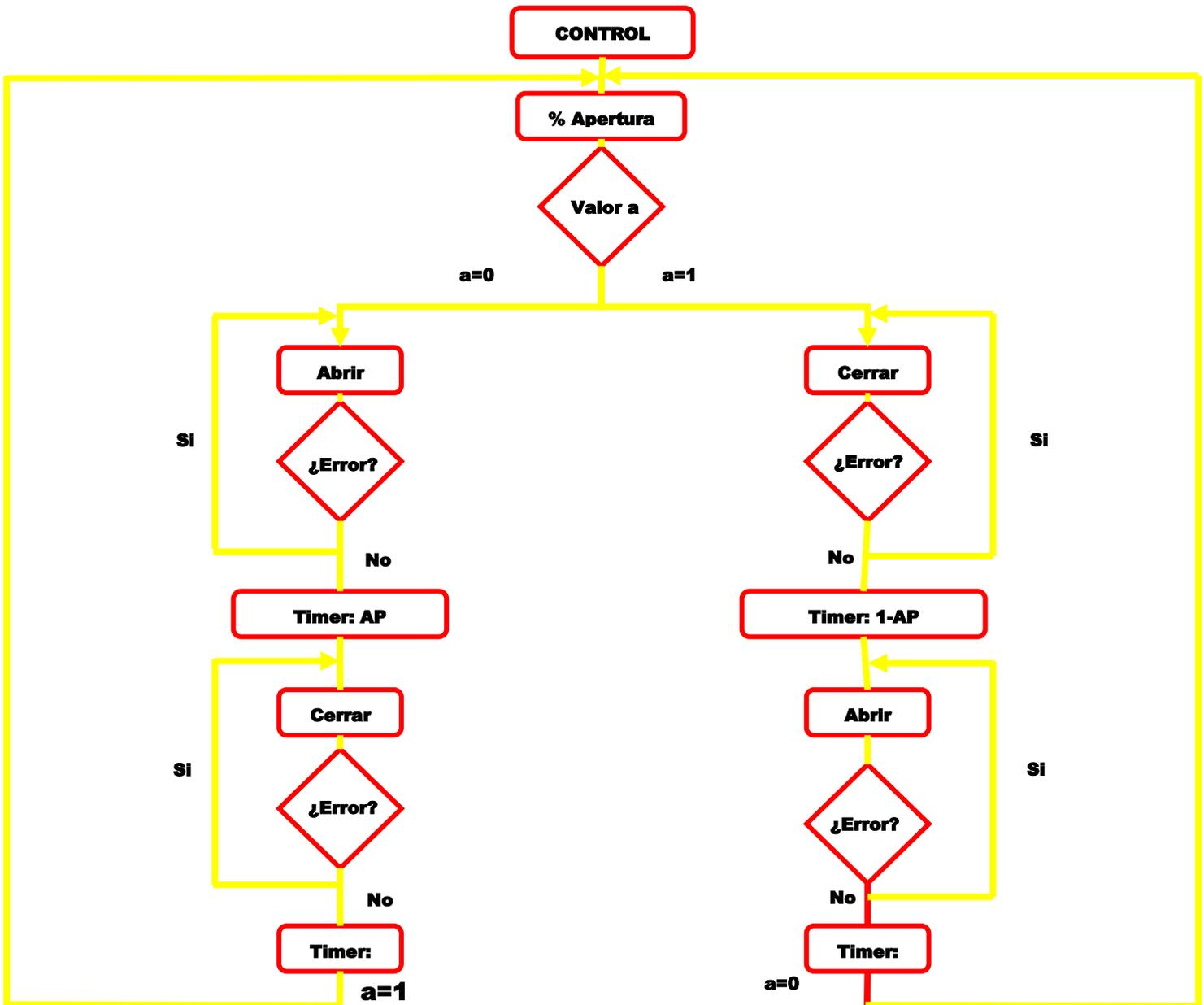


Figure 3. Valve status machine

Machine of valve states:

In order for the valve to open the opening percentage requested by the control, we have had to design a PWM module that opens the indicated percentage of the 5 min that the sampling period lasts and closes the rest of the 5 min.

To prevent excessive wear on the actuators and reduce valve openings and closures by half, we have taken into account the position in which the valve has finished in the previous action of the control. If the valve has ended open, in the next action, the control will start opening and if it has ended closed, it will start closing. This is exactly what the value a in Figure 3 indicates. If $a = 1$, it means that in the last control action the valve ended open, therefore, it will first open and then close. If instead $a = 0$, it means that the valve in its last action ended closed, therefore, it will first start closing and then opening.

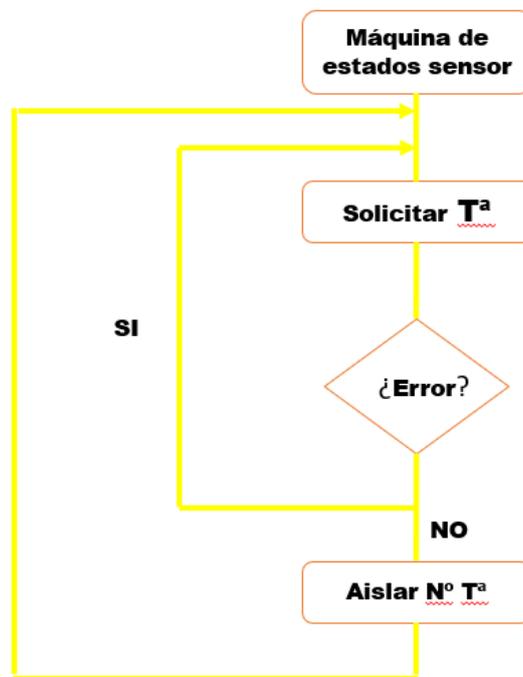


Figure 4. Sensor state machine.

Sensor state machine:

First, an http command is sent to the controller requesting the temperature measurement from the sensor. Finally, when the controller sends the response, the state machine isolates the temperature value to send it to the control.

The next task to be carried out has been to obtain a model for the MPC control from the parameters of the plant and design the predictive control through Matlab's "MPC Tool". For this design we have conducted a sensitivity study on this tool by varying the weight of both the controls and the outputs until achieving a control that suits our needs.

Finally, we have carried out the control check in simulation, using the mathematical model of the plant and the designed control.

Results and conclusions

The communication of the controller with the sensor and actuator in the Z-Wave network has been successful and without major complications.

As for the communication between the microcontroller and the controller we have had more complications since the controller communicates through http commands, while the contro does it through TCP / IP commands. Therefore we had to install the server designed in NODE JS mentioned above. Once the server was installed, no further complications were found.

On the other hand, several simulations have been carried out to verify the effectiveness of the control. Reference tracking has been simulated in a room, rejection of an abrupt change in outside temperature, a sine outside temperature simulating a winter day and finally the presence of people who provide heat in one of the rooms has been considered. After these simulations we have verified that the designed predictive control meets the expectations, presenting robustness, adequate reach and establishment times, taking into account the system time constants.

Future developments

In the face of future developments and projects, the first thing to do would be to implement the control in the raspberry and perform the relevant tests in the plant to obtain the exact parameters of the same. So that, in this way, to be able to redesign and specify control. And finally implant it and leave it in operation thus proving its effectiveness.

On the other hand, it has also been planned to carry out this same project but with adaptive control and implement it in home automation installations with different wireless communication protocols such as ISTEON or Zigbee.

Finally, investigate the probability of installing this control in other heating systems such as underfloor heating.



GRADO EN INGENIERÍA ELECTROMECAÁNICA

TRABAJO FIN DE GRADO
DISEÑO E INTEGRACIÓN DE UN CONTROL
PREDICTIVO DE TEMPERATURA EN UN SISTEMA
DOMÓTICO

Autor: Sara Puente Iglesias
Director: Juan Luis Zamora Macho

Madrid
Diciembre de 2019



Índice de contenidos

Capítulo I: Introducción.....	11
I.1 Motivación	11
I.2 Objetivos del proyecto	11
I.3 Metodología del trabajo	11
I.4 Recursos a emplear	12
I.5 Estructura de la memoria	13
Capítulo II: Estado del arte	15
II.1 Domótica	15
II.2 Configuración de las instalaciones domóticas	17
II.2.1 Sistemas centralizados	17
II.2.2 Sistemas descentralizados	18
II.3 Elementos de integración de la domótica	18
II.3.1 Sistemas de confort	19
II.3.2 Sistemas de seguridad	19
II.3.3 Sistemas de eficiencia energética	19
II.3.3 Sistemas de gestión de climatización	20
II.4 Controles	20
II.4.1 Control todo-Nada (ON/OFF)	20
II.4.2 Control	21
II.4.3 Control PID	21
II.4.4 Control predictivo (MPC)	22
Capítulo III: Sensores y actuadores	25
III.1 Comunicaciones	25
III.2 Cableados	25
III.3 Inalámbricos	25



III.4 Controladores	26
III.5 Sensores	28
III.6 Actuadores	30
III.7 Microcontroladores	31
III.8 Comunicaciones	33
III.8.1 Comunicación del controlador con el sensor y la electroválvula.....	33
III.8.2 Comunicación del controlador con control.....	34
Capítulo IV: Control MPC	49
IV.1 Control predictivo	49
IV.2 Diseño de planta	51
IV.2.1 Modelo matemático	52
IV.3 Diseño del control predictivo	58
IV.4 Estudio de sensibilidad	62
Capítulo V: Simulaciones	69
V.1 Simulación 1. Seguimiento a la referencia	69
V.2 Simulación 2. Rechazo a la perturbación T0	74
V.3 Simulación 3. Temperatura exterior real	78
V.4 Simulación 4. Rechazo a la perturbación Pd.....	82
Capítulo VI: Conclusiones	89
VI.1 Comunicaciones	89
VI.1.1 Comunicación del controlador con el sensor y el actuador	89
VI.1.2 Comunicación del controlador con Simulink	89
VI.2 Diseño del control de temperatura	90
VI.3 Simulación y prueba de control	90
Capítulo VII: Futuros desarrollos	91
Capítulo VIII: Referencias y bibliografía	93



Índice de figuras

Figura 1: Ejemplo de una vivienda domótica	16
Figura 2: Ejemplo de configuración domótica centralizada	17
Figura 3: Ejemplo de configuración domótica descentralizada	18
Figura 4: Ejemplo de controlador “todo / nada”	21
Figura 5: Ejemplo de un sistema de control de temperatura PID	22
Figura 6: Controlador Eedomus	28
Figura 7: Sensor Fibaro	30
Figura 8: Válvula Eurotronics	31
Figura 9: Raspberry pi 3 B+	32
Figura 10: Plataforma Eedomus	33
Figura 11: Añadir periférico	33
Figura 12: Simulink general	34
Figura 13: Hardware simulink	35
Figura 14: Comunicaciones Simulink	35
Figura 15: Transmisión simulink	36
Figura 16: Bloque TCP/IP send	36
Figura 17: Recepción Simulink	37
Figura 18: Bloque TCP/IP receive	37
Figura 19: Sensores Simulink	38
Figura 20: Máquina de estados del sensor	38
Figura 21: Actuadores Simulink	42
Figura 22: Máquina de estados del sensor	42



Figura 23: Elementos del MPC	50
Figura 24: Plano habitaciones	51
Figura 25: Modelo Simulink	54
Figura 26: Bus modelo	55
Figura 27: MPC Controller de Simulink	58
Figura 28: MPC Structure	60
Figura 29: Ajustes del control	60
Figura 30: Restricciones de las entradas y salidas	61
Figura 31: Pesos MPC	62
Figura 32: Escenario de simulación de MPCDesigner	63
Figura 33: Entradas simulación 1 MPCDesigner	64
Figura 34: Salidas simulación 1 MPCDesigner	65
Figura 35: Entradas simulación 2 MPCDesigner	66
Figura 36: Salidas simulación 2 MPCDesigner	67
Figura 37: PWM simulación 1	70
Figura 38: Potencias simulación 1	71
Figura 39: Temperaturas simulación 1.....	72
Figura 40: PWM simulación 2	75
Figura 41: Potencias simulación 2	76
Figura 42: Temperaturas simulación 2.....	77
Figura 43: PWM simulación 3	79
Figura 44: Potencias simulación 3	80
Figura 45: Temperaturas simulación 3.....	81



Figura 46: Pd simulación 4.....	83
Figura 47: PWM simulación 4	84
Figura 48: Potencias simulación 4	85
Figura 49: Temperaturas simulación 4.....	86



Índice de tablas

Tabla 1: Resistencias térmicas	57
Tabla 2: Capacidades térmicas	57
Tabla 3: Potencias nominales de los radiadores	58
Tabla 4: Pesos de los mandos	61
Tabla 5: Pesos de las salidas	61
Tabla 6: Pesos entradas simulación 4	65
Tabla 7: Pesos salidas simulación 4	65
Tabla 8: Pesos de los mandos	69
Tabla 9: Pesos de las salidas	69



Índice de ecuaciones

Ecuación 1: Entradas de la representación de estados	52
Ecuación 2: Variables de estados	52
Ecuación 3: Salidas de la representación de estado	52
Ecuación 4: Almacenamiento de energía térmica en un medio	52
Ecuación 5: Potencia con la que se almacena energía en el medio n	52
Ecuación 6: Transmisión de energía térmica entre dos medios	53
Ecuación 7: Potencia con la que se transmite energía del medio m al n	53
Ecuación 8: Potencia generada en la habitación n	53
Ecuación 9: Balance de potencia habitación 1	53
Ecuación 10: Balance de potencia habitación 2	53
Ecuación 11: Balance de potencia habitación 3	54
Ecuación 12: Balance de potencia habitación 4	54
Ecuación 13: Representación de estado	54
Ecuación 14: Ecuación de estado en régimen permanente	73



Capítulo I: INTRODUCCIÓN

I.1 MOTIVACIÓN

El campo de la domótica es muy amplio y con mucho recorrido. En la actualidad aún queda mucho por aportar en este ámbito, empezando por la posibilidad de acercar esta tecnología al alcance de todos. Los principales problemas de los controles de temperatura actuales son: el excesivo precio, la simplicidad, su funcionalidad limitada y que su implementación resulta complicada. Este proyecto, utilizando la tecnología z-wave, trata de desarrollar un sistema de control de calefacción de bajo coste y de fácil instalación en cualquier recinto. Implementando un control predictivo es posible conseguir un sistema fiable, eficaz y robusto.

El sistema de calefacción en el que se empleará el control es el de los radiadores de agua, el más habitual en las viviendas con lo que el proyecto tiene un amplio rango de expansión.

Su instalación es extremadamente fácil, ya que con el controlador domótico simplemente es necesario colocar un sensor de temperatura por habitación y una electroválvula por radiador.

I.2 OBJETIVOS DEL PROYECTO

Los objetivos del proyecto son los siguientes:

1. Establecer las comunicaciones entre el microcontrolador en el que se implanta el control predictivo y el controlador domótico.
2. Diseño del control predictivo de temperatura a partir de un modelo de la planta definido mediante ensayos.
3. Prueba del control mediante simulaciones

I.3 METODOLOGÍA DE TRABAJO

Para alcanzar los objetivos anteriores se realizarán las siguientes tareas:

- Primero se establecerá la correcta comunicación entre el sensor de temperatura y la válvula de actuación con el controlador domótico. A continuación, comunicaremos el microcontrolador con nuestro controlador domótico.



- Diseño del control predictivo de temperatura a partir de un modelo de la planta definido mediante ensayos. Primero estudiaremos la documentación de un control predictivo mientras realizamos los ensayos en la vivienda para obtener un modelo.
- de la planta. Posteriormente diseñaremos el control predictivo en el entorno de Matlab y Simulink.
- Por último, se realizará la implantación y las pruebas del control tanto en simulación como en la realidad.

Desglosando las tareas por orden cronológico:

Primero estudiaremos la documentación tanto del controlador eedomus como de la raspberry y a continuación estableceremos las comunicaciones entre ambos. Después comunicaremos el sensor de temperatura con la válvula de actuación. Una vez acabadas las comunicaciones daremos paso al diseño del control.

Seguidamente se estudiará la documentación del control predictivo y se diseñará el modelo matemático de la planta. Por último, diseñaremos el control. Para finalizar implantaremos el control y lo probaremos mediante simulación.

1.4 RECURSOS A EMPLEAR

Los recursos a emplear para la realización del proyecto serán los siguientes:

Hardware:

Se encargan de la organización de toda la red y pueden comunicarse con cualquier nodo. El controlador es el encargado de crear la red y solo puede haber uno por red z-wave.

Es el que permite comunicar la instalación domótica con internet y permiten la creación de escenas y ambientes. Se encuentran siempre encendidos y consumen muy poco.

- **Controlador Eedomus:** El controlador es el encargado de crear la red, se encarga de la organización de toda la red y pueden comunicarse con cualquier nodo y comunica la instalación domótica con internet.
- **Sensor Fibaro FGMS-001-ZW5:** Es un multisensor z-wave universal y es capaz de detectar movimiento, medir la temperatura además de la intensidad de la luz. Su fuente de alimentación es una batería y su diseño permite una instalación fácil y rápida en cualquier superficie; el indicador led indica el movimiento, nivel de temperatura, el modo de funcionamiento y podemos usarlo para ver si el dispositivo se encuentra dentro de la red z-wave. El sensor de movimiento permite iluminar escenas y sistemas de monitoreo de presencia. [17]



- **Cabezal termostático Eurotronic Spirit:** Es un cabezal termostático Z-wave. Se utiliza en radiadores de agua dotados de válvula termostática y permite el control de temperatura en cada estancia cortando el suministro de agua caliente al alcanzar la temperatura seleccionada. [18]
- **Raspberry pi 3 B+:** programada en el entorno Matlab, es un ordenador del tamaño de una tarjeta de crédito que funciona gracias a un procesador similar al que incorporan los terminales Smartphone. [19]

Software:

- Matlab, Simulink y toolbox de control predictivo.

I.5 ESTRUCTURA DE LA MEMORIA

Capítulo 2: En esta parte del proyecto se hace referencia al estado del arte en el cual se hace hincapié en tres puntos muy importantes como son la domótica, la climatización y el control predictivo.

Capítulo 3: En dicho capítulo se hará una descripción del software, se mostrará la configuración y la programación tanto de los sensores como de los actuadores.

Capítulo 4: En él se explica el modelo térmico para la vivienda y que usará en el control predictivo de temperatura. Además, se describe el control y diseño de los parámetros mediante la Tool box del control predictivo de Matlab.

Capítulo 5: Se habla de los resultados obtenidos en la simulación para distintos ensayos con cambios en referencia y perturbación.

Capítulo 6: Conclusiones.

Capítulo 7: Futuros desarrollos.



Capítulo II: ESTADO DEL ARTE

II.1 DOMÓTICA

El termino domótica (tal y como lo conocemos hoy en día) tiene su origen al momento de unir las palabras domus, que en latín significa casa, y tica, automática en griego. Esta palabra se define por la RAE (Real Academia Española de la Lengua) como el conjunto de sistemas encargadas de automatizar las distintas instalaciones de una vivienda.

Nuestra sociedad está experimentando un avance tecnológico sin precedentes y a una gran velocidad, afectando a todos los ámbitos de nuestra cotidianidad incluyendo algo tan básico como la vivienda que demanda necesidades hasta ahora impensables y que afectan a su diseño, construcción, red de telecomunicaciones, productos domésticos, sostenibilidad y prestación de servicios.

Las tecnologías de la información y la comunicación (TIC) han experimentado un gran auge durante éstos últimos años. Están presentes en prácticamente todos los ámbitos de nuestra vida incluyendo nuestro propio hogar.

Siempre ha existido la necesidad de dotar a la vivienda de un alto grado de control, seguridad, confort, versatilidad y ahorro. La Domótica es el conjunto de tecnologías aplicadas al control y la automatización inteligente de la vivienda y tiene como objetivo ofrecer una mejor calidad de vida en el lugar habitual de residencia. [1]

Un aspecto fundamental en el confort de nuestro hogar es la temperatura. Muchos de los termostatos actuales son bastante simples basándose en un sistema de control "todo-nada", es decir a partir de una temperatura de referencia encienden o apagan la calefacción en función de si se encuentra por encima o por debajo de la temperatura de consigna.

El objetivo de este proyecto consiste en implantar un control predictivo de temperatura dentro de un entorno domótico ya que la mayoría de los controladores actuales que hay en el mercado no cuentan con esa funcionalidad.

La ventaja principal de utilizar un control predictivo es que permite optimizar la actuación a aplicar teniendo en cuenta la predicción de la evolución futura de temperatura. Además, puede tener en cuenta el aspecto económico, si se incluye en la función objetivo que se optimiza. [2]

Entrando un poco en detalle se puede decir que un sistema domótico, está formado por una serie de equipos y una red de comunicaciones (bus domótico) que permite el intercambio de información entre estos, a través de interfaces, mediante protocolos de comunicación. Dentro del término equipo podemos englobar los siguientes dispositivos:

- Elementos de campo (sensores, captadores, detectores, etc.) los cuales son los encargados de recoger la información del entorno convirtiendo las magnitudes físicas en señales eléctricas.
- Unidad de control, la cual mediante una programación interna procesa la información recibida como señales eléctricas por los elementos de campo, gestionando el intercambio de información. El resultado es mandar que se ejecuten las acciones deseadas.
- Terminales (puntos de luz, electrodomésticos, calefacción, etc.) a los que se les aporta cierta inteligencia o capacidad de comunicación a través de un interfaz. Son los responsables de llevar a cabo las acciones mandadas por la unidad de control.
- Módulo de E/S (llamados acopladores o interfaces), cuya misión es pasar las señales de control, codificadas según protocolos de comunicación en buses domóticos a señales de salida interpretables por los actuadores.
- Hasta ahora, en un edificio todos los sistemas eran independientes, sin embargo, hoy en día la domótica ofrece soluciones y posibilidades para integrar cada uno de los componentes entre si, lo que supone una gran ventaja para el usuario.

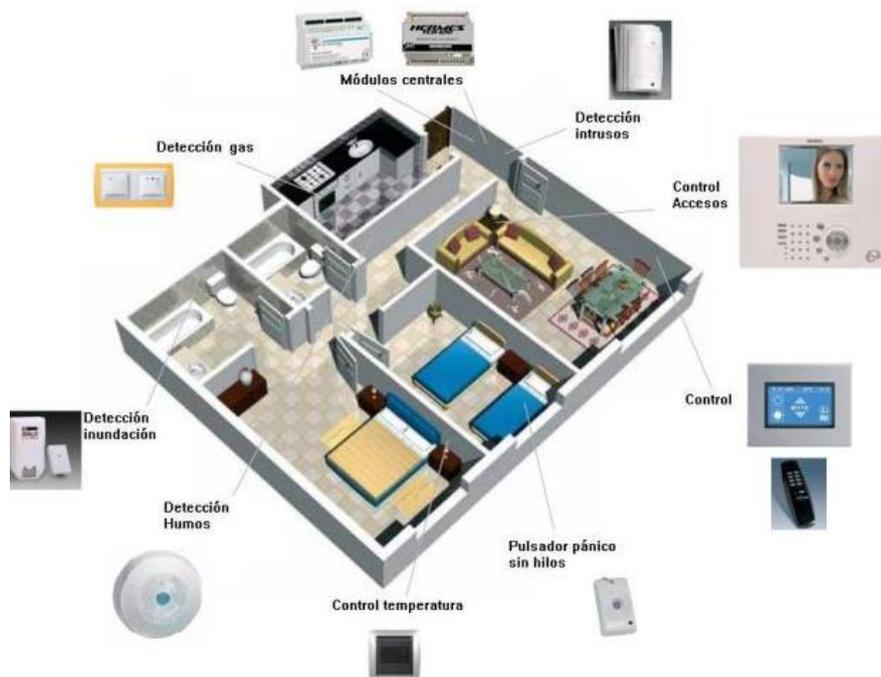


Figura 1. Ejemplo de una vivienda domótica



El uso de la domótica en la vivienda se puede aplicar a una extensa gama de aplicaciones como la seguridad, la automatización, comunicación con servidores externos, etc.

En este TFG aplicaremos la domótica con el fin de gestionar un sistema de calefacción inteligente en una vivienda que permita aumentar el confort, la seguridad y el ahorro energético.

II.2 CONFIGURACIÓN DE LAS INSTALACIONES DOMÓTICAS

Es importante señalar que un sistema domótico puede trabajar de forma centralizada o descentralizada. En este primer caso, el funcionamiento del sistema depende de la programación introducida en la central domótica. Por el contrario, en los sistemas descentralizados se dota de inteligencia a cada uno de los elementos, programándose de manera individual cada uno.

II.2.1 SISTEMAS CENTRALIZADOS

Son aquellos sistemas cableados y centralizados que requieren de una instalación específica, lo que supone un coste superior a una instalación convencional. El mayor inconveniente de estas instalaciones actualmente es la gran cantidad de cableado que se ha de realizar entre los sensores y actuadores y la unidad de control. Además, existe la posibilidad de que el sistema se venga abajo en el caso de que la unidad de control falle. La utilización de ordenadores y autómatas programados en una solución versátil pero que requiere de una instalación y configuración individualizada por lo que nos es utilizable de manera general, lo que encarece aún más su instalación.

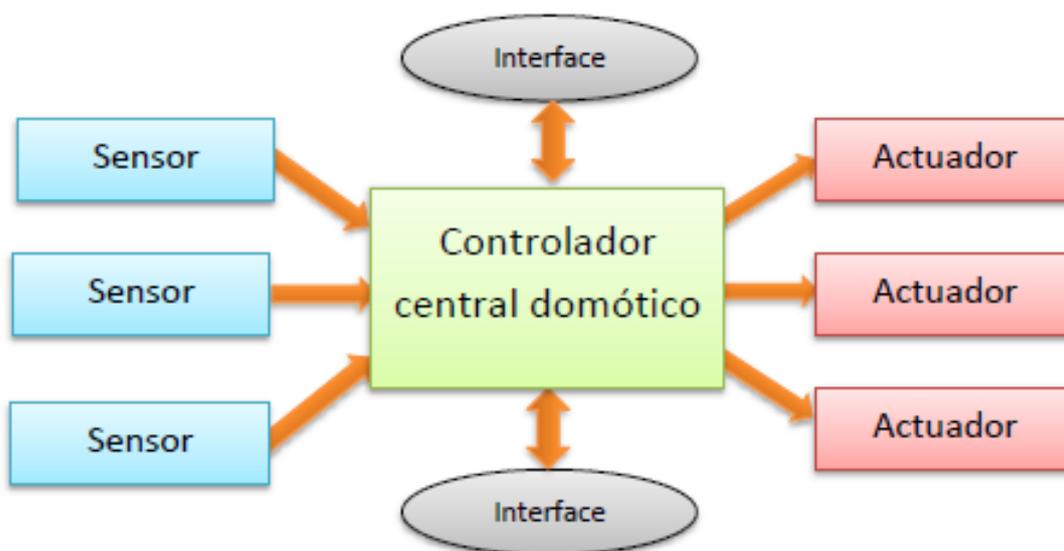


Figura 2. Ejemplo de configuración domótica centralizada

II.2.2 SISTEMAS DESCENTRALIZADOS

En los sistemas descentralizados no existe un único dispositivo que tome decisiones, sino que son varios repartidos a lo largo de la instalación. La instalación física se simplifica respecto a los sistemas centralizados y la modificación de la instalación a posteriori es más sencilla. Otra de las ventajas respecto a una instalación centralizada es que el fallo en una de las unidades de control no deja inutilizada la instalación al completo sino una parte de ella. Por lo tanto, la tendencia en la domótica ha sido la de desarrollar instalaciones descentralizadas ya que ofrecen muchas ventajas con respecto a las instalaciones centralizadas.

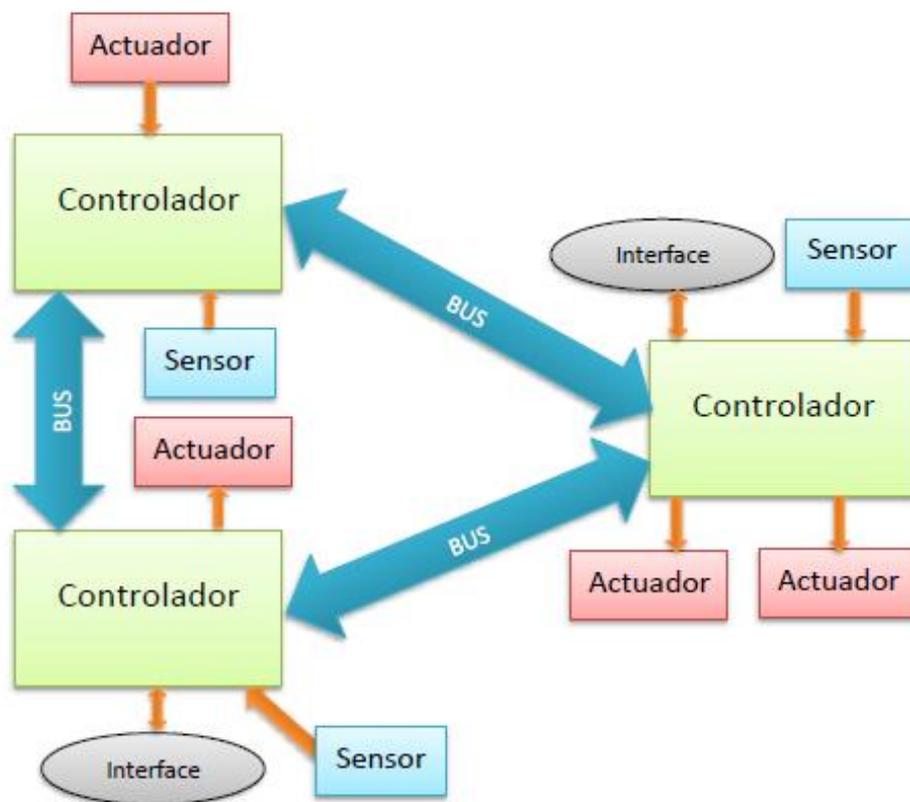


Figura 3. Ejemplo de configuración domótica descentralizada

II.3 ELEMENTOS DE INTEGRACIÓN DE LA DOMÓTICA

La integración es uno de los principales retos de la domótica puesto que los diferentes sistemas no suelen pertenecer al mismo fabricante y actualmente no existe una estandarización, por lo que, la integración de los diferentes sistemas que integran la domótica son un hándicap de las instalaciones. A continuación, describiremos algunos de los sistemas más comunes de la domótica.



II.3.1 SISTEMAS DE CONFORT

Los sistemas de automatización domotica juegan un papel de vital importancia para que las condiciones de confort sean las óptimas y adecuadas. Por tanto, la domótica en este campo trata de maximizar la comodidad de los usuarios creando ambientes agradables,

así como simplificando tareas. Para conseguir dicho confort podemos usar diversas automatizaciones como, por ejemplo:

- Control remoto de la iluminación.
- Automatización de encendido y apagado de diferentes dispositivos.
- Control y monitorización mediante elementos como el teléfono móvil.
- Encendido y apagado de la climatización de manera inteligente.

II.3.2 SISTEMAS DE SEGURIDAD

Los sistemas de seguridad domóticos son importantes a la hora de dotar a la vivienda de máxima protección posible frente a amenazas como robos en edificios o viviendas. Sensores, alarmas o control de cámaras forman parte de la domótica y hacen que nos sintamos mucho más seguros.

II.3.3 SISTEMA DE EFICIENCIA ENERGÉTICA

Hoy en día la concienciación para la reducción del consumo energético es fundamental para la disminuir la contaminación ambiental.

La implantación de la domótica en las viviendas y edificios es cada vez más común, lo que permite aumentar el confort y el ahorro energético. De hecho, la primera función de la domótica en la vivienda fue destinada a aumentar el confort de la misma, permitiendo gestionar de manera autónoma y a distancia las diversas funciones. Sin embargo, cada vez más, la domotica está siendo usada en las viviendas para aumentar el ahorro energético. Con esta finalidad de ahorro energético la domótica es usada para acciones tales como, limitar el consumo de energía en ciertas franjas horarias, desconectando los dispositivos innecesarios en algunos momentos, realizando la gestión inteligente de iluminación y climatización teniendo en cuenta entre otros factores el nivel de ocupación de la estancia, etc.



II.3.4 SISTEMA DE GESTIÓN DE CLIMATIZACIÓN

Uno de los mayores usos dados a los sistemas domóticos es el control de la climatización mediante la regulación de los sistemas de calefacción y aire acondicionado. El uso de un único termostato para toda la vivienda es el más habitual, sin embargo, es el menos eficiente. El control de la climatización mediante la domótica se basa en:

- **Programación:** se usa una programación inteligente en función de los días de la semana y las horas del día. Con ello conseguimos que la temperatura se mantenga adecuada en los momentos que no se está en la vivienda y aumente o disminuya a la temperatura de confort en los momentos de utilización de la misma.
- **Zonificación:** se utiliza para conseguir diferentes temperaturas en distintas estancias de la vivienda. Dejando por tanto una temperatura de mantenimiento en las estancias no utilizadas.

II.4 CONTROLES

Uno de los factores que más influyen en el confort de una vivienda es la climatización. En la actualidad existen múltiples sistemas de control de temperatura para una vivienda, pero en su mayoría producen fluctuaciones de temperatura que afectan al confort de los usuarios. Los más utilizados son los controles “Todo/Nada” y el PID. Otros sistemas de control como el predictivo no son habitualmente utilizados en los controladores domésticos. A continuación, se da una breve explicación de cada uno de ellos.

II.4.1 CONTROL TODO-NADA (ON-OFF)

Es la forma más simple de control. La salida del regulador se encuentra encendida o apagada sin ningún estado intermedio. El control sólo cambia la salida cuando la temperatura atraviesa la temperatura de referencia o consigna. Por ejemplo, para el calentamiento, el control se enciende si la temperatura se encuentra por debajo y se apaga si se encuentra por encima de la consigna de la temperatura.

Para evitar el uso intensivo de los actuadores se añade un diferencial a la temperatura de referencia llamado "histéresis" de manera que exige que la temperatura exceda la de referencia en una cierta magnitud antes de que se active o desactive de nuevo.

Este tipo de control se utiliza en sistemas pocos precisos y que no pueden soportar cambios frecuentes de encendidos y apagados donde las temperaturas cambian muy lentamente. Otra aplicación muy frecuente es en la gestión de alarmas. [3]

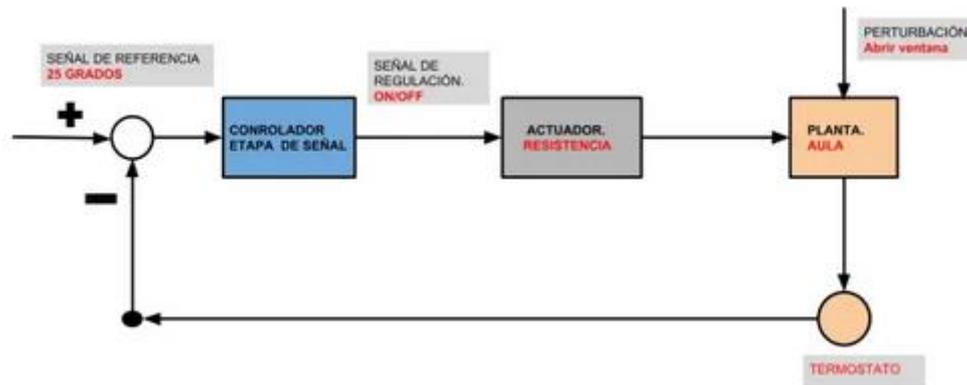


Figura 4. Ejemplo de controlador “todo / nada”

II.4.2 CONTROL P

El control proporcional disminuye la potencia media conforme se va aproximando al punto de referencia, esto produce una disminución de energía y mantiene la temperatura estable, la acción proporcional se aplica en una banda en torno a la temperatura de referencia. Fuera de esa banda el control se comporta como un todo-nada.

Dentro de la banda, la actuación depende de la posición en donde se encuentre la temperatura. Si se encuentra justo en la temperatura de referencia (en medio de la banda) la relación de encendido apagado será 1:1, es decir el tiempo de encendido y apagado es el mismo. Si en cambio la temperatura se encuentra en el extremo inferior de la banda el actuador estará encendido más tiempo. Si se encuentra justo en el extremo superior estará más tiempo apagado.

Aunque es cierto que este control proporciona una buena estabilidad y responde rápidamente, es muy sensible al ruido y puede presentar oscilaciones en la variable controlada, alterando el confort de los usuarios. [3]

II.4.3 CONTROL PID

Se trata de un control que combina las tres acciones: proporcional integral y derivativa. Calcula la desviación o error entre un valor de consigna de temperatura y la medida de la misma. Aplica las tres acciones anteriores con pesos diferentes y como resultado se obtiene la potencia media que debe que suministrar el calentador. Proporciona tiempos de respuesta rápidos, pero una mayor precisión resulta bastante sensible al ruido y puede introducir oscilaciones si se presentan retardos en el sistema.

Cuando nos encontramos con un sistema no lineal, el PID no tiene la capacidad de compensar esta no linealidad. [4]

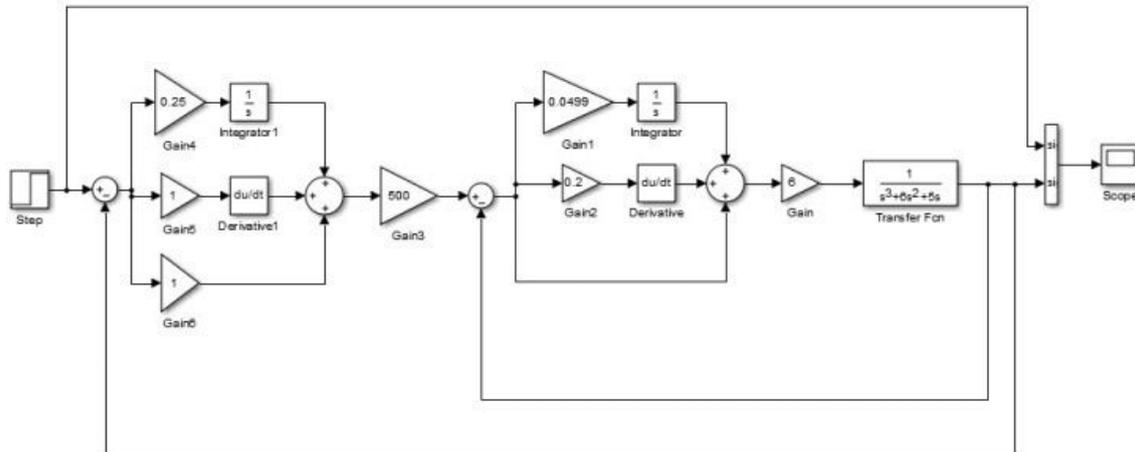


Figura 5. Ejemplo de un sistema de control de temperatura PID

II.4.4 CONTROL PREDICTIVO (MPC)

Este modelo se basa en el comportamiento futuro del sistema que se predice gracias a un modelo dinámico del mismo llamado modelo de predicción. Una de las grandes ventajas de este control es que permite la incorporación de distintos modelos de predicción, sean lineales o no lineales, mono variables o multivariables haciendo de él una estrategia muy utilizada en diversos ámbitos del control. [5]

El control predictivo permite calcular el valor estimado que deberá tomar el mando en los siguientes intervalos de tiempo. Este control es muy fiable y permite operar mucho tiempo sin intervenciones.

Pueden darse también versiones adaptativas donde el modelo de predicción cambia con el tiempo. La estrategia de este tipo de controles podemos resumirla en los siguientes pasos [6]:

1. Para cada instante, mediante el modelo de predicción utilizado (planta) se calculan las futuras salidas del proceso en función de los mandos. Esta predicción depende de salidas y entradas pasadas y las hipotéticas entradas de control futuras.
2. Las futuras acciones de control se calculan de manera que minimicen una función objetivo. Si la función objetivo es cuadrática y el modelo de predicción es lineal, se puede encontrar una solución analítica para sistema de control.
3. Se utiliza la estrategia de “horizonte móvil” donde se aplica el mando actual y se desplaza en el tiempo mostrando el horizonte de predicción.



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
GRADO EN INGENIERÍA ELECTROMECÁNICA



Se han realizado diversos estudios sobre la efectividad del control predictivo para la temperatura como por ejemplo en la regulación térmica de edificios en donde se utilizan una estructura de control predictivo para reducir el consumo de energía sin disminuir el confort térmico. [7], [8], [9]



Capítulo III: SENSORES Y ACTUADORES

III.1 COMUNICACIONES

Los diferentes dispositivos que configuran un sistema domótico. Pueden clasificarse en diferentes grupos según su funcionalidad: sensores, actuadores, controladores, BUS e interfaz.

En la actualidad, al haber cada vez más hogares interconectados, las comunicaciones entre dispositivos juegan un papel muy importante. Esa comunicación emplea un idioma propio al que denominamos “PROTOCOLO” [10], que no es más que un conjunto de normas que permiten la comunicación entre ordenadores. Existe una gran variedad de protocolos para una instalación domótica: cableados (cable PLC, cable Bus), inalámbricos (Z-Wave, Zigbee, Delta, Dore, Insteón, Homekit, y Bluetooth).

III.2 CABLEADOS

Los protocolos cableados a la red eléctrica de mayor utilización son [11]:

X-10 (Cable PLC)

Es uno de los protocolos más antiguos utiliza las redes de cableado existentes en la casa para transmitir las señales. Su principal ventaja es su bajo coste ya que la instalación necesaria es mínima. En contra, ofrece una baja velocidad de comunicación debido a la frecuencia de la corriente alterna con la que trabaja la red

KNX (Cable BUS)

Este protocolo nació de la unión de tres estándares previos (EHSA, EIBA y BCI). Se comunica mediante cableado a través de un bus de datos, para instalarlo hay que cablear todas las zonas que deseemos controlar por lo que resulta más caro que el x-10. A su favor es capaz de instalar hasta un total de 64 dispositivos, todos ellos comunicados entre sí sin necesidad de centros de control, es decir, cualquier componente puede mandar sobre otro sin ningún tipo de problema ya que se encuentran interconectados.

III.3 INALÁMBRICOS

La clara ventaja que tienen los sistemas inalámbricos frente a los cableados es que necesitan ningún tipo de obra e instalación en la vivienda. Los más utilizados en el mercado son:



Zigbee

Utiliza la radiofrecuencia para la señalización y el control. Es óptimo en redes de baja tasa de transferencia de datos. [12] Algunas de sus ventajas son que reduce tiempos de espera en el envío y la recepción, proporciona larga duración en la batería y soporta múltiples topologías de red. Su principal inconveniente es su elevado precio si se compara con sus competidores Z-wave e Insteon.

INSTEON

La principal ventaja de este protocolo es su doble malla de red, es decir, que utiliza tanto señales de radiofrecuencia como señales a través de cableado para comunicarse. [13] En la red cada dispositivo actúa como repetidor, de esta manera cuantos más dispositivos haya conectados más efectivo se vuelve el sistema domótico.

Z-wave

La tecnología z-wave es la que se va a utilizar en este proyecto. Utiliza un protocolo de comunicaciones por radiofrecuencia que permite la monitorización remota y el control específico de aplicaciones tanto en ambientes residenciales como comerciales. [14], [15] Al tratarse de tecnología inalámbrica de baja potencia se utiliza para aplicaciones que requieren un mínimo consumo de energía y una baja tasa de transferencia de datos. Trabaja en la banda de los 868MHz evitando la gran cantidad de emisoras en la banda de los 2,4GHZ y puede operar en rangos de hasta 30m en condiciones ideales. Algunas de las ventajas en la utilización de este tipo de tecnologías es que al tratarse de un protocolo estándar consigue un gran nivel de compatibilidad con diferentes marcas, es fácil de instalar y al emplear baja frecuencia tiene un gran alcance; como inconveniente cabría destacar su limitación en cuanto al número de dispositivos que se podrían conectar simultáneamente a la red y el elevado consumo eléctrico que conlleva ya que el hardware está permanentemente activo. La topología de la red es tipo malla y cada elemento se comporta como un nodo que puede ser emisor y receptor [20].

El protocolo Z-Wave tiene dos tipos básicos de dispositivos: los controladores y los esclavos.

Los dispositivos controladores son los nodos que inician los comandos de control y los envían a otros nodos. Los esclavos son los que reciben estos comandos de control contestan y los ejecutan. A su vez los esclavos pueden reenviar comandos a otros nodos permitiendo al controlador comunicarse con nodos fuera de su alcance directo por radio. A nivel funcional los dispositivos utilizados en domótica se pueden dividir en:

III.4 CONTROLADORES

Se encargan de la organización de toda la red y pueden comunicarse con cualquier nodo. El controlador es el encargado de crear la red y solo puede haber uno por red z-wave. Es



el que permite comunicar la instalación domótica con internet y permiten la creación de escenas y ambientes. Se encuentran siempre encendidos y consumen muy poco. De los controladores de la tabla los más competitivos son:

EEDOMUS +

Es uno de los mejores controladores calidad-precio y el que se utiliza en este proyecto. Trabaja con la red z-wave, el protocolo más estandarizado de la domótica inalámbrica, con el trabajan la mayoría de las marcas del sector dando solución a cualquier problema que pueda surgir. La distancia al controlador nunca será un problema, ya que los dispositivos, además de ser muy fácil de emparejar actúan de repetidores. Otra de sus ventajas es que su interfaz es bastante intuitivo y recibe bastantes actualizaciones que le añade nueva funcionalidad. Dichas actualizaciones son muy importantes ya que van puliendo el sistema y corrigiendo los pequeños errores. Tiene varios puertos USB lo que le permite ser compatible con otros protocolos y ampliar funcionalidades. [20]

Características:

- Controlador Z-wave plus GEN5
- Alimentación: DC 5V USB (adaptador suministrado)
- Funciona con la mayoría de los módulos Z-Wave
- Control total de dispositivos Z-Wave
- Interfaz de usuario amigable
- Administración multilingüe
- Soporta varias cámaras IP de soporte FTP
- Bajo consumo de Potencia (1W)
- 2 puertos USB para dispositivos con cable
- 4 puertos RJ11 para trabajar con dispositivos de otros fabricantes
- Abierto para los desarrolladores (API eedomus)
- Actualización del firmware de forma remota

Características técnicas:

- Tipo de Módulo: Z-Wave Controlador
- CPU: Samsung ARM9
- Procesador: ARM Cortex-A8 a 1 Ghz
- RAM: 512 Mb DDR3
- Almacenamiento: 4Gb eMMC
- OS: Linux
- Fuente de alimentación: AC 100/ 265V, 50/69 Hz/ adaptador 5V/2ª DC (incluido)
- Frecuencia: 868.42
- Alcance: hasta 30m
- Puertos Ethernet: 1
- Puertos USB: 2
- RJ11:4

- Consumo: 1.5 W
- Audio: altavoz 1W+micrófono de ambiente
- Uso en interiores, temperatura 0-40° C, humedad relativa 80% sin condensación
- Dimensiones: 137mm x 137mm x 29 mm
- Peso 300 g



Figura 6. Controlador Eedomus

Vera Plus

Este controlador tiene de base el protocolo zigbee. Posee una configuración fácil e intuitiva y es capaz de sincronizarse con dispositivos Bluetooth. Este controlador es más económico que el eedomus, pero es mucho más simple y limitado.

III.5 SENSORES

Los sensores o también denominados “dispositivos de entrada” se encargan de recibir la información del entorno o de la propia instalación. Entre los sensores más comunes en las instalaciones domóticas podemos destacar:

Detectan sucesos que ocurren en una vivienda y transmiten los valores detectados al controlador. También pueden controlar otros dispositivos Z-Wave directamente mediante una función de asociación. Suelen utilizar pilas y tienen una vida útil de la batería de uno a tres años dependiendo de su uso.

Entre los sensores más comunes en las instalaciones domóticas podemos destacar: termostato, sensor de temperatura, sensor de agua, sensor de iluminación, sonda de humedad, detector de fugas de gas, detector de humo y/o fuego, sensor de presencia. Son de particular interés los multisensores, los cuales un solo dispositivo es capaz de proporcionarnos diferentes medidas de temperatura, humedad, luminosidad etc.



En este proyecto se utilizará el Sensor Fibaro FGMS-001-ZW5 [21]: Es un dispositivo con tecnología Z-Wave multifunción, con capacidad de detectar: movimiento, intensidad luminosa, vibraciones y temperatura. Su instalación es rápida y sencilla. Su LED multicolor indica detección de movimiento y nivel de temperatura. El sensor de movimiento resulta ideal para crear escenas de iluminación y vigilancia-seguridad. Además, es capaz de detectar cualquier intento de manipulación (anti-sabotaje), gracias al sensor de vibraciones integrado (acelerómetro).

Funciones:

- Detector de movimiento inalámbrico
- Detecta movimientos con un sensor infrarrojo pasivo
- medición de la temperatura
- medición de la intensidad de la luz
- compatible Z-Wave Plus
- Protección contra el robo y el sabotaje
- Alertas de movimiento y temperatura que se indica mediante el parpadeo del LED
- modo de detección de terremotos
- Botón para incluir / excluir el detector
- detección de batería baja
- Muy pequeño, de tamaño compacto
- Fácil de instalar en pared, techo u otra superficie

Características técnicas:

- Tipo de módulo: Transmisor Z-Wave
- Potencia: CR123A 3,6VDC
- altura máx. de montaje recomendada: 2,4 m
- medida de rango de temperatura: -20 ° C a 100 ° C
- Precisión de la medición: 0,5 ° C
- Rango de medición Brillo: 0-32000 LUX
- Frecuencia: 868.42 MHz
- Distancia de transmisión: campo libre 50m, 30m en interiores
- Dimensiones: 4,4 cm de diámetro
- Temperatura de funcionamiento: 0 a 40 ° C

Certificaciones:

- LVD 2006/95 / WE
- EMC 2004/108 / WE
- R & TTE 1999/5 / WE
- RoHS II



Figura 7. Sensor Fibaro

III.6 ACTUADORES

[23] Son los elementos que se encargan de modificar el estado de equipos o instalaciones. Por lo general transforman la energía eléctrica en rotación, movimiento, luz, calor, sonido, etc. Los más comúnmente utilizados son: las electroválvulas, contactores, contactores para base de enchufe, sirenas o elementos zumbadores.

En este proyecto utilizaremos las electroválvulas que colocaremos en cada radiador de la habitación. Pondremos en funcionamiento el Cabezal Termostático Spirit Z-Wave Plus de Eurotronic [22].

La unidad está equipada con FLiRS (Esclavo receptor de escucha frecuente) y responde de manera inmediata a los comandos entrantes del controlador Z-Wave. Además, el dispositivo se puede conectar a un sensor de temperatura Z-Wave externo. Por lo tanto, el termostato se refiere a la temperatura del sensor externo, que se puede colocar en el medio de la habitación.

Esta cabeza termostática puede usarse como un termostato de calefacción o como un actuador dependiendo de la configuración del dispositivo. La temperatura medida o la posición del actuador del termostato se transmite al controlador Z-Wave.

Funciones:

- Reemplaza la cabeza termostática de su radiador
- Control de su radiador de forma remota
- Tecnología Z-Wave Plus
- Ahorre hasta un 30% en sus costos de calefacción
- Detección de “ventana abierta”
- Protección infantil
- 3 adaptadores (Danfoss RA, Danfoss RAV, Danfoss RAVL) incluidos
- Varios colores del compartimiento de la batería y el anillo de decoración disponibles por separado
- Instalación fácil

Características técnicas:

- Fuente de alimentación: 2 x 1.5V LR6 / AA
- Frecuencia de transmisión: 868.42MHz
- Alcance: hasta 30 m en el interior y 100 m de campo libre
- Conexión: M30 x 1.5 mm
- Frecuencia de medición de la temperatura: cada minuto
- Rango de temperatura: 6 a 28 ° C
- Temperatura de funcionamiento: 0 a 50 ° C
- Clase de protección: IP20
- Dimensiones (ancho x alto x profundidad): 56 x 68 x 89 mm
- Peso: 176 g (pilas incluidas)



Figura 8. Válvula Eurotronics

III.7 MICROCONTROLADORES

Además de los elementos domóticos anteriormente descritos, utilizaremos la raspberry pi 3 B+ la cual se comunicará con el controlador y realizará los cálculos necesarios del control predictivo diseñado mediante Matlab y simulink [24].

Esta nueva versión de la raspberry aborda las principales limitaciones de un “micrordenador” y ofrece una mayor conectividad.

En términos de potencia, tiene un procesador con una potencia aceptable para el precio del dispositivo, además de otras prestaciones, como Wi-Fi y Bluetooth.

Las características del nuevo Raspberry Pi 3 Modelo B+ son:

- CPU + GPU: Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz



- RAM: 1GB LPDDR2 SDRAM
- Wi-Fi + Bluetooth: 2.4GHz y 5GHz IEEE 802.11.b/g/n/ac, Bluetooth

- 4.2, BLE
- Ethernet: Gigabit Ethernet sobre USB 2.0 (300 Mbps)
- GPIO de 40 pines
- HDMI
- 4 puertos USB 2.0
- Puerto CSI y DSI para conectar una cámara y una pantalla táctil
- Salida de audio estéreo y vídeo compuesto
- Micro-SD
- Power-over-Ethernet (PoE)

Además de un procesador 200 Mhz más potente que su predecesor y un nuevo chip, muy importante, capaz de controlar y gestionar toda la energía que maneja el microordenador, el Raspberry Pi 3 Modelo B+ viene también con dos novedades muy importantes:

Un nuevo chip inalámbrico, fabricado por Adafruit, que ofrece Bluetooth 4.2 y BLE, además de Wi-Fi de 5 Ghz para una mayor velocidad.

Gigabit Ethernet controlado por el chip USB 2.0 que, aunque no alcanza los 1000Mbps, sí ofrece sin problema 300 Mbps, tres veces más que el modelo Pi 3 anterior.



Figura 9. Raspberry pi 3 B+



III.8 COMUNICACIONES

A continuación, se explicarán las comunicaciones entre los elementos hardware del proyecto:

III.8.1 COMUNICACIÓN DEL CONTROLADOR CON EL SENSOR Y LA ELECTROVÁLVULA

Primero se añadirá la electroválvula y el sensor a la red Z-Wave creada con el controlador. Simplemente incluiremos ambos periféricos en la red a través de la plataforma de eedomus <https://secure.eedomus.com/>:

Mediante la clave de usuario y contraseña se accede a la plataforma, configuración del sistema, añadir o eliminar un periférico y por último incluir un periférico Z-Wave

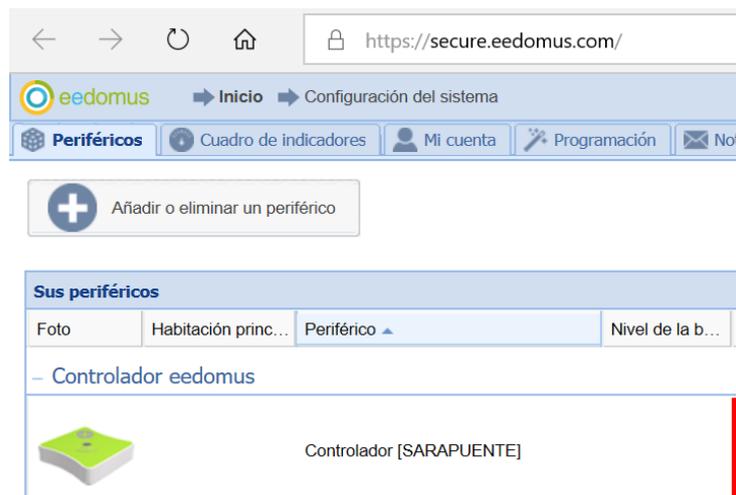


Figura 10. Plataforma Eedomus

Periféricos inalámbricos Z-Wave

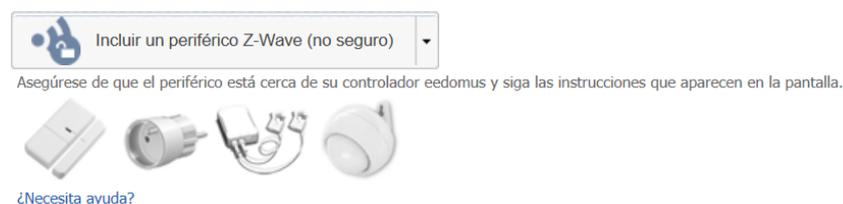


Figura 11. Añadir periférico



III.3.8.2 COMUNICACIÓN ENTRE EL CONTROLADOR Y EL CONTROL

Una vez creada la red Z-Wave y conectados los periféricos hay que establecer la comunicación entre Simulink y el controlador Eedomus

Simulink trabaja con comandos TCP/IP mientras que el controlador Eedomus maneja comandos HTTP, por lo que se programará en la raspberry un servidor hecho con NodeJS que reciba los comandos que envía simulink mediante TCP/IP y los transforme en una petición HTTP que pueda entender el controlador y viceversa

Dicho servidor se encuentra en segundo plano constantemente y ejecutará cada vez que se encienda la raspberry.

Adentrándonos ya en el archivo principal de simulink RASPI-CONTROL-SYSTEM como se puede observar hay 3 buses EEDOMUS, CONTROL y MODELO.

El bus EEDOMUS es el encargado de la comunicación contiene tanto el tamaño del buffer transmisión como el de recepción.

Centrándonos en la parte del hardware, está formado por otros 3 bloques los sensores, los actuadores y las comunicaciones:

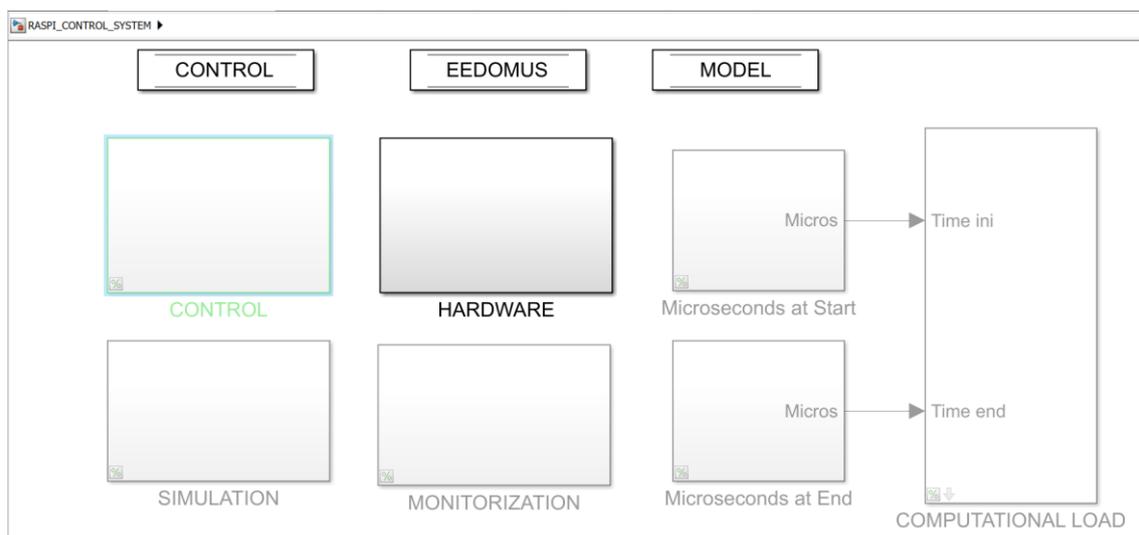


Figura 12. Simulink general

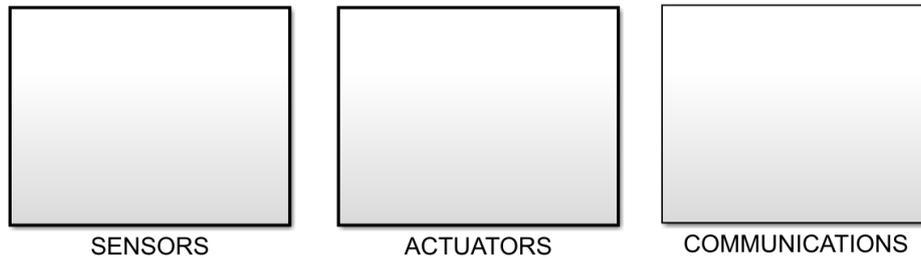


Figura 13. Hardware simulink

1. COMUNICACIONES:

El bloque de las comunicaciones se encuentra dividido en otros dos bloques el de Transmisión de datos y la recepción ambos alimentados por el bus EEDOMUS:

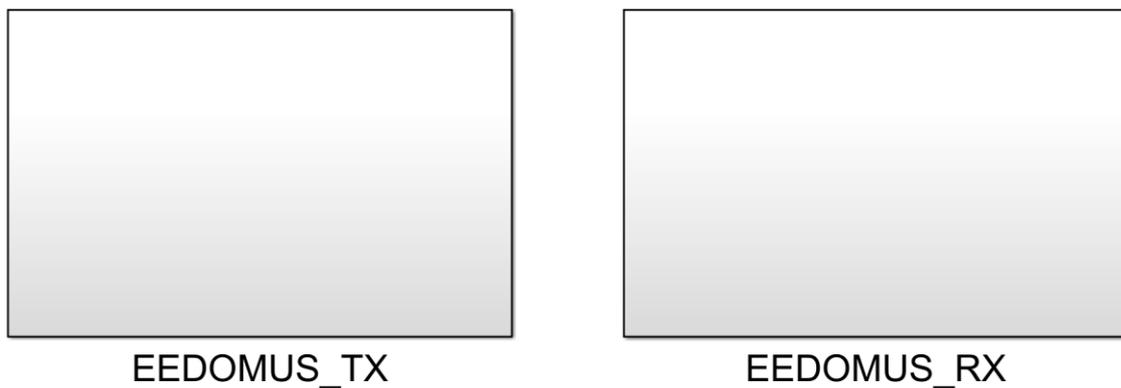


Figura 14. Comunicaciones Simulink

2. EEDOMUS_TX

En el bloque de la transmisión el buffer de transmisión del bus EEDOMUS contiene la información a transmitir, generalmente se envían comandos al sensor y al actuador, ya

sea para ajustar sus parámetros, para solicitar la medida de temperatura al sensor o para mandar un porcentaje de apertura a la válvula.

Los datos de este bus son cadenas de caracteres uint8 que contiene la petición HTTP para enviar al controlador Eedomus. El resto del bus se rellena con ceros para evitar errores. El bloque selector será el encargado de eliminar esos ceros y escoger solo el comando http para enviarlo mediante el bloque TCP/IP Send a la raspberry. Por último, la raspberry mediante el servidor creado en NodeJS transformará esos datos TCP a la petición HTTP deseada para que el Eedomus sea capaz de entenderlo.

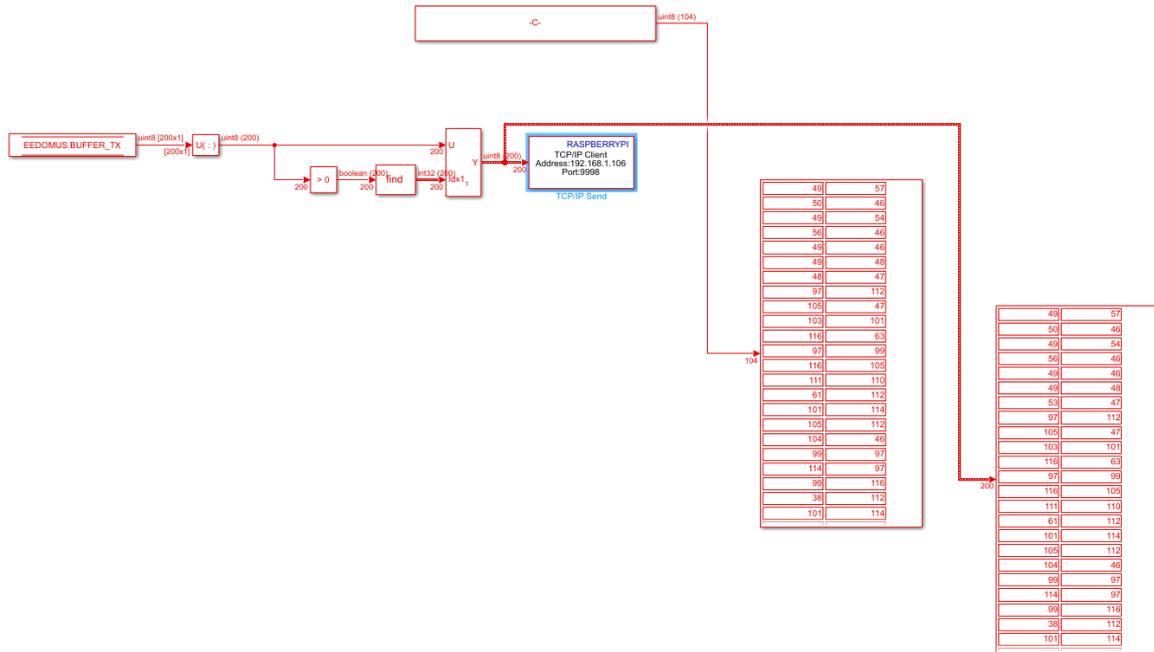


Figura 15. Transmisión Simulink

Haciendo hincapié en el bloque TCP/IP send:

Este bloque envía datos desde el modelo a equipos remotos mediante el protocolo TCP/IP.

Los parámetros que aparecen en el bloque son:

- **Adress:** Es la dirección IP del equipo a que se quiere enviar los datos, en este caso la de la raspberry.
- **Port:** el puerto remoto en el host al que necesita enviar los datos. Los valores de puerto válidos son 1 a 65535. En este caso se ha elegido el número de puerto 9998.

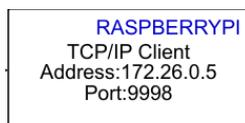


Figura 16. Bloque TCP/IP send

EEDOMUS_RX

En el bloque de la recepción generalmente se obtienen las medidas de temperatura del sensor y el porcentaje de apertura que tiene la válvula. También se reciben los mensajes de fallo o información del estado de los elementos del sistema domótico

En la recepción la raspberry recibe y transforma los comandos http del controlador Eedomus en comandos TCP/IP para que puedan ser interpretados por simulink. A continuación, son colocados en el bus Eedomus en el buffer de recepción. Se puede observar en el scope el mensaje recibido.

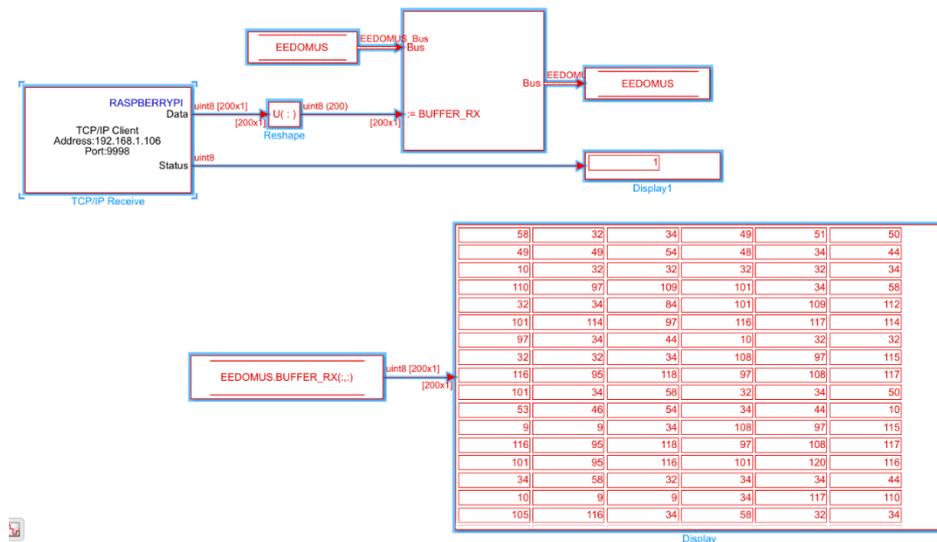


Figura 17. Recepción Simulink

Al igual que el bloque TCP/IP Send el bloque TCP/PI receive como su propio nombre indica recibe datos de un equipo especificado, en este caso la raspberry. Al igual que el bloque send contiene los parámetros de Adress y Port.

El puerto será el mismo número de puerto elegido en el bloque send, en este caso el 9998 y la dirección IP sigue siendo la de la raspberry.

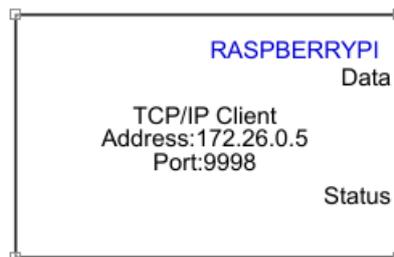


Figura 18. Bloque TCP/IP receive

3. COMUNICACIONES DEL SENSOR

En el interior del bloque del sensor se encuentran los 4 sensores de cada habitación. Su programación es exactamente idéntica.

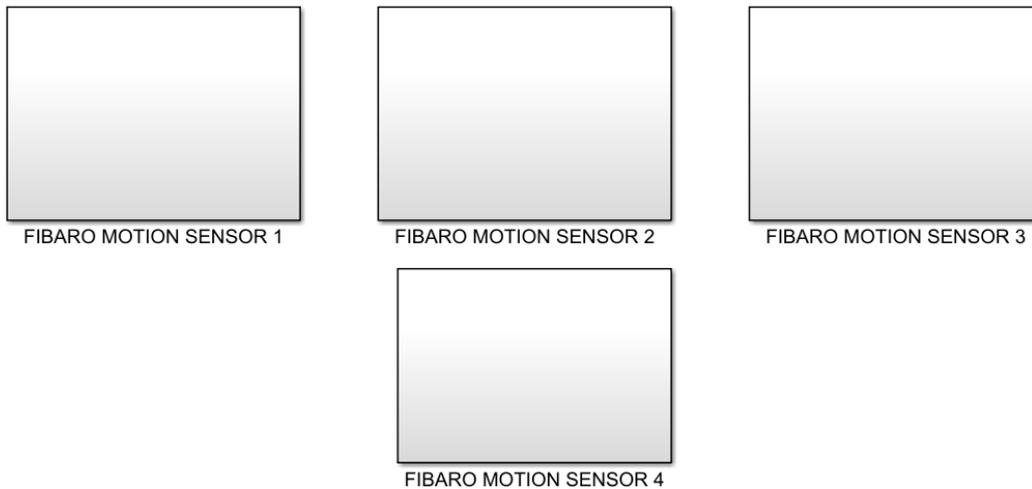


Figura 19. Sensores Simulink

El sensor se encuentra programado en un bloque de Matlab function cuyas entradas y salidas son el bus EEDOMUS y el de CONTROL.

El bus eedomus a la entrada contiene la petición http que solicita al sensor la temperatura de las habitaciones y se actualiza a la salida con la respuesta del sensor. Dicha temperatura es guardada en el bus de CONTROL para enviarla al control de temperatura.

El control solicita la temperatura al sensor cada 5 min por lo que esta función se repite cada dicho tiempo.

El control solicita la temperatura al sensor cada 5 min por lo que esta función se repite cada dicho tiempo.

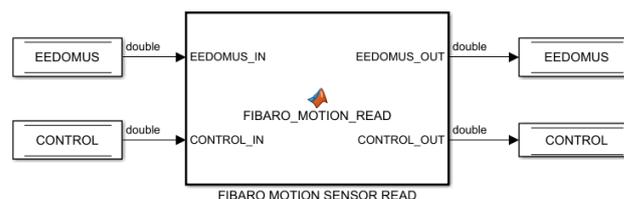


Figura 20. Máquina de estados del sensor



Adentrándonos un poco más en el código:

```
function [EEDOMUS_OUT,CONTROL_OUT] =  
FIBARO_MOTION_READ(EEDOMUS_IN,CONTROL_IN)  
  
EEDOMUS_OUT = EEDOMUS_IN;  
SAMPLING_TIME = EEDOMUS_OUT.SAMPLING_TIME;  
coder.extrinsic('findstr');  
persistent initialization STATUS TIMER  
  
if isempty(initialization)  
    STATUS = uint8(0);  
    TIMER = single(0);  
    initialization = 1;  
end  
  
switch STATUS  
    case 0 % SEND MESSAGE  
        % comand=char([EEDOMUS_INI.LOCAL_IP  
EEDOMUS_INI.GET_COMMAND_LIST(2) EEDOMUS_INI.TX_PARAM_LIST(5)  
EEDOMUS_INI.TX_PARAM_LIST(1) EEDOMUS_INI.TX_PARAM_LIST(2)]);  
  
comand=uint8('192.168.1.105/api/get?action=periph.caract&periph_  
id=1321160&api_user=QbHE74&api_secret=D3eA87NrQ3rC4H38');  
N=length(comand);  
EEDOMUS_OUT.BUFFER_TX=zeros(200,1,'uint8');  
  
EEDOMUS_OUT.BUFFER_TX(1:N)=uint8(comand);  
STATUS=uint8(1);  
    case 1 % CHECK MESSAGE  
        %ESPERAR 5 S PARA RECIBIR EL MENSAJE  
str1=EEDOMUS_OUT.BUFFER_RX;  
EEDOMUS_OUT.BUFFER_RX=zeros(200,1,'uint8');  
str0='"last_value:"';  
str0=str0(:);  
N0=length(str0);  
N1=length(str1);  
k1=0;  
for nn=1:N1-N0+1  
    if all(str1(nn:nn+N0-1)==str0)  
        k1=nn;  
        break;  
    end  
end  
  
if isempty(k1) %Si no encuentra la cadena te da una  
matriz vacía%  
    STATUS = uint8(0);  
else  
    str3=str1(k1+length(str0):end);  
    str2='"', 'last_value_text';
```



```
str2 = str2(:);
N2=length(str2);
N3=length(str3);
k2=0;
for nn=1:N3-N2+1
    if all(str3(nn:nn+N2-1)==str2)
        k2=nn;
        break;
    end
end
str4=str2(1:k2-1);
CONTROL_OUT.BUFFER_RX=uint8(str4);
STATUS= uint8(2);

end

case 2 % WAIT
    TIMER = TIMER + SAMPLING_TIME;
    if TIMER >= 300

TIMER = single(0);
    STATUS = uint8(0);
    end
otherwise
    STATUS = uint8(0);
end

return
```

El código del sensor es una máquina de estados con switch case. Primero se inicializa el timer y el estado a cero y entra en el switch.

Estado 0: SEND MESSAGE

Tras asegurarse que el buffer de transmisión del bus EEEDOMUS este vacío introduce el comando http para solicitar al sensor la temperatura de la habitación. A continuación, pasa al segundo estado.

*El comando http para solicitar la temperatura al sensor es :
192.168.1.105/api/get?action=periph.caract&periph_id=1321160&api_user=QbHE74
&api_secret=D3eA87NrQ3rC4H38*

La dirección http que aparece es la del controlador eedomus,periph.caract&periph_id=1321160 es el identificador del sensor de temperatura y el api_user y api_secret es el usuario y contraseña eedomus.

Estado 1: CHECK MESSAGE

Este estado recibe la respuesta del sensor, comprueba que es correcta y que no ha habido fallos en la comunicación. Si es correcta se obtiene el valor de la temperatura y pasa al estado 2, de lo contrario vuelve al 0.



Para comprobar si la respuesta es correcta se realiza el siguiente proceso:

Observando la respuesta http que nos tiene que devolver el sensor:

```
{ "success": 1, "body":{"periph_id": "1321160", "name": "Temperatura",  
"last_value": "25.1", "last_value_text": "", "unit": "°C", "battery": "100",  
"last_value_change": "2019-10-03 12:32:21"}}
```

Podemos ver que el valor que necesitamos de temperatura en este caso 25.1 se encuentra entre las cadenas de caracteres: *“last_value”*: y *“last_value_text”*: *“”*

Nos fijamos en una pequeña cadena de caracteres justo antes del número que indica la temperatura, en este caso: *“last_value”*:” esta cadena se guarda en la constante str0

La respuesta que recibimos del sensor se guarda en otra constante str1

A continuación, mediante el bucle for buscamos la cadena str0 en str1 y guardamos la posición en la que acaba la cadena str0 dentro de la cadena str1 en la variable k1.

Si la variable k1 es cero significa que el mensaje que nos ha devuelto el sensor es erróneo y por tanto se volvería al estado 0, en caso contrario buscaríamos en la cadena str1 el dato concreto de la temperatura que nos interesa.

A continuación nos fijamos otra cadena de caracteres que se encuentre justo detrás del valor de la temperatura : *“,”last_value_text* y la posición en donde empieza mediante un bucle for. Dicha posición se guarda en k2

Al tener la posición exacta en la cadena str0 en dónde empieza el valor de la temperatura (k1) y en donde acaba (k2) ya tenemos aislado el dato deseado que se enviará al bus de CONTROL.

Estado 2 WAIT

En este estado simplemente se encuentra un timer que cuenta los 5 min necesarios para volver a solicitar la medida de la temperatura al sensor, e inicializar el timer y la variable de estado cuando venzan dichos minutos.

Por último, se actualiza el valor de la temperatura en el bus de CONTROL.

4. COMUNICACIONES DEL ACTUADOR

Al igual que los sensores colocamos 4 electroválvulas en cada radiador de cada habitación

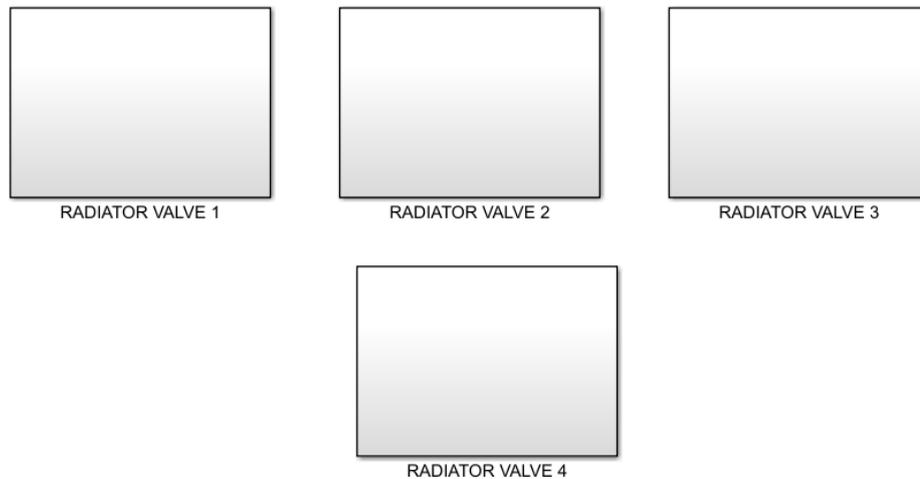


Figura 21. Actuadores Simulink

La electroválvula se encuentra programada en un bloque Matlab function cuya entrada y salida son los buses EEDOMUS Y CONTROL.

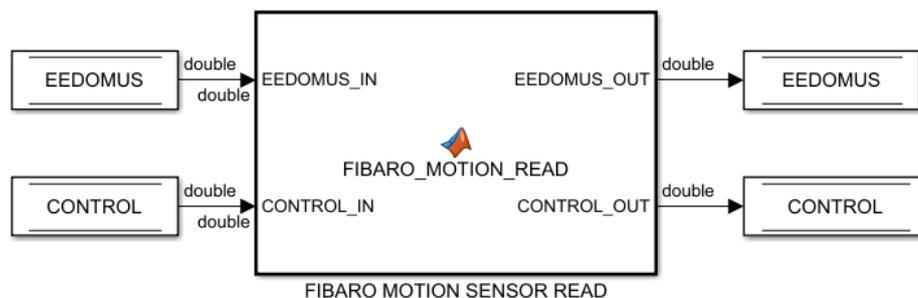


Figura 22. Máquina de estados del sensor

El control predictivo a través del bus de control enviará cada 5 minutos el porcentaje de apertura que necesita la válvula de actuación, dicho porcentaje se realizará mediante un módulo PWM.

Se programará con una máquina de estados al igual que el sensor. Para optimizar el proceso habrá que tener en cuenta en qué estado ha acabado (abriendo o cerrando) la válvula cada vez que acabe el intervalo de 5 min.



Por ejemplo, si en un intervalo la válvula ha estado primero abriendo el tiempo correspondiente y luego cerrando, en el siguiente nuevo intervalo empezará primero cerrando y luego abriendo. De esta manera, conseguimos un menor número de actuaciones en ella y por tanto un aumento de su vida útil.

Para ello hemos recurrido a la creación de dos estados de abrir, ABRIR 1 y ABRIR 2, y otros 2 de cerrar CERRAR 1 y CERRAR 2 además de un SAMPLING-COUNT que nos indica en qué estado ha acabado válvula, si abierta o cerrada.

Adentrándonos un poco más en el código:

```
function [EEDOMUS_OUT,CONTROL_OUT] =
FIBARO_MOTION_READ(EEDOMUS_IN,CONTROL_IN)

EEDOMUS_OUT = EEDOMUS_IN;
SAMPLING_TIME = EEDOMUS_OUT.SAMPLING_TIME;
apertura = 0;
SAMPLING_COUNT=single(0);

persistent initialization STATUS TIMER

if isempty(initialization)
    STATUS = uint8(0);
    TIMER = single(0);
    SAMPLING_COUNT=single(0);
    initialization = 1;
end

switch STATUS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    case 0 % Recibir el porcentaje de apertura
        %¿De dónde recibo el porcentaje?%
        % Apertura=CONTROL_OUT;
        apertura=0.7;
        TIMER = single(0);
        if rem(SAMPLING_COUNT,2)==0
            STATUS=uint8(1);
        else
            STATUS=uint8(3);
        end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

case 1 % ABRIR 1

% comand=char([EEDOMUS_INI.LOCAL_IP
EEDOMUS_INI.SET_COMMAND_LIST(1) EEDOMUS_INI.TX_PARAM_LIST(6)
EEDOMUS_INI.TX_PARAM_LIST(11) EEDOMUS_INI.TX_PARAM_LIST(1)
EEDOMUS_INI.TX_PARAM_LIST(2)]);
```



```

comand=uint8('172.26.0.5/api/set?action=periph.value&periph_id=1
302200&value=5&api_user=QbHE74&api_secret=D3eA87NrQ3rC4H38')
N=length(comand);
EEDOMUS_OUT.BUFFER_TX=zeros(200,1,'uint8');
EEDOMUS_OUT.BUFFER_TX(1:N)=uint8(comand);
%ESPERAR 5 S PARA RECIBIR EL MENSAJE
%
%   while TIMER<=3
%       TIMER=TIMER+1;
%   end
str1=EEDOMUS_OUT.BUFFER_RX;
EEDOMUS_OUT.BUFFER_RX=zeros(200,1,'uint8');

str0=""sucess": 1";
str0=str0(:);
N0=length(str0);
N1=length(str1);
k1=0;
for nn=1:N1-N0+1
    if all(str1(nn:nn+N0-1)==str0)
        k1=nn;
        break;
    end
end

if isempty(k1) %Si no encuentra la cadena te da una
matriz vacía%
    STATUS = 0;
else
    while (apertura*300) >= TIMER
        TIMER=TIMER+1;
    end
    STATUS = uint8(2);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
case 2 %CERRAR 1

comand=uint8('172.26.0.5/api/set?action=periph.value&periph_id=1
302200&value=28&api_user=QbHE74&api_secret=D3eA87NrQ3rC4H38')
N=length(comand);
EEDOMUS_OUT.BUFFER_TX=zeros(200,1,'uint8');
EEDOMUS_OUT.BUFFER_TX(1:N)=uint8(comand);
str3=EEDOMUS_OUT.BUFFER_RX;
EEDOMUS_OUT.BUFFER_RX=zeros(200,1,'uint8');
str2=""sucess": 1";

str2=str2(:);
N0=length(str2);
N1=length(str3);
k2=0;

```



```
for nn=1:N1-N0+1
    if all(str3(nn:nn+N0-1)==str2)
        k2=nn;
        break;
    end
end

if isempty(k2) %Si no encuentra la cadena te da una
matriz vacía%
    STATUS = 0;
else
    while TIMER < 300
        TIMER=TIMER+1;

        end
        SAMPLING_COUNT=SAMPLING_COUNT+1;
        STATUS = uint8(0);

    end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    case 3 % CERRAR 2

        %comand=char([EEDOMUS_INI.LOCAL_IP
EEDOMUS_INI.SET_COMMAND_LIST(1) EEDOMUS_INI.TX_PARAM_LIST(6)
EEDOMUS_INI.TX_PARAM_LIST(11) EEDOMUS_INI.TX_PARAM_LIST(1)
EEDOMUS_INI.TX_PARAM_LIST(2)]);

comand=uint8('172.26.0.5/api/set?action=periph.value&periph_id=1
302200&value=28&api_user=QbHE74&api_secret=D3eA87NrQ3rC4H38')
N=length(comand);
EEDOMUS_OUT.BUFFER_TX=zeros(200,1,'uint8');
EEDOMUS_OUT.BUFFER_TX(1:N)=uint8(comand);
%ESPERAR 5 S PARA RECIBIR EL MENSAJE
%
%   while TIMER<=3
%       TIMER=TIMER+1;
%   end
str5=EEDOMUS_OUT.BUFFER_RX;
EEDOMUS_OUT.BUFFER_RX=zeros(200,1,'uint8');
str4='"sucess": 1';
str4=str4(:);
N0=length(str4);
N1=length(str5);
k3=0;
for nn=1:N1-N0+1
    if all(str5(nn:nn+N0-1)==str4)
        k3=nn;
        break;
    end

end

end
```



```

if isempty(k3) %Si no encuentra la cadena te da una matriz
vacía%
    STATUS = 0;
else
    while (300-apertura*300) >= TIMER
        TIMER=TIMER+1;
    end
    STATUS = uint8(4);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    case 4 % ABRIR 2

comand=uint8('172.26.0.5/api/set?action=periph.value&periph_id=1
302200&value=5&api_user=QbHE74&api_secret=D3eA87NrQ3rC4H38')
    N=length(comand);
    EEDOMUS_OUT.BUFFER_TX=zeros(200,1,'uint8');
    EEDOMUS_OUT.BUFFER_TX(1:N)=uint8(comand);
    str7=EEDOMUS_OUT.BUFFER_RX;
    EEDOMUS_OUT.BUFFER_RX=zeros(200,1,'uint8');
    str6='"sucess": 1';
    str6=str6(:);
    N0=length(str6);
    N1=length(str7);
    k4=0;
    for nn=1:N1-N0+1
        if all(str7(nn:nn+N0-1)==str6)
            k4=nn;
            break;
        end
    end

    if isempty(k4) %Si no encuentra la cadena te da una
matriz vacía%
        STATUS = 0;
    else
        while TIMER < 300
            TIMER=TIMER+1;
        end
        SAMPLING_COUNT=SAMPLING_COUNT+1;
        STATUS = uint8(0);
    end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    otherwise
        STATUS = uint8(0);
end
CONTROL_OUT.INPUT.ROOM_TEMP(1) = single(str2num(str4));

return

```



Primero inicializamos la variable de estado, el timer y la variable SAMPLING_COUNT a cero, a continuación, se crea la máquina de estados mediante un switch case:

Estado 0: Recibir el porcentaje de apertura.

Del bus de control recibimos el porcentaje de apertura y lo guardamos en la variable “apertura” y se inicializa el TIMER.

A continuación, miramos si el SAMPLING_COUNT es par o impar. Si es par, la válvula significa que acabó el anterior intervalo abierta y por tanto se pasa al estado 1 (ABRIR

1), de lo contrario si SAMPLING_COUNT es impar significa que acabó cerrada y se pasa al estado 3 (CERRAR 2).

Estado 1 ABRIR 1

Enviamos la petición http al controlador Eedomus mediante el buffer de transmisión de Edomus y esperamos 3 s para que el controlador lo reciba. (192.168.1.105/api/set?action=periph.value&periph_id=1302200&value=5&api_user=QbHE74&api_secret=D3eA87NrQ3rC4H38)

Se comprueba que se ha recibido bien el mensaje. Miramos si en el buffer de recepción del bus Eedomus se encuentra la respuesta http del controlador: “*sucess*”: 1’

Para ello comparamos mediante un bucle for si la cadena “*sucess*”: 1’ se encuentra en la respuesta que nos ha proporcionado el controlador.

Si no encuentra la cadena, se vuelve al estado 0, en caso contrario, estará la válvula abierta el porcentaje de tiempo que indica la apertura de los 5 min que dura cada intervalo. Una vez finalizado dicho tiempo se pasará al estado 2 (CERRAR 1)

Estado 2: CERRAR 1

Al igual que ABRIR 1 se envía la petición http al controlador Eedomus mediante el buffer de transmisión de Edomus y esperamos 3 s para que el controlador lo reciba. (192.168.1.105/api/set?action=periph.value&periph_id=1302200&value=28&api_user=QbHE74&api_secret=D3eA87NrQ3rC4H38) y se comprueba de la misma manera que anteriormente si el mensaje ha sido recibido con éxito.

Si ha sido así, la válvula se encontrará cerrada el resto de tiempo que falte de los 5 min, la variable SAMPLING_COUNT se incrementará en 1 y se volverá al estado 0.

Si por el contrario ha habido algún error en la transmisión del mensaje se volverá al estado 0 de nuevo.

Estado 3: CERRAR 2

Funciona igual que el estado 2 enviamos mediante el buffer de transmisión Eedomus la petición para cerrar la válvula y comprobamos si se ha realizado con éxito.



Si hubiese algún error en la transmisión se pasaría al estado 0.

Si no ha habido ningún error el timer empezará a contar hasta llegar a la parte restante de los 5 min menos el porcentaje de apertura de esos 5 min (A modo de ejemplo, porcentaje es 0,7 seria $5 \text{ min} - 0,7 * 5$) la válvula estará cerrada durante ese tiempo.

A continuación, se pasaría al estado 4.

Estado 4: ABRIR 2

Al igual que los estados anteriores se envía la petición http para abrir la válvula y se comprueba si se ha realizado con éxito.

Si ha habido algún error se vuelve al estado 0 em caso contrario la válvula se queda abierta lo que queden de los 5 min, se incrementa el SAMPLING_COUNTN en uno y la variable de estado pasa a valer de nuevo 0.

Por último, añadir, que no se ha conseguido mandar un porcentaje concreto al actuador y conseguir que la válvula abriera dicho porcentaje, por lo que ha habido que crear un módulo PWM.

Lo hemos hecho de tal forma que, si en la anterior acción del control la válvula se había quedado abierta, en la siguiente acción el control empezara abriendo la válvula y luego cerrando en la siguiente acción tendría que empezar cerrando y luego abriendo y así sucesivamente. Esto lo hemos hecho para reducir a la mitad las acciones del actuador y disminuir así su desgaste.



Capítulo IV: CONTROL MPC

IV.1 CONTROL PREDICTIVO

[25] El control predictivo es una metodología de control que puede predecir las salidas futuras de la planta y con base a ello optimizar las acciones del control futuras. El control predictivo, integra toda una familia de métodos de control tales como, el control óptimo, el control de procesos con tiempos muertos, el control de procesos multivariables, etc. Esto ha permitido que el control predictivo haya tenido un desarrollo importante tanto en la comunidad científica y académica, como en el sector industrial.

Los controles predictivos más utilizados son los basados en modelo (MPC) y los adaptativo. Los adaptivos se van regulando a sí mismos a lo largo del tiempo en función de datos anteriores, por tanto, son más fiables una vez ha pasado cierto tiempo. Los basados en modelo requieren de un modelo matemático de la planta muy preciso para funcionar.

[25]El control predictivo se ha desarrollado alrededor de ciertos principios básicos comunes:

El uso explícito de un modelo del proceso para pronosticar la salida del sistema en instantes futuros de tiempo y el cálculo de una acción de control óptima basado en la minimización de una o más funciones de costo, con la posibilidad de incluir restricciones sobre las variables del proceso.

Por lo tanto, los diferentes algoritmos miembros de la extensa familia MPC, difieren entre sí principalmente en:

- El tipo de modelo utilizado para representar el proceso y sus perturbaciones.
- La función de costo que ha de ser minimizada, con o sin restricciones.

Algunas ventajas del control predictivo son:

Principios de funcionamiento intuitivos, se puede manejar problemas de control multivariable y no lineal de una manera sencilla, permite controlar procesos altamente oscilatorios o procesos inestables, introduce un control anticipativo, y de forma natural se compensan las perturbaciones medibles.

Por otro lado, estos controles presentan algunos inconvenientes. Como la necesidad de contar con un modelo matemático del proceso (planta) y la mayor computación que requiere con respecto a otros controles, como son los PID.

[26] A continuación enumeraremos los conceptos más básicos a la hora de aplicar un control predictivo:

- Referencia: Es el valor deseado de la variable que se quiere controlar. El control trata de mantener la variable física lo más parecida posible a la referencia.
- Límites de mandos: Mandos máximos y mínimos del sistema. En este caso la potencia máxima de los radiadores y la mínima, es decir 0, ya que este sistema solo consta de calefacción.
- Período de muestreo: El tiempo entre dos cálculos del mando a aplicar. En este proyecto el período de muestreo será de 5 min ya que la temperatura presenta una dinámica muy lenta.
- Horizonte de predicción: Define cuántos períodos de muestreo va a tener el control cada vez que efectúe un ciclo de cálculos.
- Horizonte de control: Es menor del horizonte de predicción y marca el instante a partir del cual la señal de control se considera constante.
- Función de optimización: Esta función intenta minimizar a la hora de calcular los mandos del sistema, el control modificará la variable física para parecerse lo más posible a la referencia respetando esta función.

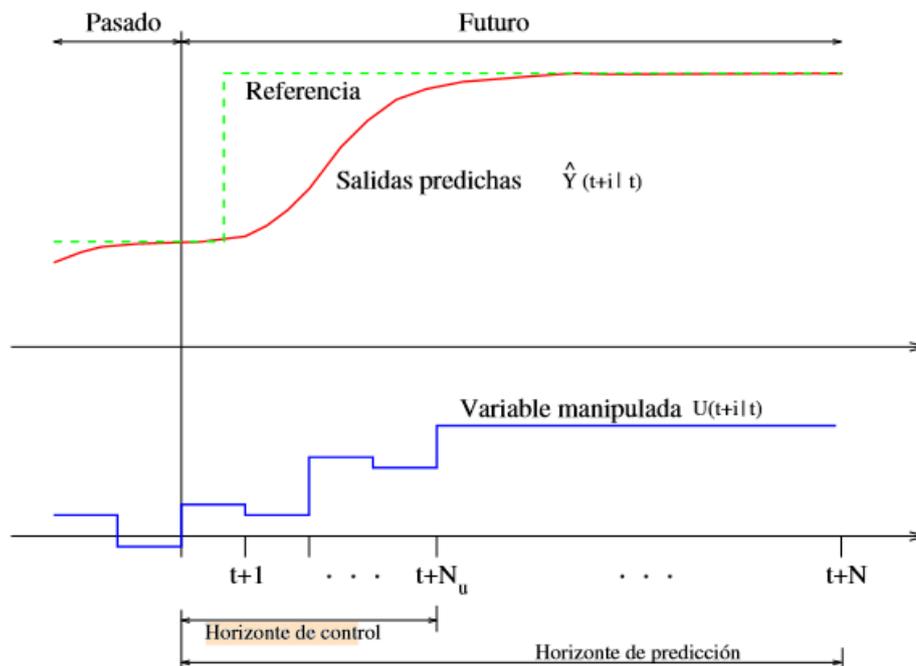


Figura 23. Elementos del MPC



IV.2 DISEÑO DE PLANTA

Para simplificar se supondrá que el sistema térmico se representa mediante un modelo de parámetros concentrados, las sustancias que se caracterizan por una resistencia al flujo de calor tienen una capacitancia térmica insignificante y que las sustancias que se caracterizan por una capacitancia térmica tienen una resistencia insignificante al flujo de calor.

Se supone además un valor de temperatura uniforme en cada habitación para así evitar las derivadas parciales que indican el almacenamiento y transmisión de energía térmica en un mismo medio.

También se supone para el modelo de la planta un sistema lineal invariante en el tiempo (LTI). Se considera que el medio exterior a la vivienda tiene la posibilidad de intercambiar energía térmica con el resto de medios del sistema sin afectar a su temperatura propia. Es decir, las variaciones de temperatura del medio exterior no se deben a su interacción con el resto de medios de la planta.

La planta en la que se va a implantar el control consiste en una vivienda de 4 habitaciones. A cada habitación le corresponde una capacidad térmica C_n : C_1 , C_2 , C_3 y C_4 . Por otra parte, entre cada habitación existe una resistencia térmica R_{nm} : R_{12} , R_{13} , R_{14} , R_{23} , R_{24} y R_{34} y una resistencia térmica de cada habitación con el exterior: R_{10} , R_{20} , R_{30} y R_{40} . Además en cada habitación hay un radiador que proporciona una potencia de entrada P_n : P_1 , P_2 , P_3 y P_4 .

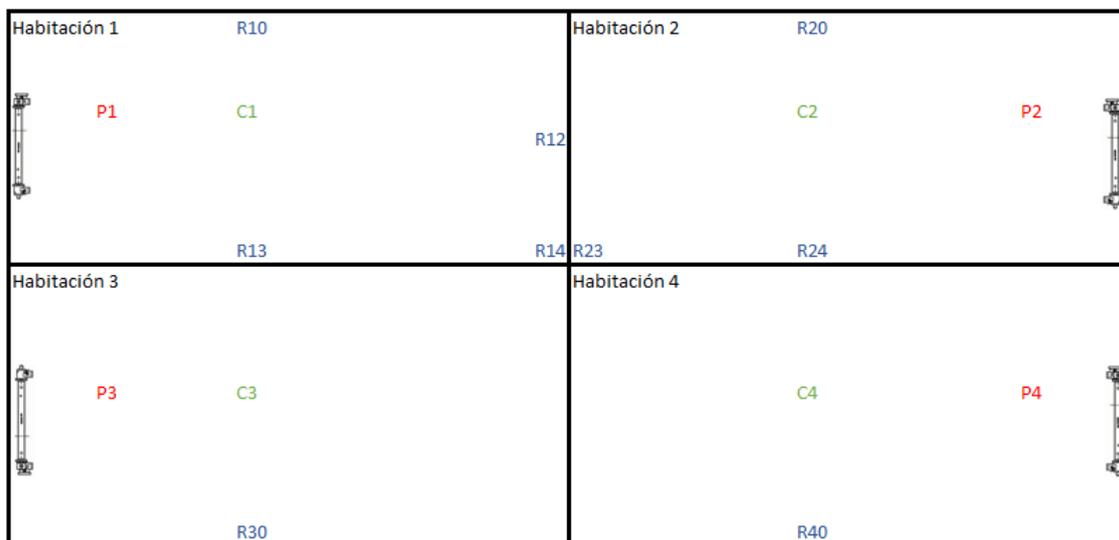


Figura 24. Plano habitaciones

No se consideran las variables correspondientes a la transmisión de calor por radiación al ser esta no lineal y solo ser significativa a muy alta temperatura.



IV.2.1 MÓDELO MATEMÁTICO

Definimos brevemente el concepto de resistencia térmica y capacitancia térmica:

$$R = \frac{\text{cambio en la diferencia de temperatura, o } C}{\text{cambio en el flujo de calor, kcal/seg}} = \frac{W}{K}$$
$$C = \frac{\text{cambio en el calor almacenado, kcal}}{\text{cambio en el flujo de calor, kcal/seg}} = \frac{W * s}{K}$$

Definimos las entradas, variable de estado y salidas del sistema:

Entradas: Son las potencias térmicas suministradas P_n , la temperatura exterior T_0 y las potencias de perturbación de cada habitación Pd_n

$$U = [P_1, P_2, P_3, P_4, T_0, Pd_1, Pd_2, Pd_3, Pd_4]'$$

Ecuación 1. Entradas de la representación de estados

Variables de estado: La temperatura de cada habitación T_n

$$X = [T_1, T_2, T_3, T_4]'$$

Ecuación 2. Variables de estados

Salidas: La temperatura de cada habitación T_n al igual que las variables de estado

$$Y = X = [T_1, T_2, T_3, T_4]'$$

Ecuación 3. Salidas de la representación de estado

A continuación, se muestra la ecuación de almacenamiento de energía en un medio físico, donde P_a es la potencia donde se almacena la energía E_a en el medio, C [W*s/K] la capacidad térmica del medio, T [°C] la temperatura del medio y t el tiempo.

$$P_a = \frac{dE_a}{dt} = C \times \frac{dT}{dt}$$

Ecuación 4: Almacenamiento de energía térmica en un medio

Por tanto, la ecuación que representa el almacenamiento de energía en la habitación n es:

$$P_{a_n} = C_n \times \frac{dT_n}{dt}$$

Ecuación 5. Potencia con la que se almacena energía en el medio n



La transmisión de energía térmica entre dos medios la rige la ecuación 6. Dónde P_t es la potencia transferida entre ambos medios T_1 y T_2 la temperatura de ambos medios y R_{12} [W/K] la resistencia térmica entre ambos medios.

$$P_t = \frac{T_1 - T_2}{R_{12}}$$

Ecuación 6. Transmisión de energía térmica entre dos medios

Por tanto, la transmisión que transmite la energía de la habitación m a la n es:

$$P_{nm} = \frac{T_n - T_m}{R_{nm}}$$

Ecuación 7. Potencia con la que se transmite energía del medio m al n

Perturbaciones: Las perturbaciones internas del medio n están representadas por cuatro potencias térmicas P_{d_n} y por la temperatura exterior T_o .

La representación de estados representado en la ecuación 8 se obtiene por balance de potencias, donde P_n es la potencia generada en la habitación n, P_{a_n} la potencia con la que se almacena energía en n, $P_{t_{nm}}$ la potencia térmica transmitida desde el medio n a m y P_{d_n} la potencia de perturbación en la habitación n.

$$P_n = P_{a_n} + \sum_{\substack{m=0 \\ m \neq n}}^4 P_{t_{nm}} - P_{d_n}$$

Ecuación 8. Potencia generada en la habitación n

Desarrollando la ecuación 8 para cada habitación obtenemos el siguiente sistema de ecuaciones:

$$P_1 = C_1 \times \frac{dT_1}{dt} + \frac{T_1 - T_2}{R_{12}} + \frac{T_1 - T_3}{R_{13}} + \frac{T_1 - T_4}{R_{14}} + \frac{T_1 - T_o}{R_{10}} - P_{d_1}$$

Ecuación 9. Balance de potencia habitación 1

$$P_2 = C_2 \times \frac{dT_2}{dt} + \frac{T_2 - T_1}{R_{12}} + \frac{T_2 - T_3}{R_{23}} + \frac{T_2 - T_4}{R_{24}} + \frac{T_2 - T_o}{R_{20}} - P_{d_2}$$

Ecuación 10. Balance de potencia habitación 2



$$P_3 = C_3 \times \frac{dT_3}{dt} + \frac{T_3 - T_1}{R_{13}} + \frac{T_3 - T_2}{R_{23}} + \frac{T_3 - T_4}{R_{34}} + \frac{T_3 - T_o}{R_{30}} - Pd_3$$

Ecuación 11. Balance de potencia habitación 3

$$P_4 = C_4 \times \frac{dT_4}{dt} + \frac{T_4 - T_1}{R_{14}} + \frac{T_4 - T_2}{R_{24}} + \frac{T_4 - T_3}{R_{34}} + \frac{T_4 - T_o}{R_{40}} - Pd_4$$

Ecuación 12. Balance de potencia habitación 4

Si se despejan las derivadas parciales de las variables de estado obtenemos la siguiente representación de estado:

$$\begin{aligned} \frac{dX}{dt} &= [A] \times X + [B] \times U \\ Y &= [C] \times X + [D] \times U \end{aligned}$$

Ecuación 13. Representación de estado

Siendo [A], [B],[C] y [D] las matrices de representación de estado.

Los cálculos anteriormente descritos se encuentran programadas en un bloque de Matlab function formando un control de espacio de estado. Alimentado por el bus MODEL.

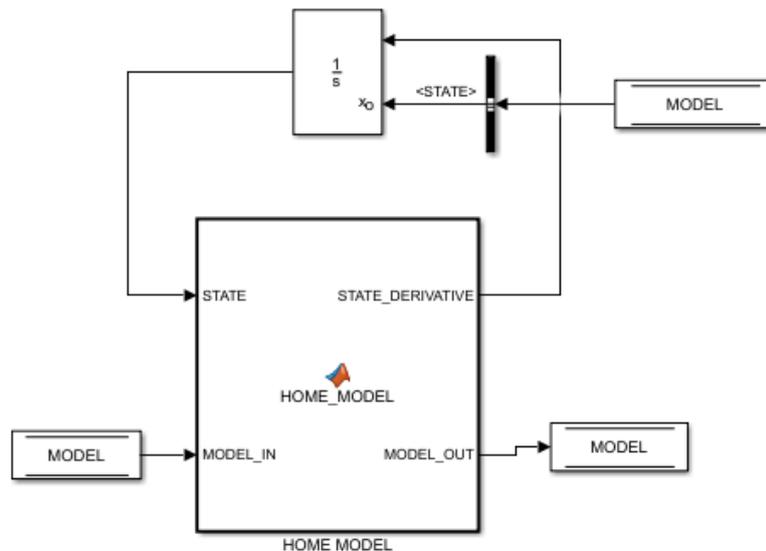


Figura 25. Modelo Simulink



El bus MODEL se alimenta de los valores que le va dando el control y van actualizando el modelo proporcionando las nuevas variables de estado, es decir, la temperatura.

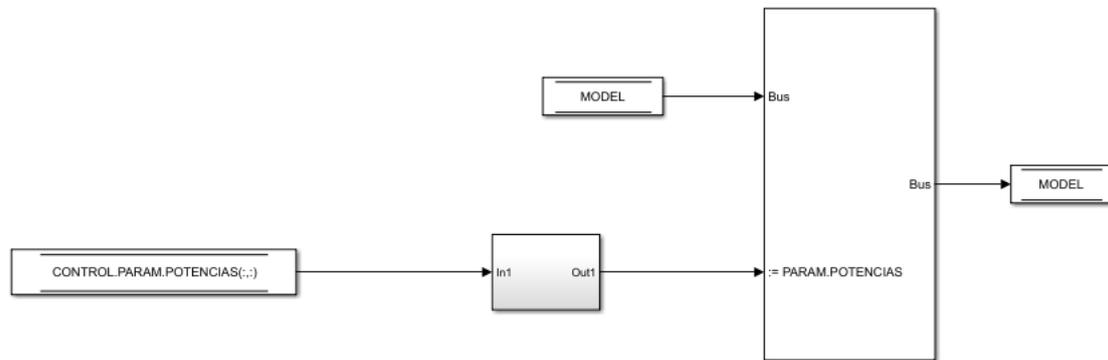


Figura 26. Bus modelo

Como podemos ver en la figura anterior el bus del modelo se actualiza con las nuevas potencias a aplicar que le proporciona el control.

El código utilizado en la Matlab Function para programar el modelo es:

```
function [STATE_DERIVATIVE,MODEL_OUT] =  
HOME_MODEL (STATE,MODEL_IN)
```

```
%% COPY MODEL_BUS  
MODEL_OUT = MODEL_IN;
```

```
%% CONTROL PARAMETERS
```

```
%%CAPACIDADES TÉRMICAS  
C1 = MODEL_INI.PARAM.C1;  
C2 = MODEL_INI.PARAM.C2;  
C3 = MODEL_INI.PARAM.C3;  
C4 = MODEL_INI.PARAM.C4;
```

```
%%RESISTENCIAS TÉRMICAS  
R10 = MODEL_INI.PARAM.R10;  
R12 = MODEL_INI.PARAM.R12;  
R13 = MODEL_INI.PARAM.R13;  
R14 = MODEL_INI.PARAM.R14;  
R20 = MODEL_INI.PARAM.R20;  
R23 = MODEL_INI.PARAM.R23;  
R24 = MODEL_INI.PARAM.R24;
```



```
R30 = MODEL_INI.PARAM.R30;  
R34 = MODEL_INI.PARAM.R34;
```

```
R40 = MODEL_INI.PARAM.R40;
```

```
%%POTENCIAS MAXIMAS RADIADORES
```

```
Pmax1 = MODEL_INI.PARAM.Pmax1;  
Pmax2 = MODEL_INI.PARAM.Pmax2;  
Pmax3 = MODEL_INI.PARAM.Pmax3;
```

```
Pmax4 = MODEL_INI.PARAM.Pmax4;
```

```
%%TIEMPO DE MUESTREO
```

```
ts_pwm = MODEL_INI.PARAM.ts_pwm;
```

```
% Matrices de la representacion de estado
```

```
% X = [T1 T2 T3 T4]'  
% U = [P1 P2 P3 P4 T0 Pd1 Pd2 Pd3 Pd4]'  
% Y = [T1 T2 T3 T4]'
```

```
matAo=[-1/(C1*R12)-1/(C1*R13)-1/(C1*R14)-1/(C1*R10) 1/(C1*R12)  
1/(C1*R13) 1/(C1*R14);...  
1/(C2*R12) -1/(C2*R12)-1/(C2*R23)-1/(C2*R24)-1/(C2*R20)  
1/(C2*R23) 1/(C2*R24);...  
1/(C3*R13) 1/(C3*R23) -1/(C3*R13)-1/(C3*R23)-1/(C3*R34) -  
1/(C3*R30) 1/(C3*R34);...  
1/(C4*R14) 1/(C4*R24) 1/(C4*R34) -1/(C4*R14)-1/(C4*R24) -  
1/(C4*R34)-1/(C4*R40)];  
matBo=[1/C1 0 0 0 1/(C1*R10) 1/C1 0 0 0;...  
0 1/C2 0 0 1/(C2*R20) 0 1/C2 0 0;...  
0 0 1/C3 0 1/(C3*R30) 0 0 1/C3 0;...  
0 0 0 1/C4 1/(C4*R40) 0 0 0 1/C4];  
matCo=eye(4);  
matDo=zeros(4,9);
```

```
%
```

```
%
```

```
PUNTO DE OPERACION
```

```
%
```

```
% Potencias de los radiadores en el punto de operacion
```

```
P10 = MODEL_INI.PARAM.P10;  
P20 = MODEL_INI.PARAM.P20;  
P30 = MODEL_INI.PARAM.P30;  
P40 = MODEL_INI.PARAM.P40;
```

```
% Temp. exterior
```

```
T0 = MODEL_INI.PARAM.T0;
```

```
% Potencias de perturbacion
```



```
Pd10 = MODEL_INI.PARAM.Pd10;
Pd20 = MODEL_INI.PARAM.Pd20;
Pd30 = MODEL_INI.PARAM.Pd30;
Pd40 = MODEL_INI.PARAM.Pd40;
```

`%% Representación de estado sin Pd`

```
matA=matAo;
matB=matBo(:,1:5);
matC=matCo;
matD=zeros(4,5);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% OUTPUT BUS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
MODEL_OUT.TEMPERATURAS =
matAo*MODEL_INI.TEMPERATURAS+matBo*MODEL_INI.INPUT;
MODEL_INI.TEMPERATURAS = MODEL_OUT.TEMPERATURAS;
CONTROL.PARAM.TEMPERATURAS = MODEL_OUT.TEMPERATURAS;

return
```

A continuación, mostraremos las resistencias y capacidades térmicas utilizadas para el modelado de la planta:

Resistencias térmicas [W/K]									
R ₁₂	R ₁₃	R ₁₄	R ₂₃	R ₂₄	R ₃₄	R ₁₀	R ₂₀	R ₃₀	R ₄₀
0,017	0,015	0,025	0,014	0,014	0,014	0,033	0,067	0,087	0,087

Tabla 1. Resistencias térmicas

Capacidades térmicas [W*s/K]			
C ₁	C ₂	C ₃	C ₄
7.5e5	3.6e5	3.6e5	3.1e5

Tabla2. Capacidades térmicas

En la tabla 3 se muestra las potencias nominales de los radiadores de cada habitación. Proporcionan la potencia nominal cuando están completamente abiertos y 0 cuando están apagados. Para obtener valores intermedios utilizaremos un PWM con valor medio la potencia que se quiera aplicar.



Potencias térmicas de los radiadores [W]			
P ₁	P ₂	P ₃	P ₄
2450	1250	1250	850

Tabla 3. Potencias nominales de los radiadores

IV.3 DISEÑO DEL CONTROL PREDICTIVO

En este proyecto se ha elegido un período de muestreo de 5 minutos, un horizonte de control de 5 períodos de muestreo, y un horizonte de predicción de 25 de períodos de muestro. Esto quiere decir que, cada 5 minutos, el control realizará un nuevo cálculo de mandos, se calcularán los primeros 25 mandos, es decir, el horizonte de control, pero este estará basado en el horizonte de predicción de 125 minutos.

Para suavizar los cálculos que tiene que realizar el control en cada período de muestreo supondrá que se aplicará el último mando del horizonte de control para el resto de valores del horizonte de predicción no calculados.

Se ha diseñado gracias a la herramienta Model Predictive Control Toolbox de Matlab. Esta herramienta proporciona funciones, una aplicación y bloques Simulink para diseñar y simular controladores predictivos modelo (MPC). Te permite especificar modelos de plantas y perturbaciones, horizontes, restricciones y pesos. Al ejecutar simulaciones de circuito cerrado, puede evaluar el rendimiento del controlador. Puede ajustar el comportamiento del controlador variando sus pesos y restricciones en tiempo de ejecución.

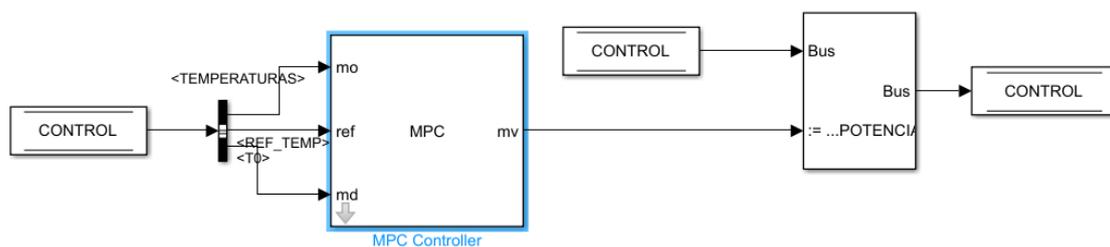


Figura 27. MPC Controller de Simulink

Lo primero que hay que hacer para utilizar el MPC Controller de Simulink es crear un objeto mpc en el espacio de trabajo de MATLAB

A partir del modelo de espacio de estado de la planta se establecen algunas de las propiedades opcionales del modelo:



```
%Linealización de los puntos de los predictivos
%Previo: definiciones para MPC designer
TimeUnit='seconds';
InputName=['P1', 'P2', 'P3', 'P4', 'T0', Pd1 , Pd2 , Pd3 , Pd4];
InputUnit={'W/h', 'W/h', 'W/h', 'C°'};
OutputName={'T1', 'T2', 'T3', 'T4'};
OutputUnit={'C°', 'C°', 'C°', 'C°'};
StateName={'T1', 'T2', 'T3', 'T4'};

P_1_mpc=ss(Pss.a,Pss.b,Pss.c,Pss.d, 'TimeUnit',TimeUnit);
P_1_mpc.InputName = InputName;
P_1_mpc.StateName = StateName;
P_1_mpc.OutputName = OutputName;
P_1_mpc.InputGroup.MV = 1:4;
P_1_mpc.InputGroup.UD = [6:9];
P_1_mpc.InputGroup.MD = [5];
P_1_mpc.OutputGroup.MO = 1:4;
P_1_mpc.OutputGroup.UO = [];
```

Como podemos observar en el código se establecen los nombres de las entradas del modelo $[P_1, P_2, P_3, P_4, T_0, P_{d1}, P_{d2}, P_{d3}, P_{d4}]$ con sus respectivas unidades W/h para las potencias y C° para la temperatura exterior, las salidas $[T_1, T_2, T_3, T_4]$ en °C y las variables de estado $[T_1, T_2, T_3, T_4]$.

A continuación, se establece cuáles de las entradas son las variables manipuladas (“MV” Manipulated variables), es decir las 4 potencias, la temperatura exterior correspondería con una perturbación medible (“MD” measured disturbance), por último las potencias de disipación serán perturbaciones no medibles (“UD” unmeasured disturbance). En cuanto a las salidas, se establecen cuáles son las salidas medibles (“MO” measured output) en este caso las cuatro temperaturas. No hay salidas no medibles (UO unmeasured output).

A continuación, como podemos observar en la figura X (MPC control system) conectamos las entradas al control a través del bus de control. Las temperaturas se conectan con las salidas medibles (MO), la temperatura exterior con la perturbación medible y en la referencia colocamos las temperaturas calculadas por el control, es decir a la temperatura que tienen que llegar la habitación. Por último, actualizamos las salidas con los nuevos mandos a aplicar, es decir, las potencias. Como podemos ver en la Figura de abajo asignamos como hemos hecho anteriormente el número correcto de variables medibles, perturbaciones y salidas medibles y no medibles para ajustar sus entradas y salidas.

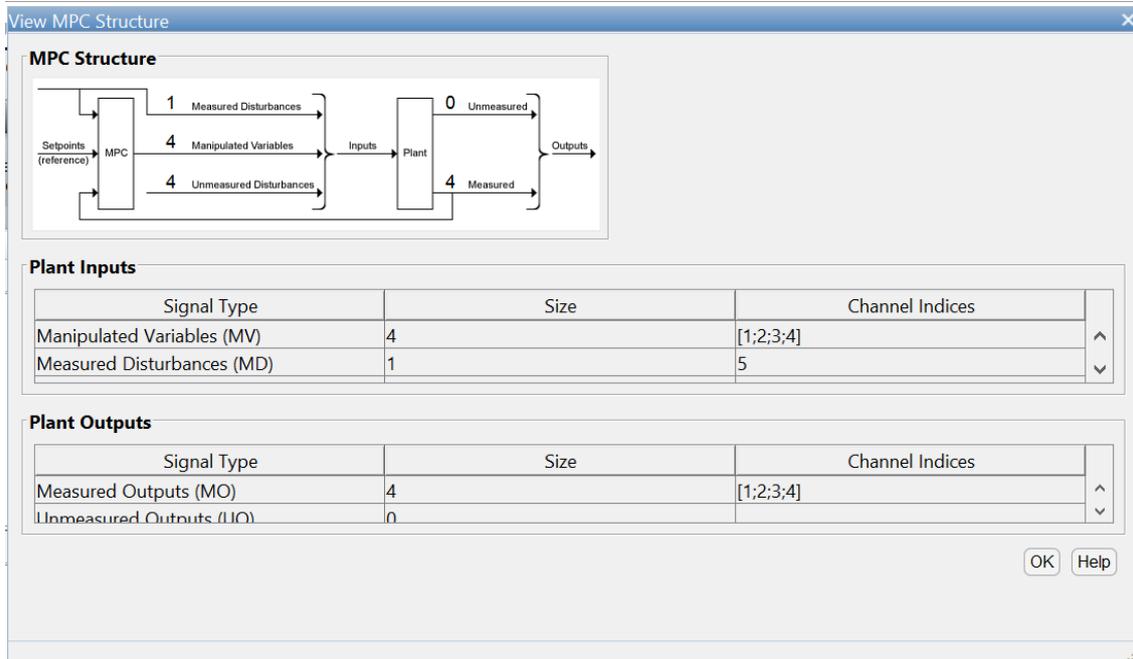


Figura 28. MPC Structure

Una vez asignadas las señales correctamente pulsamos definir y linealizar creándose un controlador MPC predeterminado utilizando la planta linealizada como modelo de predicción interna además de un escenario de simulación determinado.

En TUNING colocamos el sampling time del control (300 s), el horizonte de predicción (25) y el horizonte de control.

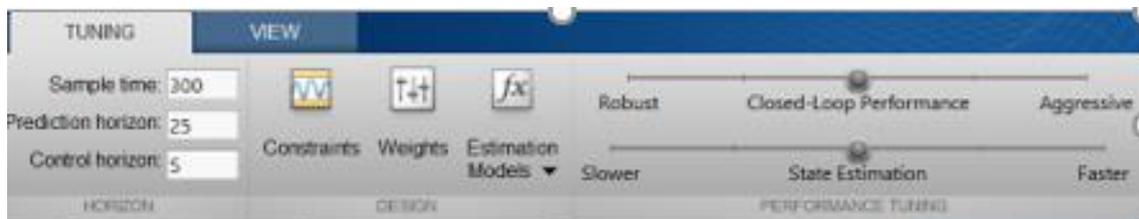


Figura 29. Ajustes del control

Ajustamos las restricciones del control tanto de los mandos como de las salidas.

Los mandos tienen como restricción mínima la potencia 0, es decir apagados y como máxima la potencia nominal.

Las salidas no tienen restricciones.

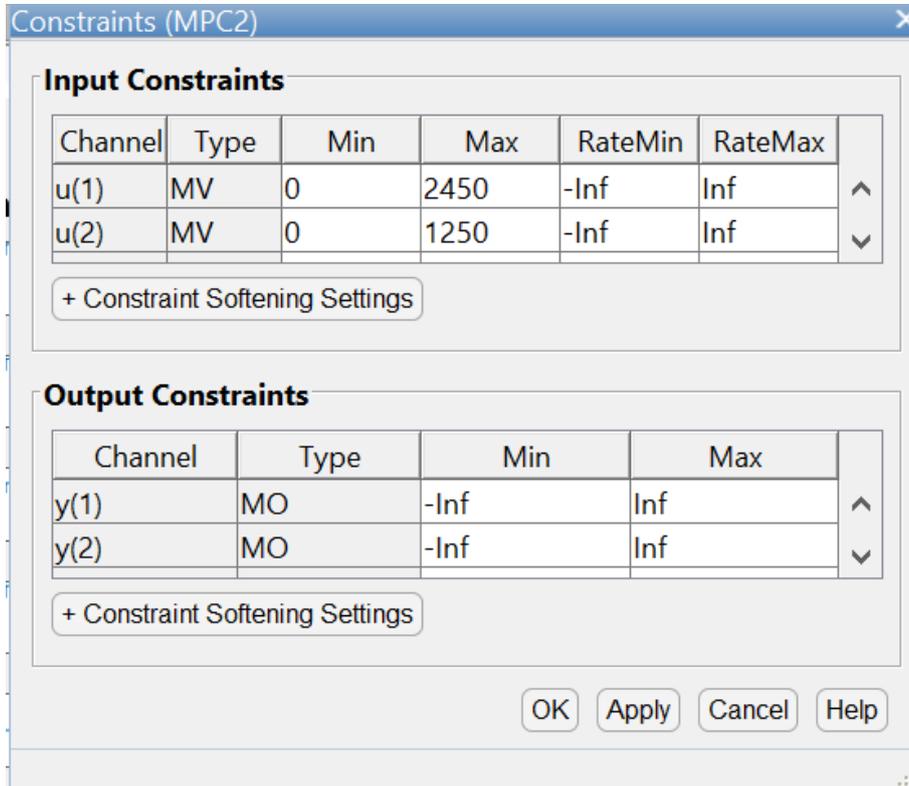


Figura 30. Restricciones de las entradas y salidas

Por último, asignamos los pesos a las entradas y salidas que se encargan de relajar y restringir las variables. Estos pesos son los que utiliza la función objetivo que se minimiza en el control. Tras hacer varias pruebas los pesos óptimos para cada variable son:

Inputs	P ₁	P ₂	P ₃	P ₄
Weights	0,0073	0,016	0,016	0,023

Tabla 4. Pesos de los mandos

Outputs	T ₁	T ₂	T ₃	T ₄
Weights	1	1,2	1,4	1,27

Tabla 5. Pesos de las salidas

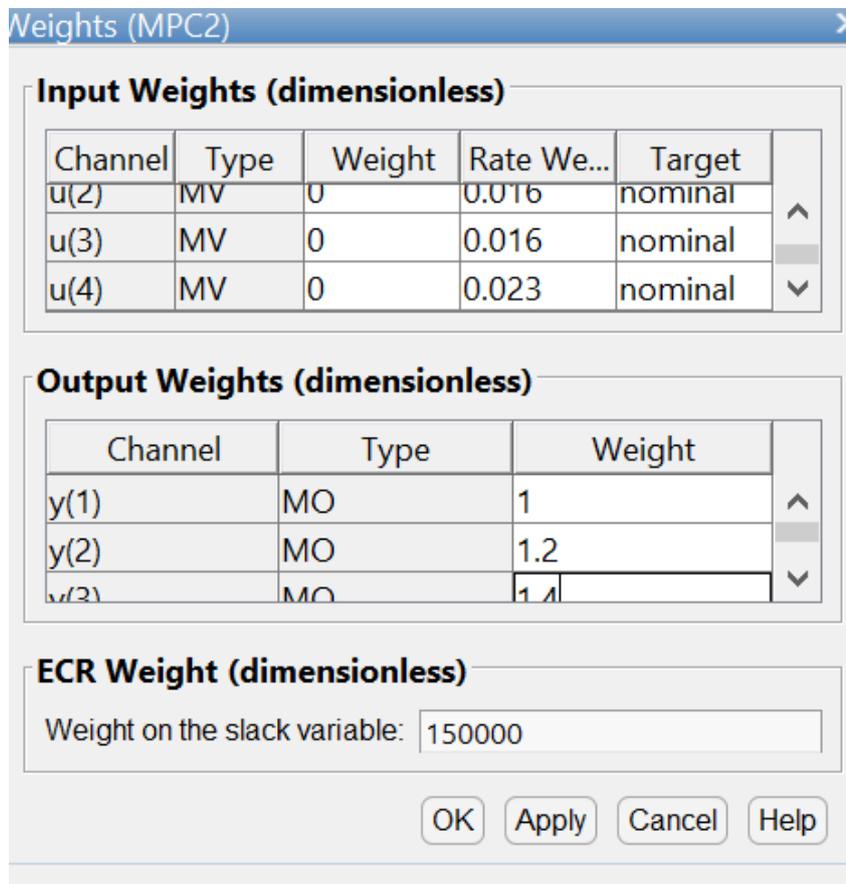


Figura 31. Pesos MPC

IV.4 ESTUDIO DE SENSIBILIDAD

Por último, vamos a realizar un estudio de sensibilidad creando diferentes escenarios en la herramienta del MPCDesigner para poder simular en ella y hacernos una idea del funcionamiento del control.

En la figura 33 ver el escenario de simulación de la herramienta MPCDesigner. En él podemos ajustar las características de la simulación a nuestro gusto.

Simulation duration: Tiempo de duración de la simulación.

Reference Signals: Son las referencias de temperatura que debe seguir el control. En este caso están representadas por un escalón que asciende nada más empezar la simulación de 15°C a 21°C. Establece el tiempo de duración de simulación. En este caso se ha elegido 86400s (24h).



Measured Disturbances: Se refiere a las perturbaciones medibles. En nuestro caso la temperatura exterior. Para hacerlo lo más semejante a la realidad la hemos simulado como un seno que varía entre 5°C y 11°C según la hora del día.

Unmeasured Disturbances: Hace referencia a las perturbaciones no medibles. En nuestro caso las potencias de dispersión. Les hemos dado valores de: 40W, 30W, 30W y 20W para Pd₁, Pd₂, Pd₃ y Pd₄ respectivamente.

Simulation Settings

Plant used in simulation: Default (controller internal model) ▼

Simulation duration (seconds) 86400

Run open-loop simulation Use unconstrained MPC

Preview references (look ahead) Preview measured disturbances (look ahead)

Reference Signals (setpoints for all outputs)

Channel	Name	Nominal	Signal	Size	Time	Period	
r(3)	Ref of T3	15	Step ▼	6	1		^
r(4)	Ref of T4	15	Step ▼	6	1		v

Measured Disturbances (inputs to MD channels)

Channel	Name	Nominal	Signal	Size	Time	Period	
u(5)	T0	5	Sine ▼	6	300	170000	

Unmeasured Disturbances (inputs to UD channels)

Channel	Name	Nominal	Signal	Size	Time	Period	
u(8)	Pd3	0	Pulse ▼	30	0	5000	^
u(9)	Pd4	0	Pulse ▼	20	0	5000	v

Output Disturbances (added at MO channels)

Channel	Name	Nominal	Signal	Size	Time	Period	
y(1)	T1	0	Const... ▼				^
y(2)	T2	0	Const... ▼				v

Load Disturbances (added at MV channels)

Channel	Name	Nominal	Signal	Size	Time	Period	
u(1)	P1	0	Const... ▼				^
u(2)	P2	0	Const... ▼				v

Figura 32. Escenario de simulación de MPCDesigner.

A continuación, hemos realizado la simulación para tener una idea del funcionamiento del control y hemos obtenido:

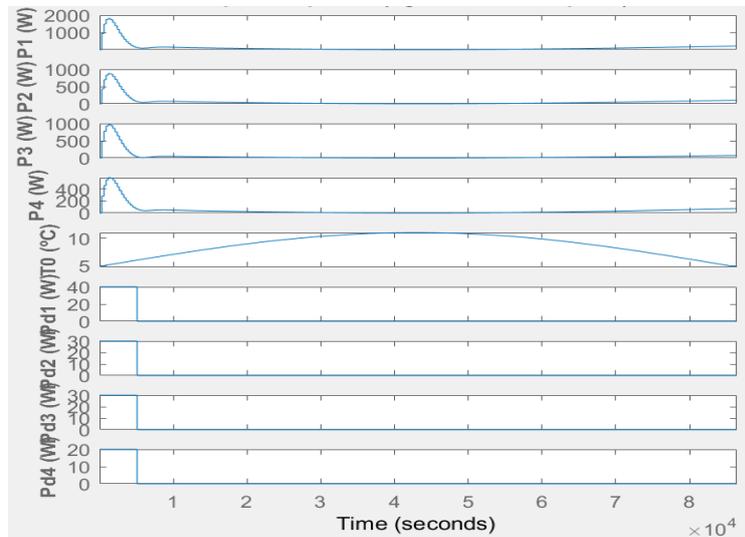


Figura 33. Entradas simulación 1 MPCDesigner

En la habitación 1 la potencia máxima alcanza los 1810 W, el 73,9% de su capacidad nominal, conforme aumenta la temperatura exterior, la potencia 1 llega a alcanzar los 0 W en torno a los 41900s (11,6h) a partir de ahí la potencia vuelve a crecer ya que la temperatura exterior vuelve a caer, alcanzando un valor de 203 W al final de la simulación

En la habitación 2 la potencia máxima alcanza los 878 W, el 70,2% de su capacidad nominal, conforme aumenta la temperatura exterior, la potencia 2 llega a alcanzar los 0 W en torno a los 42700s (11,9h) a partir de ahí la potencia vuelve a crecer ya que la

temperatura exterior vuelve a caer, alcanzando un valor de 103 W al final de la simulación.

En la habitación 3 la potencia máxima alcanza los 969 W, el 77,5% de su capacidad nominal, conforme aumenta la temperatura exterior, la potencia 3 llega a alcanzar los 0 W en torno a los 42500s (11,8h) a partir de ahí la potencia vuelve a crecer ya que la temperatura exterior vuelve a caer, alcanzando un valor de 71,9 W al final de la simulación.

En la habitación 4 la potencia máxima alcanza los 593 W, el 69,7% de su capacidad nominal, conforme aumenta la temperatura exterior, la potencia 4 llega a alcanzar los 0 W en torno a los 42900s (11,9h) a partir de ahí la potencia vuelve a crecer ya que la temperatura exterior vuelve a caer, alcanzando un valor de 71,3 W al final de la simulación.

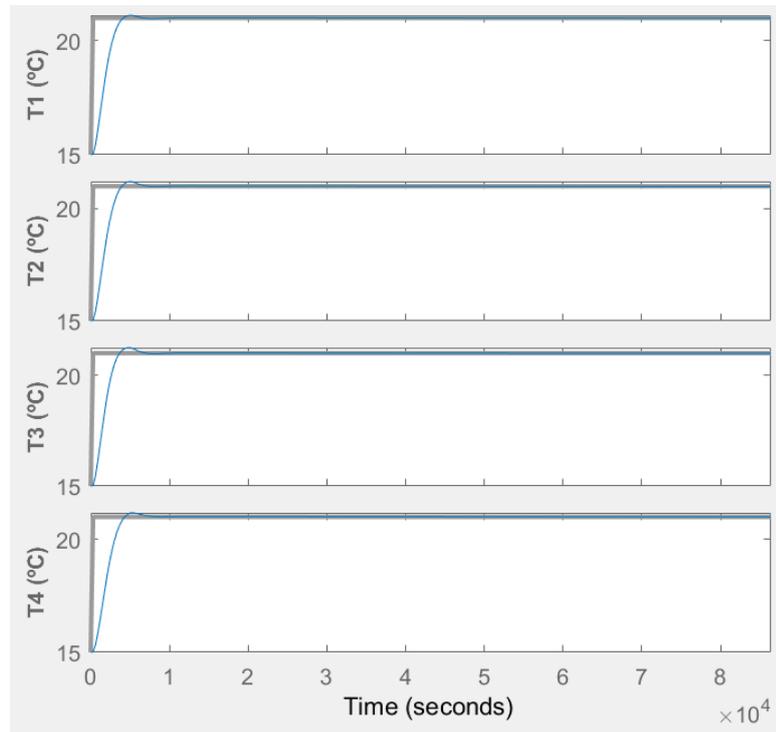


Figura 34. Salidas simulación 1 MPCDesigner

La temperatura 1 tiene un sobrepaso de 0.1 y llega al valor deseado a los 4140 (69 min).

La temperatura 2 tiene un sobrepaso de 0.2 y llega al valor deseado a los 3940 (65,6 min).

La temperatura 3 tiene un sobrepaso de 0.2 y llega al valor deseado a los 3670 (61,2 min).

La temperatura 4 tiene un sobrepaso de 0.2 y llega al valor deseado a los 4360 (72.6 min)

El control ha respondido de forma eficaz con los valores de los pesos elegidos. A continuación, vamos a cambiar el valor de los pesos a:

Inputs	P_1	P_2	P_3	P_4
Weights	0,009	0,03	0,03	0,02

Tabla 6. Pesos entradas simulación 4

Outputs	T_1	T_2	T_3	T_4
Weights	1	1,2	1,3	1,1

Tabla 7. Pesos salidas simulación 4

Hemos aumentado los pesos de las entradas y disminuido los de las salidas, de esta manera se consigue un control más lento, pero más robusto y sin sobrepasos.

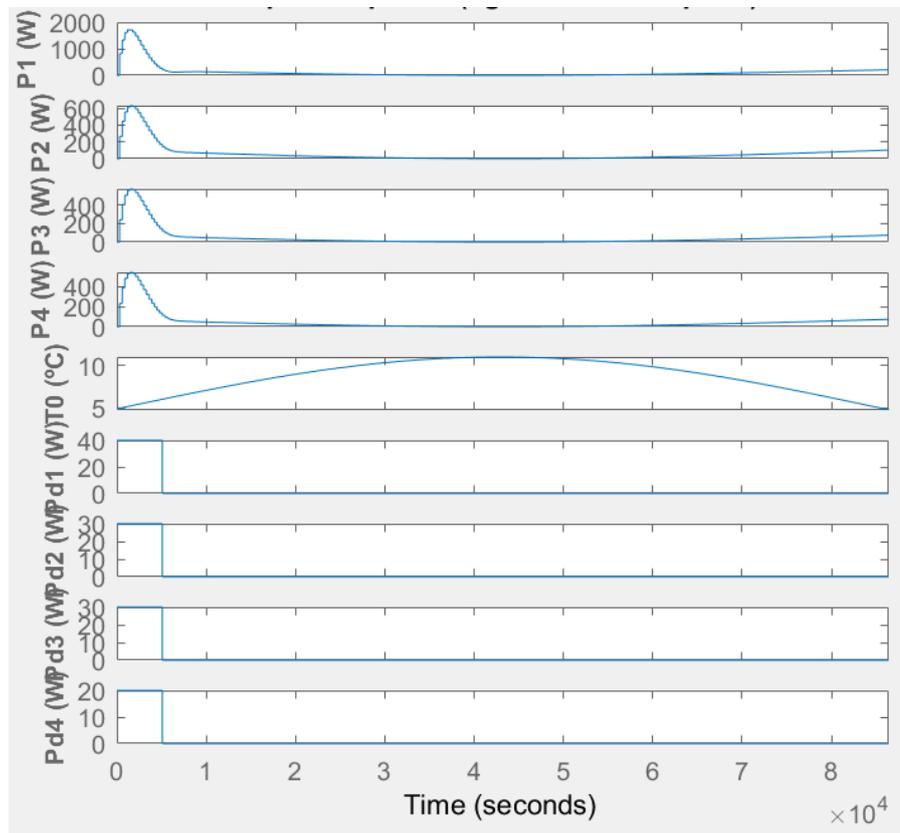


Figura 35. Entradas simulación 2 MPCDesigner

En las entradas ya podemos observar la acción del cambio de pesos,

El valor máximo de las 4 potencias es menor que en la simulación anterior:

Valor máximo P1: 1720 W, el 70% de su capacidad nominal

Valor máximo P2: 632 W, el 50,6% de su capacidad nominal

Valor máximo P3: 571W, el 45,7% de su capacidad nominal

Valor máximo P4: 539 W, el 63,4% de su capacidad nominal

Las potencias de las habitaciones 2 y 3 han bajado considerablemente respecto a la anterior simulación. Se han modificado los pesos de entrada de 0,025 a 0,03 en cada una, pero han sido los cambios en los pesos de las salidas lo que más ha afectado de 1.3 para P2 y 1.5 para p3 a 1.2 y 1.3 respectivamente.

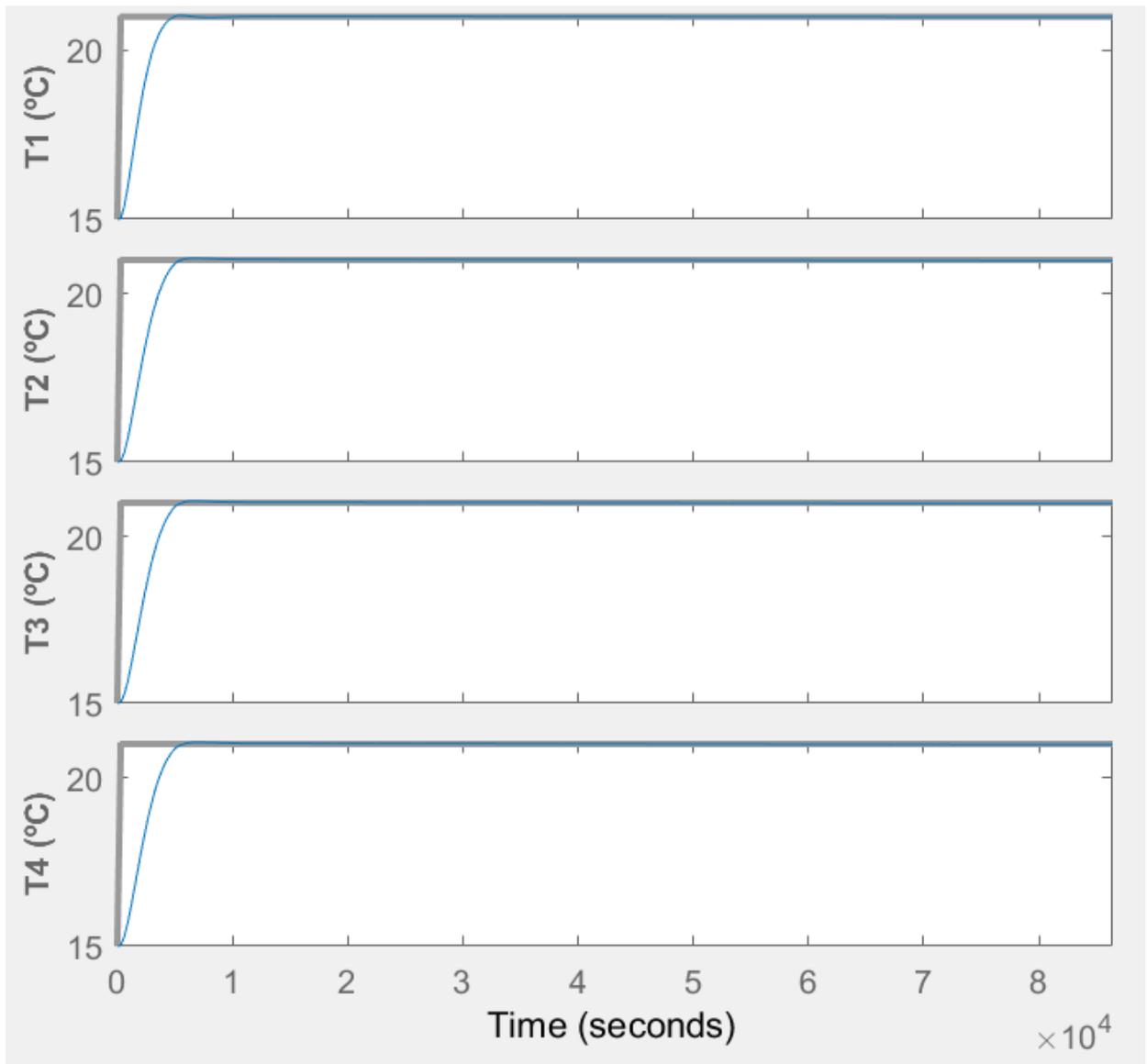


Figura 36. Salidas simulación 2 MPCDesigner

En cuanto a las salidas, podemos observar que su sobrepaso es prácticamente nulo (0.01 en cada una de ellas). En contraposición, los tiempos de respuesta se han vuelto más lentos:

La temperatura 1 llega a referencia a los 4910s, 82 min

La temperatura 2 llega a referencia a los 5540s, 92 min.

La temperatura 3 llega a referencia a los 5610s, 93,5 min

La temperatura 4 llega a referencia a los 5710s, 95 min



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
GRADO EN INGENIERÍA ELECTROMECÁNICA



Como hemos podido observar manipulando los pesos podemos influir en el control, si bajamos el peso de las entradas y subimos el de las salidas conseguimos un control más rápido y agresivo es un control más inestable y con más sobrepasos, mientras que si subimos el peso de las entradas y bajamos el de las salidas conseguimos un control más robusto y con menos sobrepaso, pero más lento.



Capítulo V: SIMULACIONES

Una vez probado el control predictivo en la herramienta de MPCDesigner de la Toolbox de Matlabn en este capítulo realizaremos diferentes simulaciones con el modelo diseñado y el control predictivo propuesto.

Los pesos elegidos para las simulaciones serán:

Inputs	P ₁	P ₂	P ₃	P ₄
Weights	0,0073	0,016	0,016	0,023

Tabla 8. Pesos de los mandos

Outputs	T ₁	T ₂	T ₃	T ₄
Weights	1	1,2	1,4	1,27

Tabla 9. Pesos de las salidas

V.1 SIMULACIÓN 1. SEGUIMIENTO A LA REFERENCIA

Como primera simulación realizaremos un seguimiento a las temperaturas de referencia. Las cuatro habitaciones se encontrarán a una temperatura de 15 grados y una vez pasados 10 min se encenderá la calefacción de la habitación 1 y tendrá que alcanzar los 21°C sin que el resto alteren su temperatura.

La referencia estará representada por un escalón de valor inicial 15°C que aumentará su valor a 21°C a los 10 min del comienzo de la simulación.

La temperatura exterior se encontrará constante a 0°C.

Las potencias de dispersión serán 0.

El tiempo de simulación será de 3h.

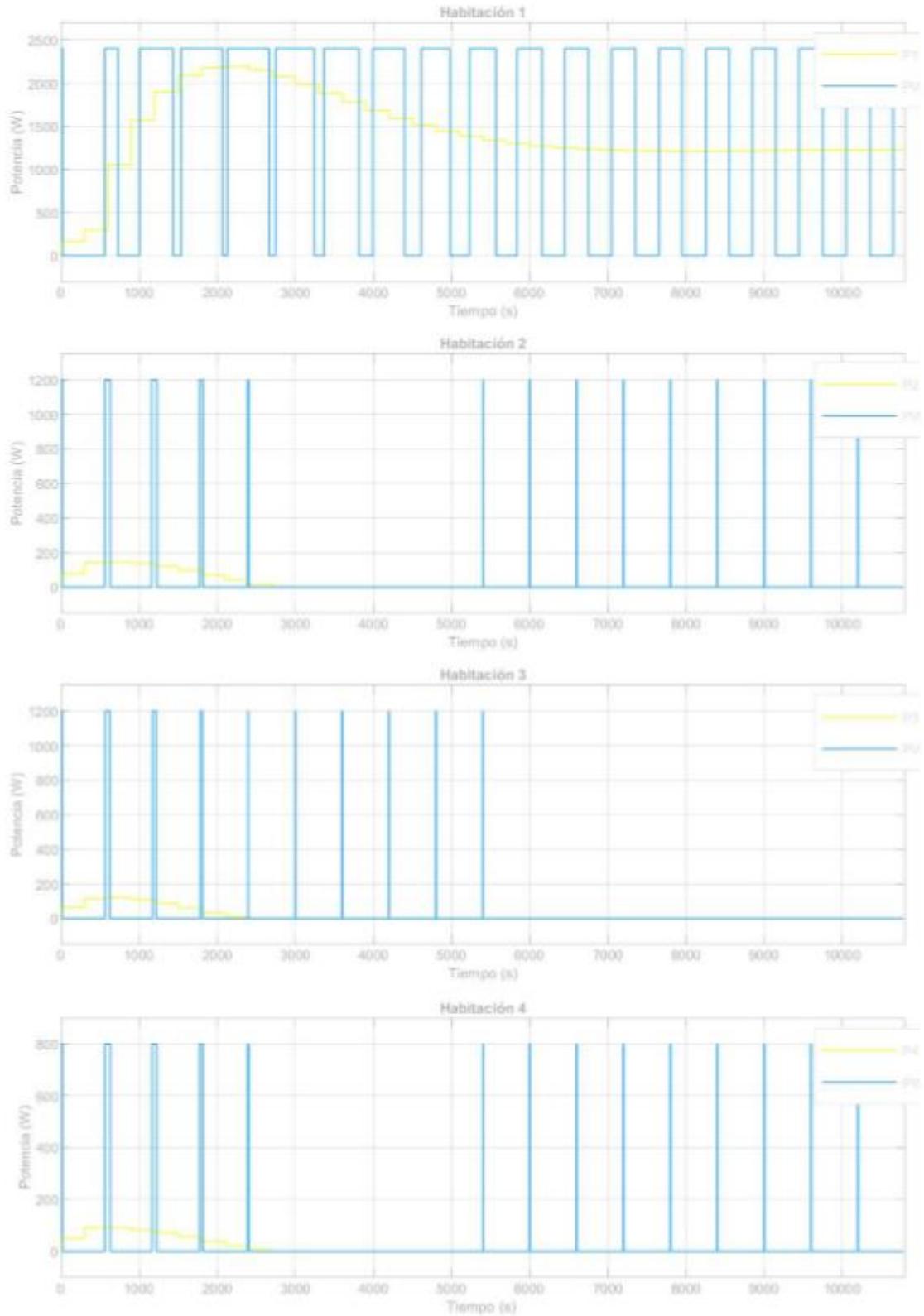


Figura 37. PWM simulación 1.

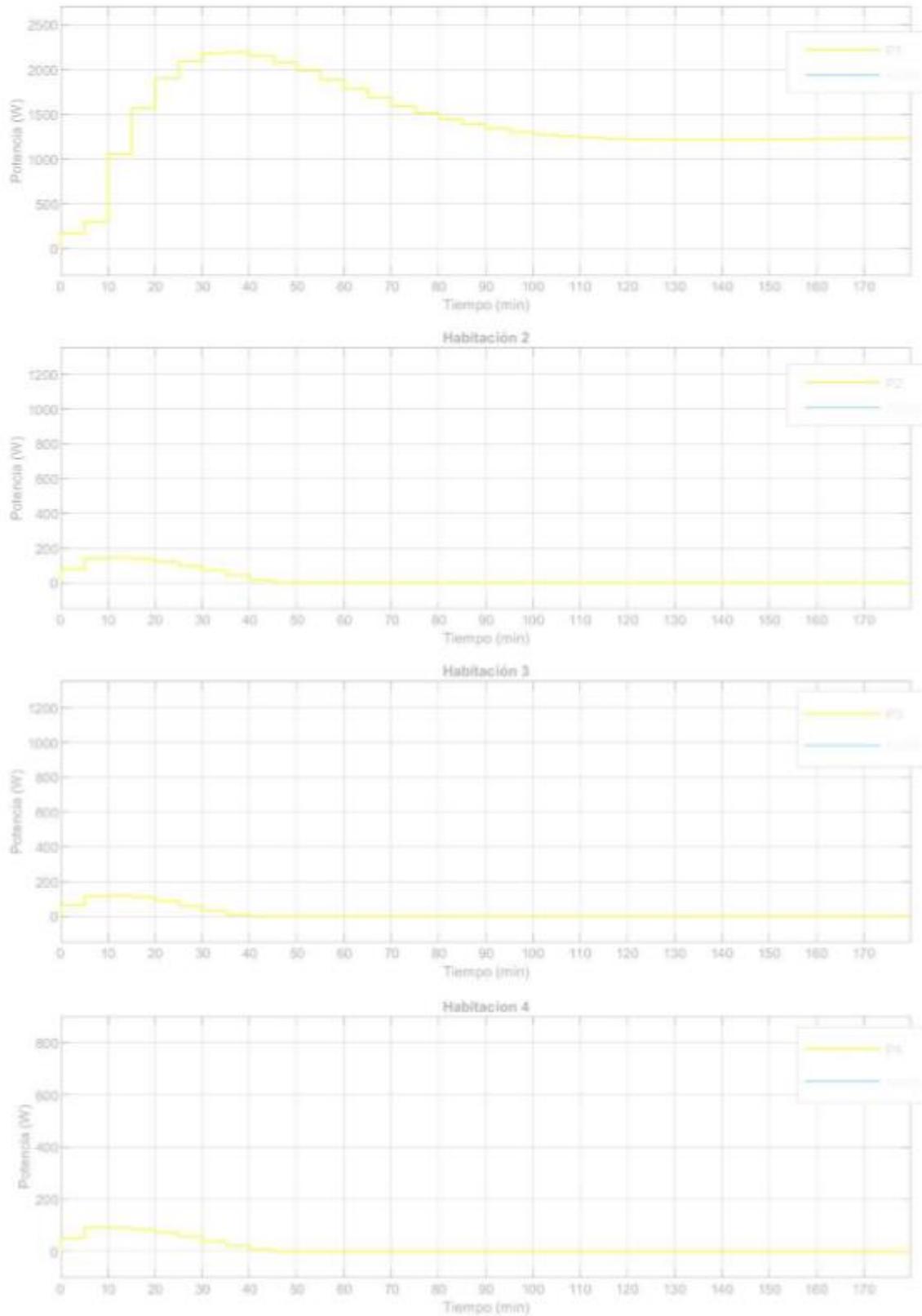


Figura 38. Potencias simulación 1.

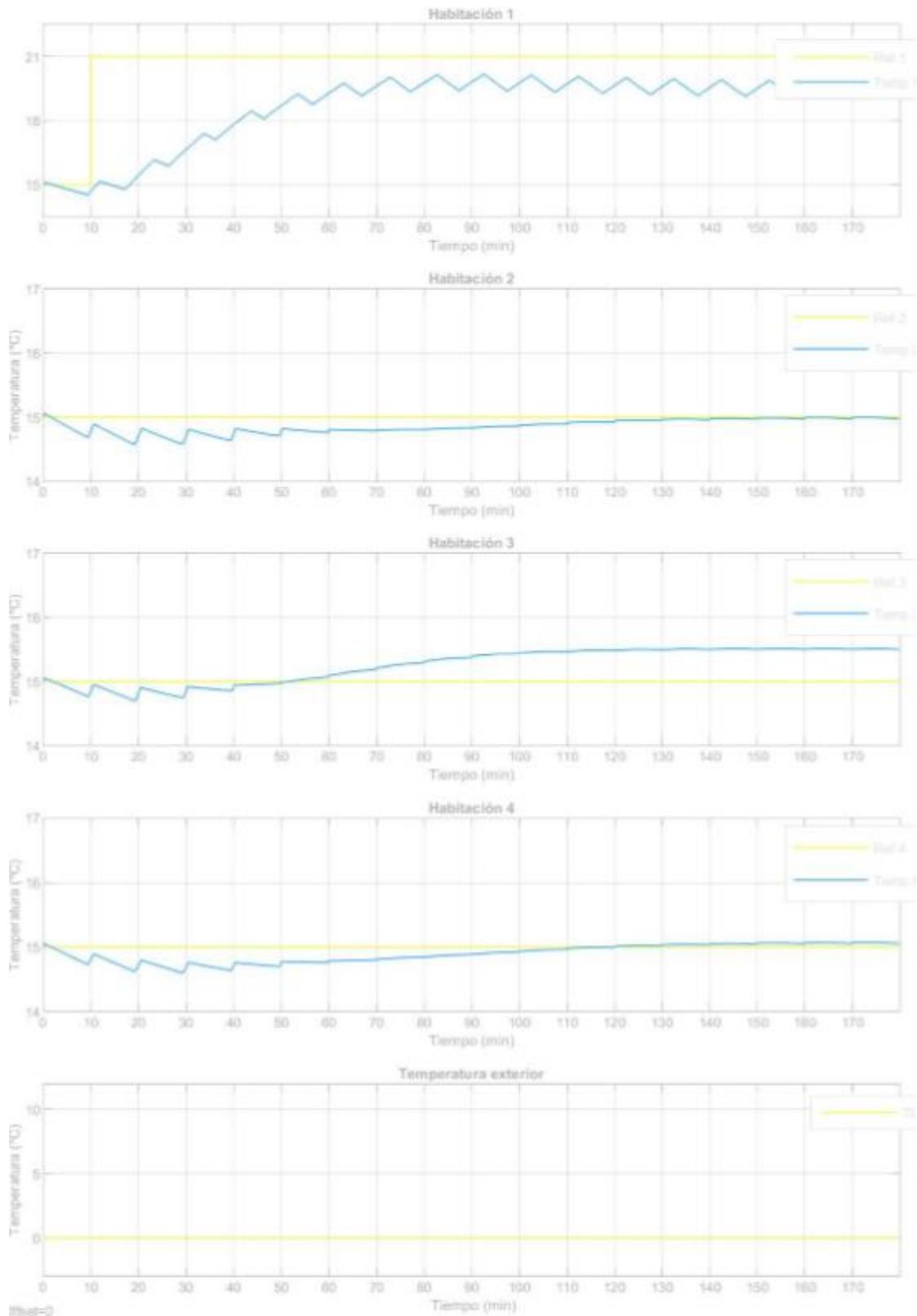


Figura 39. Temperaturas simulación 1



Resultados simulación 1

Potencias de entrada de los radiadores

La potencia máxima que alcanza el radiador cuya referencia asciende a 21° C es 2193 W, el 89,5 % de su capacidad nominal.

El resto de radiadores alcanzan una temperatura máxima 145,1 W, 119,3 W y 92,5 W para P₂, P₃ y P₄ respectivamente.

Temperaturas

La temperatura de la habitación 1 no llega a alcanzar los 21°C, alcanza la estabilidad en torno a los 20,6 °C.

Las temperaturas de las habitaciones 2 y 4 mantienen los 15 °C sin apenas alterarse. Por último, la temperatura de la habitación 3 asciende ligeramente la temperatura a unos 15,5 °C.

Como podemos observar en la simulación la temperatura de la habitación 1 no llega a la referencia de 21°.

Se podría pensar que la potencia del radiador 1 ha saturado, es decir ha llegado a la potencia máxima que puede ofrecer, pero no es así, ya que en la simulación obtenemos una potencia máxima de 2193W y el límite del radiador es de 2450W.

Por eso hemos recurrido a la ecuación de estado en régimen permanente:

$$P_j = A * T_j + B * T_0$$

Ecuación 14. Ecuación de estado en régimen permanente

En donde A y B son las matrices de estado en régimen permanente, $T_j = [22^{\circ}C \ 15^{\circ}C \ 15^{\circ}C \ 15^{\circ}C]^T$, $T_0 = 5^{\circ}C$ y P_j las potencias de los radiadores que queremos obtener. Resolviendo dicha ecuación alguna de las potencias de las habitaciones da un resultado negativo, como es imposible que ninguna potencia de ningún radiador sea negativa (tenemos un límite físico de 0W) no existe ninguna solución en la cuál $T_1 = 21^{\circ}C$, por ello la solución más próxima a la realidad que ofrece el control es de $P_j = [20,6W, 0W, 0W, 0W]^T$ y se obtiene un vector de temperaturas en régimen permanente de $T_j = [20,6^{\circ}C \ 15^{\circ}C \ 15,5^{\circ}C \ 15^{\circ}C]^T$



V.2 SIMULACIÓN 2. RECHAZO A LA PERTURBACIÓN DE T0

Como segunda simulación veremos como el control rechaza las perturbaciones de la T0. Las cuatro habitaciones se encontrarán a una temperatura de 15 grados y deberán mantener dicha temperatura durante toda la simulación a pesar del brusco cambio de temperatura exterior que se producirá

La temperatura exterior se encontrará constante a 15°C hasta los 3000s (50 min) que baje a 0 °C.

Las potencias de dispersión serán nulas.

El tiempo de simulación será de 3h.



Figura 40. PWM simulación 2

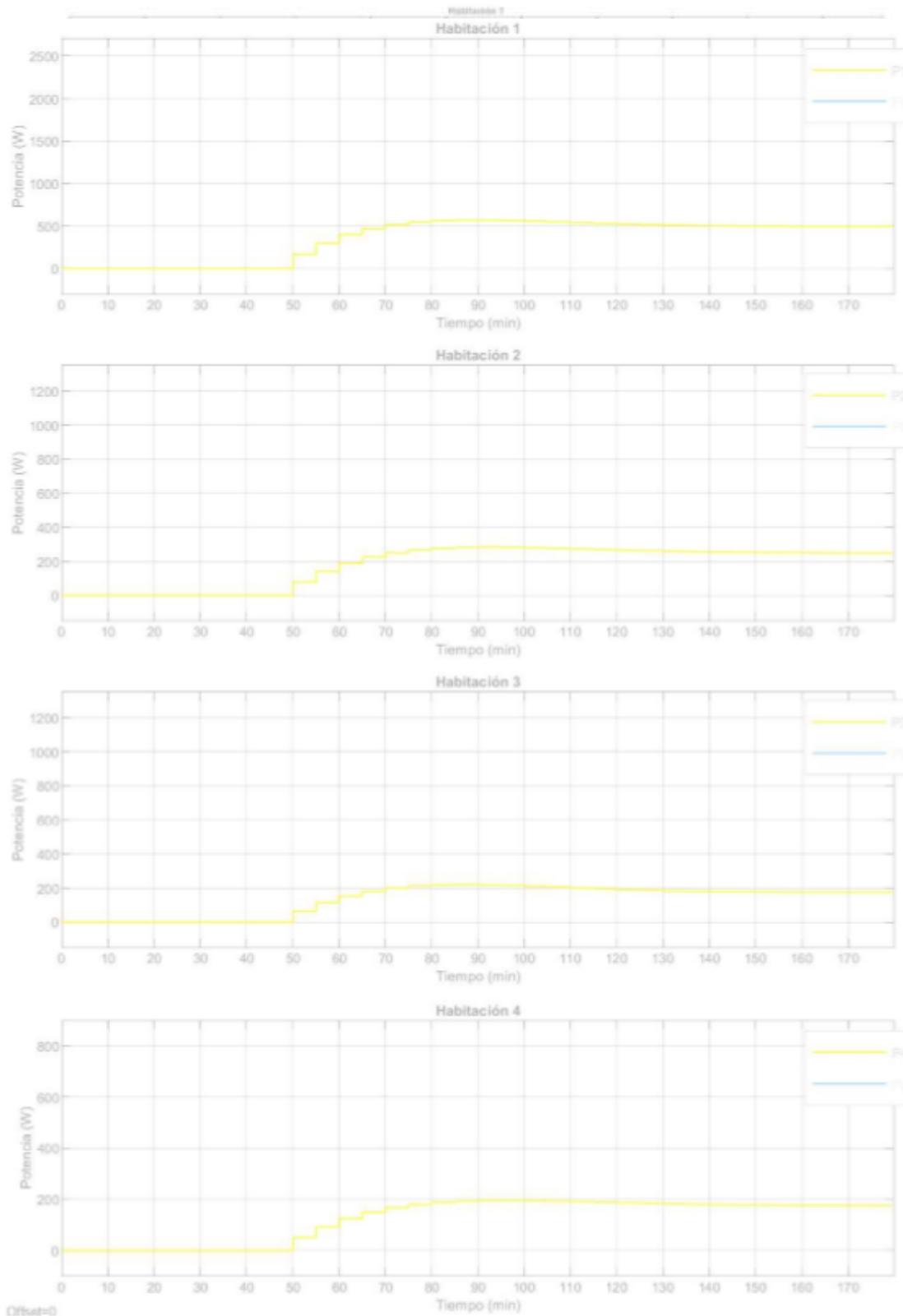


Figura 41. Potencias simulación 2

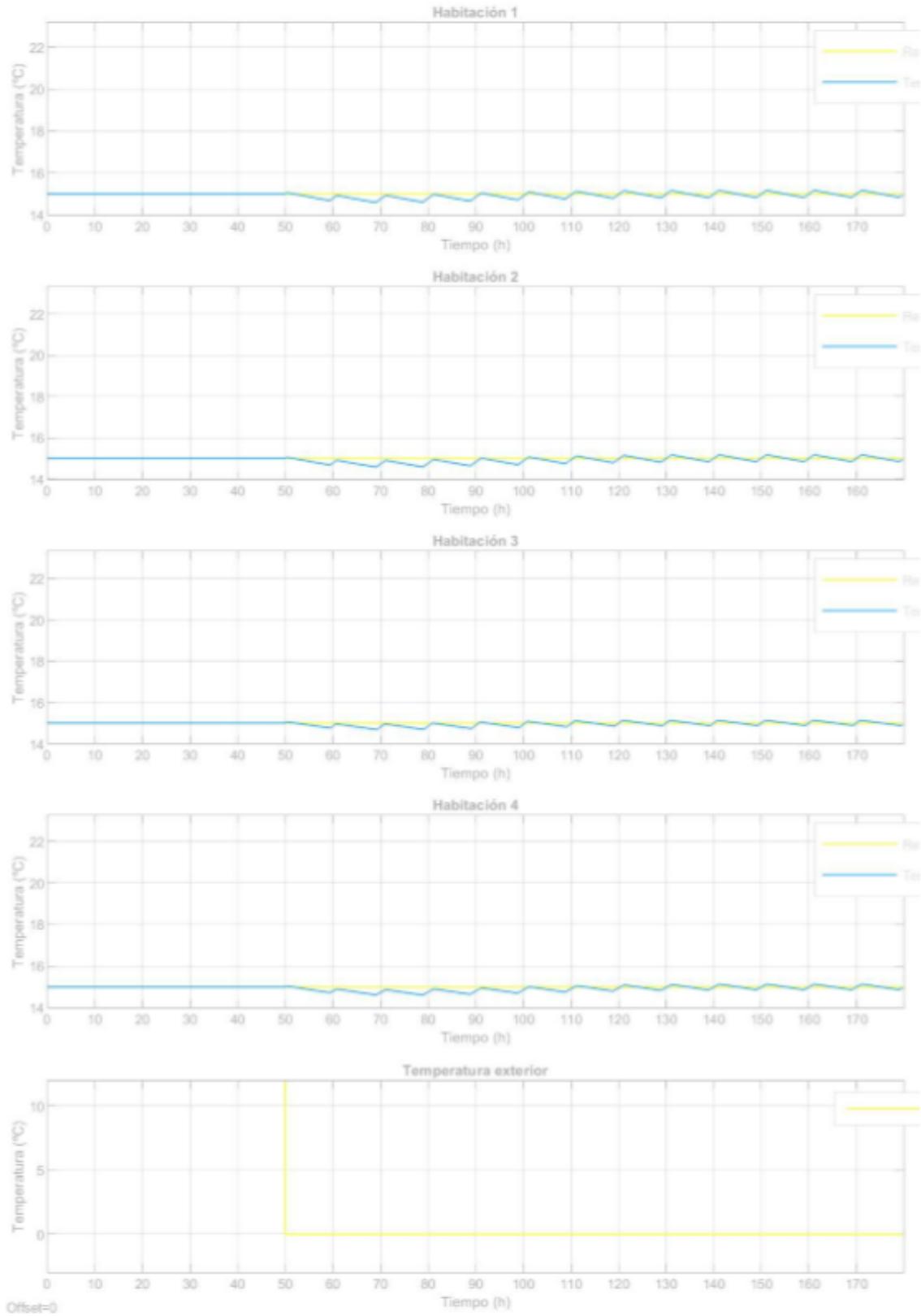


Figura 42. Temperaturas simulación 2



Potencias de entrada de los radiadores

Pasados 3000s (50 min) la temperatura exterior baja bruscamente a 0°C y las potencias empiezan a aumentar alcanzando un valor máximo de 570,4 W, 283,1 W, 219,8 W y 196,1 W para P₁, P₂, P₃ y P₄ respectivamente.

Temperaturas

Las salidas mantienen el valor de la referencia durante toda la simulación esto significa que el control rechaza de forma adecuada la perturbación.

V.3 SIMULACIÓN 3. TEMPERATURA EXTERIOR REAL

En esta simulación temperatura inicial de las habitaciones será de 15°C y pasados 10 min desde el comienzo de la simulación tendrán que alcanzar los 22°C.

En este caso la simulación tendrá una duración de 24h.

La temperatura exterior se asemejará a la de un día de invierno, se encuentra representada por un seno que oscilará a lo largo del día entre los 0° y 10°C según la hora que sea.

Las potencias de dispersión son nulas.

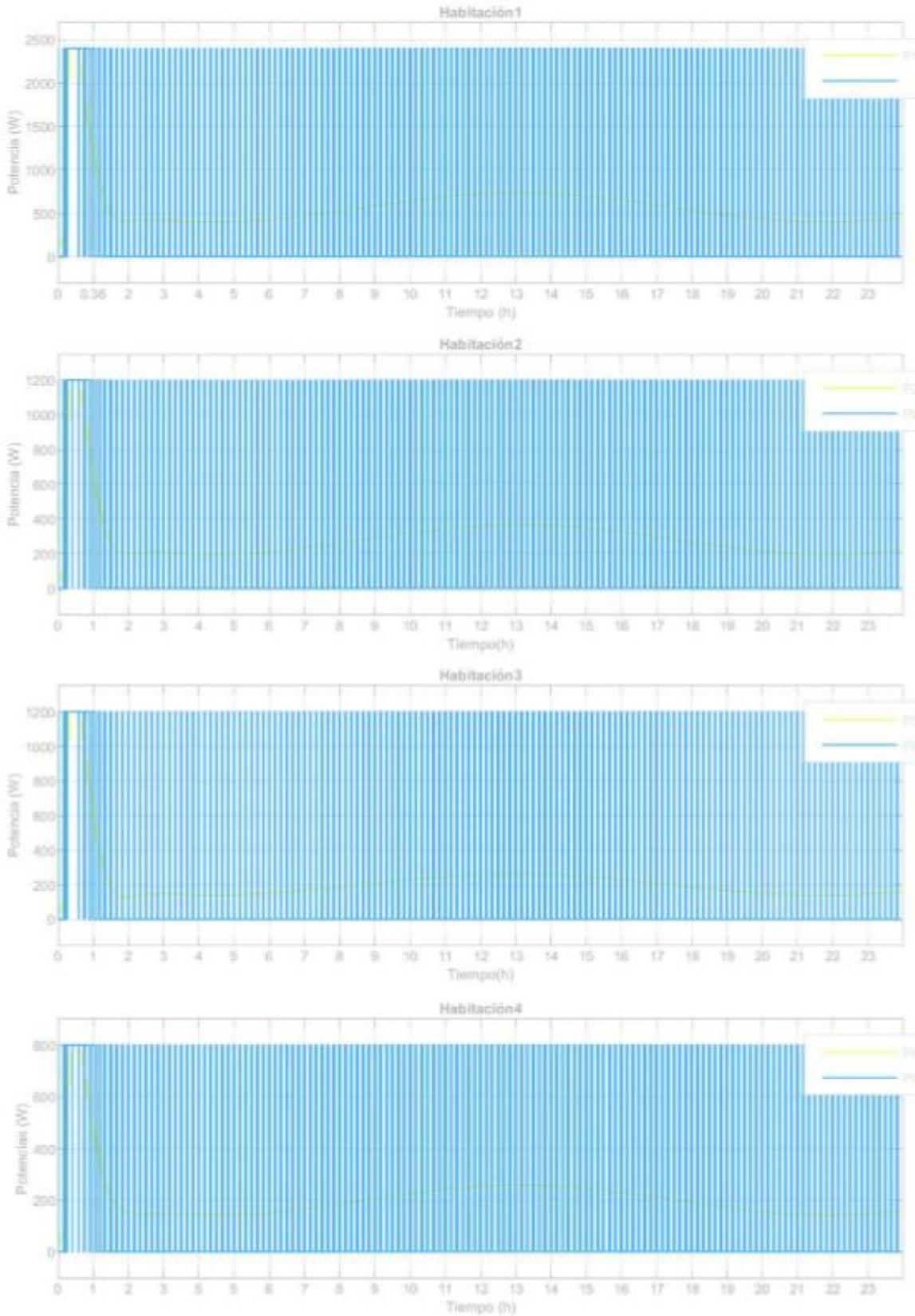


Figura 43. PWM simulación 3

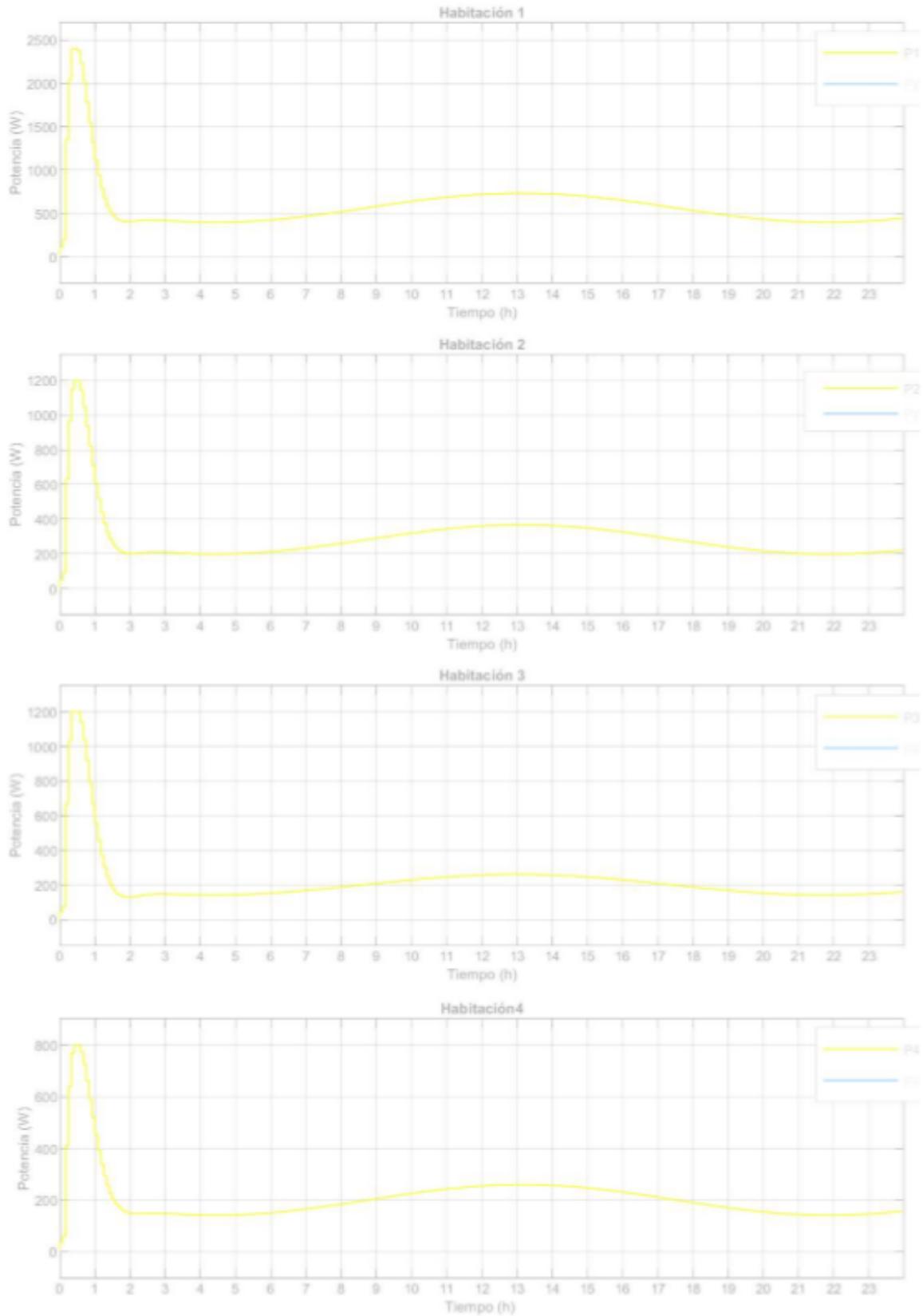


Figura 44. Potencias simulación 3

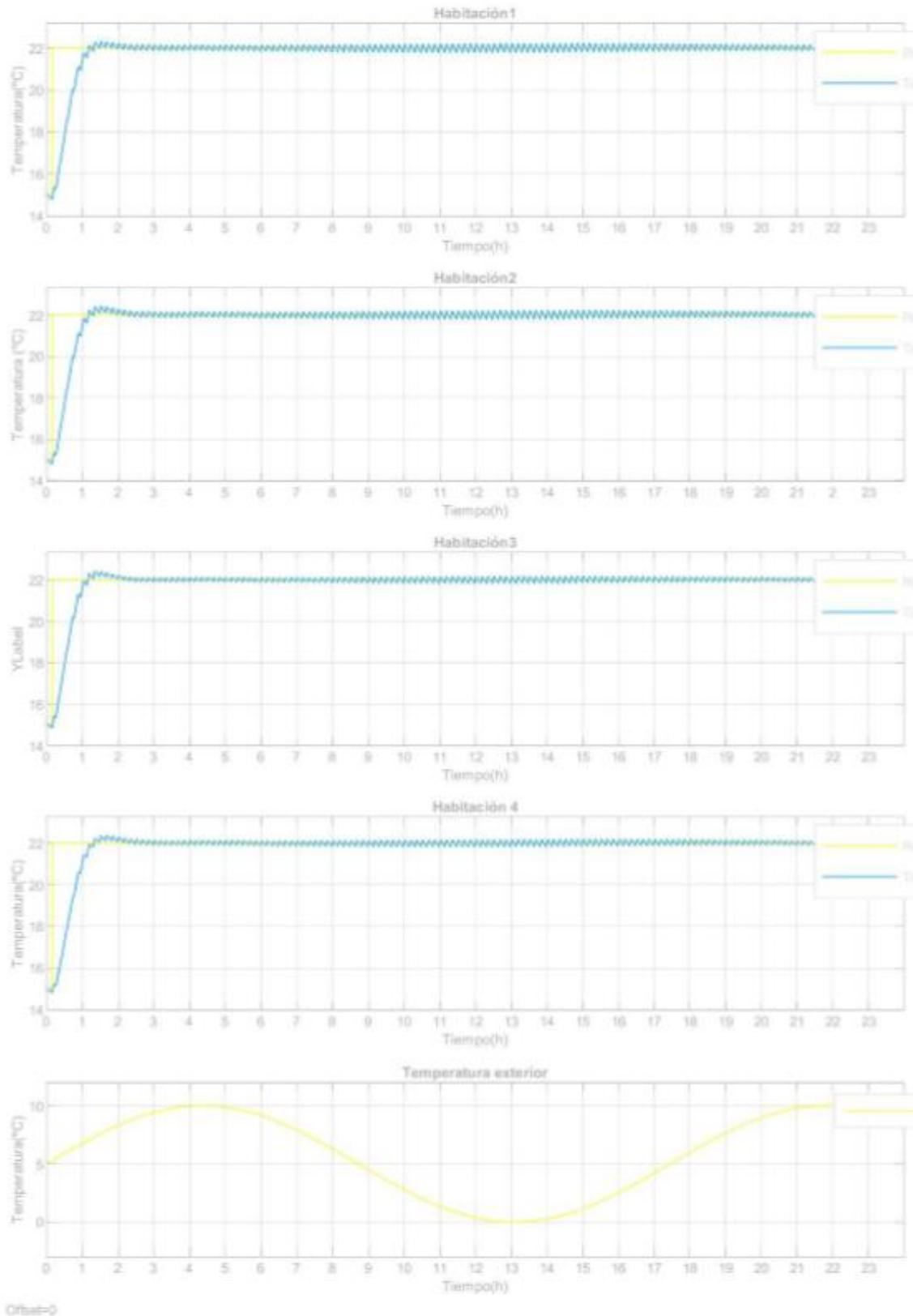


Figura 45. Temperaturas simulación 3



Potencias de entrada de los radiadores

Nada más que comienza la simulación las potencias ya comienzan a aumentar, ya que tienen que compensar el cambio de la temperatura exterior, a los 600s (10 min) hay un salto en la referencia de 15°C a 22°C.

Las potencias máximas que alcanzan son: 2400W, 1200W, 1200W y 800W para P₁, P₂, P₃ y P₄ respectivamente. A continuación, la potencia va aumentando y disminuyendo siguiendo una sinusoidal según el valor de la temperatura exterior, conforme aumenta la T₀ disminuye la potencia y viceversa.

Temperaturas

Las temperaturas llegan a su referencia a los 4222s (70 min), 4192s (70 min), (61,4 min) y 4340s (72 min) y mantiene la referencia durante toda la simulación.

V.4 SIMULACIÓN 4. RECHAZO A LA PERTURBACIÓN Pd.

En esta simulación consideraremos que en el interior de la habitación 3 hay cuatro personas que disipan su propio calor. La Pd₃ estará representada por un escalón que asciende de 0W a 600W a la hora de comenzar la simulación.

Temperatura inicial de las habitaciones: 15°C.

La temperatura de las habitaciones se mantendrá constante a 20°C.

La temperatura exterior se encuentra constante a 5 °C

El tiempo de simulación será de 5h.

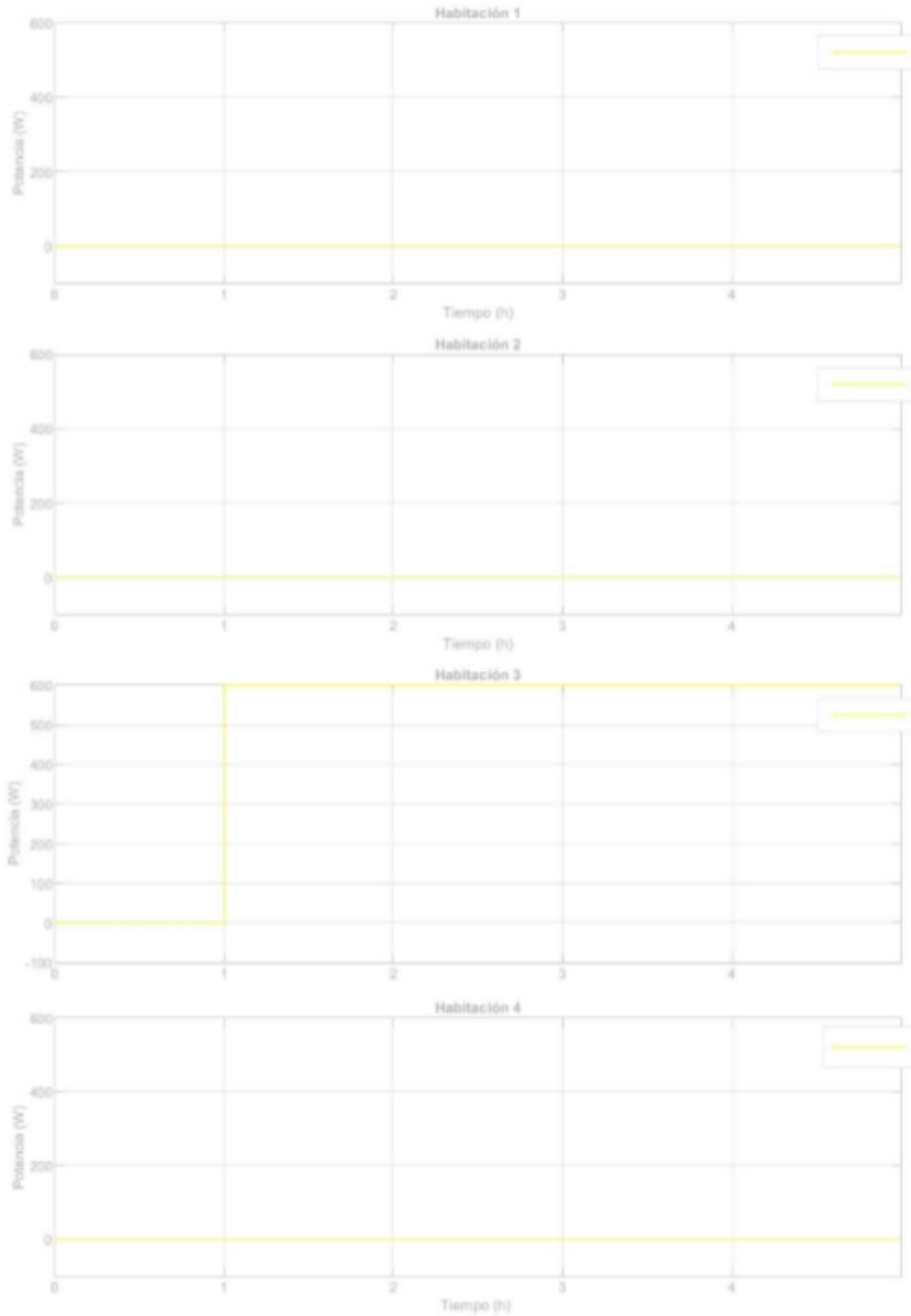


Figura 46. Pd simulación 4



Figura 47. PWM simulación 4

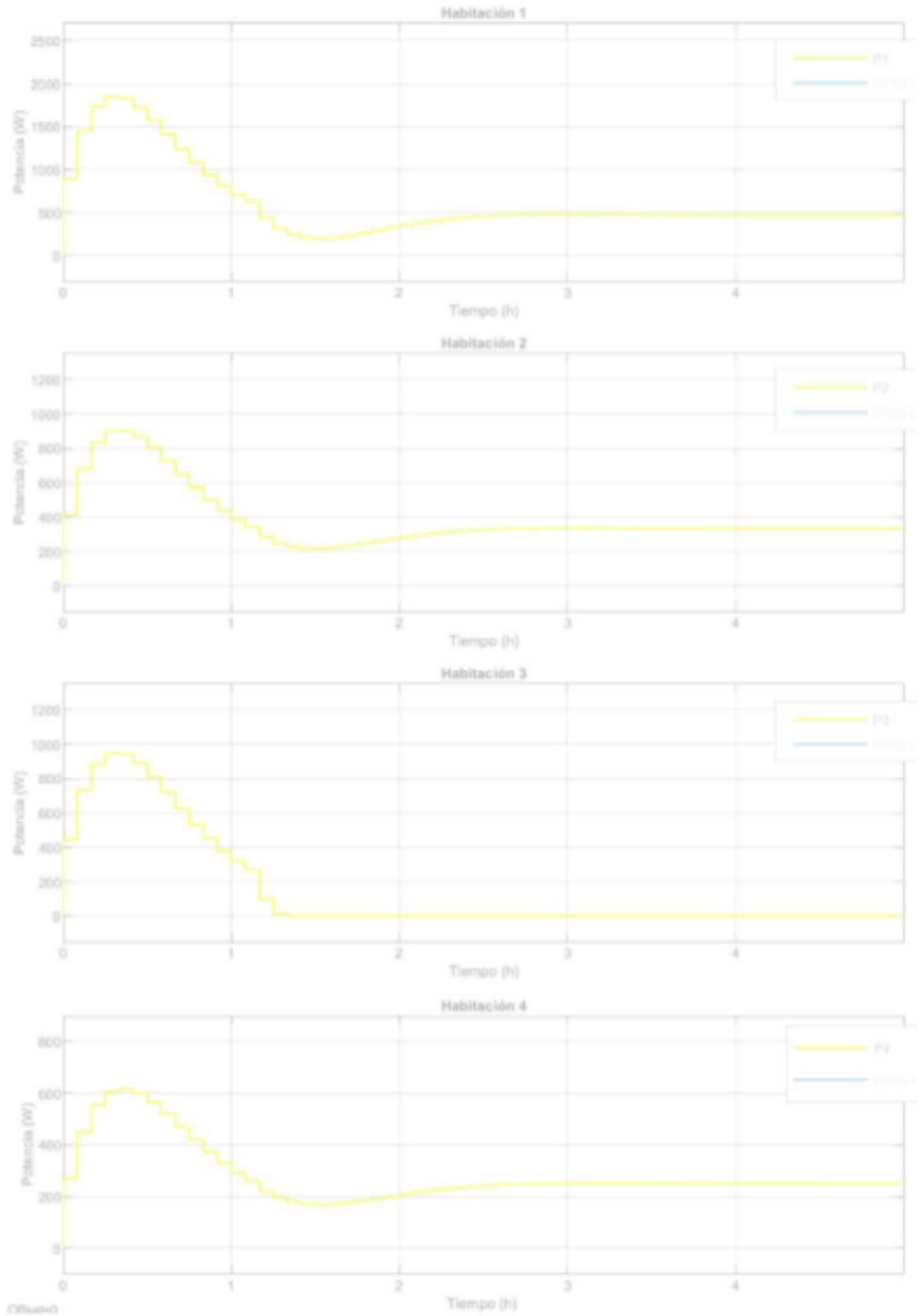


Figura 48. Potencia simulación 4

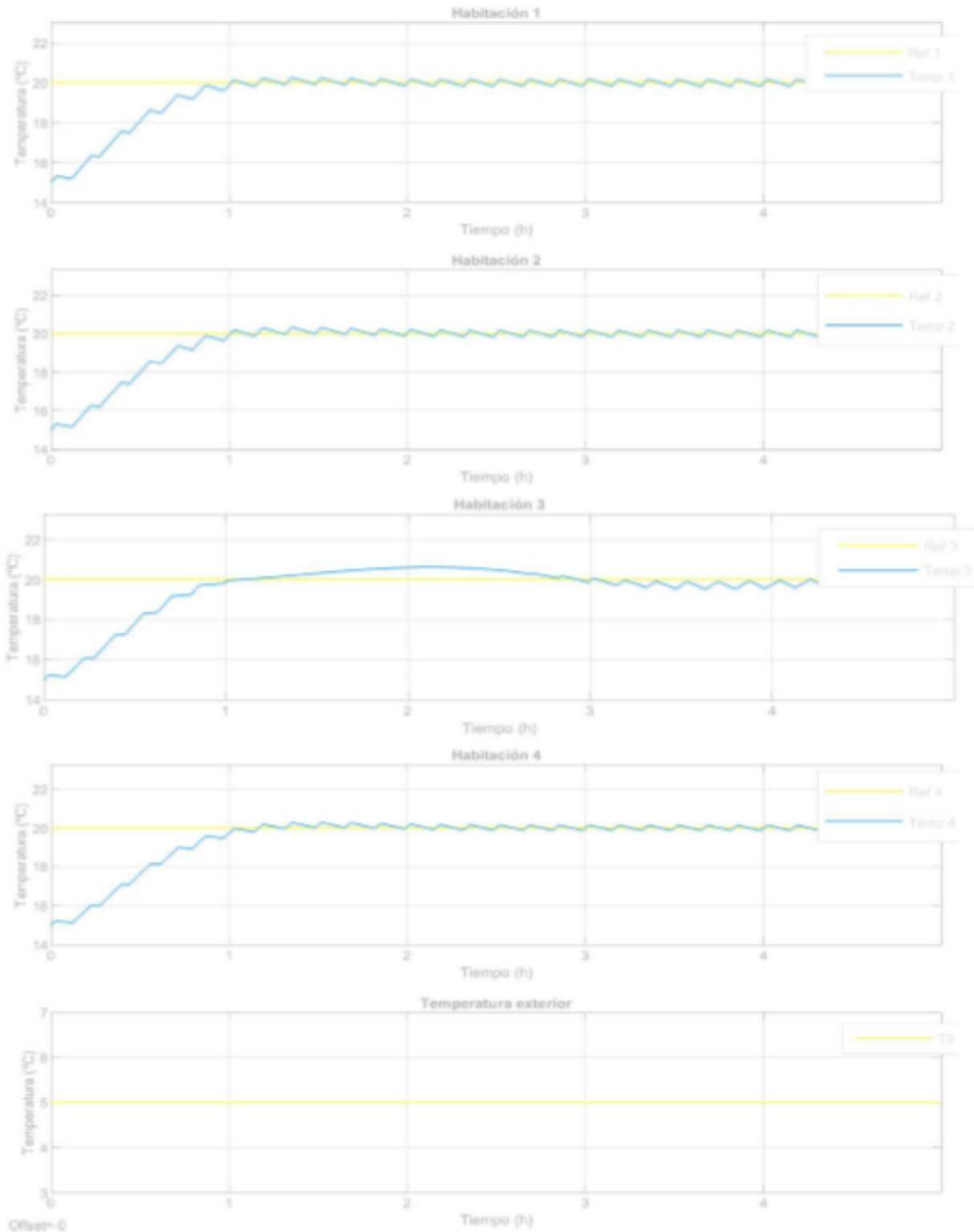


Figura 49 Temperaturas simulación 4



Potencias de entrada de los radiadores

Nada más comenzar la simulación los radiadores empiezan a dar potencia para ascender la temperatura de las habitaciones a los 20°C. Todas alcanzan su valor máximo en torno a los 25 min, dichos valores son: 1825W, 902W, 940W y 603W para P₁, P₂, P₃ y P₄ respectivamente.

Por último, las potencias se establecen en: 439W, 284W y 222W para P₁, P₂, y P₄. Al entrar las cuatro personas en la habitación 3 y aportar 600W al ambiente La P₃ comienza a disminuir hasta hacerse 0 para evitar que la temperatura se dispare.

Temperaturas

Las 4 temperaturas alcanzan la referencia en torno a los 50 min.

La temperatura de la habitación 3 comienza a subir ligeramente a la hora de comenzar la simulación, en el momento en que la P_{d3} asciende a 600W, alcanzando un valor máximo de 20.7°C. El control actúa sobre esta perturbación y la temperatura vuelve a alcanzar la referencia.



Capítulo VI: CONCLUSIONES

En este capítulo se exponen las conclusiones de cada una de principales tareas realizadas en el proyecto

VI.1 COMUNICACIONES

Se han dividido las comunicaciones en dos partes, por un lado, la comunicación entre el controlador Eedomus con el sensor y con el actuador y por el otro la comunicación entre el simulink y el controlador.

VI.1.1 COMUNICACIÓN DEL CONTROLADOR CON EL SENSOR Y EL ACTUADOR

Ha sido una tarea bastante sencilla, se comunican a través de una red z-wave que crea el controlador, de una forma sencilla se incluye cada periférico en la red a través de la plataforma de Eedomus y se comunican a través de líneas de comandos http.

Resulta una comunicación fácil rápida y sencilla, no se ha encontrado ninguna complicación.

El único inconveniente que se podría destacar es la limitación a la hora de configurar el dispositivo. Varias veces al intentar configurar el sensor y la válvula (tiempo entre cada medida del sensor, forma de apertura de la válvula etc...) no daba resultado, o tardaba mucho en recibirlo.

VI.1.2 COMUNICACIÓN DEL CONTROLADOR CON SIMULINK

Como hemos mencionado en capítulos anteriores, con esta comunicación tuvimos más problemas.

El simulink utiliza a través de comandos TCP/IP mientras que nuestro controlador a través de comandos http por lo que ha habido que programar e implantar en la raspberry un servidor con NodeJS que reciba los comandos que envía simulink mediante TCP/IP y los transforme en una petición HTTP.

El único inconveniente es que si por algún casual el nodo fallara interrumpiéndose la comunicación habría que reiniciar la raspberry para que vuelva a arrancar.



VI.2 DISEÑO DEL CONTROL DE TEMPERATURA

Primero obtenemos el modelo matemático de la planta una vivienda de cuatro habitaciones a través de las ecuaciones de estado de un modelo térmico de temperatura.

A continuación, se diseñará el control predictivo a través de la herramienta Tool Box de Matlab estableceremos el tiempo de muestreo, los horizontes de control y predicción, los pesos, las referencias y las restricciones y comprobaremos en la propia herramienta los resultados.

En cuanto a los pesos, hemos comprobado que bajando el peso de los mandos y subiendo el de las salidas para las mismas condiciones del control obtenemos un control más rápido, pero con más sobrepaso y oscilaciones, en cambio al aumentar el peso de los mandos y disminuir el de las salidas para las mismas condiciones de control hemos comprobado que obtenemos un control más lento, pero mucho más robusto sin sobrepasos y oscilaciones.

VI.3 SIMULACIÓN Y PRUEBA DEL CONTROL

Una vez diseñado el modelo de la planta y el control, se realizan diferentes simulaciones, como el seguimiento en referencia en una habitación, el rechazo de un cambio brusco de temperatura exterior brusco, una temperatura exterior senoidal simulando un día de invierno y por último se ha considerado la presencia de personas que aportan calor en una de las habitaciones.

Analizando los resultados obtenidos el control predictivo diseñado cumple con las expectativas, presentando robustez y tiempos de alcance y establecimiento adecuados, teniendo en cuenta las constantes de tiempo del sistema.



Capítulo VII: FUTUROS DESARROLLOS

De cara a futuros desarrollos y proyectos, lo primero que habría que hacer sería implantar el control en la raspberry y realizar los ensayos pertinentes en la planta para obtener los parámetros exactos de la misma. Para que, de esta manera, poder rediseñar y precisar el control. Y finalmente implantarlo y dejarlo en funcionamiento probando así su efectividad.

Otro posible avance sería mejorar la comunicación entre la raspberry y el simulink retocando el archivo implantado en la raspberry creado en NodeJs.

También se ha pensado en realizar un control adaptativo de manera que se vaya regulando a lo largo del tiempo en función de los datos anteriores y se haga más fiable cada vez.

Otro gran desarrollo sería ampliar este proyecto a otros controladores con diferentes protocolos de comunicación, como por ejemplo INSTEON o Zigbee.

Por último, una vez perfeccionado todos los puntos exteriores podría investigarse en probar este tipo de control para otros sistemas de calefacciones como las calefacciones de suelo radiante.



Capítulo VIII: REFERENCIAS Y BIBLIOGRAFÍA

- [1] <http://www.cedom.es/sobre-domotica/que-es-domotica> [último acceso: 5 de enero del 2019]
- [2] <https://controlautomaticoeducacion.com/control-predictivo/control-gpc/> [último acceso: 7 de enero del 2019]
- [3] <https://es.omega.com/prodinfo/controladores-de-temperatura.html> [último acceso: 9 de enero del 2019]
- [4] <https://simulaciondeprocesosudenar.wordpress.com/control-proporcional-integralderivativo-pid-y-estabilidad-de-sistemas-dinamicos/> [último acceso: 14 de enero del 2019]
- [5] <http://bibing.us.es/proyectos/abreproy/70242/fichero/Contenido+del+Trabajo+Fin+de+Master%252F2.+Control+Predictivo+Basado+en+Modelo+%28MPC%29+aplicado+a+sist.pdf> [último acceso: 17 de enero del 2019]
- [6] <https://es.slideshare.net/sergioluissanchezbarrios/modelo-de-control-predictivo> [último acceso: 21 de enero del 2019]
- [7] Petru-Daniel Moroşan, Romain Bourdais, Didier Dumurb, Jean Buissona, Building temperature regulation using a distributed model predictive control, September 2010. (HAL, Archives-Ourverters.fr)
- [8] Yudong Ma, Garrett Anderson, Francesco Borrelli, A distributed predictive control approach to building temperature regulation, San Francisco, USA: Proceedings of the 2011 American Control Conference, 18 August 2011
- [9] Ion Hazyuk, Christian Ghiaus, David Penhouet, Optimal temperature control of intermittently heated buildings using Model Predictive Control, Energy and Buildings, November 2011. (12th Conference of International Building Performance Simulation Association, Sydney, 14-16 November).
- [10] https://es.wikipedia.org/wiki/Protocolo_de_comunicaciones [último acceso: 28 de enero del 2019]
- [11] <https://www.monografias.com/trabajos104/protocolos-domotica/protocolos-domotica.shtml> [último acceso: 4 de febrero del 2019]
- [12] <https://sites.google.com/site/ingsoftwares5a/ventajas-y-desventajas-de-zigbee> [último acceso: 7 de febrero del 2019]
- [13] <https://alhenaing.wordpress.com/protocolo-insteon/> [último acceso: 11 de febrero del 2019]
-



- [14] <https://www.domodesk.com/162-a-fondo-z-wave-sin-cables.html> [último acceso: 16 de febrero del 2019]
- [15] <https://ricveal.com/blog/z-wave/> [último acceso: 20 de febrero del 2019]
- [16] <https://domoconnect.com/central-domotica/> [último acceso: 22 de febrero del 2019]
- [17] <https://manuals.fibaro.com/es/motion-sensor/> [último acceso: 25 de febrero del 2019]
- [18] <https://www.domoticalia.es/es/termostatos/969-eurotronic-spirit-cabezal-termostatico-zwave-radiadores-4260012711301.html> [último acceso: 28 de febrero del 2019]
- [19] <https://www.zonamovilidad.es/noticia/8398/novedades/raspberry-pi-b-el-ordenador-deltamano-de-una-tarjeta-de-credito.html> [último acceso: 5 de marzo del 2019]
- [20] http://oa.upm.es/48426/1/PFC_CARLOS_SERRANO_CANO.pdf
- [21] <https://www.domoticalia.es/es/multisensores/554-detector-de-movimiento-multisensor-4-en-1-de-fibaro-z-wave-plus-5902020528579.html>
- [22] <https://zwave.es/index.php?route= Spirit-Eurotronic>
- [23] <https://books.google.es/books?hl=es&lr=&id=V6IzqqDcfF8C&oi=fnd&pg=PR1&dq=ACTUADORES+DOMOTICA&ots=trOktsh7bG&sig=EMcbf6kwQpJE2TLIDI58McvvFmo#v=onepage&q=ACTUADORES&f=false> PONER LO DE QUE ES DE GOOGLE SCHOLAR : J.M.HUIDOBRO MOYA – R.J.MILLÁN TEJEDOR, Manual de Domótica, Febrero 2010, CREACIONES COPYRIGHT.
- [24] <https://hardzone.es/2018/07/02/review-raspberry-pi-3-modelo-b/>
<https://es.mathworks.com/help/mpc/gs/design-controller-using-the-command-line.html>
<https://es.mathworks.com/help/mpc/gs/designing-a-model-predictive-controller-for-a-simulink-plant.html>
- [25] <http://oaji.net/articles/2017/5082-1501183303.pdf> Sendoya Diego Fernando, ¿Qué es el Control Predictivo y Hacia Dónde se Proyecta?, 12 de Octubre de 2012, Escuela de Ciencias Básicas, Tecnología e Ingeniería, Universidad Nacional Abierta y a Distancia, Neiva, Colombia, Revista Especializada en Ingeniería [26] Javier García Aguilar, Implementación de un control predictivo de temperatura en una vivienda, 20 de julio de 2016, Universidad Pontificia de Comillas, Icai.



COMILLAS
UNIVERSIDAD PONTIFICIA

ICAI

PROYECTO DE FIN DE GRADO

PRESUPUESTO

DISEÑO E INTEGRACIÓN DE UN CONTROL
PREDICTIVO DE TEMPERATURA EN UN
SISTEMA DOMÓTICO

Autor:

Puente Iglesias, Sara.

Director:

Zamora Macho, Juan Luis

Capítulo I: RECURSOS EMPLEADOS

Para comenzar se resumen las cantidades de los recursos empleados, tanto materiales, de software y humanos.

I.1 COMPONENTES

<i>COMPONENTE</i>	<i>CANTIDAD</i>
CONTROLADOR EEDOMUS	1
SENSOR FIBARO	1
ELECTROVÁLVULA EUROTRONICS	1
RASPBERRY PI 3B+	1
CABLE RJ45	1
PILA SENSOR 881C-2	1

I.2 SOFTWARE

<i>ELEMENTO</i>	<i>CANTIDAD</i>
LICENCIA MATLAB USO ACADÉMICO	1
LICENCIA SIMULINK USO ACADÉMICO	1
MICROSOFT OFFICE	1

I.3 MANO DE OBRA

<i>ELEMENTO</i>	<i>CANTIDAD (H)</i>
ESTUDIO COMUNICACIONES HTTP	4
DESARROLLO COMUNICACIÓN ENTRE EL CONTROLADOR CON EL SENSOR Y EL ACTUADOR	80
PRUEBA COMUNICACIÓN Y AJUSTES DE PARÁMETROS DEL SENSOR Y ACTUADOR	3
ESTUDIO COMUNICACIONES TCP/IP	10
DESARROLLO COMUNICACIONES ENTRE EL MICROCONTROLADOR Y EL CONTROLADOR	70
PRUEBA COMUNICACIONES	2
ESTUDIO CONTROLES MPC	5
DISEÑO DEL CONTROL	15
DISEÑO DEL FICHERO DEL SIMULINK	100
SIMULACIONES	10
REDACCIÓN DOCUMENTOS	100
PREPARACIÓN PRESENTACIÓN	10
TOTAL	409

Capítulo II: PRECIOS UNITARIOS

II.1 COMPONENTES

<i>COMPONENTE</i>	<i>PRECIO/UD (€)</i>
CONTROLADOR EEDOMUS	295
SENSOR FIBARO	50,29
ELECTROVÁLVULA EUROTRONICS	48,41
RASPBERRY PI 3B+	42,18
CABLE RJ45	1,7
PILA SENSOR 881C-2	3

II.2 SOFTWARE

<i>ELEMENTO</i>	<i>PRECIO/UD (€)</i>
LICENCIA MATLAB USO ACADÉMICO	400
LICENCIA SIMULINK USO ACADÉMICO	400
MICROSOFT OFFICE	350

II.3 MANO DE OBRA

<i>ELEMENTO</i>	<i>PRECIO/UD (€/H)</i>
ESTUDIO COMUNICACIONES HTTP	5
DESARROLLO COMUNICACIÓN ENTRE EL CONTROLADOR CON EL SENSOR Y EL ACTUADOR	15
PRUEBA COMUNICACIÓN Y AJUSTES DE PARÁMETROS DEL SENSOR Y ACTUADOR	18
ESTUDIO COMUNICACIONES TCP/IP	5
DESARROLLO COMUNICACIONES ENTRE EL MICROCONTROLADOR Y EL CONTROLADOR	15
PRUEBA COMUNICACIONES	18
ESTUDIO CONTROLES MPC	5
DISEÑO DEL CONTROL	15
DISEÑO DEL FICHERO DEL SIMULINK	15
SIMULACIONES	17
REDACCIÓN DOCUMENTOS	25
PREPARACIÓN PRESENTACIÓN	10

Capítulo III: Sumas parciales

III.1 COMPONENTES

<i>COMPONENTES</i>	<i>PRECIO/UD (€)</i>	<i>UNIDADES</i>	<i>TOTAL (€)</i>
CONTROLADOR EEDOMUS	295	1	295
SENSOR FIBARO	50,29	1	50,29
ELECTROVÁLVULA EUROTRONICS	48,41	1	48,41
RASPBERRY PI 3B+	42,18	1	42,18
CABLE RJ45	1,7	1	1,7
PILA SENSOR 881C-2	3	1	3
TOTAL	---	---	440,58

III.2 SOFTWARE

<i>ELEMENTO</i>	<i>PRECIO/UD (€)</i>	<i>UNIDADES</i>	<i>TOTAL (€)</i>
LICENCIA MATLAB USO ACADÉMICO	400	1	400
LICENCIA SIMULINK USO ACADÉMICO	400	1	400
MICROSOFT OFFICE	350	1	350
TOTAL	---	---	1150

III.3 MANO DE OBRA

<i>ELEMENTO</i>	<i>PRECIO/UD (€/H)</i>	<i>TIEMPO (H)</i>	<i>TOTAL (€)</i>
ESTUDIO COMUNICACIONES HTTP	5	4	20
DESARROLLO COMUNICACIÓN ENTRE EL CONTROLADOR CON EL SENSOR Y EL ACTUADOR	15	80	1200
PRUEBA COMUNICACIÓN Y AJUSTES DE PARÁMETROS DEL SENSOR Y ACTUADOR	18	3	54
ESTUDIO COMUNICACIONES TCP/IP	5	10	50
DESARROLLO COMUNICACIONES ENTRE EL MICROCONTROLADOR Y EL CONTROLADOR	15	70	1050
PRUEBA COMUNICACIONES	18	2	36
ESTUDIO CONTROLES MPC	5	5	25
DISEÑO DEL CONTROL	15	15	225
DISEÑO DEL FICHERO DEL SIMULINK	15	100	1500

SIMULACIONES	17	10	170
REDACCIÓN DOCUMENTOS	25	100	2500
PREPARACIÓN PRESENTACIÓN	10	10	100
TOTAL	---	409	6930

Capítulo IV: Sumas totales

<i>APARTADO</i>	<i>COSTE</i>
HARDWARE	440,58
SOFTWARE	1150
MANO DE OBRA	6930
PEM	8520,58
GASTOS GENERALES (10%)	852,06
IVA (21%)	1789,32
PEC	11161,96

