

# Trabajo Fin de Grado

## Grado en Ingeniería de Tecnologías Industriales

### Modelado y Control de Sistemas de Climatización de edificios en TRNSYS

Autor: Alberto Oliveros Iglesias

Tutor: Daniel Limón Marruedo

Joaquín García Ordoñez

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

Sevilla, 2020





Trabajo Fin de Grado  
Grado en Ingeniería de Tecnologías Industriales

# **Modelado y Control de Sistemas de Climatización de edificios en TRNSYS**

Autor:

Alberto Oliveros Iglesias

Tutor:

Daniel Limón Marruedo

Catedrático de Universidad

Joaquín García Ordoñez

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

Sevilla, 2020



Trabajo Fin de Grado: Modelado y Control de Sistemas de Climatización de edificios en TRNSYS

Autor: Alberto Oliveros Iglesias  
Tutor: Daniel Limón Marruedo  
Joaquín García Ordoñez

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

Presidente:

Vocal/es:

Secretario:

acuerdan otorgarle la calificación de:

El Secretario del Tribunal

Fecha:



*A mi familia*  
*A mis amigos*  
*A mis profesores*









































## 2 Definición del sistema

---

*La ciencia puede divertirnos y fascinarnos, pero es la Ingeniería la que cambia el mundo.*

ISAAC ASIMOV

En este primer capítulo, se persigue la definición del sistema sobre el que se basa toda la realización de dicho trabajo. Es por ello que se trata de una parte fundamental, ya que de esta dependerá la calidad de los resultados obtenidos. En las próximas páginas, se podrá visualizar y caracterizar el sistema, y posteriormente, se ilustrará como se deben implementar correctamente todas las características necesarias para su construcción.

Este capítulo se apoya fundamentalmente en la herramienta TRNSYS, un software comercial desarrollado por la universidad de Wisconsin-Madison (EE.UU) orientado, entre otras cosas, a la simulación energética de edificios, considerando aspectos realistas como pérdidas por flujo de aire, ganancias internas del sistema (personas, luces...), así como pérdidas propias del edificio [18].

Una vez se ha definido el entorno de trabajo, ya se está en disposición de comenzar el diseño del edificio sobre el cual se apoyará el trabajo restante.

### 2.1 Diseño del edificio

El diseño del edificio que se va a realizar está basado en la construcción de los laboratorios ubicados en la Escuela Técnica Superior de Ingeniería de la Universidad de Sevilla, más concretamente, se puede asimilar a los que se encuentran en el Laboratorio 1, planta 2.

Para el diseño, si lo asemejamos a los laboratorios, el esquema estará compuesto por un total de seis despachos, tres de ellos con ubicación norte y los otros tres con ubicación sur, además de un pasillo común para lograr la comunicación entre despachos. Para poder ilustrar de una forma más clara la configuración, se ha realizado el esquema visible en la Figura 2.1, el cual ha sido creado con la herramienta online de Google SketchUp [16].

Como se puede observar en la configuración propuesta, todas las estancias están equipadas de una ventana para poder aportar al edificio un intercambio de calor provocado por la radiación solar, y que podrá ser considerado como una fuente de perturbaciones en el modelo.

Con respecto a las dimensiones de los despachos se han diseñado todos de la misma dimensión con un área de unos  $16m^2$ , siendo cuadrados de  $4m$  por cada dimensión, la altura de los techos, se sitúa a unos  $3.5m$  sobre el nivel del suelo. Todos ellos, como ya se ha enunciado anteriormente, disponen de una ventana localizada en las paredes exteriores (norte y sur) con una dimensión de  $1.75m$  de ancho por  $1.20m$  de alto.

En relación al pasillo, como es lógico, tendrá una dimensión a lo largo de  $12m^2$ , mientras que su ancho es algo más pequeño que el de los despachos, siendo este de  $3m$ . Al igual que los despachos, el techo está situado a una altura de  $3.5m$ . Además, también dispone de una ventana situada en la cara oeste del edificio, con las mismas dimensiones que las instaladas en los despachos.

Para finalizar con el diseño preliminar del edificio, se ha de saber que para facilitar comunicación entre los despachos y el pasillo se disponen de unas puertas de dimensión de  $2.07m$  de alto por  $0.91m$  de ancho. Al

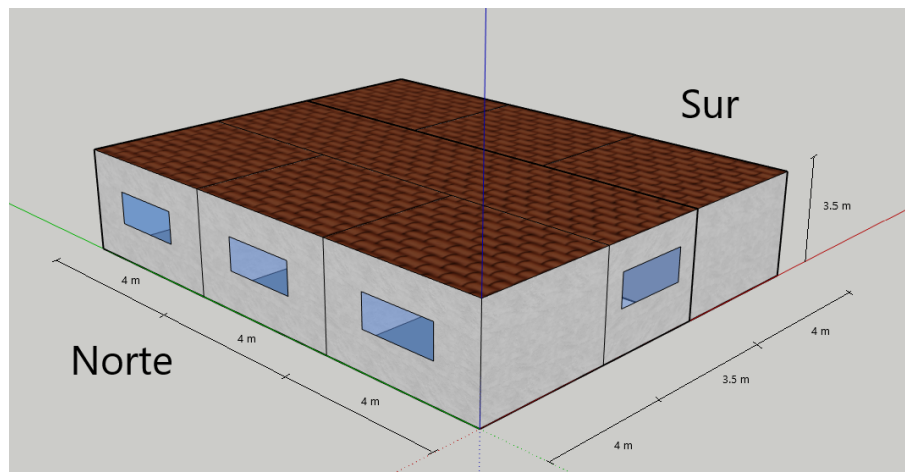


Figura 2.1 Configuración del edificio.

igual que ocurre con las ventanas, estas puertas serán una fuente de perturbación, ya que cuando dos espacios se encuentran a diferente temperatura, se produce entonces un intercambio de calor, donde el nodo caliente cede calor al nodo con una menor temperatura.

De aquí en adelante, para facilitar el manejo del llamamiento a cada una de las salas, se referirán a ellas de la siguiente manera como se puede observar en la Figura 2.2, donde se tiene que:

- La sala 1 es la que está situada en la cara noroeste.
- La sala 2 se define como la sala central que puede verse desde la orientación norte del edificio.
- La sala 3 posee orientación noreste.
- La sala 4 está en la cara sur del edificio con orientación suroeste.
- La sala 5 es la conjugada a la sala 3, por lo que se sitúa a la derecha de la sala 4.
- La sala 6 se caracteriza por tener orientación sureste.
- La sala 7 o también referida como pasillo, está situada entre las dos filas de despachos.

Una vez se han definido de forma real las distintas salas, ya se está en disposición de poder comenzar la implementación correspondiente en el software de simulación elegido.

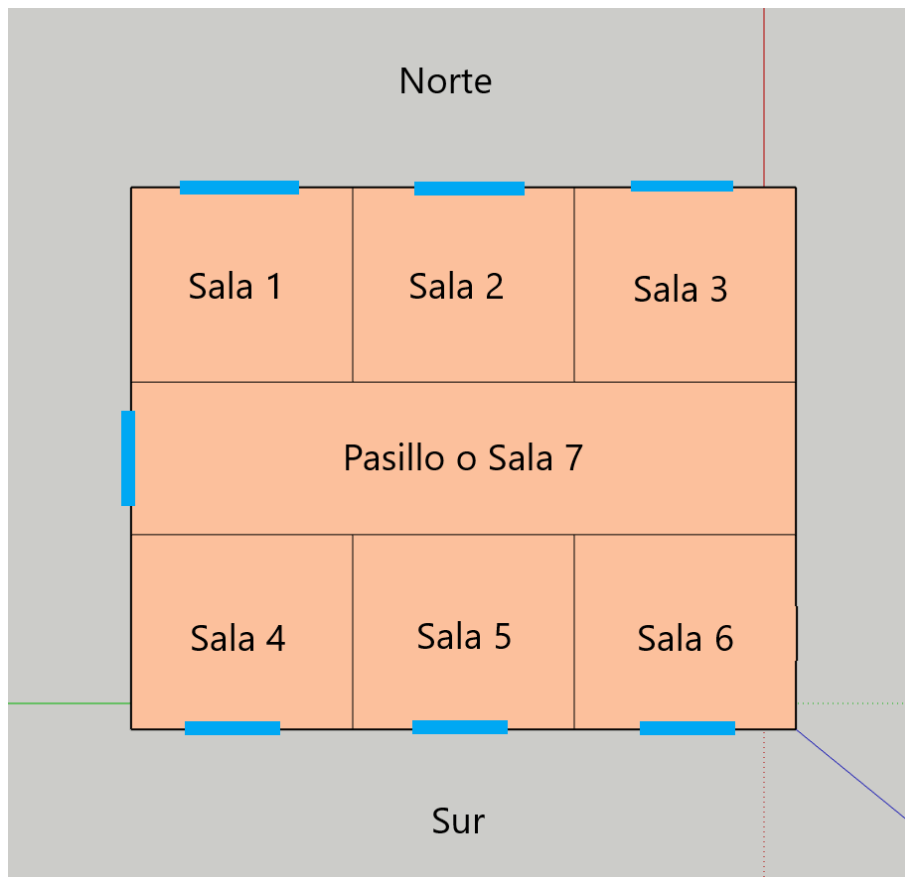
## 2.2 Creación de espacios en TRNBuild

En este apartado, se va a estudiar cómo se ha modelado el sistema en TRNSYS, el cual ha sido elegido entre otras cosas, por permitir una gran capacidad de extensión con programas como Matlab. Para el diseño del edificio se hace uso de la herramienta disponible en TRNSYS [8], llamada TRNBuild [20].

Una vez se está en disposición de iniciar el modelado del edificio, lo primero que se debe crear es un nuevo proyecto en la herramienta TRNBuild. Para ello, una vez se ha abierto dicho software, seleccionamos la pestaña *File/New*. A continuación, hay que tomar la primera decisión acerca de la localización geográfica del edificio. Para ello, se tiene que elegir si situarlo en el hemisferio norte o en el hemisferio sur. En este caso concreto, se elige el hemisferio norte, para adaptar el edificio a una ubicación lo más próxima posible al emplazamiento real.

Una vez se ha creado el proyecto correctamente, se abren dos ventanas, de las cuales la referida al *Project*, es mera información relacionada con el proyecto (Título, descripción, fecha, etc). Por otro lado, también encontramos la ventana llamada *TRNBuild Navigator*, donde, a modo de esquema se pueden visualizar dos ramas principales:

- En la rama denominada *Project*, la información relevante que se encuentra es la relativa a las correspondientes entradas y salidas del bloque del edificio que posteriormente será insertado en TRNSYS. Es ahí donde se debe corroborar que todas han sido definidas correctamente. Si se observa la ventana de *Properties*, en ella, se permite la modificación de los distintos parámetros del aire como son su densidad y calor específico, entre otras.



**Figura 2.2** Asignación de nombre a cada sala.

- El otro epígrafe, que figura como *Zones*, será el más importante en lo que incumbe en este apartado, el diseño de las zonas térmicas, por ello, aquí deben aparecer todas las salas que se han definido en el edificio.

Cuando se ha identificado y ya se tiene claro las funciones de cada ventana, se puede proceder a la incorporación de todas las zonas o espacios de la instalación. Para ello, se debe hacer botón derecho en el apartado *Zones* y seleccionar *Add Zone*. Es entonces cuando se abre una nueva pestaña donde hay que introducir el nombre de la zona y el nombre del nodo de aire (*Airnode Name*). Los nodos de aire no son más que dos subzonas dentro de una zona que poseen diferentes temperaturas, es decir, dentro una única zona pueden existir dos '*Airnodes Zones*'. En nuestro caso, todas las habitaciones van a ser definidas como zonas, ya que todas van a estar aisladas con sus respectivas paredes. Una vez se han definido todas las zonas, el esquema resultante estará compuesto de todas las zonas del edificio.

Finalizada la creación de dichas zonas, si se accede a una de ellas, aparece una ventana como la mostrada en la Figura 2.3, donde a modo resumen, se puede acceder a realizar las siguientes modificaciones:

- En la parte izquierda de la ventana, dentro del apartado *Airnodes*, se permite el cambio de sala a una de las que ya ha sido creada, e incluso la creación de nuevas.
- El segundo gran bloque es el relativo a *Airnodes Regimen Data*, el cual es de gran importancia, ya que es donde se van a realizar cambios para la caracterización del edificio.
  - En primer lugar, se observan los campos de el volumen total y la capacidad calorífica, los cuales son calculados automáticamente una vez se hayan definido las dimensiones de la sala.
  - El campo de Infiltración (*Infiltration*) hace referencia a las pérdidas de aire que se producen a través de las paredes cuando se realiza la simulación.
  - La casilla de ventilación (*Ventilation*), permite habilitar un intercambio forzado del aire interior con el exterior para producir así pérdidas energéticas en el edificio.

- Los iconos de ventilación (*Cooling*) y calefacción (*Heating*) introducen en el sistema un acondicionamiento de la sala ya sea enfriándola o calentándola. Esto no va a ser objeto de estudio en este trabajo, pues el objetivo es la implementación de controladores para lograr la climatización de las distintas salas.
  - El siguiente, llamado ganancias (*Gains*), sí que va a ser de gran utilidad, pues en él se debe introducir las distintas ganancias internas del sistema, las cuales serán utilizadas como fuentes de perturbación en el algoritmo de control.
  - El relativo al confort (*Confort*), permite modificar aspectos más avanzados como son la tasa metabólica, la velocidad relativa del aire o el factor de ropa.
  - Para finalizar, encontramos la casilla donde se establecen las condiciones iniciales (*Initial Values*) de la sala en cuestión, estableciendo así una temperatura y una humedad inicial.
- En el tercer bloque, se debe especificar todas las características de cada uno de los muros de cada sala. Esto se estudiará de forma más detallada en las próximas páginas.
  - El módulo de ventanas aparece como un bloque independiente, pero este puede verse como una extensión al de los muros, pues ahí deben definirse todas las características de las ventanas.
  - Por último, en el lado derecho se visualiza el llamado (*Therm. Zones*), pero de este sólo se hará uso cuando el modelo del edificio se haya realizado con SketchUp.

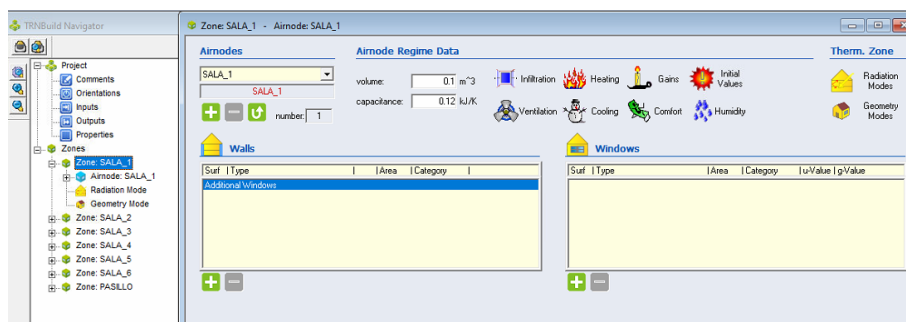


Figura 2.3 Ventana para la modificación de características de una sala..

Una vez se han creado correctamente los espacios o zonas del sistema, se procede a la definición de los muros o paredes que definen y delimitan el sistema de estudio.

### 2.3 Definición de muros en TRNBuild

En este apartado, se va a centrar la explicación acerca de cómo se deben crear los muros que constituyen una sala del modelo del edificio. Antes de comenzar, es preciso saber que cada espacio debe estar compuesto de las paredes verticales que limitan el área de cada sala, así como de los dos muros horizontales que conforman tanto el suelo como el techo. A modo de ejemplo, a continuación, se ilustra paso a paso como se debe crear cada uno de los elementos.

En primer lugar, se debe identificar cada uno de los muros que componen el espacio y definir de qué tipo de muros se trata. Si uno se fija en la sala 1, dicha definición sería la siguiente: La sala, posee un muro exterior con orientación norte y otro muro exterior con orientación oeste. Además, tiene dos muros interiores o adyacentes, donde uno de ellos, el de orientación sur, separa la sala 1 del pasillo, mientras que el de orientación este, lo separa de la sala 2. Por último, no se debe olvidar ni el techo ni suelo, donde el techo o cubierta se considera un muro exterior, pero con diferentes propiedades a las de los verticales, y el suelo se establece como perímetro de todos los muros verticales. Se comienza, a continuación, con la definición uno a uno de todos los muros expuestos.

Para el primero de ellos, el de orientación norte, como vemos se ha definido como muro exterior, entonces la primera modificación que se debe realizar la encontramos en el campo *Wall type* donde se ha elegido el tipo *FA\_VENT\_01*, el cual es el correspondiente a una fachada ventilada compuesta de ladrillo perforado, una capa de 5 cm de aislamiento, una cámara de aire y por último un acabado de placas, siendo su transmitancia térmica total igual a  $0.51 \text{ W/m}^2\text{K}$ . La segunda variable a elegir es el área, que como se explicó en apartados

anteriores es de  $4\text{ m}$  de ancho por  $3.5\text{ m}$  de alto, resultando un total de  $14\text{ m}^2$ . Prosiguiendo, encontramos el rótulo *Category*, en el cual debemos especificar la opción de *EXTERNAL*, ya que se trata de un muro exterior. En el campo *geosurf*, se ha de especificar el porcentaje de radiación de onda corta que atraviesa dicho muro, en este caso concreto se toma un valor de 0.1. Los dos campos siguientes no se verán modificados, ya que su objetivo es la introducción de ganancias en una o ambas caras de la pared. Para finalizar, se debe establecer la orientación del muro que, para este, como ya se ha dicho es *N\_180\_90*, además del *view factor to sky*, que, como en este caso es perpendicular al suelo y este se supone horizontal, la fracción del muro que ve al cielo es de 0.5. Todas estas configuraciones pueden ser observadas en la Figura 2.4.

Surf	Type	Area	Category	Orientation
1	FA_VENT_01	14.00	EXTERNAL	N_180_90

Additional Windows

Surface-ID: 1

wall type: FA\_VENT\_01

area: 14 m<sup>2</sup> incl. windows

category: EXTERNAL

geosurf: 0.1

surf. gain inside: 0 kJ/h

surf. gain outside: 0 kJ/h

orientation: N\_180\_90

view fac. to sky: 0.5

Figura 2.4 Características del muro exterior norte de la Sala 1.

El segundo muro exterior, presenta las mismas propiedades que el anterior, salvo por una excepción ya que dicho muro presenta una orientación oeste, por lo que se debe seleccionar la opción *W\_90\_90* para configurarlo correctamente.

Se continúa con el muro adyacente de la sala 2, es decir el de orientación este. Para este muro el *Wall type* seleccionado ha sido el *MD\_MACI\_01* que no es más que una mediana de ladrillo perforado con una transmitancia de  $2.38\text{ W/m}^2\text{ K}$ . La categoría del muro, como se enunciado con anterioridad, es *ADJACENT* y para indicar el espacio con el que es adyacente, se debe modificar la casilla *adjac. to airnode* donde se indicará el número de sala correspondiente. Con esto, ya se tiene definido el muro de separación entre las salas 1 y 2.

Con respecto al muro adyacente con el pasillo, se define de igual forma que acabamos de exponer, sabiendo que hay que indicar que, en este caso el espacio adyacente es el pasillo. Además, como se precisó en el diseño del edificio, este muro posee una puerta que comunica ambos espacios. Una vez se llega a este punto hay que tomar la decisión de modelar las perturbaciones en las temperaturas de cada habitación cuando se abre y cierra una puerta. Habiendo analizado distintas posibles soluciones, se llega a concluir que la mejor opción es el modelado de dicha puerta como si de una ventana se tratase. Para ello, una vez se ha creado la pared adyacente nos centramos en la sección *Windows* que aparece a mano derecha. En dicha sección se especifica el *window type* como el *SCO2* ya que su transductancia térmica es muy similar a la de la madera, además del área de la puerta, siendo esta de  $1.88\text{ m}^2$ , como la puerta se encuentra entre dos espacios, se debe seleccionar la opción adyacente como su categoría. Por último, siendo este paso el más importante, se debe modelar una infiltración (*coupling air flow*), identificada como el caudal de aire en  $\text{Kg/h}$  que pasa de un espacio a otro, provocando así una alteración en las temperaturas de ambos espacios.

Ahora se pasa a introducir el techo o cubierta de la sala. Para las cubiertas, el tipo de muro empleado es el *CB\_AISS\_01* que representa una cubierta plana invertida no transitada con bovedilla cerámica unidireccional y  $5\text{ cm}$  de aislamiento, con una transmitancia de  $0.5\text{ W/m}^2\text{ K}$ . El área de la parte superior de la sala 1 se debe configurar como  $16\text{ m}^2$ . Al igual que los muros exteriores, la configuración de la categoría es la misma, sin embargo, la orientación de dicho muro es horizontal, concretamente la *H\_0\_0* con una fracción de muro

que ve al cielo igual a 1, ya que dicho muro al ser horizontal y estar ubicado en la parte superior, es el más expuesto a la radiación solar.

Antes de acabar con la sala 1 es necesaria la implementación del suelo, que también es considerado como otro muro. Para ello, el *Wall type* empleado ha sido el *SO\_AISS\_01* el cual simula una solera maciza de 4 cm de aislamiento sobre forjado con una transmitancia de  $0.61 \text{ W/m}^2\text{K}$ . La definición de este muro es mucho más sencilla, ya que al seleccionar *BOUNDARY* como la categoría del muro, la única modificación a realizar es el cambio de el porcentaje de radiación de onda corta que atraviesa el suelo, que en este caso tomará un valor de 0.6.

Por último, solo es necesaria la incorporación de la ventana existente en dicho muro y que comunica la sala con el ambiente exterior. Esta ventana se encuentra en el muro exterior norte, donde para ello hay que dirigirse a la modificación del muro y centrarse en este caso en la ventana *Windows*, donde se elegirá el tipo de ventana llamado *CLA2* que representa una ventana cuyas propiedades son las de un cristal doble con un 15% de marco. El área de la ventana, especificada en el primer apartado, es de unos  $2.1 \text{ m}^2$ . La ventana, al igual que el muro donde se encuentra, es externa con orientación *N\_180\_90*, con el la fracción de muro que ve al cielo igual a 0.5.

Una vez se han realizado todos estos pasos, se concluye con la definición de todos los muros, ventanas y puertas que componen la sala 1 y como se puede corroborar el volumen de la sala que ha sido calculado de forma automática, debe coincidir con el valor real, lo cual puede ser un punto de comprobación para garantizar que se está realizando correctamente.

Para la incorporación de todas las salas restantes, se debe proceder de igual forma a como se ha realizado con esta última, prestando especial atención a la localización geográfica y tipo de muro de cada uno de ellos. Todos los muros que han sido definidos y su etiqueta correspondiente se pueden observar en la Figura 2.5, donde los números de color negro indican las paredes verticales, los representados en color azul las ventanas, los rojos los correspondientes a las cubiertas y, por último, los de color marrón son los correspondientes a la solera.

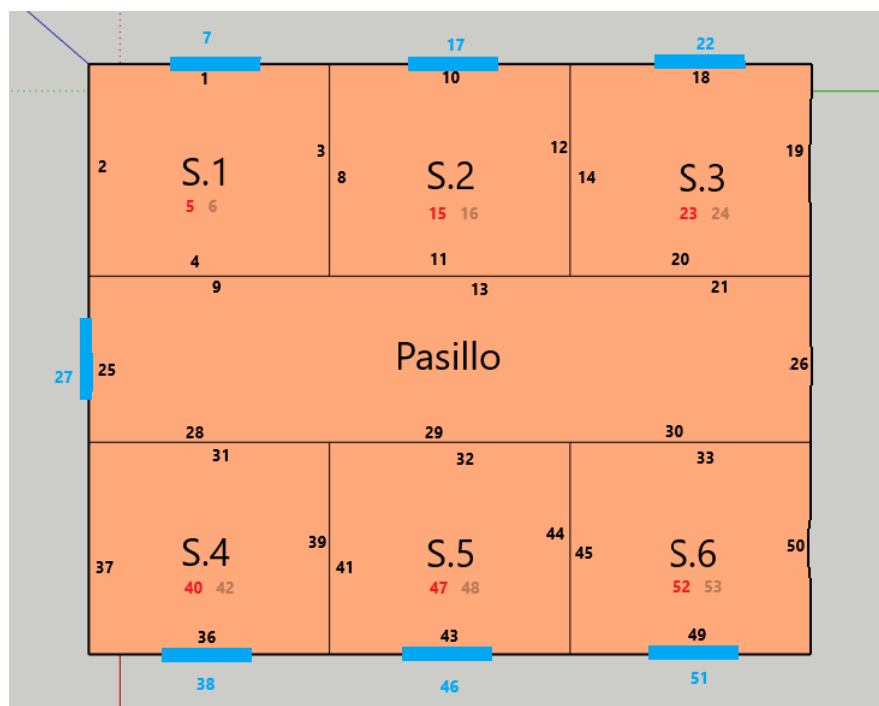


Figura 2.5 Numeración resultante tras la definición de los muros.

## 2.4 Declaración de las infiltraciones

En esta sección, lo que se realiza básicamente es la introducción de las infiltraciones que se producen por las paredes verticales y que modelan las pérdidas energéticas propias del edificio. Para ello, se ha llevado a cabo un proceso de documentación [12] para establecer un valor coherente de dicho parámetro siendo el elegido, el de un aislamiento medio igual a  $4.13 \text{ l/h}$ .

Para su definición en el software TRNBuild, este parámetro será definido como una variable global y facilitar así su modificación en todas las salas y su posible posterior redefinición si se desea. Para ello, hay que dirigirse a la barra superior de iconos y elegir el correspondiente a las infiltraciones (siendo este el tercero comenzando por la izquierda), una vez seleccionado, en la pestaña emergente que aparece, debe hacerse click sobre el botón *New Type* localizado en la esquina inferior derecha. Es entonces cuando se elige un nombre que caracterice a esa infiltración (*INFIL001*) y su valor correspondiente. De esta forma, se habrá creado una variable global que almacene dicho valor. Habiendo realizado esto, la caracterización de las salas será mucho más sencilla puesto que basta con seleccionar el icono correspondiente y elegir la ventilación que ha sido creada.

## 2.5 Simulación en TRNSYS

En este apartado se tratará, como se acaba de decir, la incorporación y posterior simulación en TRNSYS del edificio creado con TRNBuild. Una vez se haya creado un nuevo proyecto en blanco, se deberán introducir <sup>1</sup> los siguientes componentes [21]:

- En primer lugar, para añadir el edificio que se ha diseñado se usa el bloque *Type 56*, el cual modela el comportamiento térmico de un archivo con la extensión *.bui*. Para indicarle al bloque el archivo a usar, se debe hacer doble click sobre él y una vez dentro, hay que dirigirse a la ventana *External Files*, donde habrá que introducir la ruta del archivo correspondiente. Con esto ya es suficiente para la creación del bloque.
- Como es lógico, dicho edificio se verá influido por la acción del tiempo atmosférico, para que puedan producirse interacciones energéticas entre el edificio y el ambiente exterior. Para ello empleamos el bloque *Type 15-6*, que tiene la función de leer los datos meteorológicos de un archivo externo que los contenga y hacerlos compatibles con el resto de bloques de un proyecto. Una vez se ha arrastrado el bloque al proyecto se hace doble click y dirigirse a la pestaña *External Files* donde se elige la ruta deseada del archivo. Hay que destacar que TRNSYS incorpora una base de datos meteorológicos en distintas localizaciones, aunque esta no es muy amplia, la más próxima al emplazamiento real está situada en el aeropuerto Madrid-Barajas Adolfo Suárez, por lo que este será el empleado durante la realización del proyecto. Este archivo puede localizarse en: *Weather-Meteororm-EuropeES-Madrid-Barajas-82210.tm2*.
- Por último, pero no menos importante, se debe incluir el *Type 65d*, que nos permite mostrar la simulación de las variables requeridas en una ventana auxiliar. Hay que resaltar que, con este bloque no se genera ningún archivo de datos externo para guardar dichos datos, por lo que si desean obtener, habrá que hacer uso de otro.

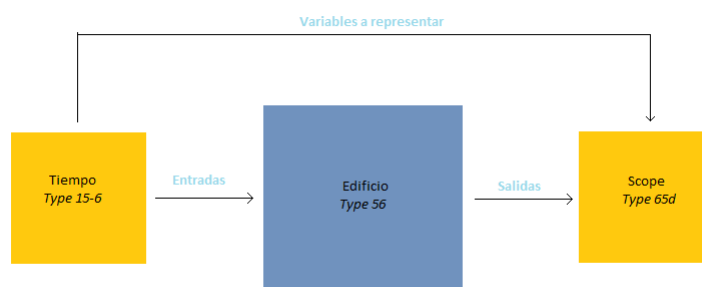



Figura 2.6 Esquema conceptual de conexiones.

Cuando todos los iconos de los respectivos bloques se encuentren sobre la hoja en blanco, se debe proceder a la conexión entre ellos, siendo este un paso más complejo que los realizados hasta ahora, puesto que entran en juego conceptos más avanzados. Para ello, lo primero que se debe seleccionar es el icono que habilita la unión, ese icono lo encontramos en la barra de herramientas situada en la parte izquierda y representado

<sup>1</sup> Para añadir componentes estos se deben arrastrar de la librería localizada en la columna derecha.

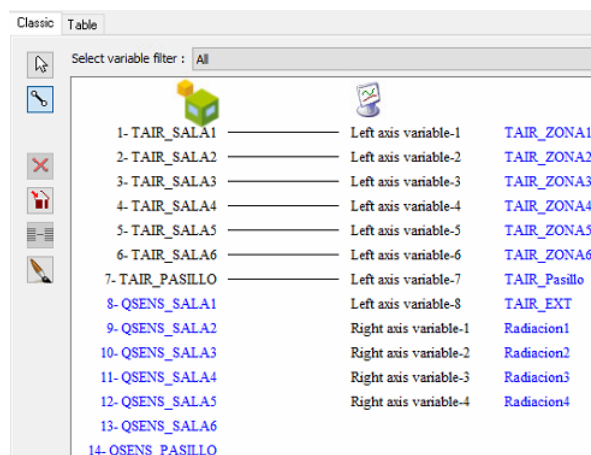
con el símbolo . Entonces si se hace click sobre este y luego uno se dirige a realizar la unión verá que cuando se sitúa encima de uno de los bloques previamente insertados, aparece una nube 8 puntos. A pesar de la existencia de tantos puntos de unión, lo cual se ha realizado para permitir mejores visualizaciones en el archivo, su significado es el mismo. Lo que sí es realmente importante es la dirección en la que se realiza dicho enlace, puesto que primero se debe pulsar sobre el bloque que queramos que nos aporte unas ciertas salidas sobre el segundo bloque que las tomará como entradas.

Comenzamos, en primer lugar, con la conexión más sencilla, siendo esta la que se da entre el bloque que simula al edificio y el scope insertado. Una vez se han realizado la unión (desde el edificio hasta el scope), aparecerá una ventana emergente, donde se muestran por un lado las salidas del bloque del edificio y por el otro, las entradas al scope. Las entradas al scope son variables (número variables en cada eje y límites valores del eje Y) que pueden ser sometidas a modificación en el interior del propio bloque. Las variables de salida del bloque del edificio son las mostradas en la Tabla 2.1, de las cuales solo serán de interés las temperaturas de las habitaciones. Para realizar la conexión basta con unir los nombres de las temperaturas con las que han sido creadas automáticamente en el bloque de la representación.

**Tabla 2.1** Salidas del bloque del edificio (*Type 56*).

Definición	Notación
Temperatura de cada una de las salas	$T_1, T_2 \dots T_7$
Calor intercambiado por cada una de las salas	$Q_{Sens1}, Q_{Sens2} \dots Q_{Sens7}$

Para enlazar de una forma adecuada las variables y obtener una buena representación, se deben realizar las conexiones mostradas en la Figura 2.7. De esta forma, ya se tienen correctamente conectados ambos bloques y se puede continuar con la otra conexión.



**Figura 2.7** Conexiones realizadas entre el edificio y el scope.

Para poder establecer todas las conexiones, ya solo es necesario realizar la correspondiente al bloque atmosférico y el edificio. Para ello, se vuelve a habilitar el icono de las conexiones y se hace la unión desde el bloque del tiempo hacia el edificio. Al igual que antes, aparece una ventana emergente donde entre las variables que se pueden visualizar, hay que destacar las siguientes:

- La temperatura de bulbo seco (*Dry bulb temperature*), es la temperatura medida con un termómetro, el cual se encuentra protegido de fenómenos como la radiación solar y la humedad. Esta debe ser conectada tanto a la temperatura ambiente ( $T_{AMB}$ ), como a la temperatura de la superficie del suelo ( $T_{SGRD}$ ), ambas disponible entre las múltiples entradas del edificio.
- La temperatura efectiva del cielo (*Effective sky temperature*) es un concepto bastante abstracto, pero aquí basta con realizar la conexión entre esta y la entrada del edificio que representa lo mismo, denominada con  $T_{SKY}$
- La humedad relativa del ambiente viene dada por la variable *Percent relative humidity*, que como se sabe representa la cantidad de agua o vapor que está presente en el aire. En el bloque del edificio



existe una variable que representa lo mismo, denominada *RELHUMAMB*, por lo que ambas deben ser conectadas.

- Para la determinación de la radiación incidente en el edificio, la base de datos dispone de las distintas variaciones que se producen en el ángulo cenital (es el formado con la normal del punto donde nos encontramos) y el acimutal (es el que forma la proyección del Sol con la dirección sur). Estas dos variables están representadas por *Solar zenith angle* y *Solar azimuth angle* respectivamente, mientras que en el edificio están definidas como *AZEN* y *AAZM*.
- Para el cálculo de la radiación total horizontal, se ha de especificar la cantidad de radiación incidente (*Total horizontal radiation*) y la *Horizontal beam radiation*, que la cantidad de radiación solar recibida antes de ser dispersada por la atmósfera. Además, también se distingue el parámetro que determina el ángulo de incidencia horizontal. Para ello, estas tres variables deben ser conectadas con *IT\_H\_0\_0*, *IB\_H\_0\_0* y *AI\_H\_0\_0* respectivamente.
- Al igual que sucede con la cara horizontal, el resto de orientaciones (Norte, Sur, Este y Oeste) deben ser caracterizadas con las correspondientes cantidades totales de radiación solar, así como con la beam radiation y el ángulo de incidencia sobre cada superficie. En este caso, las superficies aparecen numeradas desde la 1 hasta la 4, por lo que hay que realizar la conexión sabiendo la relación entre los números y las superficies, del tal forma que:
  - La superficie 1 se corresponde con la orientación norte (*N\_180\_90*).
  - La superficie 2 con la cara este del edificio (*S\_0\_90*).
  - La superficie 3 con la orientación oeste (*E\_270\_90*).
  - Por último, la 4 con la orientada hacia el sur (*W\_90\_90*).
- El último parámetro a tener en cuenta la reflectancia de la tierra (*Ground reflectance*), que es la fracción de la radiación solar que es reflejada por el suelo exterior al edificio. Esta variable, debe estar conectada a la que en el edificio aparece ilustrada como (*GRDREF*), que esta representa el mismo concepto.

Una vez se han aclarado estos conceptos algo más tediosos, en la Tabla 2.2 se muestra un resumen de todas las variables de entradas al bloque que representa el edificio diseñado. Por su parte, en la Figura 2.8, se muestra el resultado de realizar todo el conexionado correspondiente. Debido a las limitaciones de espacio en la impresión de la pantalla las conexiones que no se muestran son las correspondientes a los ángulos de incidencia sobre las superficies.

**Tabla 2.2** Entradas del bloque del edificio (*Type 56*).

Definición	Notación
Temperaturas exteriores	TAMB, TSKY
Humedad relativa del ambiente	RELHUMAAMB
Ángulos cenitales y acimutal	AZEN AAZM
Incidencias totales en cada orientaciones	IT_N_180_90 <sup>a</sup>
Beam radiation	IB_N_180_90 <sup>a</sup>
Ángulo de incidencia sobre cada superficie	AI_N_180_90 <sup>a</sup>
Reflectancia del suelo	GRDRED

<sup>a</sup> Esta variable se da para todas las orientaciones del edificio (norte, sur, este y oeste).

Una vez se han realizado las conexiones correspondientes, se puede proceder a la simulación del archivo y comprobar como el edificio se ve afectado por su interacción con el ambiente. Como se expuso en el mapa conceptual del enlace entre los distintos bloques, el resultado de realizarlo en TRNSYS es el que se puede visualizar en la Figura 2.9. Antes de realizar las simulaciones, habría que manipular los parámetros de la simulación.

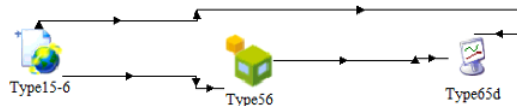
Esto se realiza en la ventana resultante que se genera cuando se accede a *Assembly/Control Cards*, donde se deberá establecer un periodo de simulación, así como un tiempo de muestreo. Para el periodo de simulación, hay que saber que la base de datos del tiempo posee valores almacenados siendo el primer valor el del primer día del año, por lo que para realizar una simulación en una época estival, se deberán introducir valores medios, situados estos alrededor de 5000hr. Para ello, deben modificarse los campos *Simulation start time* y

Variable	Component	Value
Dry bulb temperature	1- TAMB	0
Dew point temperature	2- RELHUMAMB	0
Wet bulb temperature	3- TSKY	0
Effective sky temperature	4- TSGRD	0
Mains water temperature	5- AZEN	0
Humidity ratio	6- AAZM	0
Percent relative humidity	7- IT_N_180_90	0
Wind velocity	8- IT_E_270_90	0
Wind direction	9- IT_W_90_90	0
Atmospheric pressure	10- IT_S_0_90	0
Total sky cover	11- IT_H_0_0	0
Opaque sky cover	12- IB_N_180_90	0
Extraterrestrial solar radiation	13- IB_E_270_90	0
Global horizontal radiation (not interpolated)	14- IB_W_90_90	0
Direct normal radiation (not interpolated)	15- IB_S_0_90	0
Solar zenith angle	16- IB_H_0_0	0
Solar azimuth angle	17- AI_N_180_90	0
Total horizontal radiation	18- AI_E_270_90	0
Horizontal beam radiation	19- AI_W_90_90	0
Sky diffuse radiation on the horizontal	20- AI_S_0_90	0
Ground diffuse radiation on the horizontal	21- AI_H_0_0	0
Total diffuse radiation on the horizontal	22- GRDFEF	0
Angle of incidence for horizontal	23- PERSONAS_S1	0
Total tilted surface radiation for surface-1	24- PERSONAS_S2	0
Total tilted surface radiation for surface-2	25- PC_S1	0
Total tilted surface radiation for surface-3	26- PC_S2	0
Total tilted surface radiation for surface-4	27- PERSONAS_S3	0
Beam radiation for surface-1	28- PC_S3	0
Beam radiation for surface-2	29- PERSONAS_S4	0
Beam radiation for surface-3	30- PC_S4	0
Beam radiation for surface-4	31- PERSONAS_S5	0

**Figura 2.8** Conexiones realizadas entre el tiempo y el edificio.

*Simulation stop time.* En relación al tiempo de muestreo, este puede establecerse en unas unidades distintas a las que se han especificado los límites de la simulación, en el caso de estudio un tiempo de muestreo de *1min* es suficiente ya que los sistemas donde se analizan temperaturas de espacios son más lentos que muchos otros. Para realizar dicha modificación, se debe hacer en el campo *Simulation time step*.

De aquí en adelante, se mostrarán los diversos comportamientos del sistema creado cuando este se ve influido por las condiciones atmosféricas que se dan en el exterior. Para ello, con el esquema realizado y habiendo definido los parámetros de simulación de forma adecuada y coherente, basta con ejecutarlo pulsando F8. Si todo se ha realizado correctamente, se debe visualizar el comportamiento en una ventana auxiliar generada por TRNEXE.



**Figura 2.9** Resultado de realizar todas las conexiones.

Para la presentación en esta memoria descriptiva, dichas gráficas resultantes, van a ser las realizadas con MATLAB®, ya que las obtenidas con la herramienta de TRNSYS, no son de tan buena calidad como las obtenidas con la otra herramienta. Para ello, obviamente se debe realizar una comunicación entre ambos programas. Esta comunicación, en esta ocasión, se supone conocida, ya que será objeto de estudio en Capítulo 3. Estas simulaciones tendrán una duración de un día (24 hr), por lo que se podrán observar todos los cambios producidos en cada una de las salas cuando se modifican los valores de las distintas incidencias de la radiación solar a lo largo del día.

Como se puede observar en la Figura 2.10 y Figura 2.11, durante las primeras horas del día, las radiaciones incidentes a primera horas de la mañana son mayores en la cara este del edificio, mientras que a última hora, este fenómeno se produce en la orientada hacia el oeste como es lógico. Sin embargo, en las caras norte y sur, las radiaciones son más duraderas en el tiempo pero presentan una menor incidencia.

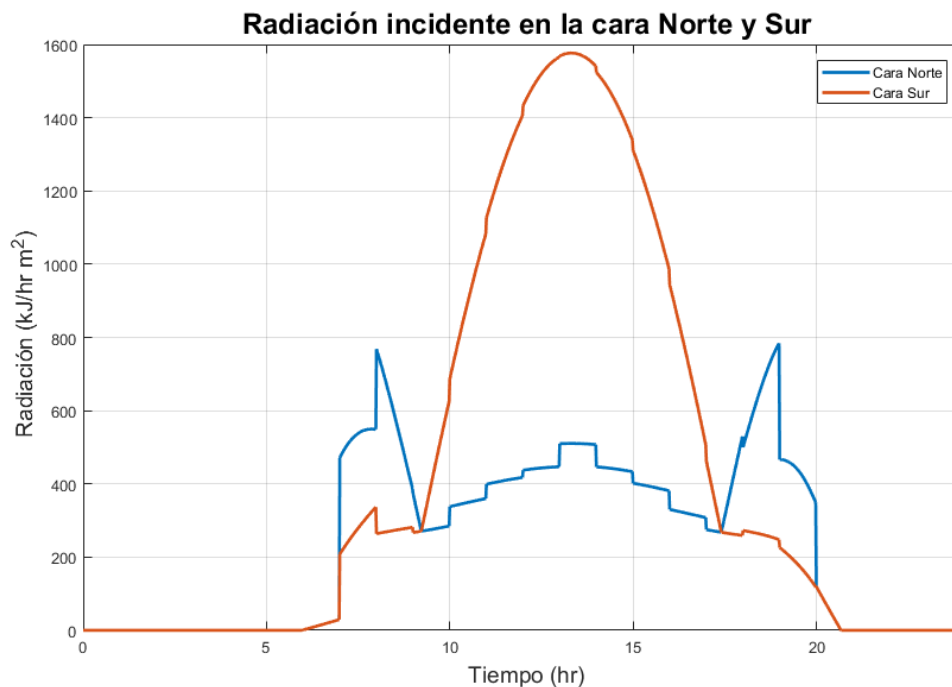


Figura 2.10 Radiación incidente en las caras Norte y Sur.

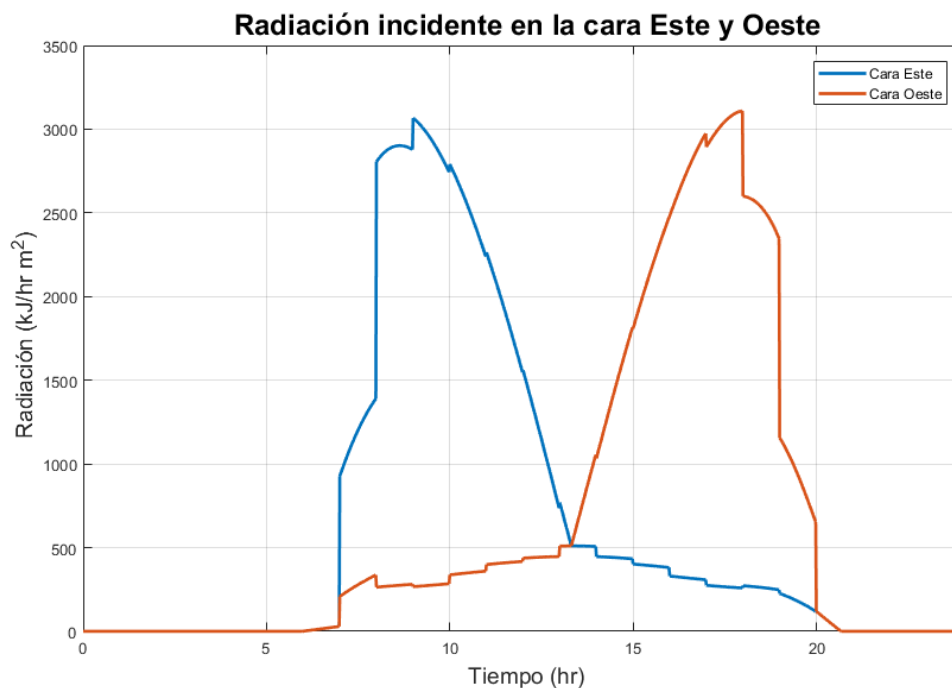


Figura 2.11 Radiación incidente en las caras Este y Oeste.

Con esta incidencia, las temperaturas de las habitaciones, representadas en la Figura 2.12 y Figura 2.13, resultan:

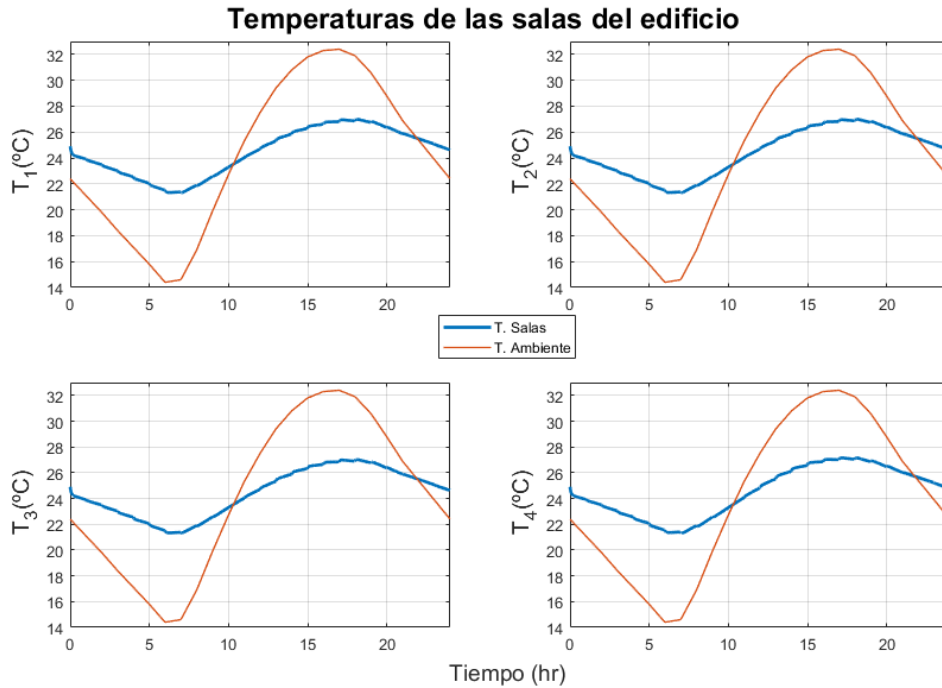


Figura 2.12 Temperaturas de la sala 1 hasta la 4 considerando la influencia atmosférica.

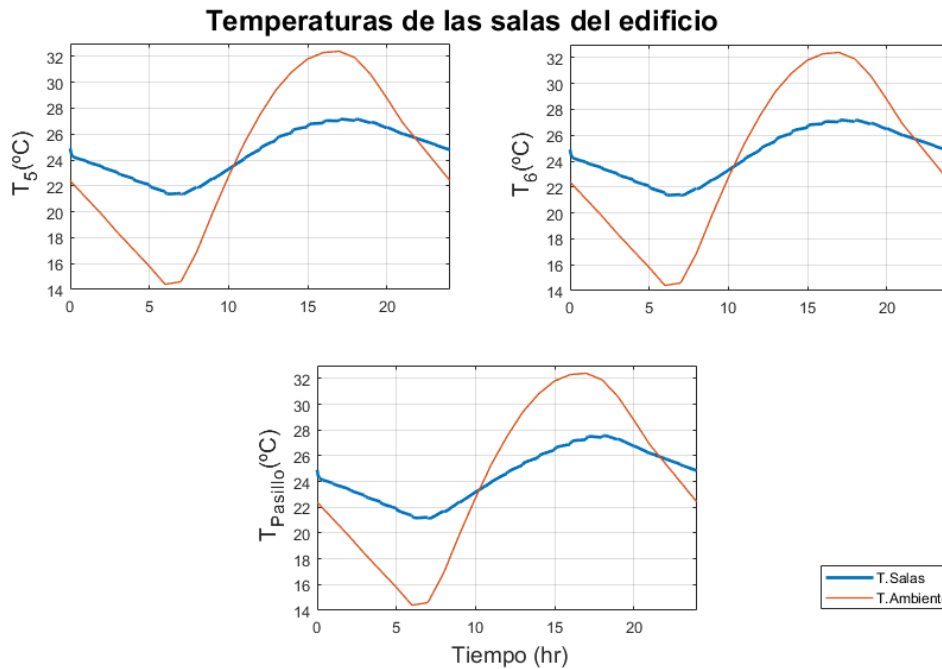


Figura 2.13 Temperaturas de la sala 5 hasta la 7 considerando la influencia atmosférica.

Como se puede observar, las gráficas resultantes son coherentes, y habiendo una temperatura exterior que ronda los 32°C en su pico máximo, las temperaturas de las salas ascienden a un valor comprendido entre los 27°C y 28°C. Además, aunque no se aprecie a simple vista, cuando la radicación incidente es máxima en la

cara oeste (última hora de la tarde), es en esa sala donde se produce un repunte en la temperatura alcanzando un valor de  $27.57^{\circ}\text{C}$ , siendo este superior al del resto de los espacios en ese instante.

Con esto se da por concluido la redacción del Capítulo 2, donde se ha explicado, el modelado del sistema y, se han expuesto los resultados obtenidos tras el modelado, observando que son coherentes, considerando así que el sistema es apto para proceder con los siguientes pasos.



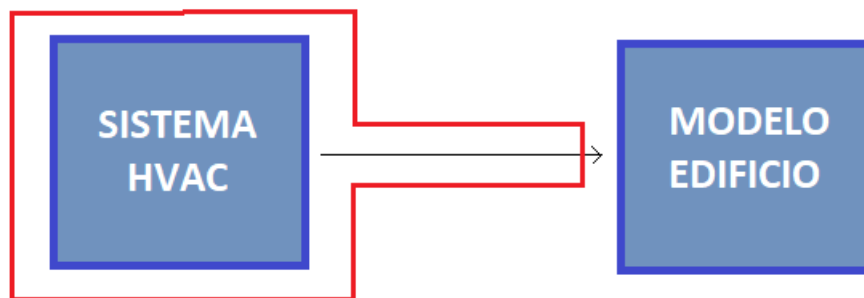
### 3 Incorporación de Equipos

---

**D**urante la redacción del correspondiente capítulo se pretende dar a conocer los distintos elementos que componen el sistema HVAC (*Heating, Ventilation and Air Conditioning*) que será empleado para acondicionar la temperatura de uno de los espacios que componen el edificio.

Para ello, durante todo el capítulo se va a ir exponiendo uno a uno todos los elementos empleados y, una vez se disponga del esquema completo y el modelado de cada uno de ellos, se va a estudiar cómo debe implementarse en TRNSYS, usando para ello la herramienta MATLAB<sup>®</sup>, por lo que también debe explicarse cómo se comunican y que variables usan ambos programas. Por último, para poder comprobar si todo ello se ha realizado correctamente, se procede a la presentación de los resultados obtenidos y el correspondiente análisis de todos ellos.

A modo de ejemplo, el objetivo de este capítulo puede expresarse en la Figura 3.1, donde todo lo que se encuentra bajo el rectángulo de color rojo, será lo que vaya a ser tratado.



**Figura 3.1** Objeto de estudio en el Capítulo 3.

Para comenzar el desarrollo del capítulo, se partirá por la definición de cada uno de los componentes y cada uno de los modelos que se van a implementar para poder lograr la construcción del sistema que permita acondicionar las temperaturas de cada una de las salas [11] [1].

#### 3.1 Descripción del sistema y modelo de esqupos

En primer lugar, antes de entrar en materia es conveniente saber qué es un sistema HVAC. Un sistema HVAC básicamente es un conjunto de equipos que deben cumplir tres funciones: la primera de ellas, hace referencia al calentamiento de una estancia para poder aumentar así su temperatura; el segundo objetivo es la ventilación de espacios, que consiste en la renovación de aire interior de un habitáculo mediante la aportación del aire exterior con el objetivo de asegurar su calidad; la última función es la de acondicionar la sala enfriándola para lograr así una disminución de su temperatura. Todos estos, son los pilares básicos sobre los que se apoyan los sistemas HVAC, pero durante la realización de este proyecto, el trabajo se centrará exclusivamente sobre uno de ellos, que se basará en la refrigeración.

Lo que se pretende es tener la posibilidad de mantener la temperatura de cada uno de los espacios, que se encuentran en un modelo diseñado, en unos niveles razonables cuando la temperatura exterior es superior a la deseada, es decir, cuando el edificio en cuestión se encuentra en una época estival. Dicho esto, ya se deja claro que el objetivo a conseguir es el enfriamiento mediante una extracción de energía, de las salas que componen un edificio.

Para el desarrollo del resto del capítulo, el desarrollo explicativo se realizará para el caso particular que ha sido elaborado en el Capítulo 2, donde de forma resumida, se dispone de un edificio compuesto de seis salas (tres de ellas ubicación norte y las otras tres con orientación sur), todas ellas comunicadas entre sí mediante un pasillo.

Volviendo al objeto del capítulo, como se ha dicho, el objetivo primordial es el enfriamiento de las habitaciones. Para ello se propone un sistema de refrigeración basado en un circuito cerrado de tuberías de agua. El funcionamiento de dicho circuito es el siguiente: el agua de la que se dispone en dicho circuito de hace pasar, mediante el impulso provocado por una bomba, por una enfriadora o *chiller* que permite disminuir su temperatura. Una vez se ha conseguido enfriar el agua, esta se distribuye uniformemente en un sistema de siete tuberías que desemboca en cada una de las salas. Para poder realizar el intercambio energético que permita la disminución de temperatura se hace uso de un *fancoil* o intercambiador de calor. La potencia intercambiada en cada espacio depende fundamentalmente del caudal de agua que circula por cada fancoil, dicho caudal se encuentra regulado por sendas válvulas de tres vías controlables que permite la modificación del caudal que circula por cada intercambiador. Por último, una vez el agua ha atravesado el intercambiador y su correspondiente temperatura ha aumentado, todos los caudales de agua son recogidos mediante un mezclador para lograr que el agua vuelva a ser enfriada por el chiller. Mediante este ciclo, se consigue extraer una determinada potencia a cada una de las salas y conseguir así disminuir la temperatura de cada una de ellas.

A modo de resumen, lo anteriormente expuesto puede visualizarse en la Figura 3.2. En dicho esquema se ha simplificado el número de válvulas para poder ilustrar su composición en profundidad y poder detallar así donde se ubica cada uno de los componentes.

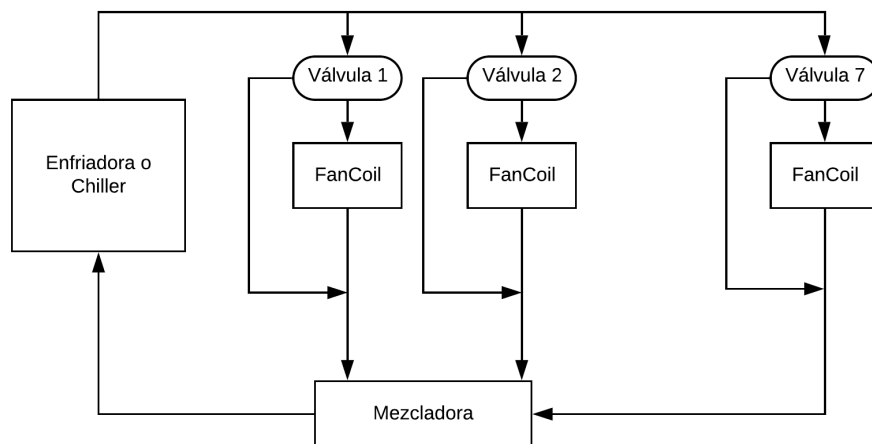


Figura 3.2 Esquema del circuito refrigerador.

A continuación, se va a ir exponiendo uno a uno cada uno de los elementos y las ecuaciones que modelan cada uno de ellos, dichas ecuaciones han sido elaboradas con datos de fabricantes comerciales para un determinado modelo [11].

En primer lugar, el elemento más básico que se encuentra es la válvula de tres vías, cuya función es básicamente regular la cantidad de caudal que se deriva al fancoil y por el canal de derivación. El parámetro que caracteriza su comportamiento se denomina apertura y puede tomar valores comprendidos entre cero y uno. El valor de dicho parámetro representa el porcentaje del caudal parcial (es el que se distribuye por cada módulo, es igual al que atraviesa cada intercambiador más el que es derivado) que pasa por el intercambiador, de forma que el resto es desviado por el canal de derivación manteniendo su temperatura constante. En las ecuaciones que describen el comportamiento, se denota como  $q_{ps}$  al que pasa por el intercambiador,  $q_{pd}$  al



que es derivado y  $q_p$  al caudal parcial que pasa por cada unidad. Con esto, las ecuaciones que modelan el comportamiento son

$$\begin{aligned} q_{ps} &= q_p A \\ q_{pd} &= q_p (1 - A) \end{aligned} \quad (3.1)$$

donde, el parámetro  $A$ , representa la apertura de cada una de las válvulas. Hay que indicar que, durante todo el desarrollo que se va a realizar, las unidades en las que se expresan los caudales serán  $l/min$ . Por otro lado, se tiene que el caudal total que circula por esquema, representado por  $q$ , cumple

$$q = \sum_{i=1}^7 q_{pi} \quad (3.2)$$

que, para el caso de estudio aplicado, se repartirá de forma equitativa por cada una de las salas en cuestión, por lo que el caudal parcial de cada uno de los módulos será una séptima parte del caudal total de partida que se ha establecido en el diseño del equipo.

El siguiente elemento básico que se encuentra en el esquema es la mezcladora, cuya función es garantizar la mezcla de todos los conductos disponibles, los cuales incluyen los que han atravesado el fancoil tanto como los que han sido desviados. A la salida de la enfriadora el caudal es conocido ya, pues es la suma de todos los caudales parciales, denotado como  $q$ , sin embargo, el cálculo de la temperatura ( $T_{we}$ ) es algo más tedioso, pues se tienen distintos caudales a distintas temperaturas. La temperatura resultante del agua se puede obtener realizando una media ponderada en base a los caudales, de cada una de las temperaturas que llegan a la mezcladora. De forma matemática, esto puede expresarse como

$$T_{we} = \frac{\sum_{i=1}^7 (T_{ps} q_{ps} + T_{pe} q_{pd})}{q}, \quad (3.3)$$

donde,  $T_{ps}$  ( $^{\circ}C$ ) representa la salida de la temperatura del agua del intercambiador y  $T_{pe}$  ( $^{\circ}C$ ) es la temperatura del agua del canal de derivación, o lo que es lo mismo la temperatura del agua a la salida de la enfriadora, puesto que se supone que no se producen pérdidas energéticas en dicho canal.

En relación a la enfriadora, esta se puede definir como una máquina frigorífica cuyo objetivo es el de enfriar un medio líquido. Lo que interesa con respecto a este equipo es, que para dada una capacidad nominal ( $CAP_N$ ) y un factor de carga parcial ( $fcp$ ), se pueda determinar la temperatura ( $T_{ws}$ ) a la que es enfriado el caudal de agua que atraviesa dicho equipo. El caudal que entra está definido por la variable  $q$  ( $l/s$ ) y por la temperatura  $T_{we}$  ( $^{\circ}C$ ) que, si no se consideran pérdidas en el conducto, esta coincide con la de la salida de la mezcladora.

El factor de carga parcial que afecta a la enfriadora puede obtenerse mediante el cálculo que puede observarse en

$$fcp = \frac{c_p \rho q (T_{we} - T_{ws})}{CAP} \quad (3.4)$$

donde, la  $c_p$  es la capacidad calorífica del fluido,  $q$  ( $l/s$ ) representa el caudal del trabajo de la máquina,  $T_{we}$  la temperatura a la entrada,  $T_{ws}$  la de la salida, y la variable  $CAP$ , representa la capacidad de la enfriadora, la cual depende de su valor nominal ( $CAP_N$ ) y de los valores que toman la temperatura del aire exterior,  $T_{ae}$  ( $^{\circ}C$ ), y de la temperatura del agua a la salida de la enfriadora,  $T_{ws}$ . En forma de ecuación, todo ello puede expresarse como

$$CAP = CAP_N (c_0 + c_1 T_{ws} + c_2 T_{ws}^2 + c_3 T_{ae} + c_4 T_{ae}^2 + c_5 T_{ws} T_{ae}), \quad (3.5)$$

en la cual pueden observarse algunos parámetros  $c_i$ , los cuales han sido obtenidos por identificación por datos aportados por el fabricante. Los valores de estos coeficientes son los que pueden observarse en la Tabla 3.1.

A modo de resumen, la enfriadora puede ser considerada como un bloque, al cual se le aporta un valor del factor de carga parcial ( $fcp$ ) y una temperatura de entrada del agua ( $T_{we}$ ), y este devuelve, en función de la temperatura ambiente ( $T_{ae}$ ), una temperatura de salida del agua ( $T_{ws}$ ).

El último elemento que se va a emplear es el *fancoil* o intercambiador de calor. Este elemento está constituido por un conjunto de tubos por los que circula el fluido en cuestión, dicho fluido ve modificada su

**Tabla 3.1** Coeficientes  $c_i$  de la CAP de la enfriadora..

$c_i$	Valor	$c_i$	Valor
$c_0$	0.90491362	$c_3$	0.00284939
$c_1$	0.04080004	$c_4$	-0.00016049
$c_2$	0.00052899	$c_5$	-0.00048742

temperatura cuando interacciona energéticamente con el aire de la sala que atraviesa el ventilador. De esta forma, el fluido que está a una temperatura más baja que la del aire, absorbe energía en forma de calor del aire, para poder llegar a un equilibrio energético donde la temperatura del aire de la sala ha disminuido y por tanto, la del fluido ha aumentado. Para llevar a cabo el modelado de dicho equipo, se procede de igual forma que en la enfriadora, a partir de una identificación por los datos aportados por el fabricante. Este equipo se caracteriza esencialmente por la potencia  $P (W)$  que absorbe de la habitación, la cual depende del caudal que atraviese el intercambiador,  $q_{ps} (l/min)$ , así como de la temperatura del fluido,  $T_{pe} (°C)$ . Para ello, se emplea una ecuación como

$$P = p_{00} + p_{10}T_{pe} + p_{01}q_{ps} + p_{20}T_{pe}^2 + p_{02}q_{ps}^2 + p_{11}T_{pe}q_{ps} + p_{30}T_{pe}^3 + p_{21}T_{pe}^2q_{ps} + p_{12}T_{pe}q_{ps}^2 + p_{03}q_{ps}^3 \tag{3.6}$$

Los valores de los coeficientes que pueden observarse en la Ecuación 3.6, son los obtenidos por la identificación realizada con los datos aportados por los fabricantes y, sus valores son los expuestos en la Tabla 3.2.

**Tabla 3.2** Coeficientes  $p_{ij}$  de la Potencia del intercambiador ..

$p_{ij}$	Valor	$p_{ij}$	Valor
$p_{00}$	467.3	$p_{02}$	19.54
$p_{10}$	-158.3	$p_{30}$	-0.4593
$p_{01}$	-902.7	$p_{21}$	0.0156
$p_{20}$	15.95	$p_{12}$	-0.3116
$p_{11}$	26.61	$p_{03}$	-0.139

Por convenio, la potencia intercambiada para enfriar el sistema se toma de signo negativo, es decir, es el intercambiador el que gana calor del aire de la sala. Además, de forma lógica, se sabe que cuando circula un mayor caudal a una temperatura constante, la potencia intercambiada es mayor. Por otro lado, si la temperatura del fluido aumenta siendo el caudal constante, la cantidad de energía intercambiada disminuye y se produce así menos transferencia de calor de un lado a otro.

A modo de resumen, todos los equipos explicados y cada una de sus variables pueden visualizarse en la Figura 3.3. Para poder interpretar cada elemento como un bloque del que se dispone de una serie de entradas, que proporcionan una salida, se ha elaborado la Tabla 3.3.

**Tabla 3.3** Tabla resumen de las variables empleadas por cada equipo..

Equipo	Entradas	Salidas	No modificadas	Var. manipulables
Válvulas	$q_p$	$q_{ps}, q_{pd}$	$T_{ws}$	Aperturas(A)
Mezcladora	$q_{ps}, q_{pd}, T_{ws}, T_{ps}$	$q, T_{we}$		
Enfriadora	$T_{we}, T_{ae}$	$T_{ws}$	$q$	$fcp$
FanCoil	$T_{ws}, q_{ps}$	$P, T_{ps}$		

Con ello, ya se ha definido el sistema capaz de extraer la energía necesaria que permita mantener las temperaturas de cada una de las habitaciones en torno a unos niveles razonables.

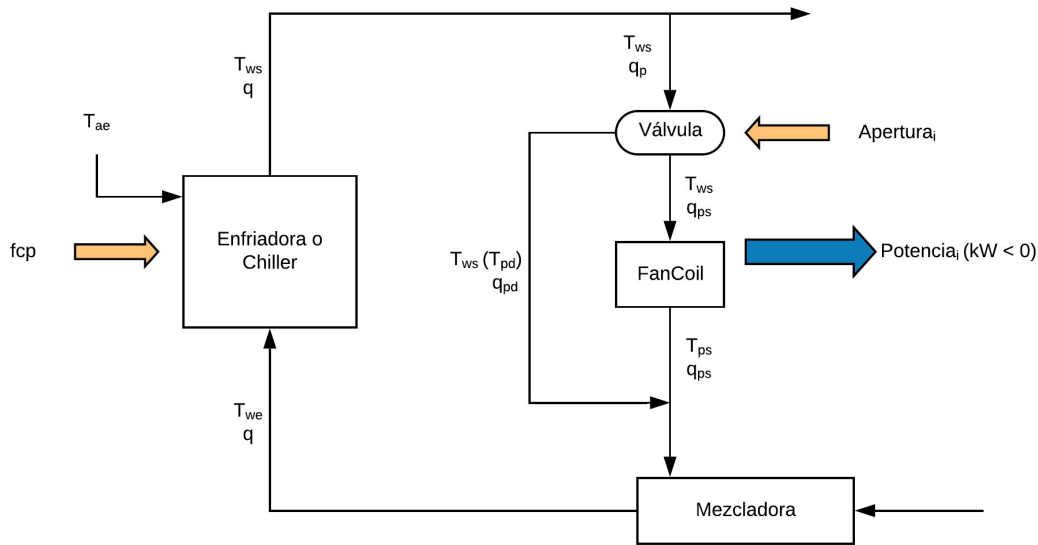


Figura 3.3 Esquema del circuito representando las variables.

### 3.2 Comunicación TRNSYS-MATLAB®

Cuando se dispone ya de cada uno de los modelos que simulan el funcionamiento de todos los equipos, se debe saber que la herramienta empleada para su incorporación en TRNSYS, va a ser MATLAB®. Para ello, se hace uso de un bloque disponible en TRNSYS, más concretamente el denominado como "Type 155" o "Calling Matlab", en dicho bloque en la pestaña *Special Cards*, se debe indicar la ruta del archivo ".m" que contiene el código que se desea ejecutar.

Para la ejecución se emplea un script en el cual se hace uso de diversas variables internas aportadas por TRNSYS, que permiten saber en qué parte de la ejecución se encuentra dicha simulación en cada tiempo de muestreo. En primer lugar, es conveniente saber qué partes se pueden encontrar en el archivo fuente [22] y cuál es la función de cada una de ellas:

- En primer lugar, se define una primera parte en la cual se deben realizar todas las inicializaciones necesarias para las posteriores ejecuciones. Dicho fragmento de código, solo se ejecutará una vez, por lo que si es necesario tomar algunos parámetros del esquema de TRNSYS, también deben ser leídos como entradas. Todas las variables que se definan serán almacenadas y se podrá disponer de ellas en cualquier momento de la simulación.
- De forma complementaria, también se encuentra una parte que se ejecuta en último lugar, es decir una vez se ha producido la última iteración. En esta parte se suelen realizar almacenamiento de variables, para poder disponer de ellas en cualquier momento tras la finalización de la simulación.
- La parte fundamental es la correspondiente al código que se debe ejecutar de forma iterativa, la cual se ejecuta cada tiempo de muestreo. Es aquí donde se debe introducir todas las llamadas a los equipos que se han modelado.

Para poder realizar de forma exitosa la posible división del código, se hace uso de diferentes variables aportadas por TRNSYS. Entre ellas, las más importantes son:

- Una de las variables más importantes es *trnInfo*, un vector de quince componente en la que cada una de ellas aporta una determinada información, entre ellas se deben destacar: las componentes 7 y 8 indican en qué parte de la simulación se encuentra (inicialización, llamada iterativa o final); la componente número 3 aporta la cantidad de entradas que son introducidas en el *Type 155*, mientras que la 6 indica el de salidas.
- Para poder determinar los valores de cada una de las entradas y salidas que son aportadas y calculadas, se dispone de los vectores *trnInputs* y *trnOutputs*, en los cuales se almacena dicha información.

- Para poder establecer en instante inicial de la simulación este se puede encontrar en la variable denotada como *trnStartTime*. De forma análoga, para indicar el instante final se hace uso de *trnStopTime*. Para indicar el tiempo de muestreo que se ha establecido en la simulación, este se encuentra disponible en *trnTimeStep*.
- Adicionalmente, en el código se puede observar una variable llamada *mFileErrorCode*, cuya función no es más que la de poder identificar un error cuando se realiza la simulación, ya que uno de los inconvenientes que presenta la comunicación es la dificultad para poder localizar dichos errores, puesto que estos no son indicados por MATLAB®.
- Para indicar el tiempo de muestreo en el que se encuentra la simulación, se emplea una variable definida por el usuario y denotada como *nStep*, la cual se debe incrementar cada tiempo de muestreo.

Con todo ello, el código empleado para realizar las inicializaciones correspondientes en el que se puede ver el uso que se hace de cada una de las variables explicadas, es el siguiente:

```
% Estructura del código de inicialización
if ( (trnInfo(7) == 0) & (trnTime-trnStartTime < 1e-6) )
    % Se define la variable nStep
    nStep = 1;

    % Aquí se realizan el resto de inicializaciones necesarias

    % El código para identificar posibles errores
    mFileErrorCode = 120
end
```

En la parte que se ejecuta una vez por cada tiempo de muestreo, es decir, en el ciclo iterativo, se ha de incrementar la variable que indica el tiempo de muestreo en el que se encuentra el sistema, *nStep*. Además de esto, se quiere emplear un controlador sobre el sistema en cuestión, el lugar correcto para hacerlo sería este. El fragmento de código disponible es:

```
% Se incrementa el instante
if ( trnInfo(7) == 0 )
    nStep = nStep+1;
end

% Se indica que comienza el proceso iterativo
mFileErrorCode = 130;

% Aquí se hacen los cálculos necesarios

% Se indica que acaba el proceso iterativo
mFileErrorCode = 0;
```

Como se ha podido observar, la componente número siete de *trnInfo* es la que marca que se deben realizar las inicializaciones y la llamada inicial. Sin embargo, esta última debe cumplir además la condición de que la ejecución se debe encontrar justo en el momento que ha comenzado la simulación, mientras que para la fase iterativa, únicamente se hace uso de la componente siete. De esta forma, se llega a la conclusión de que la última llamada se indicará mediante la componente número ocho.

Para finalizar, el último extracto de código es el relativo a la última llamada que hace TRNSYS a MATLAB® justo antes de acabar con la simulación. Este se compone de:

```
% Código correspondiente a la última llamada
if ( trnInfo(8) == -1 )
    % Se indica que empieza la última llamada
    mFileErrorCode = 1000;

    % Aquí se hacen los cálculos necesarios
```

```

% Se indica que se ha llegado al final
mFileErrorCode = 0;
return
end

```

Con todo ello, ya se puede entender de una forma algo más clara como se estructura el código que se emplea para realizar la comunicación, aunque conocer todo esto no es suficiente para realizarla de forma correcta. Como se ha enunciado, el bloque de TRNSYS que ejecuta el archivo de MATLAB®, se debe conectar a otros bloques presentes para generar así una serie de salidas y entradas al bloque en cuestión. El objetivo de las siguientes líneas es conseguir realizar dicha modificación.

Para poder establecer el número de salidas y entradas deseadas en el bloque *Type 155*, en la pestaña que se crea cuando se hace doble click sobre este (llamada *Parameter*), aparecen una serie de variables que se pueden manipular. Entre ellas, hay que destacar las llamadas *Number of inputs* y *Number of outputs*, en las cuales bastas con indicar el número de variables de cada una de ellas. Para modificar el nombre que aparece cuando se va a realizar el conexionado y por tanto, evitar posibles confusiones, este se puede modificar en la pestaña *Input* para las entradas y *Output* para las salidas. Sin embargo, este nombre no es determinante, pues para poder hacer referencia a ellas en MATLAB®, estas se encuentran almacenadas en los vectores *trnInputs* y *trnOutputs*.

De esta forma se ha abordado la explicación de cómo realizar la comunicación entre las dos herramientas software disponibles, las cuales serán de gran interés para los próximos apartados.

### 3.3 Implementación en TRNSYS y MATLAB® del equipo propuesto

Durante la realización de la Sección 3.1 se ha llevado a cabo la definición y explicación de cada uno de los modelos empleados para la caracterización de los equipos que componen el sistema de refrigeración empleado. De esta sección se pueden realizar diversas funciones de MATLAB® que implementen los desarrollos matemáticos explicados.

Las funciones de las que se disponen, presentan una serie de entradas y se generan ciertas salidas. La función relativa al comportamiento de la válvula es

$$[qsi, qbi] = Valve(qi, A),$$

donde *qsi* representa el caudal que pasará por el fancoil, *qbi* es el derivado por el canal adyacente, *A* hace referencia a la apertura de la válvula y *qi* al caudal parcial que circula por cada módulo. En el caso en el que se está llevando a cabo el análisis, dicho caudal es una séptima parte (hay siete estancias) del caudal total, el cual es un parámetro de diseño del equipo. Este ha sido elegido con un valor de  $70l/min$ , de formar que por cada módulo circulen un total de  $10l/min$ . Para la enfriadora o chiller se emplea

$$[Twsnext, solTws] = chiller(Twsant, Twe, Tae, fcp, q),$$

en la cual, *Twsnext* es la temperatura a la que va a ser enfriada el agua a la salida en el proximo tiempo de muestreo, es decir, en un minuto, *Twsant* es la temperatura a la salida en el anterior tiempo de muestro, *Twe* es identificada como la temperatura a la entrada, *Tae* es la del ambiente exterior, *fcp* representa el factor de carga parcial y *q* el caudal total de trabajo (establecido como  $70l/min$ ). En el contenido de dicha función se deben realizar dos puntualizaciones necesarias para aclarar su contenido:

- En dicha función se debe establecer la capacidad nominal de la máquina ( $CAP_N$ ), la cual identifica la potencia disponible para enfriar el agua. Para el edificio en cuestión con una superficie aproximada de unos  $138m^2$ , y sabiendo que para poner obtener los resultados deseados se suelen necesitar unas  $100\text{ frigorías/h}$  por cada  $m^2$ , y sabiendo que  $1W$  equivale a  $0.86\text{ frigorías/h}$ , Se estima una potencia necesaria de unos  $16kW$ , pero para aporta una mayor seguridad y no saturar el sistema, dicho parámetro tomará un valor de  $24.5kW$ .
- El segundo aspecto a destacar es cómo se realiza el cálculo de la temperatura de la salida de la enfriadora. Para ello se utiliza una aproximación mediante una función de transferencia de primer orden con una constante de tiempo ( $\tau = 2min$ ), que indica que el e cambio de temperatura que se produce no es instantáneo y necesita de un tiempo para poder alcanzarse.

La última función elaborada es la correspondiente al intercambiador de calor o *fancoil*, la cual presenta como entradas y salidas las establecidas en la definición de la función

$$[Pout, Tout] = fanCoil(Tin, qin),$$

en ella, *Pout* es la potencia ganada por el agua cuando se produce el intercambio energético, *Tout* se puede definir como la temperatura del agua a la salida del intercambiador, *Tin* es a la que se encuentra cuando entra al intercambiador y *qin* el caudal que circula por los conductos del intercambiador.

Atendiendo a la estructura que se ha planteado en cada una de las ecuaciones, se debe plantear el orden en el que se deben realizar cada una de las llamadas a cada una de las funciones. Para ello, basta con realizar un análisis lógico del planteamiento del problema en cuestión, el cual puede dividirse en varios pasos o fases:

1. Antes de comenzar se debe realizar una inicialización de las distintas variables que van a ser empleadas: una de ellas es el caudal total que circulará por el equipo completo siendo  $q = 70 l/min$ , la temperatura inicial del agua que se encuentra en el conducto se va a considerar igual a  $15^{\circ}C$  siendo esta igual en todos los puntos del conducto ( $T_{ws} = T_{we}$ ), se deben establecer unos parámetros iniciales de *fcp* y apertura para que el sistema comience a funcionar y con dicho parámetros también se conoce (usando la función *valve*) la cantidad de caudal que circula por el intercambiador y cuanta se desvía.
2. Para comenzar una simulación recurrente de todos los equipos que componen el sistema, se va a comenzar con la válvula, la cual únicamente depende de la apertura inicial que ha sido establecida.
3. Cuando se dispone de la cantidad de caudal que circula por cada una de las ramas, se ejecuta la función del intercambiador que calcula la potencia que se cede a cada una de las salas del edificio y la temperatura a la que sale el agua una vez ha atravesado el equipo.
4. Antes de entrar en la enfriadora, se debe aplicar la ecuación de la mezcladora, usando para ello la definida en las primeras páginas de dicho capítulo.
5. Por último, solo es necesario realizar el cálculo de la temperatura a la que es enfriada el agua cuando atraviesa el *chiller*, mediante la función elaborada para ello.

Todos estos, son los pasos a realizar, de forma que en cada tiempo de muestreo se debe ejecutar desde el paso 2 hasta el 5, siendo este el orden correcto de llamada a cada una de las funciones y conseguir así que el sistema sea ejecutado y se comporte de manera deseada, ya que si se emplea un orden distinto al indicado, hay variables que se necesitan y todavía no han sido calculadas, por lo que no se podría realizar la simulación. Para realizar todo esto, se ha empleado el siguiente fragmento de código, el cual está ubicado en la parte del archivo que se debe ejecutar cada tiempo de muestreo.

```

%% Tratamos en primer lugar a la valvula
for i=1:7
    [qsi(i,nStep) qbi(i,nStep)] = Valve(qin(i,1),Apertura(i,nStep));
end

%% Ahora se intercambia calor con la habitacion y el fancoil
for i=1:7
    [Potencias(i,nStep) Tout(i,nStep)] = fanCoil(Twsnext(1,nStep-1),qsi(i,nStep)
    );
end

%% Una vez pasa por el fancoil se mezclan todos los flujos
Tmix = (qsi(:,nStep).*Tout(:,nStep)+qbi(:,nStep)*Twsnext(1,nStep-1))./(qsi(:,
    nStep)+qbi(:,nStep));
Twe(1,nStep) = (Tmix'*qin(:,1))/sum(qin(:,1));

%% Lo ultimo en ejecutar es el chiller
[Twsnext(1,nStep) solTws] = chillerUnit(Twsnext(1,nStep-1),Twe(1,nStep),Tae,fcp
    (1,nStep),caudal_total);

```

Una vez se ha abordado el problema de la implementación del equipo en MATLAB<sup>®</sup>, todo el equipo refrigerador puede verse como un bloque al cual se le aporta la temperatura exterior del ambiente y genera unas salidas en forma de potencia que se deben aplicar a cada una de las salas que componen el edificio en cuestión. De esta forma, para solucionar el problema completo, solo es necesario realizar la conexión del equipo con el archivo ".m" en TRNSYS.

Para empezar a realizar el conexionado, hay que indicar en el bloque del edificio, que se va a introducir una potencia en cada una de las salas, de forma que esta será la encargada de refrigerarlas. Para ello, dichas potencias van a ser tratadas como propias ganancias internas de cada uno de las salas que dispone el edificio.

Para la definición de dicho parámetro como una entrada del bloque *Type 56* hay que realizar la operación que se va a exponer sobre cada una de las salas. Para comenzar, se debe seleccionar la sala sobre la que se desea añadir, y en la ventana que aparece hay que dirigirse a los parámetros denominados como *Gains* (marcada con el símbolo de una vela). Una vez se pulse, aparece un cuadro donde se pueden especificar varios tipos de ganancias, donde el correspondiente en este caso es *Other Gains*, donde se debe pulsar sobre el icono "+" y definir un nuevo tipo de ganancia. En la ventana emergente que aparece indica el tipo de ganancia a incorporar, se sabe que el intercambio de calor entre el aire y el fluido que circula por los conductos se produce por el fenómeno de convección, por lo que este será el tipo elegido. Para ello se pulsa sobre la flecha verde en el apartado *Convective Power* y se selecciona como variable una de tipo entrada (*Input*). Realizando este procedimiento para cada una de las salas, se llega a disponer de siete nuevas entradas al bloque del edificio, las cuales caracterizan la potencia aportada por cada uno de los intercambiadores sobre cada sala del edificio.

Cuando se llega a este punto, hay que darse cuenta de que dichas potencias definidas hay que introducirlas en *kJ/hr*, y si se observa la salida del bloque de la máquina enfriadora, estas son calculadas en *W*, por lo que se debe realizar una conversión de unidades para realizar la aportación de forma correcta. Dicha conversión se realiza como

$$P_{TRNSYS}(kJ/hr) = P_{MATLAB}(J/s) \frac{3600(s)}{1(hr)} \frac{1(kJ)}{1000(J)}. \quad (3.7)$$

Por ello, es necesario realizar este cálculo antes de pasar los datos a los vectores de salida.

De la forma explicada, el bloque *Type 155*, que es el correspondiente al archivo de MATLAB<sup>®</sup>, debe estar compuesto de una entrada y siete salidas. De esta forma, el esquema resultante al realizar la conexión debe ser el expuesto en la Figura 3.4.



Figura 3.4 Esquema del conexionado a realizar.

De esta forma, ya se puede realizar diversas simulaciones para observar cómo se comporta el equipo implementado cuando se producen modificaciones en las variables que modelan su funcionamiento, es decir cambios en los valores que tanto el fcp como la apertura de cada una de las válvulas que componen el sistema.

### 3.4 Resultados de las simulaciones

Para llevar a cabo un análisis de los resultados que se obtienen tras la implementación de todo lo que ha sido explicado hasta ahora, se van a realizar concretamente dos simulaciones en TRNSYS. En ellas se van a producir variaciones en las variables manipulables, es decir, en las aperturas de cada una de las válvulas y en el valor del factor de carga parcial (*fcp*). En todas las simulaciones, se toma una temperatura exterior del ambiente constante e igual a unos 30°C.

En la primera simulación que se ha realizado, se han modificado dos de las aperturas de las válvulas a un valor arbitrario entre cero y uno, dejando en un valor constante el fcp, siendo este igual a 0.1. Para ello, se parte de un instante donde el valor de las aperturas es 0.3. Para ilustrar correctamente los resultados, se van a representar las temperaturas de cada una de las salas, esto se puede visualizar en las Figuras 3.5 y 3.6.

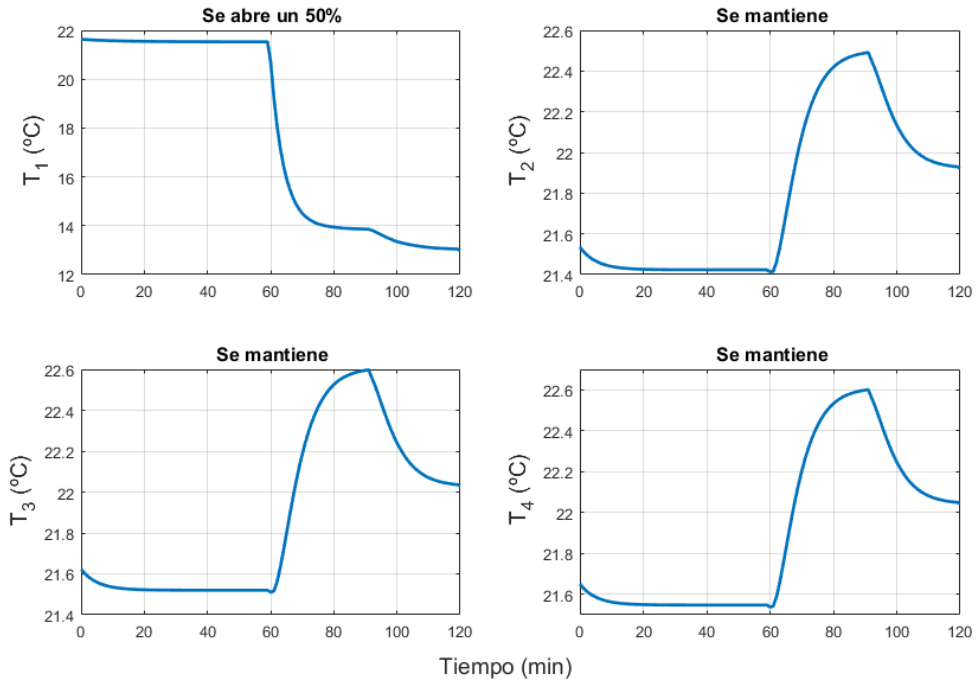


Figura 3.5 Resultados obtenidos tras la modificación de las Aperturas (Salas 1 a 4).

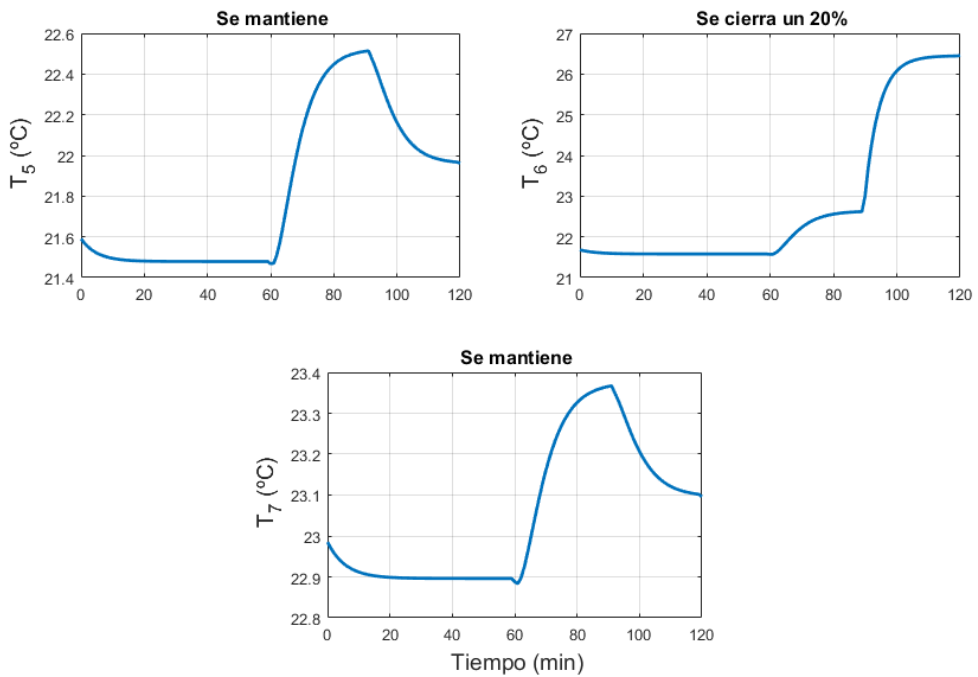


Figura 3.6 Resultados obtenidos tras la modificación de las Aperturas (Salas 5 a 7).

Una vez se obtienen los resultados de las simulaciones que han sido realizadas, es conveniente destacar y obtener algunas conclusiones de los resultados obtenidos:

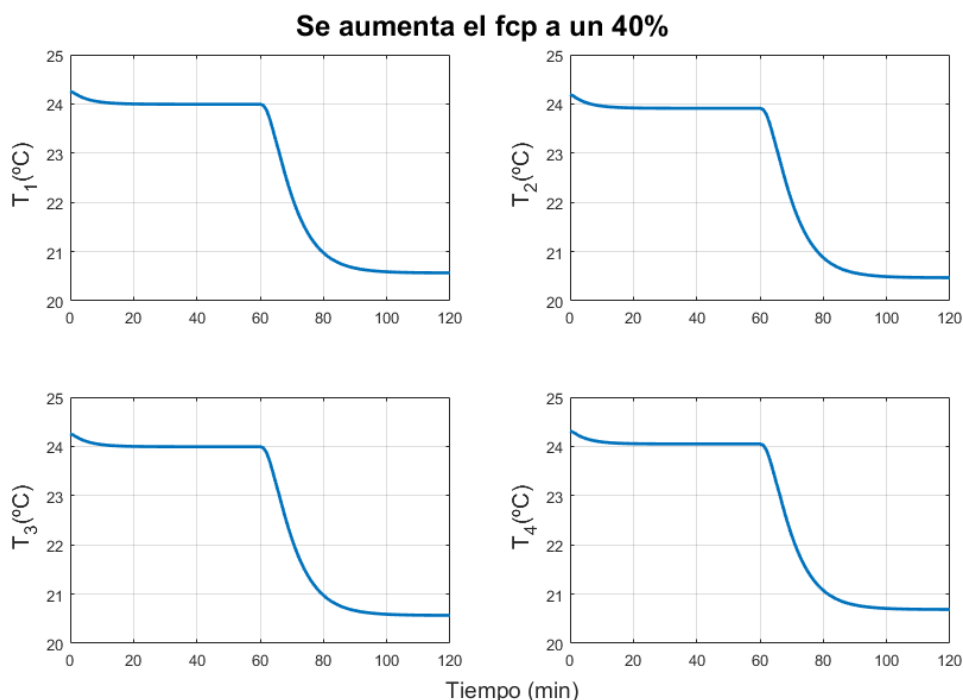
- La primera conclusión que se puede obtener con una vista rápida sobre las gráficas, se encuentra en que se puede afirmar que, cuando la apertura de una válvula se modifica hacia un valor superior (se abre más la válvula), la temperatura de la sala o espacio disminuye, mientras que si esta se cierra se produce un aumento de la temperatura. Explicar este fenómeno de forma física no es complicado,



pues lo que realmente ocurre cuando se produce una modificación en alguna de las aperturas, es una variación del caudal que atraviesa el intercambiador, de forma que la potencia que absorbe el fluido será mayor cuanto mayor sea la apertura y viceversa.

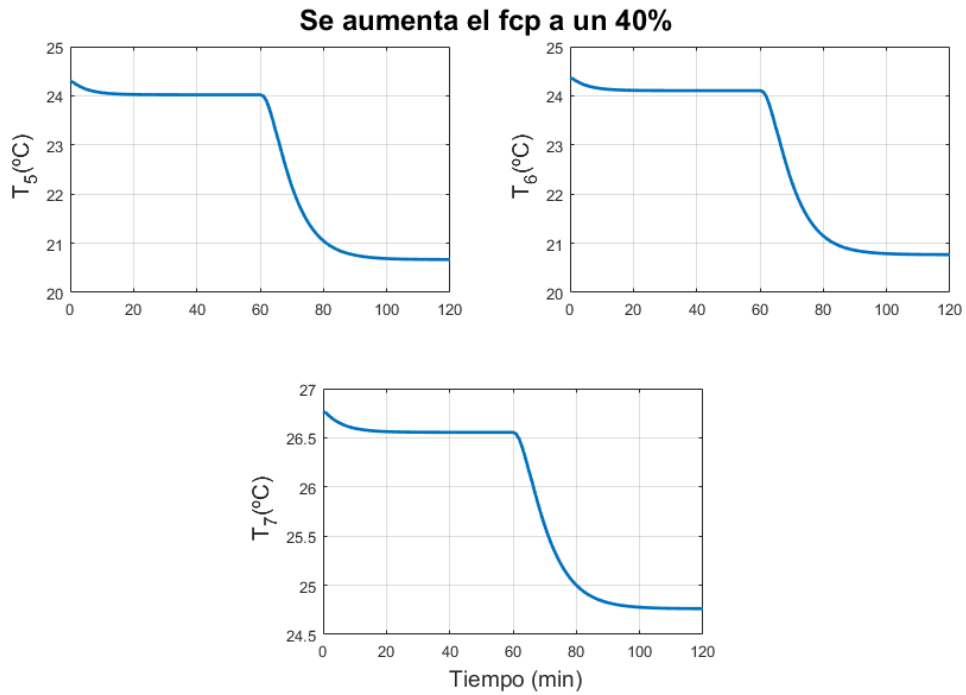
- Si uno se centra en lo que pasa en el caso concreto de la sala 5 (Figura 3.6), en el cual la apertura se mantiene constante, se puede observar que, a pesar de ello, la temperatura de la sala aumenta, aunque en menor medida que cuando se produce una modificación. Esto viene causado porque en la simulación se produce un cambio en algunas de las aperturas en el mismo instante de tiempo, modificando entonces el valor de la temperatura del agua que entra en el fancoil, ya que, aunque el  $f_{cp}$  se mantiene constante (la capacidad de enfriamiento es la misma), la temperatura a la salida de los intercambiadores ya no es la misma, por lo que tampoco será la misma a la entrada de la enfriadora, generando así una temperatura distinta a la salida de este, causado todo ello por el principio de conservación de la energía, ya que si la máquina enfriadora genera la misma potencia frigorífica en todo momento, el que una sala se lleve más potencia implica que se la quita a la otras.
- La última nota aclaratoria que sería necesario realizar, se puede centrar en las sobreoscilaciones que se producen en algunos casos cuando se aumenta la apertura de las válvulas, como ocurre en el caso de la sala 1 (Figura 3.5), donde se puede apreciar una pequeña sobreoscilación. Este fenómeno se produce debido a que en las simulaciones, TRNSYS autoajusta los valores mediante funciones de transferencia de primer orden cuando los parámetros no sufren muchas variaciones. Como se puede ver, las aperturas permanecen constantes durante casi toda la simulación excepto cuando se produce el cambio, por lo que tiende a ajustar las curvas. Cuando se produce así el cambio, justo en el instante anterior, el sistema vuelve a ajustar, generando así una pequeña sobreoscilación que es corregida posteriormente.

Para comprobar que ocurre cuando se modifica el valor del  $f_{cp}$ , se realiza otra simulación en la que, manteniendo el valor de las aperturas constantes, se produce una única variación en el valor del  $f_{cp}$ . Los resultados pueden observarse en las Figuras 3.5 y 3.6.



**Figura 3.7** Resultados obtenidos tras la modificación del  $f_{cp}$  (Salas 1 a 4).

En las simulaciones que se han realizado variando los valores que toma el  $f_{cp}$ , se observa que, cuando dicho valor se aumenta, se puede ver que las temperaturas de las habitaciones disminuyen. Esto puede explicarse si se entiende cual es la función de la enfriadora, ya que cuando se produce un aumento en el  $f_{cp}$ , la capacidad de la enfriadora aumenta, y por tanto la temperatura a la que sale el agua de la enfriadora disminuye, logrando



**Figura 3.8** Resultados obtenidos tras la modificación del fcp (Salas 5 a 7).

así, que la potencia absorbida por el agua tras su paso por el fancoil también aumente, generando por tanto una disminución en la temperatura de la sala.

También se puede destacar que el enfriamiento producido en la sala 7, que es la que representa al pasillo, es inferior al del resto de salas, ya que el espacio que representa el pasillo es superior al de las salas, por lo que este necesita mayor potencia para lograr el mismo enfriamiento.

## 4 Identificación del sistema

---

Uno de los pasos más importantes para poder lograr una correcta caracterización del sistema corresponde a la identificación del mismo, ya que los modelos térmicos del edificio son muy complejos por tener muchos factores a tener en cuenta y obtener ecuaciones físicas de ellos es muy difícil. Sin embargo, para las técnicas de control que van a ser empleadas es necesario disponer de estas ecuaciones para tener un conocimiento del sistema. Es por ello que se quiere realizar una identificación en espacio de estados que explique la dinámica del sistema [10], el cual puede definirse como un sistema de ecuaciones diferenciales que se apoyan en el uso de un vector de estados. El correspondiente modelo deberá presentar una forma como

$$x(k+1) = Ax(k) + Bu(k) \quad (4.1a)$$

$$y(k) = Cx(k), \quad (4.1b)$$

donde  $x \in \mathbb{R}^{nx}$  representa el vector de estados,  $u \in \mathbb{R}^{mu}$  al conjunto de variables de control e  $y \in \mathbb{R}^{my}$  representa las salidas del sistema. Para poder llevar a cabo dicha identificación, se va a hacer uso de un paquete de identificación de MATLAB<sup>®</sup>. Para ello, se debe definir cuáles son las entradas y salidas del sistema en cuestión, preparar las pruebas necesarias para lograr la identificación, y por último, aplicar las herramientas necesarias para lograr el objetivo en cuestión.

### 4.1 Definición de entradas y salidas

Antes de poder proceder con la identificación de un sistema cualquiera, es importante saber que variables del sistema van a ser consideradas como entradas y cuales como salidas, ya que esto define las dimensiones de cada uno de los vectores y matrices que caracterizan un espacio de estados.

Para el caso de estudio de este proyecto, se parte de un sistema al cual se le quieren implementar distintos tipos de estrategias de control para poder obtener referencias deseadas en los valores de las temperaturas de cada una de las salas, siendo en este caso concreto un total de siete salas. Para poder aportar la energía capaz de producir un enfriamiento en dichas salas, se hace uso de un equipo frigorífico compuesto de una enfriadora, un intercambiador o fancoil en cada una de las salas, siete válvulas que modelan el flujo de fluido por cada uno de los intercambiadores, y una mezcladora que permite la unión de todos los flujos para poder impulsarlos de nuevo en el chiller o enfriadora. Cada uno de estos equipos poseen distintas variables que permiten modelar el comportamiento del mismo así como la potencia que se aporta a cada una de las salas.

En este caso se toman como parámetros de control las válvulas que modelan el caudal que circula por cada fancoil, y el *fcp* que caracteriza a la enfriadora, el cual permite un mayor o menor enfriamiento del fluido que lo atraviesa. De esta forma, se puede definir un bloque llamado sistema que englobe tanto el modelo del edificio como el del conjunto de equipo que constituyen la máquina enfriadora. Dicho sistema, se puede modelar mediante las variables de entradas definidas, siendo estas un total de ocho, las cuales generan distintas salidas. Todo esto que acaba de ser enunciado, es lo que se puede extraer de la Figura 4.1.

Con esto, ya se puede empezar a dimensionar algunos de los elementos que constituyen el modelo de espacio de estados. Más concretamente, el vector  $y$  estará compuesto de un total de 7 componentes que representan las temperaturas; el vector  $u$  es el relativo a las variables de control, y por tanto almacena las aperturas (de la componente uno a la siete) y el *fcp* (en la última componente) en un total de 8 componentes;

el vector de estados  $x$  todavía no se le puede determinar su dimensión exacta, pues tendrá que ser ajustado para obtener los mejores resultados posibles.

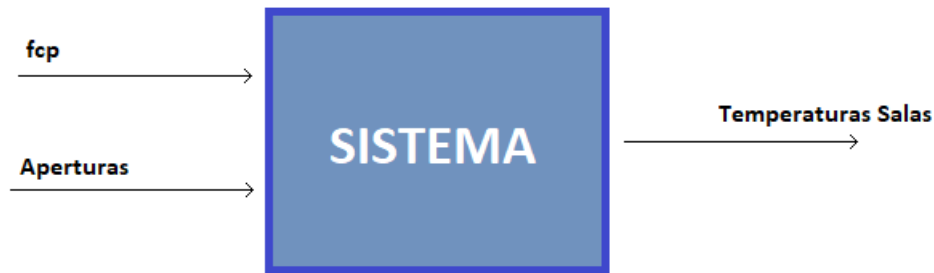


Figura 4.1 Esquema entradas y salidas del sistema.

## 4.2 Procedimiento identificación

Para poder realizar la identificación correspondiente al sistema, se debe hacer uso de un toolbox de identificación disponible en MATLAB<sup>®</sup>, más concretamente el denominado como *System Identification Toolbox*. Antes de entrar a conocer debe usarse dicho paquete, hay que saber qué tipo de datos y la forma que deben adoptar estos para que la identificación que se produzca sea exitosa.

El objetivo es poder aportar unos datos al *toolbox* en el que se pueda ver cómo afecta la modificación de cada una de las entradas sobre cada una de las salidas del sistema. Para ello, se pretende realizar tantas simulaciones como número total de entradas se dispongan. En cada una de esas simulaciones se debe ir generando un escalón sobre cada una de las entradas para comprobar así qué efecto produce esa entrada sobre cada una de las salidas. En este tipo de pruebas, las entradas que permanecen constantes lo hacen sobre un valor, siendo este el punto de funcionamiento. De esta forma la identificación obtenida es considerada como un sistema lineal en espacio de estados en torno a un punto de operación.

Dicho esto, se puede comenzar a realizar cada una de las pruebas en cuestión, las cuales deben implementarse en TRNSYS mediante el esquema visible en la Figura 4.2. El modelado de cada una de las entradas en todas las pruebas se realiza mediante MATLAB<sup>®</sup>, el cual genera unos valores de aperturas y  $fcp$  que son introducidos en el modelo del equipo enfriador y este genera cada una de las potencias cedidas en las habitaciones del edificio.

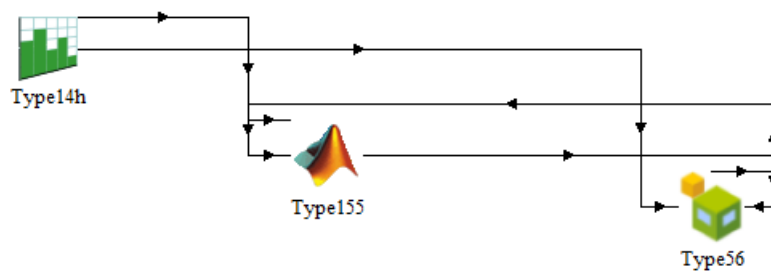


Figura 4.2 Esquema obtención de datos en TRNSYS.

Para poder realizar esta identificación se va a considerar una temperatura exterior del ambiente constante, de forma que cuando el tiempo atmosférico sea introducido, este sea considerado como una perturbación no conocida del sistema. Otros de los parámetros que deben ser modelados en el sistema son los puntos de operación elegidos para cada una de las salidas, los cuales han tomado un valor de 0.3 para las aperturas de las válvulas y 0.1 para el factor de carga parcial. El momento en el que se produce un cambio en las entradas del sistema será justo en la mitad de la simulación, momento en el que se produce un cambio en las entradas del sistema es cuando el sistema se ha estabilizado correctamente antes de producir nuevas variaciones.

Para ilustrar todo esto desde otra perspectiva, el código propuesto para modelar los distintos escalones y general variaciones sobre cada una de las salidas es:

```

%% Se diseñan las entradas para el modelado
%% Se haran 8 pruebas, donde se modificaran las entradas del sistema
Apertura = 0.3*ones(7,nTimeSteps+1);
fcp = 0.1*ones(1,nTimeSteps+1);

%% Establecemos 14 pruebas %%
% DESCOMENTAR LO QUE PROCEDA para cada prueba %
Apertura(1,:) = [0.3*ones(1,round((nTimeSteps+1)/2)) 0.6*ones(1,nTimeSteps-
    round((nTimeSteps+1)/2-1))]; % Prueba 1
% Apertura(2,:) = [0.3*ones(1,round((nTimeSteps+1)/2)) 0.6*ones(1,nTimeSteps-
    round((nTimeSteps+1)/2-1))]; % Prueba 2
% Igual hasta la prueba 8

```

Cuando se dispone de cada una de las entradas definidas, ya solo es necesario introducir el valor que toman cada una de ellas en un tiempo de muestreo en el modelo de la máquina que se ha implementado en el Capítulo 3 y almacenando cada una de las temperaturas de las salas del edificio, que son las salidas del sistema. Para poder almacenar los datos y disponer de ellos cuando se desee, se puede hacer uso del comando `save` de MATLAB®, el cual permite almacenar algunas o todas las variables que están disponibles en el *workspace*, tal y como se muestra a continuación:

```

%% Se almacenan los resultados de la simulación %%
save('prueba_1','time','Apertura','fcp','temperatura_habitaciones');

```

Esta línea de código debe ejecutarse cuando ya se ha acabado de realizar la simulación, ya que será cuando se disponga al completo de todos los datos obtenidos. Estas simulaciones se tienen que realizar tantas veces como número de entradas tenga el sistema, siendo para este caso concreto un número igual a ocho. De esta forma se deben disponer de un total de ocho resultados en los que se dispongan de todas las variables de entradas y sus correspondientes influencias en cada una de las salidas. Con estos resultados, ya se puede proceder a realizar la identificación de un espacio de estados del sistema completo.

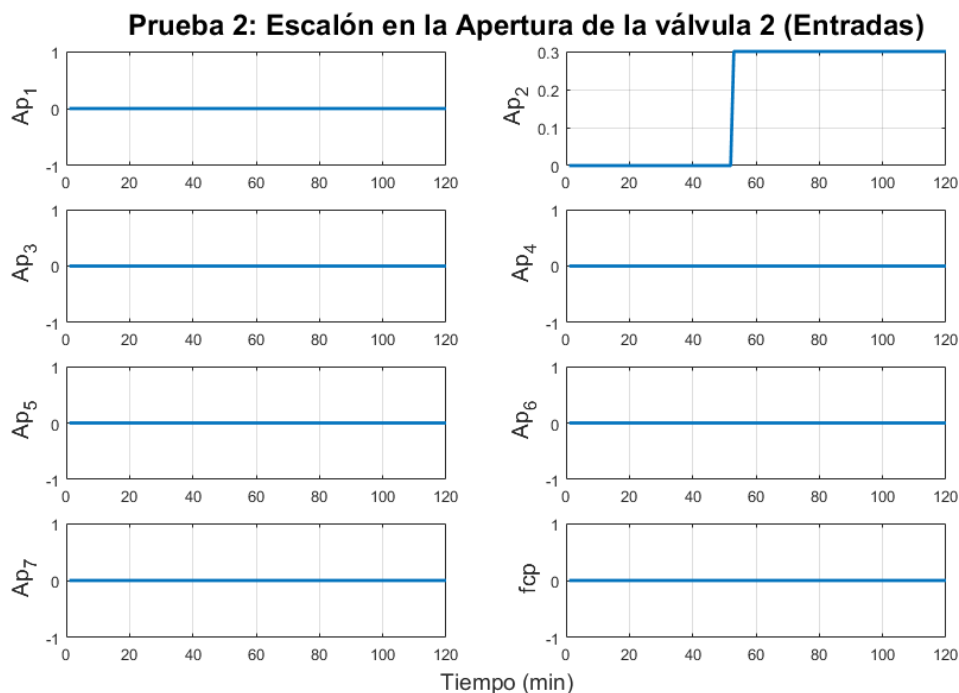
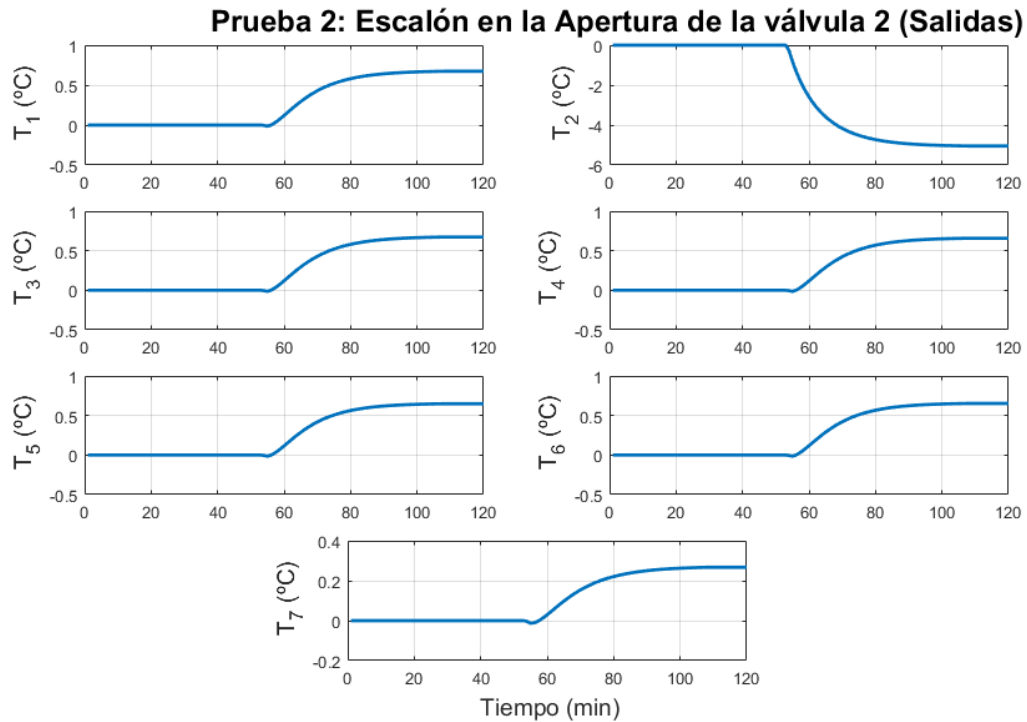


Figura 4.3 Entradas del sistema de la Prueba 2 .

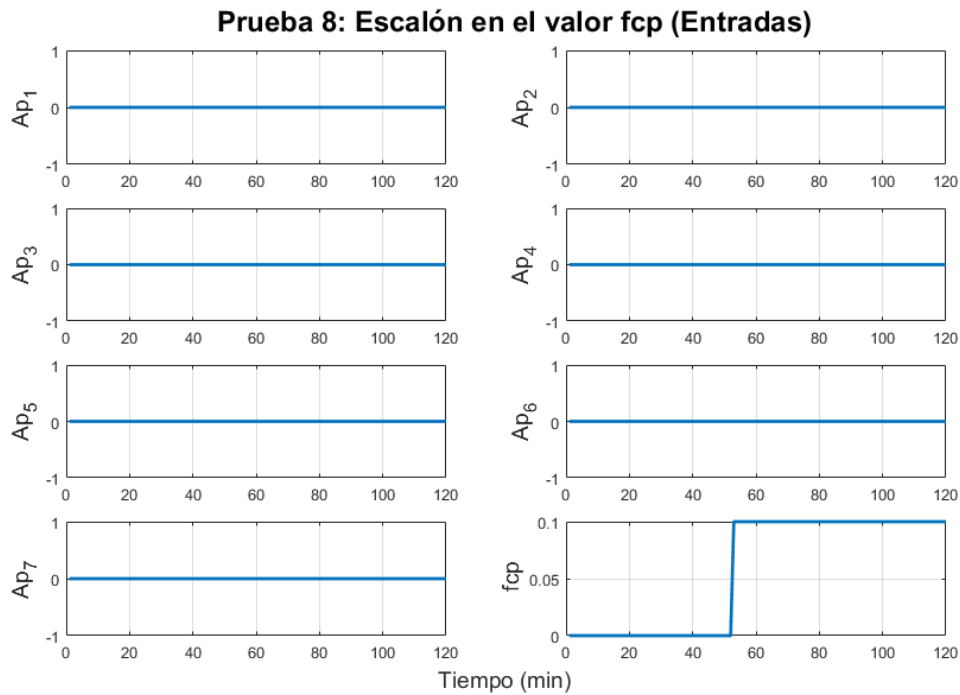
Los datos de partida para realizar la identificación, como ya se ha dicho son un total de ocho pruebas realizadas, las cuales deben ser cargadas una a una en MATLAB®. Los datos cargados, son los correspondientes a la simulación completa realizado con TRNSYS, por lo que en ellos hay algunos periodos de la simulación que no son de interés puesto que en los primeros tiempos se debe alcanzar un régimen permanente para los puntos de operación. Esto hecho genera la necesidad de recortar los valores de los vectores de entrada y salida del sistema. Por otro lado, para lograr una identificación en torno al punto de equilibrio, los valores de las entradas y salidas deben estar centrados en cero, de forma que habrá que restarle el punto de operación a las entradas y salidas del sistema. El punto de operación de las salidas puede ser considerado como el valor que toma la primera componente del vector de las temperaturas tras haber realizado la selección necesaria. A modo de ejemplo se van a mostrar dos de las pruebas tanto con las entradas y salidas tras haber realizado la selección y el centrado necesario.



**Figura 4.4** Salidas del sistema de la Prueba 2 .

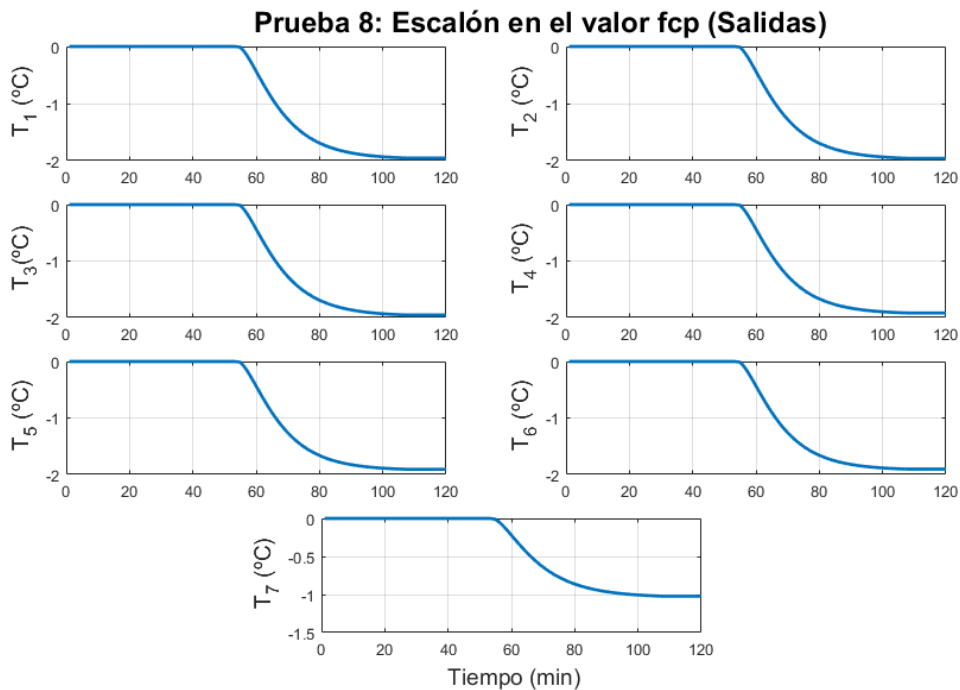
En las Figuras 4.3 y 4.4 se ilustran los resultados obtenidos tras la realización de la prueba 2, en la que el escalón de las entradas se produce en la apertura de la segunda válvula y se ve como el resto se mantienen completamente constante durante el resto del tiempo mostrado. Para comprobar que las modificaciones que se producen en las salidas son coherentes, se ve como la temperatura de la sala 2 baja algunos grados y el resto sufren un pequeño aumento, lo cual es lógico porque la temperatura a la que sale el agua del segundo intercambiador es algo superior a la de los instantes anteriores y como el fcp permanece constante, la capacidad de enfriamiento es la misma. Por lo que la potencia aportada a cada sala disminuye, produciendo así un aumento de temperatura.

Por otro lado, en las Figuras 4.5 y 4.6 se muestran los resultados relativos a la última de las pruebas a realizar en la que se produce un escalón en el factor de carga parcial (fcp) de la enfriadora manteniendo todas las aperturas en el valor del punto de operación que se ha establecido. Con respecto al resultado obtenido, para poder entenderlo, hay que saber que cuando se produce un aumento en el valor del fcp lo que se produce es un aumento de la capacidad de enfriamiento del *chiller*, es decir, una disminución de la temperatura del agua cuando llega a la salida de éste. De esta forma aunque el caudal de agua que circula por cada intercambiador no varía (no se modifica ninguna de las aperturas), si lo hace la temperatura del fluido, logrando así que el intercambio energético que se produce entre la sala y el fancoil sea superior a los instantes anteriores, logrando así que la temperatura de cada una de las salas sufra un descenso siendo menor en el pasillo por las dimensiones de éste.



**Figura 4.5** Entradas del sistema de la Prueba 8 .

A la vista de los resultados obtenidos tras la selección correcta de los datos de interés y su correspondiente centrado en torno al punto de equilibrio, ya se está en disposición de hacer uso del toolbox de identificación disponible en MATLAB<sup>®</sup>. Con la respectiva explicación de cómo hacer uso de él, se va a ir explicando cada una de las funciones de interés y como hacer un uso correcto de ellas. Dichas funciones deben ser aplicadas sobre los datos que se han obtenidos hasta ahora.



**Figura 4.6** Salidas del sistema de la Prueba 8 .

Para poder crear un modelo en espacio de estados, se deben encapsular todos los datos de las medidas del sistema que se quieren identificar, creando para ello un vector fila cuyas variables son de tipo *IDdata*, donde cada una de las componentes almacena las entradas y salidas de cada una de las pruebas realizadas, por lo que en caso en cuestión se creará uno de ocho componentes. Para ello basta con hacer uso de uno de los comandos de los que se disponen en el *toolbox*, más concretamente la función *iddata*, a la cual hay que pasarle un vector columna con todas las salidas obtenidas, otro vector columna con las diferentes entradas y, por último, hay que indicar que son datos discreto con un tiempo de muestreo igual a uno. De esta forma, la función debe presentar el esquema siguiente:

$$IDdata\{i\} = iddata(Salidas, Entradas, 1).$$

De esta forma, se dispone de un vector formado por objetos de tipo *iddata*, los cuales deben ser integrados en un único objeto de tipo *iddata*, el cual albergue todos los datos de las pruebas realizadas. Para ello, se dispone de una función conocida con el nombre de *merge*, del inglés unir, la cual permite realizar dicha acción. Para ello solo es necesario pasar como argumentos cada una de las componentes del vector construido anteriormente

$$IDmerged = merge(IDdat\{1 : nexp\}).$$

Por último, ya solo falta obtener el espacio de estados resultante de la identificación, para lo cual el *toolbox* dispone de una función que hace dicha identificación. La función a la que se debe llamar es *ssest* que es capaz de estimar un espacio de estados tanto en discreto como continuo para un número de estados determinado a partir de los datos aportados. Uno de los parámetros de diseño es el número de estados a usar. Para poder estimar dicho número MATLAB<sup>®</sup> dispone de un asistente que te asigna a priori la mejor opción llamando a la función

$$ssmodel7 = ssest(IDmerged, 7 : 14, 'Ts', 1),$$

donde los argumentos de entrada son el conjunto de datos agrupados en un único objeto, los estados entre los que se desea buscar mejor opción y se debe indicar que el sistema a obtener se quiere en tiempo discreto con un tiempo de muestreo igual a uno. Si se prueba con unos valores que varíen entre uno y catorce estados, se obtienen los resultados de la Figura 4.7. En ella se puede observar que la mejor de las opciones propuesta es la de ocho estados, aunque esta habría que compararla con las versiones de siete y ocho, pues los resultados que se obtienen haciendo esto generan una primera idea, la cual no debe ser tomada como definitiva y se debe perfilar un poco más.



**Figura 4.7** Resultados obtenidos para decidir el número de estados a considerar.

Tras realizar pruebas para los tres valores propuestos, se llega a la conclusión de que el mejor de los resultados que se obtienen es para un vector de estados formado por ocho componentes. A continuación, se van a analizar los resultados que se han obtenido.



### 4.3 Resultados obtenidos

Como se acaba de decir ya se ha acordado que el espacio de estados que va a ser usado es el compuesto de un vector de ocho estados, pues este es el que mejor se comporta y modela el comportamiento real. Dicho modelo es un sistema de ecuaciones como

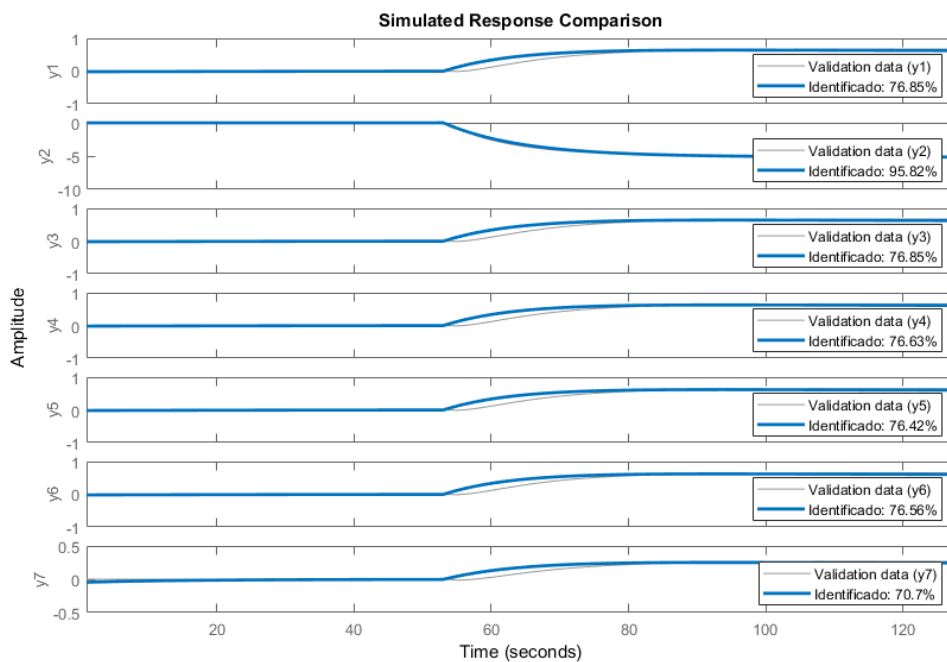
$$x(k+1) = Ax(k) + Bu(k) \quad (4.2a)$$

$$y(k) = Cx(k), \quad (4.2b)$$

en el cual ya se pueden caracterizar cada una de las componentes: donde  $x \in \mathbb{R}^8$ ,  $u \in \mathbb{R}^8$  e  $y \in \mathbb{R}^7$ , de forma que las matrices  $A \in \mathbb{R}^{8 \times 8}$ ,  $B \in \mathbb{R}^{8 \times 8}$  y  $C \in \mathbb{R}^{7 \times 8}$ , lo cual permite definir por completo el modelo matemático del sistema.

Para poder afirmar que la identificación realizada se comporta de forma similar a como lo realizaría el sistema, se van a representar algunas comparaciones entre el sistema identificado y los datos reales. Además, una forma de observar el comportamiento del sistema en espacio de estados obtenidos se puede visualizar aplicando un escalón sobre cada una de las entradas y ver cómo afecta a cada una de las salidas.

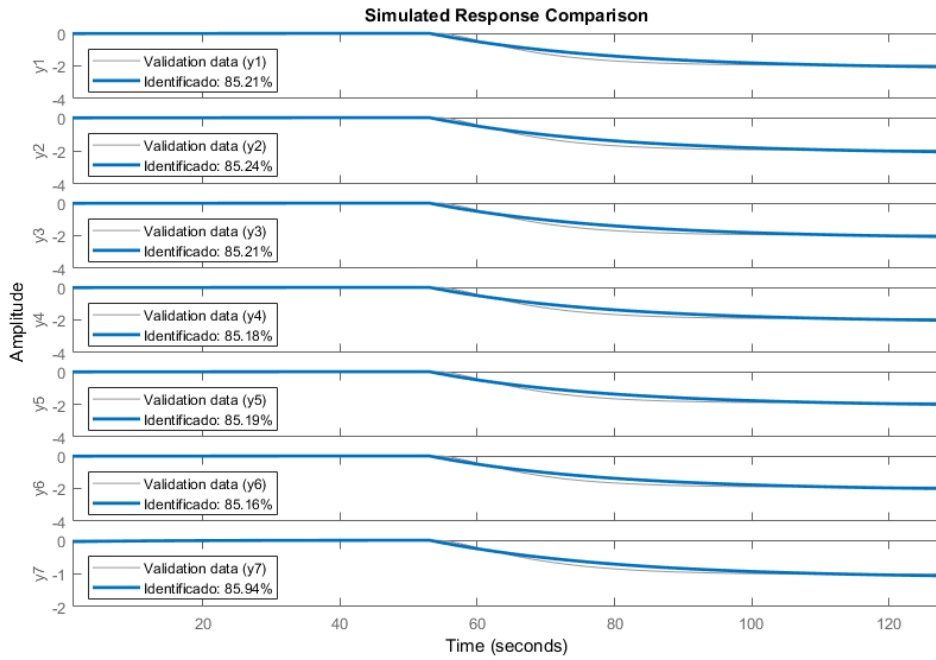
Para comenzar con el análisis en la Figura 4.8 se muestra la comparación entre los resultados obtenidos tras la identificación de la prueba 2, en la que el escalón se produce en la apertura de la segunda válvula. Con los resultados que se observan en la imagen, se puede afirmar que son buenos, ya que para todas las salidas el régimen permanente se ajusta por completo y en la cual se produce el escalón, el ajuste es prácticamente ideal. El transitorio de las otras salidas no se ajusta tan bien debido a que presenta un cero de fase no mínima, los cuales no pueden identificarse con los estados establecidos, de todas formas esto no causa ningún inconveniente ya que interesa la constante de tiempo y el régimen permanente (sin error).



**Figura 4.8** Resultados en la identificación de la Prueba 2.

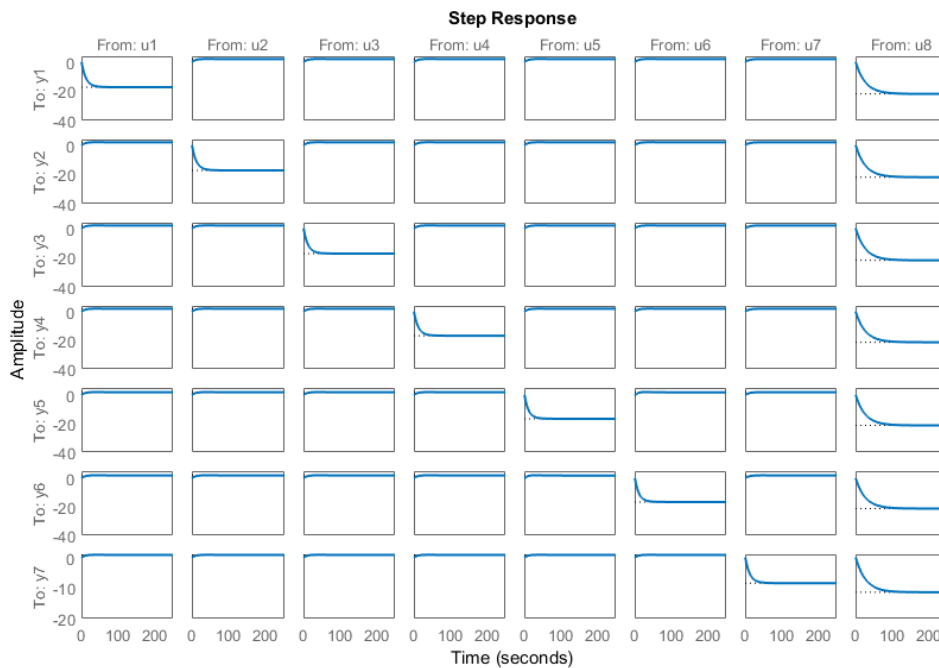
Por otro lado, en la Figura 4.9, se pueden ver los resultados obtenidos para la prueba 8 en comparación con los datos reales. En ellos se puede observar que el ajuste en régimen permanente es casi perfecto y durante el transitorio es ajuste que se produce casi que se ajusta al valor real.

Con estos resultados, se puede afirmar que el modelo obtenido es bueno, ya que los resultados restantes que no se han mostrado en esta memoria por cuestión de espacio, son muy similares a los obtenidos en el caso de la prueba dos, ya que el comportamiento de las variaciones en las aperturas de las válvulas es muy similar en las habitaciones, así como su influencia sobre las demás salas.



**Figura 4.9** Resultados en la identificación de la Prueba 8.

Por último, en la Figura 4.10 se observa el resultado de aplicar un escalón en el tiempo 0 sobre cada una de las entradas que definen el sistema y se puede ver cómo afecta a cada una de las entradas. Esta información es muy ilustrativa para conocer la influencia de cada una de las variables sobre cada una de las salidas. Para cada una de las válvulas ( $u_1$  hasta  $u_7$ ) se ve que la influencia se produce principalmente sobre la sala donde se encuentra, mientras que  $f_{cp}$  ( $u_8$ ) afecta a todas de una forma más general.



**Figura 4.10** Resultados tras la aplicación de escalones unitarios en el espacio de estados.

# 5 Simulador de Perturbaciones

---

Para aportar un toque mucho más realista y funcional a la simulación, en este capítulo se va a elaborar un simulador que aporte la capacidad de incorporar al sistema algunas perturbaciones que modificarán las temperaturas de las salas existentes. Para ello, se pretende añadir la posibilidad de que sean consideradas las ganancias térmicas que producen la existencia de personas, luces e incluso ordenadores en cada uno de los espacios.

Adicionalmente, se pretende también incorporar un modelo matemático que permita facilitar el cálculo de las fluctuaciones de aire que se producen cuando se abre una puerta que comunica dos de los espacios disponibles. Dicho esto, en el capítulo en cuestión se va a abordar la definición de cada uno de los objetivos a lograr, la implementación de todos ellos tanto en TRNSYS como en MATLAB® y por último, como se viene haciendo hasta ahora, aunar todos los conocimientos expuestos con el fin de poder presentar los resultados obtenidos.

## 5.1 Objetivos del Simulador

Antes de comenzar con las implementaciones de los parámetros en las herramientas disponibles para ello, es conveniente que se limiten, acoten y definan los objetivos a cumplir por el simulador. Se quiere conseguir uno lo más realista posible en cuanto a la ocupación del edificio, por lo que se incorporará la posibilidad de considerar ganancias como son las correspondientes a personas, luces, ordenadores y aperturas de puertas. Para ello, a continuación, se definirá cada una de ellas.

En primer lugar, se comenzará con las personas, las cuales desarrollan su trabajo en el edificio. Estas, al igual que ocurre en el caso de una oficina, tendrán diversos turnos de llegada, dependiendo del día y la hora, de forma que se estima que puedan llegar entre las 6:00 y las 11:00. En cada sala o despacho, es posible que haya entre 0 y 3 personas en cada momento, siempre destacando que todas ellas están trabajando. Para abandonar las salas, y por tanto el edificio, al igual que ocurre con las llegadas se estima que el horario de salidas se encuentra entre las 14:00 y 18:00.

Con respecto a las luces, en este sentido es mucho más sencillo, puesto que cómo es lógico, siempre estarán encendidas cuando haya una persona en la sala en cuestión, y se apagarán mientras que dicha sala no se encuentre ocupada.

Otro de los parámetros que se deben establecer, es el correspondiente al posible uso que haga cada uno de los trabajadores de los ordenadores disponibles en las salas. Para ello, se establece que cuando una persona llegue a cada una de las salas, pueda o no encender el ordenador (de forma aleatoria) y este se mantendrá encendido hasta que una persona abandone la sala y pueda apagarlo.

Por último, en relación a las infiltraciones, lo que se pretende es obtener una cuantificación de las perturbaciones que producen las aperturas de puertas cuando una persona entra o sale de cada una de las salas. Esta infiltración provoca un contacto de dos masas de aire que se encuentran a diferente temperatura, lo cual produce un intercambio energético que es interesante modelar.

En este subapartado, se han definido todos los parámetros de una forma muy genérica para que se pueda ir creando una base de trabajo que después va a ser profundizada y ampliada cuando se proceda a la implementación en las herramientas de trabajo.

En forma de resumen, los aspectos más importantes de todas las variables que van a ser consideradas, pueden ilustrarse en la Tabla 5.1

Tabla 5.1 Resumen de parámetros a considerar..

Variable	Situación
Personas	Puede haber de 1 a 3 con horario de entrada y salida
Luces	Se encuentran encendidas mientras haya una persona en la sala
Ordenadores	Las personas pueden encenderlo aleatoriamente a su llegada
Infiltraciones	Permiten modelar apertura de puertas

## 5.2 Implementación en TRNSYS

Para comenzar un poco con la aplicación del caso de estudio, se va a proceder a la implementación y definición de todos los parámetros expuestos anteriormente, con el fin de poder disponer de ellos en la herramienta TRNSYS. Una de las ventajas que presenta este software es que permite la posibilidad de definir una serie de ganancias térmicas entre las que se encuentran las deseadas. De esta forma, se va a estudiar cómo se han de definir cada una de ellas de forma correcta.

Para comenzar, se partirá con la definición de la variable que indique el número de personas que haya en cada una de las salas. Para ello, en la herramienta TRNBuild [20], una vez se han creado las distintas zonas existentes en el edificio (objeto de estudio en Capítulo 2), se debe seleccionar una de ellas para comenzar la definición. Una vez aparece en detalle la ventana de las propiedades de la sala, se debe hacer click sobre la pestaña llamada *Gains*, marcada con el dibujo de una vela. Una vez en ella, aparecen una serie de divisiones que permiten añadir distintos tipos de ganancias, donde en este caso concreto, hay que centrarse en la denotada como *Persons*. En ella inicialmente se puede observar marcado un icono que indica *Off*, si este es pulsado aparecen nuevos cuadros de dialogo que son que se deben modificar.

En las casillas emergentes, se puede realizar una selección entre la norma ISO-7730 y la guía VDI-2078. Para este caso, se selecciona la norma ISO, en la cual se trata tanto la ergonomía como el bienestar del ambiente térmico. Justo al lado de dicha norma, se puede observar un botón llamado *Table*, al cual hay que acceder pulsando sobre él. En la ventana que aparece, se pueden observar distintas situaciones de las personas que se encuentran en el interior del espacio de cada sala. Para el caso de estudio se escogerá la situación número 4, en la cual dichas personas se encuentran trabajando (similar a oficinas). Con esto, ya se tiene definida el tipo de ganancia, aunque todavía es necesario definir el nombre de la variable que calculada mediante MATLAB®, será introducida en TRNSYS e indicará el número de personas que se encuentran en cada sala.

Esto se puede realizar pulsando sobre la flecha verde que aparece justo en el margen derecho de lo que es denominado como *scale*. En el desplegable que aparece, se permite seleccionar que dicha variable va a ser considerada como una entrada (*Input*), por lo que es necesario elegir el nombre de la variable entre uno de los existentes si este ha sido creado con anterioridad, y si no ha sido así, es necesaria su creación.

Para comprobar que todo ha sido realizado correctamente, el cuadro resultante debe ser igual o similar al representado en la Figura 5.1. Todo este procedimiento tiene que realizarse para cada una de las salas que contemple la posibilidad de personas en ellas, prestando especial cuidado en el uso de una variable distinta en cada una de ellas.

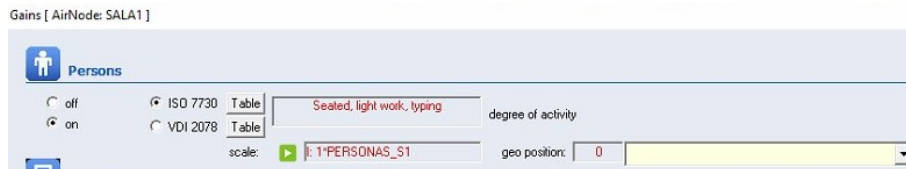


Figura 5.1 Cuadro resultante tras la implementación de personas en TRNSYS.

Para proceder con la implementación de las ganancias ocasionadas por las luces se debe acceder a la pestaña *Gains* que aparece sobre la ventana de las propiedades en cada espacio disponible. En ella, se observa que una de las divisiones que aparecen corresponde a la iluminación (*Artificial Lighting*). Para poder definirla, se debe habilitar dicha división, lo cual genera una serie de cuadros para indicar ciertos parámetros acerca de esta. El primero a rellenar es el correspondiente al *related floor area*, en el cual se debe indicar el área del espacio sobre el que se está aplicando dicha iluminación, que en el caso de estudio, es  $16m^2$  para los despachos y  $42m^2$  en el caso del pasillo.

Para proseguir con la definición se debe elegir una ganancia para bombilla a utilizar, la cual ha sido elegida de unos  $16 \text{ W}/\text{m}^2$ . Otros de los parámetros a seleccionar es la parte convectiva de la ganancia establecida, que ha sido elegida como si el tipo de bombilla a utilizar fuese un tubo fluorescente, en el cual un 40% de la ganancia establecida es de tipo convectivo. Por último, se debe definir la variable a utilizar, de forma que al igual que ocurre con las personas, esta sea una entrada del bloque del edificio aportada mediante MATLAB<sup>®</sup>, pulsando para ello sobre las flechas verdes que aparecen y seleccionando tipo *Input*, y posteriormente su variable, la cual debe crearse si no se dispone de ella.

El resultado que se debe obtener si todo se ha realizado de forma correcta, es visible en la Figura 5.2. Como ocurre con las personas, todo esto debe realizarse para todos los espacios donde se quiera considerar la iluminación.



**Figura 5.2** Cuadro resultante tras la implementación de las luces en TRNSYS.

Dentro de la ventana *Gains*, el otro parámetro que se puede definir es el correspondiente a los ordenadores disponibles en cada sala. Para ello, en la división correspondiente a *Computer*, se debe habilitar la casilla para poder implementarlo. Cuando se ha producido la habilitación, se puede seleccionar entre varias opciones el tipo de ordenador a implementar, qué en el caso de estudio, el escogido ha sido uno de 230 W compuesto de un PC y un monitor de color. La implementación de los ordenadores como se puede ver es algo más sencilla que en el resto de los casos, pues ya solo es necesario establecer la variable de entrada al igual que se ha hecho con los parámetros anteriores. De esta forma, el resultado obtenido es el que puede verse en la Figura 5.3.



**Figura 5.3** Cuadro resultante tras la implementación de los ordenadores en TRNSYS.

Visto esto, ya solo es necesario definir cómo se van a implementar las infiltraciones de aire que se producen con las aperturas de las puertas de cada una de las salas. La decisión de como implementarlo no ha sido una tarea fácil, pues de esta forma, hay buscar una forma matemática para poder cuantificarlas e implementarlas.

La forma escogida se basa en interpretar que el intercambio energético que se produce se logra gracias a una infiltración de aire a través de la puerta que comunica los dos espacios en cuestión. Como se ve en el Capítulo 2, las puertas se han definido como ventanas, entonces en el cuadro de definición de dicha ventana, cabe la posibilidad de elegir un caudal de infiltración, mediante el campo *coupling air flow*, en el cual se indica que cantidad de aire pasa de un lado a otro. Para poder usar este campo como si se tratase de una infiltración, se deben definir las variables que van a ser calculadas con el programa de MATLAB<sup>®</sup> y que caractericen el fenómeno en cuestión. Para ello, se han de definir como entradas al bloque del edificio.

Como se ha explicado hasta ahora, el sistema de definición de las perturbaciones en TRNSYS, no es más que una definición de variables de entrada al archivo creado por TRNBuild y cuyos valores serán aportado por MATLAB<sup>®</sup>. En el caso concreto de estudio los nombres de las variables que han sido utilizadas, pueden verse recogidos en la Tabla 5.2

**Tabla 5.2** Variables definidas en TRNBuil.

Variable	Nombre utilizado
Personas	$PERSONAS_{S1}, PERSONAS_{S2} \dots PERSONAS_{S7}$
Luces	$LUZ_{S1}, LUZ_{S2} \dots LUZ_{S7}$
Ordenadores	$PC_{S1}, PC_{S2} \dots PC_{S7}$
Infiltraciones a Salas	$INFILTR_{S1}, INFILTR_{S2} \dots INFILTR_{S7}$
Infiltraciones a Pasillo	$INFILTR_{P1}, INFILTR_{P2} \dots INFILTR_{P7}$

### 5.3 Implementación en MATLAB®

Como se ha visto, es necesario definir cada una de las variables creadas en TRNSYS, en un código de MATLAB®, el cual intentará asignar de una forma lo más genérica posibles valores a todas esas variables. En esta sección se va a abordar ese problema de forma más profunda y generalizando al máximo posible todas ellas. Al igual que se ha venido realizando hasta ahora, se va a dividir el trabajo por tipos de variables y explicando en profundidad cada una de ellas.

Antes de comenzar, para poder generalizar al máximo posible el código, se deben realizar una serie de cálculos que permitan obtener unas variables globales que serán posteriormente utilizadas para poder lograr dicho fin.

Una de estas variables, representa el número de días totales en los que se basa la simulación, mientras que otra de ellas permite saber a cuántos tiempos de muestreo equivale un día, para poder saber así en que día se encuentra la simulación. Para lograr dicho cálculo, se debe hacer uso de los parámetros internos de la simulación, que son los almacenados en las variables *trnStopTime*, *trnStartTime* y *trnTimeStep*. Además, también es preciso conocer el número de tiempos de muestreos de los que va a estar compuesta la simulación, para poder diseñar así algunos vectores en la etapa de inicialización. Para realizar el cálculo se supone que todos los parámetros tienen unidades de horas. De esta forma, el código empleado para ello es el siguiente:

```
% Se definen una serie de parámetros globales
nTimeSteps = (trnStopTime-trnStartTime)/trnTimeStep + 1;
numero_dias = ceil((trnStopTime-trnStartTime)/24);
dia = 24/trnTimeStep;
```

Otras de las modificaciones previas que se deben realizar si se quiere obtener un simulador completamente aleatorio es resetear el RandStream, para evitar que en diferentes simulaciones, todos los parámetros adquieran el mismo valor. Para ello, basta con ejecutar el siguiente comando:

```
% Se debe resetear la semilla del sistema
reset(RandStream.getGlobalStream,sum(100*clock))
```

Con estas respectivas inicializaciones, ya se puede proceder a la incorporación de una de las variables, el número de las personas que se encuentran en cada una de las salas (la explicación se va a llevar a cabo con una única sala para simplificarla, ya que todo es extrapolable al número deseado). Como ya se dijo en el primer apartado del capítulo se van a establecer unos posibles turnos de entrada y salida de personas en cada una de las salas, así como una hora de cierre general del edificio para que todo se apague cuando este se produzca. Estableciendo también la ocupación inicial de ellas en cero personas, el código que resulta es:

```
% Se establecen los horarios de llegada
primera_hora_llegada = 6/trnTimeStep-1;
ultima_hora_llegada = 11/trnTimeStep-1;

% Se establecen los horarios de salidas
primera_hora_salida = 14/trnTimeStep;
ultima_hora_salida = 18/trnTimeStep;
hora_cierre = 20/trnTimeStep;

% Se inicializa el numero de personas a cero en todas las salas
num_personas_sala1 = 0;
```

El objetivo ahora es poder establecer ahora tres posibles turnos de llegada aleatorios comprendidos entre las 6:00 y las 11:00. En esas posibles horas de llegada, puede llegar una persona o no, de forma que se dispongan de un vector que almacene las posibles horas (denotado como *llegada<sub>i</sub>*) y otro que indique si ha llegado o no una persona en cada turno (*num\_personas\_llegada<sub>i</sub>*). Como uno de los objetivos es poder generalizar al máximo posible el código, todo ello se debe realizar tantas veces como número de días se haya establecido en la simulación (usando para ello *todas\_llegada<sub>i</sub>* y *todas\_personas\_llegada<sub>i</sub>*). De esta forma, ya se dispone a qué hora pueden llegar las personas y si dicha persona llega o no.

De igual forma que se ha realizado con las llegadas, esto puede realizarse para las salidas. De forma análoga se crea  $salida_i$  que genera tres horarios de salidas distintos para cada sala (comprendidos entre los horarios de salida) y  $num\_personas\_salida_i$  que indica si se ha producido o no una salida en la hora aleatoria establecida. Para poder generalizarlo se emplean los vectores  $todas\_salida_i$  y  $todas\_personas\_salida_i$  que almacenan los datos generales de la simulación. Antes de llegar al punto de almacenarlos, hay que darse cuenta de que la posibilidad aleatoria de que en cada turno de salida abandone una persona, no contempla el número de personas que hay en la sala, por lo que puede producirse que salgan más personas de las que hay. Para ello, se deben limitar los valores del número de personas a la salida a un mínimo de cero personas en cada sala.

Todo lo expuesto, se puede expresar median te el código:

```
for k = 1:numero_dias
    % Se crean aleatoriamente tres turnos de llegada
    llegada1 = round((ultima_hora_llegada-primera_hora_llegada).*rand(1,7)+
        primera_hora_llegada);
    llegada2 = round((ultima_hora_llegada-llegada1).*rand(1,7)+llegada1);
    llegada3 = round((ultima_hora_llegada-llegada2).*rand(1,7)+llegada2);
    % Se almacenan para la simulación de varios días
    todas_llegada1 = [todas_llegada1 llegada1+(k-1)*dia*ones(1,7)];
    todas_llegada2 = [todas_llegada2 llegada2+(k-1)*dia*ones(1,7)];
    todas_llegada3 = [todas_llegada3 llegada3+(k-1)*dia*ones(1,7)];

    % Se genera aleatoriamente la posibilidad de que llegue o no una persona
    num_personas_llegada1 = round((1-0).*rand(1,7)+0);
    num_personas_llegada2 = num_personas_llegada1+round((1-0).*rand(1,7)+0);
    num_personas_llegada3 = num_personas_llegada2+round((1-0).*rand(1,7)+0);
    % Se generaliza para varios días
    todas_personas_llegada1 = [todas_personas_llegada1 num_personas_llegada1];
    todas_personas_llegada2 = [todas_personas_llegada2 num_personas_llegada2];
    todas_personas_llegada3 = [todas_personas_llegada3 num_personas_llegada3];

    % Se crean aleatoriamente tres turnos de salida
    salida1 = round((ultima_hora_salida-primera_hora_salida).*rand(1,7)+primera
        _hora_salida);
    salida2 = round((ultima_hora_salida-salida1).*rand(1,7)+salida1);
    salida3 = round((ultima_hora_salida-salida2).*rand(1,7)+salida2);
    % Se almacenan para la simulación de varios días
    todas_salida1 = [todas_salida1 salida1+(k-1)*dia*ones(1,7)];
    todas_salida2 = [todas_salida2 salida1+(k-1)*dia*ones(1,7)];
    todas_salida3 = [todas_salida3 salida1+(k-1)*dia*ones(1,7)];

    % Se genera aleatoriamente la posibilidad de que salga o no una persona
    num_personas_salida1 = num_personas_llegada3-round((1-0).*rand(1,7)+0);
    num_personas_salida2 = num_personas_salida1-round((1-0).*rand(1,7)+0);
    num_personas_salida3 = num_personas_salida2-round((1-0).*rand(1,7)+0);
    % Se generaliza para varios días
    todas_personas_salida1 = [todas_personas_salida1 num_personas_salida1];
    todas_personas_salida2 = [todas_personas_salida2 num_personas_salida2];
    todas_personas_salida3 = [todas_personas_salida3 num_personas_salida3];
end
```

Con ello, ya solo es necesaria la creación de los vectores (denotados  $num\_personas\_sala_i$ ) de datos que serán los que se tengan que pasar a TRNSYS. Estos vectores son el resultado de una simple combinación de todos los cálculos que han sido realizados hasta ahora, de forma que se tengan que crear por franjas con el objetivo de indicar en cada momento de la simulación, cuantas personas se encuentran en cada una de las salas. Todo esto, debe realizarse para cada una de las salas de las que está compuesta el edificio. La constitución de dichos vectores puede realizarse como:

```

for k = 1:numero_dias
    num_personas_sala1 = [num_personas_sala1 0*ones(1,llegada1(1)) num_personas_
        _llegada1(1).*ones(1,(llegada2(1)-llegada1(1))) num_personas_llegada
        2(1).*ones(1,(llegada3(1)-llegada2(1))) num_personas_llegada3(1).*ones
        (1,salida1(1)-llegada3(1)) num_personas_salida1(1).*ones(1,salida2(1)-
        salida1(1)) num_personas_salida2(1).*ones(1,salida3(1)-salida2(1)) num_
        personas_salida3(1).*ones(1,hora_cierre-salida3(1)) 0*ones(1,dia-hora_
        cierre)];
end

```

Con ello, ya se puede dar por concluida la forma de implementar el número de personas. Se pasa ahora al estudio en profundidad de cómo establecer si las luces de cada una de las salas se encuentran encendidas o no. Como se expuso anteriormente, las luces de cada sala se encontrarán encendidas siempre y cuando haya una persona en el interior de la sala en cuestión. Para poder determinar esto, se debe hacer uso de los vectores anteriormente creados que indican el número de personas que llegan y salen en cada momento de cada una de las salas (*num\_personas\_llegada<sub>i</sub>* y *num\_personas\_salida<sub>i</sub>*). Cuando se disponga de dicha información, ya se puede construir el vector completo que indica cual es el estado de las luces en cada tiempo de muestreo. En código siguiente únicamente se detallará con la llegada 1 para reducir el texto introducido.

```

for i = 1:7
    if num_personas_llegada1(i)>0
        luces_llegada1(i) = 1;
    end
end
luz_S1 = [luz_S1 0*ones(1,llegada1(1)) luces_llegada1(1).*ones(1,(llegada2(1)-
llegada1(1))) luces_llegada2(1).*ones(1,(llegada3(1)-llegada2(1))) luces_
llegada3(1).*ones(1,salida1(1)-llegada3(1)) luces_salida1(1).*ones(1,salida
2(1)-salida1(1)) luces_salida2(1).*ones(1,salida3(1)-salida2(1)) luces_
salida3(1).*ones(1,hora_cierre-salida3(1)) 0*ones(1,dia-hora_cierre)];

```

Dicho esto, ya se tiene identificado por completo cual es el estado de cada una de las luces de las salas en cada momento. Para poder determinar el estado de los ordenadores en cada momento de la simulación, se parte de la hipótesis de que cuando una persona llega a la sala, esta puede encender o un ordenador de forma completamente aleatoria. Para ello, se generan vectores (*num\_ordenadores\_llegada<sub>i</sub>* y *num\_ordenadores\_salida<sub>i</sub>*) para cada una de las llegadas en los que se indica si se enciende o no un ordenador, siempre con la condición de que haya llegado una nueva persona a la sala. Adicionalmente, se han de saturar los valores respectivos, de forma que el número de ordenadores nunca pueda ser negativo en los distintos turnos de salida. Esto resulta:

```

% Se crean los posibles encendidos de los ordenadores
num_ordenadores_llegada1 = round((1-0).*rand(1,7)+0).*num_personas_llegada1;
num_ordenadores_llegada2 = num_ordenadores_llegada1+round((1-0).*rand(1,7)+0)
.*(num_personas_llegada2-num_personas_llegada1);
% Se crean los posibles apagados de los ordenadores
num_ordenadores_salida1 = num_ordenadores_llegada3 - (num_personas_llegada3 -
num_personas_salida1);
num_ordenadores_salida2 = num_ordenadores_salida1 - (num_personas_salida1 - num_
personas_salida2);

% Se saturan los valores para cada una de las salidas
for i = 1:7
    if num_ordenadores_salida1(i) < 0
        num_ordenadores_salida1(i) = 0;
    end
end
end

```

Con ello, ya solo es necesario formar el vector completo del estado de los ordenadores en cada momento, para poder aportar al modelo del edificio los valores necesarios en cada tiempo de muestreo.



```
ordenadores_sala1 = [ordenadores_sala1 0*ones(1,llegada1(1)) num_ordenadores_
llegada1(1).*ones(1,(llegada2(1)-llegada1(1))) num_ordenadores_llegada2(1)
.*ones(1,(llegada3(1)-llegada2(1))) num_ordenadores_llegada3(1).*ones(1,
salida1(1)-llegada3(1)) num_ordenadores_salida1(1).*ones(1,salida2(1)-
salida1(1)) num_ordenadores_salida2(1).*ones(1,salida3(1)-salida2(1)) num_
ordenadores_salida3(1).*ones(1,hora_cierre-salida3(1)) 0*ones(1,dia-hora_
cierre)];
```

De esta forma, ya solo hace falta indicar cómo se va a realizar el modelado de las infiltraciones que se producen cuando abren o cierran las puertas. Para ello, tras la búsqueda de diversa información, se ha llegado a la conclusión de que la mejor opción para cuantificar el valor de dicho caudal de aire que pasa de una sala a otra se puede determinar mediante la fórmula de Pham and Oliver (1983) [4], en la cual permite el cálculo de la infiltración como

$$I(\text{kg/hr}) = 3600 \cdot 0.226A \rho_i (\text{gH})^{0.5} \left( \frac{\rho_i - \rho_o}{\rho_i} \right)^{0.5} \left( \frac{2}{1 + (\rho_o/\rho_i)^{0.333}} \right)^{1.5}, \quad (5.1)$$

donde  $I$  representa a las infiltraciones producidas,  $A$  es el área de la puerta por la que se produce dicha infiltración,  $g$  indica el valor de la gravedad en  $m/s^2$ ,  $H$  hace referencia a la altura de la puerta, mientras que  $\rho_i$  y  $\rho_o$  son los valores de las densidades en cada uno de los espacios que se comunican. Como se puede observar, solo son necesarios los cálculos de cada una de las dos densidades.

Como el cálculo de las infiltraciones depende de la densidad del aire de la sala en cada momento, dicho procedimiento se debe realizar junto con las determinaciones de la densidad en cada tiempo de muestreo. Para poder obtener las densidades, se recurre a la fórmula de la densidad revisada (CIPM-2007) [14] que se establece como

$$\rho_a = \frac{pM_A}{ZR T} \left[ 1 - x_v \left( 1 - \frac{M_v}{M_a} \right) \right], \quad (5.2)$$

siendo  $p(\text{Pa})$  la presión a la que se encuentran las salas,  $T(\text{K})$  la temperatura de cada una de las salas,  $M_a(\text{g/mol})$  la masa molar del aire seco,  $M_v(\text{g/mol})$  la masa molar del vapor de agua,  $R(\text{J/molK}) = 8.314475$  y, por último,  $Z$  y  $x_v$ , que representan el factor de compresibilidad y la fracción molar de vapor de agua respectivamente, las cuales se calculan con

$$x_v = h(\alpha + \beta p + \gamma t^2) \frac{\exp(AT^2 + BT + C + D/T)}{p} \quad (5.3a)$$

$$Z = 1 - \frac{p}{T} [a_0 + a_1 t + a_2 t^2 + (b_0 + b_1 t)x_v + (c_0 + c_1 t)x_v^2] + \frac{p^2}{T^2} (d + e x_v^2), \quad (5.3b)$$

donde  $h$  (tanto por uno) representa la humedad relativa de la sala en cuestión,  $T$  son las temperaturas en grafos K mientras que  $t$  en  $^\circ\text{C}$  y los valores de los coeficientes son los expresados en la Tabla 5.3

**Tabla 5.3** Coeficientes utilizados para el cálculo de la densidad..

Coeff	Valor	Coeff	Valor
$A$	$1.2378847 \times 10^{-5} K^{-2}$	$a_0$	$1.58123 \times 10^{-6} KPa^{-1}$
$B$	$-1.9121316 \times 10^{-2} K^{-1}$	$a_1$	$-2.9331 \times 10^{-8} Pa^{-1}$
$C$	33.93711047	$a_2$	$1.1043 \times 10^{-10} K - 1Pa^{-1}$
$D$	$-6.3431645 \times 10^3 K$	$b_0$	$5.707 \times 10^{-6} KPa^{-1}$
$\alpha$	1.00062	$b_1$	$-2.051 \times 10^{-8} Pa^{-1}$
$\beta$	$3.14 \times 10^{-8} Pa^{-1}$	$c_0$	$1.9898 \times 10^{-4} KPa^{-1}$
$\gamma$	$5.610 \times 10^{-7} K^{-2}$	$c_1$	$-2.376 \times 10^{-6} Pa^{-1}$
$e$	$-0.765 \times 10^{-8} K^2 Pa^{-2}$	$d$	$1.83 \times 10^{-11} K^2 Pa^{-2}$

Con estas ecuaciones, ya se está en disposición de comenzar el cálculo en cada uno de los tiempos de muestreo, en los cuales se debe leer en primer lugar, cada una de las temperaturas de las salas y con ellas,

proceder al cálculo de la densidad. Este cálculo implementado, se puede expresar como:

```
% Densidad para la Sala 1
Xv_S1 = (RH/100) * (alpha + beta*P_salas + gamma*T_S1^2) *(exp (A*Tk_S1^2 + B*
    Tk_S1 + C+ D/Tk_S1) / P_salas);
Z = 1-(P_salas/Tk_S1)*(a0+a1*T_S1+a2*T_S1^2+(b0+b1*T_S1)*Xv_S1+(c0+c1*T_S1)*Xv_
    S1^2)+(P_salas^2/Tk_S1^2)*(d+e*Xv_S1^2);
densidad_S1 = (P_salas*Ma)/(Z*R*Tk_S1)*(1-Xv_S1*(1-Mv/Ma));
```

Hay que tener en cuenta que previamente se han tenido que inicializar todas las variables y coeficientes correspondientes para que el cálculo se pueda llevar a cabo. Para poder establecer cuando si y cuando no se va a producir una infiltración, se debe hacer uso de los que almacenan los datos de llegada y permiten, por tanto, saber cuándo se abre y cierra una puerta. Cuando este fenómeno se produce, el caudal de aire que pasa de un lado a otro depende de las densidades del aire en cada uno de los dos lados, de forma que se produce un flujo de aire de zonas de mayor densidad hacia zonas de menor valor. Para ello, en cada momento se debe comprobar cuál es el valor de la densidad en cada sala y establecer el sentido de flujo para poder otorgar el valor en el sentido correcto. Esto puede traducirse en código como lo siguiente:

```
if ((todas_llegada1(1+7*w) == nStep && infiltraciones_llegada1(1+7*w) == 1) ||
    (todas_llegada2(1+7*w) == nStep && infiltraciones_llegada2(1+7*w) == 1) ||
    (todas_llegada3(1+7*w) == nStep && infiltraciones_llegada3(1+7*w) == 1) ||
    (todas_salida1(1+7*w) == nStep && infiltraciones_salida1(1+7*w) == 1) || (
    todas_salida2(1+7*w) == nStep && infiltraciones_salida2(1+7*w) == 1) || (
    todas_salida3(1+7*w) == nStep && infiltraciones_salida3(1+7*w) == 1) || (
    hora_cierre == nStep && infiltraciones_cierre(1+7*w) == 1))
    if densidad_S1 < densidad_Pasillo
        infiltraciones_a_S1 = 0;
        infiltraciones_Pasillo_de_S1 = 3600*0.226*Area_puerta*sqrt(g*H)*
            densidad_Pasillo*sqrt((densidad_Pasillo-densidad_S1)/densidad_
                Pasillo)*(2/(1+(densidad_S1/densidad_Pasillo)^0.333))^1.5
    else
        infiltraciones_a_S1 = 3600*0.226*Area_puerta*sqrt(g*H)*densidad_S1*sqrt
            ((densidad_S1-densidad_Pasillo)/densidad_S1)*(2/(1+(densidad_Pasillo
                /densidad_S1)^0.333))^1.5
        infiltraciones_Pasillo_de_S1 = 0;
    end
end
end
```

Hecho esto, ya se puede concluir con la explicación un poco más en detalle de cómo se ha elaborado el simulador, otorgando un valor a todas las variables de entrada en el bloque del edificio. Ya solo falta comprobar que todo funciona correctamente y establecer la conexión entre todos los bloques en un modelo de TRNSYS.

## 5.4 Resultados obtenidos

Para poner a prueba todo lo que ha sido elaborado hasta ahora, se van a implementar todos los archivos que han sido elaborados hasta ahora en TRNSYS, para poder realizar las posteriores simulaciones. Para empezar en una plantilla de TRNSYS se deben realizar las distintas conexiones entre los bloques del tiempo atmosférico, el bloque del edificio y la llamada a MATLAB®, como se puede ver en la Figura 5.4

Las conexiones que se deben realizar entre todos los componentes son:

- La conexión que se realiza entre el bloque del tiempo atmosférico y el edificio es la relativa a la incidencia de cada una de las variables exteriores sobre el edificio en cuestión, las cuales ya han sido tratadas de forma mucho más detallada con anterioridad.
- La unión Simulador-Edificio, aporta al edificio todos los valores instantáneos de cada una de las variables que han sido detalladas con los distintos códigos de MATLAB® y las cuales indican el

número de personas, ordenadores, estado de las luces y las infiltraciones producidas a través de las puertas.

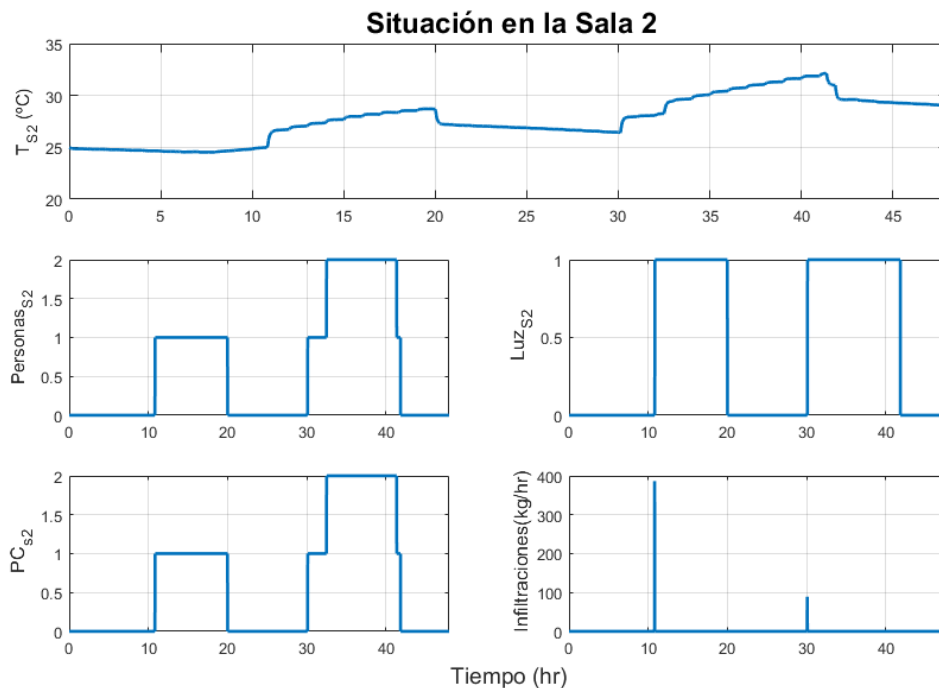
- El edificio únicamente debe aportar al simulador el valor que toma cada una de las temperaturas de las salas creadas en cada tiempo de simulación y que son necesarias para el cálculo de las densidades.



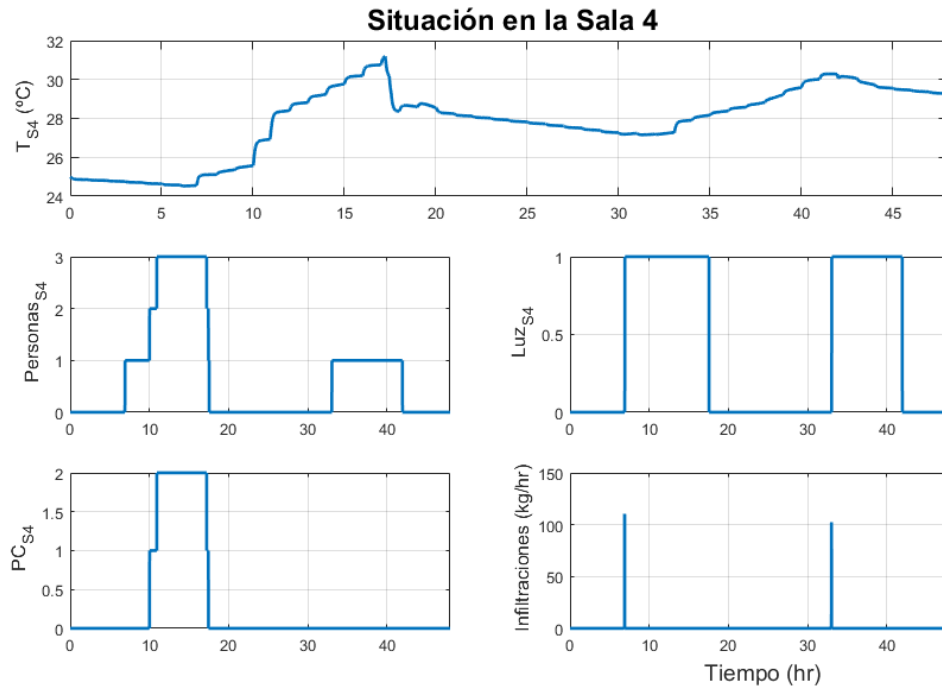
**Figura 5.4** Esquema del conexionado a realizar para introducir el simulador.

Implementando este esquema de conexión ya se puede proceder a la simulación para observar cómo se comportan cada una de las salas. Dicha simulación corresponderá a un día completo con un tiempo de muestreo, y cuyo fin es observar cómo todos los parámetros se generan de forma aleatoria y cómo afectan a cada una de las temperaturas de cada sala. Para ello se va a estudiar que ocurre en tres de las salas durante la simulación completa.

En la Figura 5.5 puede observarse qué ocurre en la Sala número dos durante un periodo igual a 2 días, en el primer día solo asiste una persona a la sala y permanecen en ella hasta la hora del cierre, además enciende el ordenador. Puede verse como la luz permanece encendida durante todo el periodo. Cuando dicha persona entra en la sala, se produce una infiltración hacia ella ya que la densidad de la sala es mayor a la del pasillo, mientras que cuando esta abandona la sala, no se produce ninguna infiltración hacia la sala y si se produce en el pasillo, ya que las densidades han cambiado. Se puede observar que cuando una persona llega a la sala y enciende el ordenador se produce un lógico aumento de la temperatura de dicha sala, la cual disminuye cuando ya no se encuentra nadie en su interior. En el segundo día ocurre algo similar, pero en este caso llega haber un total de 2 personas y dos ordenadores en la sala, por lo que se observa que se produce un mayor aumento de la temperatura durante ese periodo. Sin embargo, la infiltración provocada es menor, lo cual indica que las temperaturas de las salas y el pasillo en ese instante no son muy dispares.

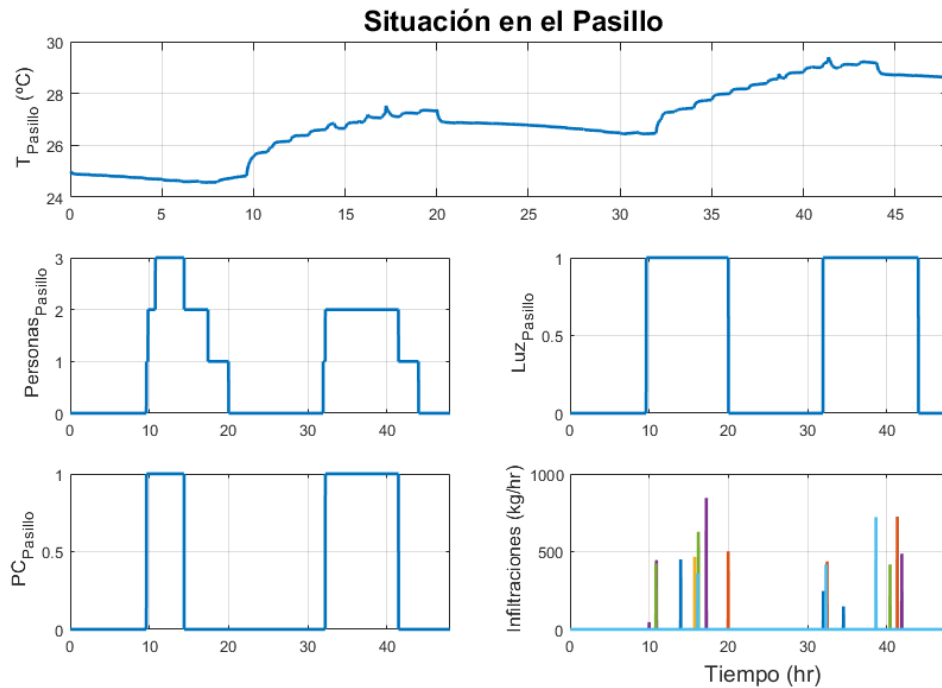


**Figura 5.5** Situación de la Sala 2 durante la simulación.



**Figura 5.6** Situación de la Sala 4 durante la simulación.

En la Figura 5.6 se observa cómo evoluciona la temperatura de la sala 4 y como varían las distintas variables. Una de las cosas que hay que destacar durante el primer día es cómo afecta la llegada de 3 personas a la temperatura, se puede observar como aumenta drásticamente y como aumenta también la pendiente con la que crece la temperatura. También se observa cómo aumenta en mayor medida cuando se produce la llegada de una persona que enciende el ordenador. Para comprobar el correcto funcionamiento se puede observar el comportamiento de las luces, que permanecen encendidas siempre que haya una persona.



**Figura 5.7** Situación del Pasillo durante la simulación.

Por último, en la Figura 5.7 se puede observar cómo el tamaño de sala muestra gran influencia sobre las evoluciones de la temperaturas, ya que el pasillo casi triplica al tamaño de una sala y las temperaturas alcanzadas son algunos grados inferiores a la del resto de las salas. Esto puede explicar que cuando se produce la salida de las personas, las infiltraciones se produzcan hacia el pasillo, puesto que posee una menor temperatura, mientras que a primera hora de la mañana cuando se producen las primeras llegadas, es el pasillo el que tiene mayores temperaturas, debido que presenta un menor contacto con el exterior, y por ello las infiltraciones se producen en sentido contrario.

Como se puede observar en las correspondientes simulaciones, se puede afirmar el correcto funcionamiento del simulador y el comportamiento completamente aleatorio que este presenta, de forma que se puede aplicar a cualquier simulación que se realice independientemente de la duración de ésta.



## 6 Control LQR

---

En este capítulo, se pretende aplicar al sistema diseñado un controlador LQR para el seguimiento de referencias[5]. Un controlador LQR puede ser definido como un controlador en bucle cerrado para sistemas MIMO (Multiples Inputs Multiples Outputs). Para su diseño se parte del sistema identificado en el Capítulo 4. Dicho sistema era enunciado como

$$x(k+1) = Ax(k) + Bu(k) \quad (6.1a)$$

$$y(k) = Cx(k), \quad (6.1b)$$

donde, como ya se dijo en su momento,  $x \in \mathbb{R}^{n_x}$  representa el vector de estados,  $u \in \mathbb{R}^{m_u}$  al conjunto de variables de actuación presentes en el sistema, por último,  $y \in \mathbb{R}^{n_y}$  que hace referencia a las temperaturas de las habitaciones, las cuales se desean controlar.

Durante este capítulo se llevará a cabo una explicación teórica del controlador LQR, así como de la versión del controlador LQR con efecto integral. Para ello, el primer lugar, se debe introducir un observador de estados, que, a partir de la salida del sistema y la actuación del controlador, este sea capaz de calcular el estado del sistema. Al final de capítulo, se exponen los resultados obtenidos de la aplicación de dicho controlador al sistema elaborado en capítulos anteriores.

### 6.1 Observador de estados

Un observador de estados [7] se puede definir como un bloque que estima las variables de estado de un sistema, usando como base las variables de control y salida de un sistema. Para su explicación se parte del sistema expuesto en la Ecuación 6.1 que representa el sistema en cuestión.

Se sabe que el modelo matemático del observador es igual que el modelo de la planta al que se le debe añadir un término de error que representa la diferencia entre la salida medida y la observada. Se debe saber que  $\hat{x}$  hace referencia durante todo el capítulo al vector de estados observado. Representando lo expuesto, se tiene

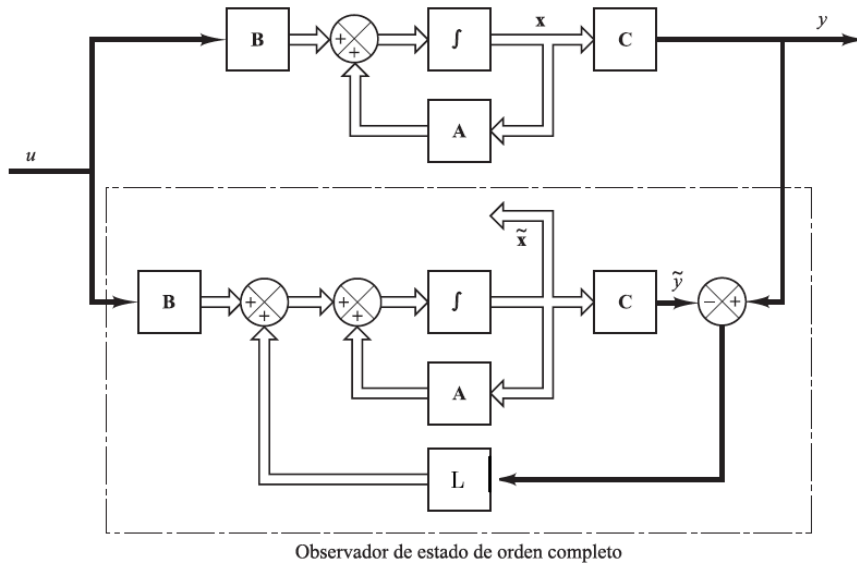
$$\begin{aligned} \hat{x}(k+1) &= A\hat{x}(k) + Bu(k) + L(y - C\hat{x}(k)) \\ &= (A - LC)\hat{x}(k) + Bu(k) + Ly(k). \end{aligned} \quad (6.2)$$

En la Ecuación 6.2, el segundo y tercer término pueden agruparse de forma que construyan un vector  $\mathbf{u}$  como composición de las variables de control y las salidas del sistema, tal que

$$\mathbf{u} = \begin{bmatrix} u(k) \\ y(k) \end{bmatrix}. \quad (6.3)$$

La estructura que se ha implementado puede visualizarse gráficamente en la Figura 6.1. Realizando la composición del vector  $\mathbf{u}$ , la ecuación del observador puede reescribirse de una forma más compacta como

$$\hat{x}(k+1) = (A - LC)\hat{x}(k) + [B \quad L] \mathbf{u}. \quad (6.4)$$



**Figura 6.1** Esquema del observador utilizado.

Con esto, ya se dispone de la ecuación del observador de estados (Ecuación 6.4), donde si se ha determinado correctamente el valor de L, se puede obtener una estimación del estado actual del sistema. La determinación de dicha ganancia en este caso se obtiene mediante la ayuda de MATLAB<sup>®</sup>, haciendo uso del comando *dlqr* cuyas entradas son dos matrices que simulan el comportamiento del sistema, además de dos matrices que ponderas la intensidad con la que se modelan las variables. En este caso concreto, para el cálculo de dicha ganancia L, el comando a ejecutar sería

$$L = \left( dlqr(A^T, C^T, W, V) \right)^T. \quad (6.5)$$

De esta forma, se concluye con la explicación del primer paso necesario para la elaboración del esquema de control. La existencia del observador es algo fundamental debido a como se ha definido el sistema, ya que sin él no sería posible la estimación del vector de estados que lo caracteriza en cada momento. El siguiente paso, como es lógico es el cálculo de la variable de control.

## 6.2 Control LQR

El controlador LQR del inglés Linear Quadratic Regulator, es un algoritmo de control óptimo, a partir del cual se obtiene una acción de control para lograr resolver el problema de control de un sistema, que en este caso será de tiempo discreto [6]. Para enunciar el problema de resolución del controlador, se parte de las ecuaciones del sistema en cuestión, representado como

$$x(k + 1) = Ax(k) + Bu(k) \quad (6.6a)$$

$$y(k) = Cx(k). \quad (6.6b)$$

El objetivo del control LQR, es el cálculo de una acción de control proporcional al estado del sistema. En forma de ecuación, esto puede verse como

$$u(k) = -Kx(k), \quad (6.7)$$

donde, esa matriz de ganancias (K), se obtiene de la minimización de una función de coste, J, sujeta a la dinámica del sistema en cuestión. Esta función J, es elaborada ponderando el peso de los valores de tanto los estados como de las señales de actuación, todos ellos al cuadrado, de la siguiente manera

$$J = \sum_{k=0}^{\infty} \left( x_k^T Q x_k + u_k^T R u_k \right), \quad (6.8)$$



siendo las matrices  $Q > 0$  y  $R > 0$  pesos que permiten variaciones en la forma de controlar. Para entender esto mejor se exponen dos ejemplos:

- Si se toman unos valores de  $Q = 1$  y  $R = 100$ , se produce una acción de control mucho más suave, ya que se pondera mucho más el término de las variables de control, por lo que para que se minimice correctamente, ese término debe ser lo más pequeño posible.
- Por el contrario, con unos valores de  $Q = 100$  y  $R = 1$ , la acción de control debe llevar el vector de estados a 0, por lo que como no se pondera tanto la acción de control, se producirá una acción más agresiva sobre las variables a controlar.

Observando la importancia que toman ambos parámetros, cuando se proceda al diseño del controlador se debe buscar un compromiso entre la brusquedad en el controlador, que puede provocar sobreoscilaciones, y la velocidad de control, de forma que el sistema alcance el punto deseado en un tiempo razonable.

De la Ecuación 6.8, se demuestra que la constante proporcional que minimiza la función de coste, cumple la siguiente ecuación:

$$K = (R + B^T S B)^{-1} (B^T S A). \quad (6.9)$$

Además se identifica la matriz  $S$  como solución de la ecuación de Riccati en tiempo discreto, la cual puede enunciarse como

$$A^T S A - S - (A^T S B)(B^T S B + R)^{-1}(B^T S A) + Q = 0. \quad (6.10)$$

Por lo que, una vez se dispone de dicha matriz  $S$ , habiendo solucionado la ecuación de Riccati, es trivial el cálculo de la acción de control que se debe aplicar al sistema. Sin embargo, este tipo de controlador no contempla el problema de seguimiento o tracking, ya que de lo que se encarga es de la minimización de los estados del sistema. Por ello, se recurre al control LQR con efecto integral, el cual permite abordar el seguimiento de una referencia.

### 6.3 Control LQR con efecto integral

Para abordar el problema del seguimiento de referencias con control LQR, lo que se conoce como control con efecto integral [2], se comienza planteando las referencias como si fuesen perturbaciones que afectan al sistema con el objetivo de eliminar el error, definido como la diferencia entre la referencia y la salida del sistema, cuando el sistema en cuestión se encuentra en régimen permanente. Además, se define el sistema en términos incrementales haciendo las siguientes declaraciones:

$$\begin{aligned} \Delta x(k) &= x(k) - x(k-1) \\ \Delta u(k) &= u(k) - u(k-1) \\ \Delta y(k) &= y(k) - y(k-1). \end{aligned} \quad (6.11)$$

Haciendo esta definición, el error puede ser definido como  $\Delta e(k) = \Delta r(k) - \Delta y(k)$ , Si además se trata la referencia o perturbación como un escalón donde en régimen permanente se cumple que  $r(k+1) - r(k) = 0$ , resulta que el incremento del término del error puede expresarse como

$$\Delta e(k) = \Delta r(k) - \Delta y(k) = -\Delta y(k). \quad (6.12)$$

Por otro lado, el sistema identificado del que se parte, por linealidad, puede expresarse también en términos incrementales, de forma que se tiene

$$\begin{aligned} \Delta x(k+1) &= A \Delta x(k) + B \Delta u(k) \\ \Delta y(k) &= C \Delta x(k). \end{aligned} \quad (6.13)$$

Combinando las ecuaciones 6.12 y 6.13, el término del error puede escribirse en una como  $e(k) = e(k-1) - C \Delta x(k)$ , lo que permite reconstruir la primera ecuación del sistema, en una análoga pero ampliada

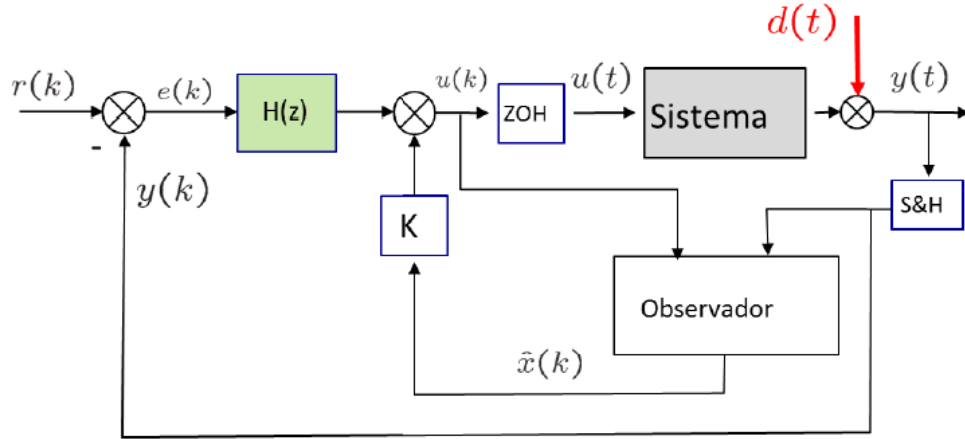


Figura 6.2 Diagramas de bloques del control LQR.

con términos del error, de forma que el nuevo sistema ampliado es

$$\begin{bmatrix} \Delta x(k+1) \\ e(k) \end{bmatrix} = \begin{bmatrix} A & 0 \\ -C & I \end{bmatrix} \begin{bmatrix} \Delta x(k) \\ e(k-1) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} \Delta u(k)$$

$$\Delta y(k) = C \Delta x(k). \quad (6.14)$$

Con este nuevo sistema que se ha definido, obviamente la acción de control calculada se modifica y su nueva estructura es

$$\Delta u(k) = K_z \begin{bmatrix} \Delta x(k) \\ e(k-1) \end{bmatrix} = K_x \Delta x(k) + K_e e(k-1). \quad (6.15)$$

Esta ley de control también puede expresarse en términos reales de las variables y no de forma incremental, como

$$u(k) = K_x x(k) + K_e \sum_{j=0}^{k-1} e(j). \quad (6.16)$$

Como se puede observar, la nueva ecuación de la acción de control presenta dos términos, el primero de ellos es el genérico que se genera con minimización de los estados, mientras que el segundo de ellos depende del error que se produce entre la referencia y la salida, por lo que de esta forma se garantiza el seguimiento o tracking de la referencia. Agrupando todos los conocimientos abordados en los apartados anteriores, el diagrama de bloques resultante es el que se puede observar en la Figura 6.2. Este control será el objetivo de implementación en el sistema de estudio.

## 6.4 Aplicación al caso de estudio

Como se definió en el Capítulo 4, el sistema identificado obtenido está compuesto de: siete variables a controlar, donde cada una de ellas representa la temperatura de cada una de las habitaciones ( $T_i$ ); por otro lado, las variables manipulables o controlables son ocho, las cuales representan las aperturas de cada una de las válvulas y el factor de carga parcial (fcp) de la enfriadora; por último, el número de estados del sistema, que se considera igual a ocho tras los resultados de la identificación. El sistema identificado, por tanto es

$$x(k+1) = Ax(k) + Bu(k) \quad (6.17a)$$

$$y(k) = Cx(k), \quad (6.17b)$$

donde  $x \in \mathbb{R}^8$  representa el vector de estados,  $u \in \mathbb{R}^8$  al conjunto de variables de control, por último,  $y \in \mathbb{R}^7$  que hace referencia a las temperaturas de las habitaciones.

Para comenzar con el algoritmo de control, se debe saber que este está dividido en tres etapas que se deben realizar cíclicamente cada tiempo de muestreo, en primer lugar, se deben leer las salidas del sistema, con ellas se calcula la acción de control, y por último, estas se pasan al sistema.

Pero antes de poder realizar todo esto, hay que realizar una serie de cálculos previos como son por ejemplo, la ampliación de las matrices A y B para el correspondiente cálculo de la constante proporcional de control. Además, se debe calcular la L del observador. Todo ello, se realiza con el siguiente código:

```
% Ampliamos el sistema
A_gorro = [A zeros(nx,ny);-C eye(ny)];
B_gorro = [B; zeros(ny,nu)];

% Calculamos la matriz L del observador
W = 0.01.*eye(nx);
V = 0.001.*eye(ny);
L = dlqr(A',C',W,V);
L = L';

% Calculamos la matriz K para la calcular la actuación
Q = 1.*eye(size(A_gorro));
R = 100.*blkdiag(10,10,10,10,10,10,10,1);
K = dlqr(A_gorro,B_gorro,Q,R);
Kx = K(:,1:nx);
Ke = 0.5.*K(:,nx+1:end);
```

Para la confección de los valores de Q y R, se modifica el valor de R que es la que afecta a las variables de control para que genere un control más suave, de igual forma ocurre con la última componente de la matriz R que es la que afecta al fcp, que se desea que este sea inferior a las aperturas para lograr así disminuir lo máximo posible el consumo eléctrico de la enfriadora.

Cuando se hayan realizado estos cálculos necesarios para el bloque de control, se procede al primer paso de los enunciados anteriormente. En él básicamente se deben leer los valores de las salidas del sistema, es decir, cada una de las temperaturas de las habitaciones. Esto se obtiene de la siguiente forma:

```
% Leemos las Temperaturas
Tae = trnInputs(1);
y_actual(1,nStep) = trnInputs(2);
y_actual(2,nStep) = trnInputs(3);
y_actual(3,nStep) = trnInputs(4);
y_actual(4,nStep) = trnInputs(5);
y_actual(5,nStep) = trnInputs(6);
y_actual(6,nStep) = trnInputs(7);
y_actual(7,nStep) = trnInputs(8);
```

Cuando ya se dispone de los valores de las salidas, se procede al paso más importante, donde se calculan los valores de las señales de control. Hasta este punto, lo único de lo que se dispone información es de los valores de las señales de actuación en tiempo de muestreos anteriores y de las temperaturas de las habitaciones. Con esta información, se puede calcular el estado actual del sistema mediante el observador ( $\hat{x}$ ), esto se realiza, construyendo el vector ampliado con las actuaciones y salidas e, introduciendo este en la ecuación del observador. Cuando se dispone del valor de  $\hat{x}$ , este junto con el error entre la referencia y la salida, son introducidos en la ecuación del cálculo de la señal de control. Todo esto, se ha realizado en términos incrementales, ya que su implementación es más sencilla y conlleva menos coste computacional que si se realizase en términos absolutos. Para ello, el código elaborado es el siguiente:

```
% Se calcula la x del observador
deltax_observador(:,nStep-1) = [deltax(:,nStep-1);deltay_actual(:,nStep-1)];
deltax_observador(:,nStep) = (A-L*C)*deltax_observador(:,nStep-1) + [B L]*
    deltax_observador(:,nStep-1);
```

```

x_observador(:,nStep) = x_observador(:,nStep-1) + deltax_observador(:,nStep);

% Se calcula la señal de actuacion
deltax(:,nStep) = -Kx*deltax_observador(:,nStep) - Ke*error(:,nStep-1);
u_actual(:,nStep) = u_actual(:,nStep-1) + deltax(:,nStep);
% Se saturan los valores
for w=1:nu
    if u_actual(w,nStep)<0
        u_actual(w,nStep) = 0;
    end
    if u_actual(w,nStep)>1
        u_actual(w,nStep) = 1;
    end
end
end

```

Como se observa en el código anterior, también se limitan los valores de la actuación, ya que, tanto la válvula como el fcp toman valores entre 0 y 1. Por último, para completar el bucle de control, se deben aplicar dichas señales de control calculadas a sus respectivos equipos. Esto se realiza con una simple asignación de los elementos correspondientes a las variables deseadas:

```

% Se asignan las señales de control
Apertura(:,nStep) = u_actual(1:nu-1,nStep);
fcp(1,nStep) = u_actual(end,nStep);

% Preparan las variables para la siguiente ejecucion
error(:,nStep) = ref(:,nStep) - y_actual(:,nStep);
deltax_actual(:,nStep) = y_actual(:,nStep) - y_actual(:,nStep-1);

```

Hay que destacar que, antes de finalizar el bucle de control, se deben actualizar las componentes relativas al error y a la variación de las salidas, puesto que estas serán usadas en la siguiente iteración para el cálculo de la  $u$  del observador.

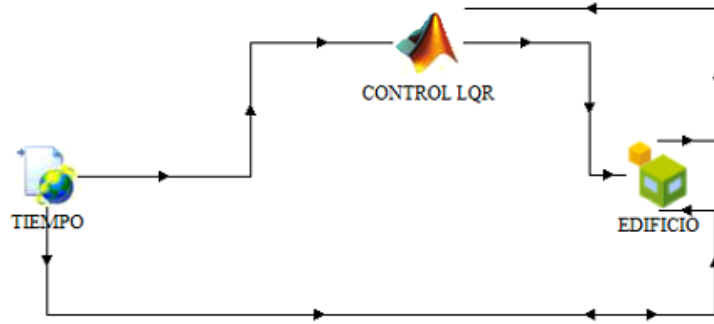
Como se ve, este controlador genera unos valores de los parámetros de Apertura y fcp óptimos para realizar un control sobre el sistema en cuestión. Estos valores deben ser aplicados, en primer lugar, sobre los equipos que generan o calculan la potencia necesaria que se debe aportar sobre cada una de las habitaciones. Para ello, las conexiones entre los bloques deben ser las siguientes:

- Una conexión entre el bloque del tiempo y controlador LQR que permita la toma del dato de la temperatura exterior, la cual es necesaria para caracterizar el funcionamiento de la enfriadora.
- Por otro lado, la conexión realizada entre el edificio y el controlador que permite la aportación de cada una de las temperaturas de las habitaciones a dicho controlador. Estas son las entradas del controlador.
- De forma análoga, pero en sentido contrario, el bloque de control tiene que aportar cada una de las potencias calculadas a través de los datos de aperturas y fcp del controlador. Estas potencias son las encargadas de regular la temperatura de las habitaciones.
- Por último, se tiene la conexión correspondiente entre el tiempo atmosférico y el edificio, la cual es necesaria para su funcionamiento.

De esta forma, con todo lo expuesto anteriormente, el esquema resultante se observa en la Figura 6.3.

## 6.5 Resultados obtenidos

Para realizar una comprobación de los resultados obtenidos y garantizar que el controlador funcione adecuadamente, se va a realizar una simulación donde se producirán diversos cambios en la referencia de las temperaturas de las habitaciones para observar así la respuesta del sistema y, por tanto, comprobar cómo se llega a dicha referencia. Para ello, se realizan simulaciones de una duración de  $800 \text{ min}$ , en las que se producen cuatro cambios aleatorios en la referencia cada  $200 \text{ min}$ . En estas simulaciones el control se aplica



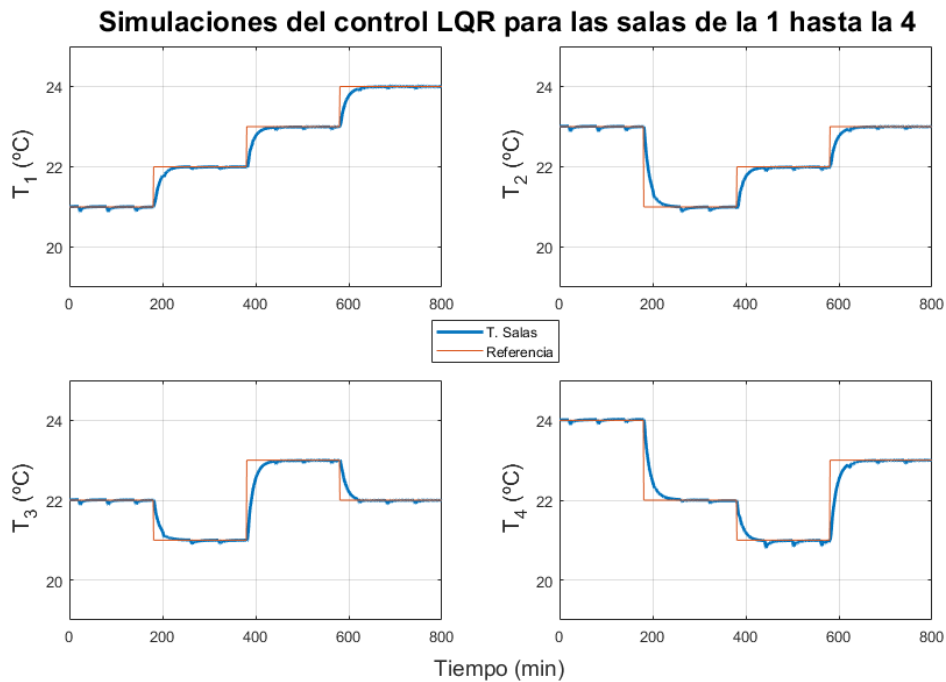
**Figura 6.3** Esquema de implementación del controlador LQR en TRNSYS.

siempre sobre el sistema real y no sobre el linealizado. Para observar los resultados se ilustran los resultados de todas las habitaciones en las Figuras 6.4 y 6.5.

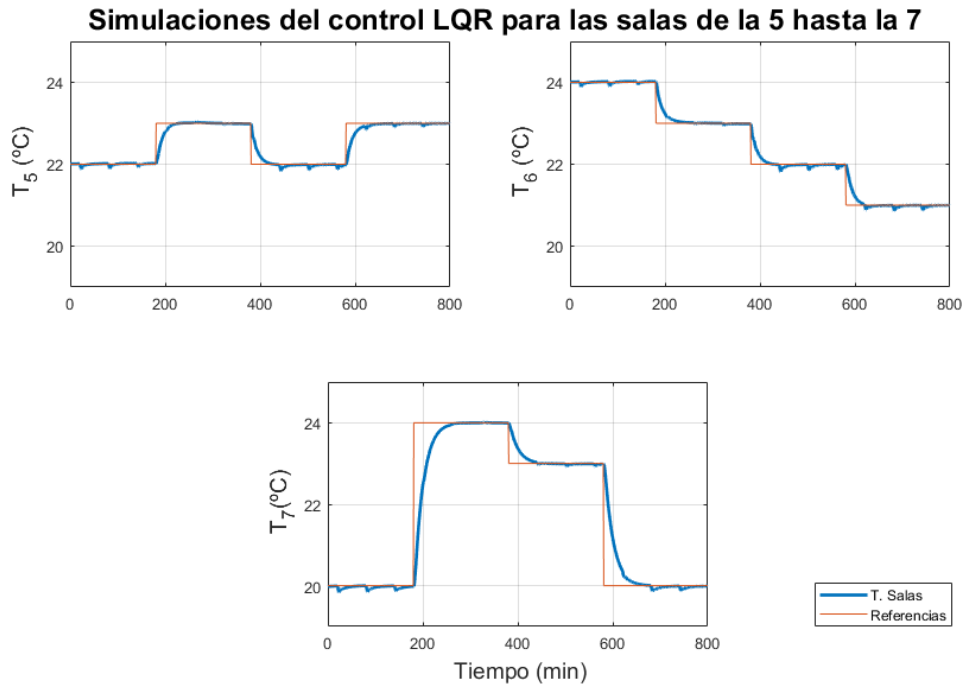
Como se observa, las referencias son alcanzadas y mantenidas correctamente incluso cuando se producen las pequeñas perturbaciones ocasionadas por las dinámicas que representan los flujos de aire a través de las paredes. Se debe visualizar también qué ocurre con los valores que toman las aperturas de las válvulas así como el del valor del factor de carga parcial (fcp). Para ello se han obtenido las gráficas visibles en las Figuras 6.6 y 6.7.

Como se observa, los resultados obtenidos son coherentes puesto que cuando se desea aumentar la temperatura de una sala, la válvula se cierra un poco provocando así que circule menos caudal de agua, lo que conlleva que la potencia intercambiada con la sala sea menor. En el caso contrario, cuando se desea disminuir la temperatura, la válvula debe abrirse más para permitir el paso de mas agua fría.

La pendiente negativa que se puede observar en los valores del factor de carga parcial, se debe principalmente a varias razones donde, la primera de ellas, se produce porque las simulaciones expuestas representan un extracto de la simulación total realizada, por lo que el agua que atraviesa la enfriadora ha sido enfriada

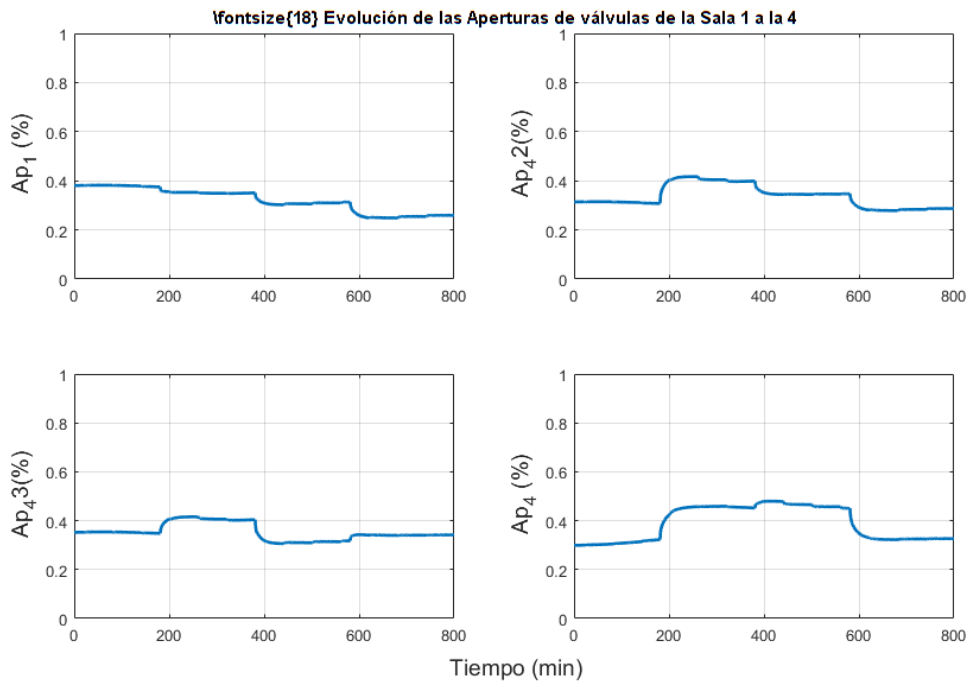


**Figura 6.4** Resultados obtenidos de la implementación del LQR (Salas 1 hasta 4).

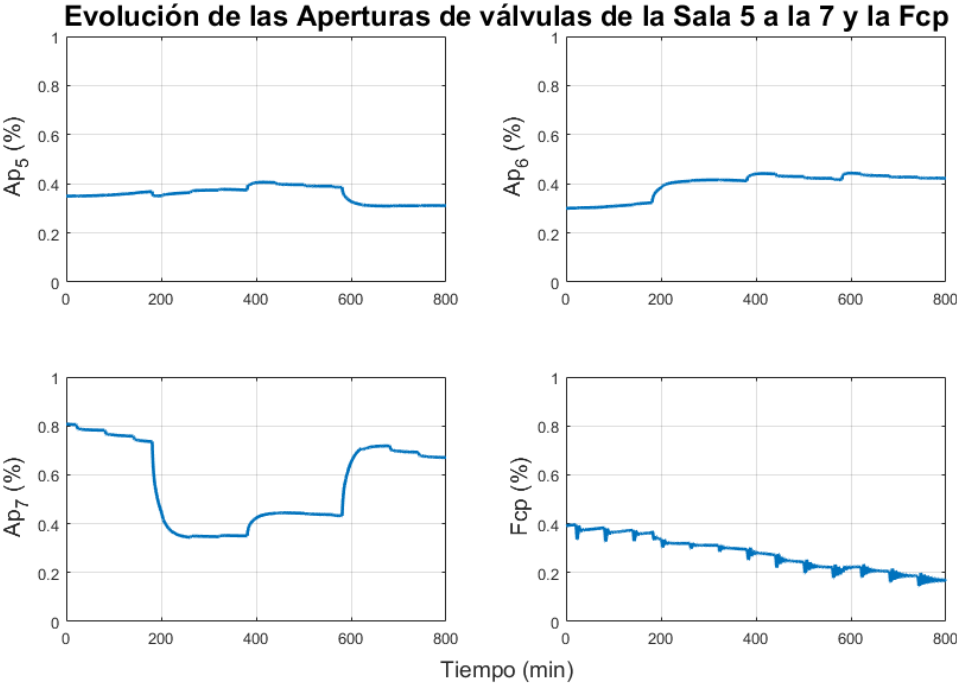


**Figura 6.5** Resultados obtenidos de la implementación del LQR (Salas 5 hasta 7).

mucho al principio y está entorno a la temperatura mínima de trabajo, por lo que el sistema se permite el lujo de minimizar aún más el fcp aumentando así la temperatura del agua, hasta que llegue un punto que no se disponga de la potencia suficiente, y sea entonces cuando dicha variable tenga que aumentar su valor para poder seguir enfriando las salas. La segunda razón se encuentra en los valores que se han aportado en los parámetros del controlador (matriz R), otorgándole al fcp un mayor peso, de forma que el controlador intenta modificar y mantener los valores de las aperturas en valores superiores al del fcp.



**Figura 6.6** Resultados obtenidos de la implementación del LQR (Aperturas 1 hasta 4).



**Figura 6.7** Resultados obtenidos de la implementación del LQR (Aperturas 5 hasta 7 y fcp).

Otro de los factores que puede llegar a influir en dicha pendiente negativa viene dado por la dinámica del edificio, que se va enfriando poco a poco con el tiempo y "almacenando" el frío en sus componentes más lentas (paredes, suelos, y demás aislantes), y así cada vez hace falta menos potencia para mantener la temperatura de las salas

Los resultados obtenidos mediante las simulaciones són por tanto, una muestra clara de que el controlador realiza su función correctamente y que el sistema responde de forma adecuada a dicho control.





# 7 Control MPC

---

En este capítulo, se abordará el diseño y sintonización de un controlador MPC (Model Predictive Control), que representa un controlador predictivo, para el sistema desarrollado en el Capítulo 4, el cual podía ser definido como

$$x(k+1) = Ax(k) + Bu(k) \quad (7.1a)$$

$$y(k) = Cx(k), \quad (7.1b)$$

donde, como ya se dijo en su momento,  $x \in \mathbb{R}^{n_x}$  representa el vector de estados,  $u \in \mathbb{R}^{m_u}$  al conjunto de variables de actuación presentes en el sistema, por último,  $y \in \mathbb{R}^{n_y}$  que hace referencia a las temperaturas de las habitaciones, las cuales se desean controlar.

Para ello, en el capítulo presente, se enunciará en que consiste el control MPC [15] y el necesario uso de un observador de estados para obtener el estado actual del sistema en cada tiempo de muestreo, posteriormente se aplicarán dichos conocimientos al caso concreto de estudio y poder diseñar así una ecuación de optimización y, por último se analizarán los resultados obtenidos.

## 7.1 Observador de estados

Al igual que ocurre con el controlador LQR, para el controlador MPC, es necesaria la existencia de un observador de estados [7]. Para el diseño de dicho observador se puede tomar como referencia la Sección 6.1, aunque se enuncie brevemente la estructura de dicho observador.

El observador de estados permite calcular, como ya se ha enunciado en otras ocasiones, el estado actual en el que se encuentra el sistema, tomando como datos de partida el valor de las señales de salida del sistema y los valores de las variables de control, se calculan los estados en los que se encuentra el sistema en cada momento. Para dicho observador se emplea una estructura ampliada que permite realizar el cálculo de las perturbaciones existentes en el sistema que serán de utilidad para la función de coste del controlador. Para ello, usa una estructura del tipo

$$\hat{x}(k+1) = \begin{bmatrix} \hat{x}(k+1) \\ d(k+1) \end{bmatrix} = \left( \begin{bmatrix} A & 0 \\ 0 & I \end{bmatrix} - L \begin{bmatrix} C & I \end{bmatrix} \right) \hat{x}(k) + \begin{pmatrix} \begin{bmatrix} B \\ 0 \end{bmatrix} & L \end{pmatrix} \mathbf{u}, \quad (7.2)$$

donde el vector  $\mathbf{u}$  es una composición de los vectores que agrupan variables de control y las salidas del sistema, es decir, presenta una estructura del tipo

$$\mathbf{u} = \begin{bmatrix} u(k) \\ y(k) \end{bmatrix}. \quad (7.3)$$

Al igual, el cálculo de la matriz  $L$ , se puede realizar mediante la resolución del problema LQR que representa el sistema y, en un caso más práctico con MATLAB<sup>®</sup>, con ayuda del comando `dlqr` (usado para sistemas en tiempo discreto) y del paso de los correctos argumentos, se pueden obtener sin ningún tipo de problema los valores correspondientes.

## 7.2 Control MPC

El control MPC, del inglés Model Predictive Control, es una estrategia de control que presenta sus bases en el control óptimo, el cual se basa en tres puntos claves: el uso de un modelo para predecir la salida del sistema en una serie de tiempos de muestreo futuros; el cálculo de la mejor serie de secuencias futuras que minimicen una función de coste correspondiente; la aplicación en cada tiempo de muestreo de la primera señal de control, obviando el resto que han sido calculadas.

Entre las ventajas que presentan este tipo de esquemas de control [3], se destaca que es aplicable a casi cualquier tipo de sistema, trata de forma sencilla los casos de sistemas multivariables, así como la implementación intrínseca de restricciones a tener en cuenta en la fase de minimización. Sin embargo, también presenta algunos inconvenientes como es la necesidad de requerir un modelo muy preciso para la predicción de la salida del sistema además de la carga computacional que conlleva dicho algoritmo.

Como se ha dicho, la primera fase del control MPC consiste en la realización de una predicción de la salida y estado del sistema, cuyo objetivo es que con datos anteriores de los que se disponga, obtener y determinar el estado actual en el que se encuentra el sistema. Para ello, se debe disponer de un modelo que proporcione información del tipo

$$\frac{dx}{dt} = f(x, u, t) \quad (7.4a)$$

$$y = h(x, u, t) \quad (7.4b)$$

$$x(t_0) = x_0, \quad (7.4c)$$

donde  $x \in \mathbb{R}^{n_x}$  representa los estados,  $u \in \mathbb{R}^{n_u}$  son las variables de control,  $y \in \mathbb{R}^{n_y}$  las salidas del sistema. Además, se indica una condición inicial del estado del sistema, la cual en el instante cero suele definirse como nula.

Como el caso de estudio es un sistema discreto, el modelo de predicción que se usa es el disponible para tiempos discretos o también conocido como modelo de espacio de estados, dicho modelo ya ha sido expuesto en numerosas ocasiones, pues también ha sido objeto de estudio en capítulos anteriores. El modelo es el siguiente

$$x(k+1) = Ax(k) + Bu(k) \quad (7.5a)$$

$$y(k) = Cx(k), \quad (7.5b)$$

Con dichas ecuaciones, se dispone ya de un modelo de predicción bastante correcto para la realización del controlador. Como se dijo, la segunda fase del problema consiste en la minimización de una función de coste con el objetivo de obtener unas secuencias de actuaciones óptimas. Para su diseño se debe analizar el problema en cuestión y elegir una que acorde con las características de cada uno.

Para ello, se suele partir de una básica que satisfaga un *Linear Quadratic Problem* el cual permita la minimización de una función de coste básica a la que se le vayan añadiendo diversos términos. Esta función de coste básica puede verse como la medición de la desviación entre la diferencia entre los estados y variables de control con un valor cero. Por ello, dicha función puede representarse en su forma más básica como

$$V(x(0), u) = \sum_{k=0}^{N-1} [x(k)^T Q x(k) + u(k)^T R u(k)] + x(N)^T P x(N) \\ s.a. = x(k+1) = Ax(k) + Bu(k), \quad (7.6)$$

donde N representa el número de tiempos de muestreos futuros a considerar, conocido como horizonte de predicción. El objetivo con dicha función es encontrar una secuencia de actuaciones futuras que minimicen lo máximo posible su valor. Estas secuencias de actuaciones, están representadas con

$$u = \{u(0), u(1), u(2), \dots, u(N-1)\}. \quad (7.7)$$

Como puede observarse, además del diseño de la función objetivo, se deben sintonizar los parámetros Q y R, donde en función de ambos se puede conseguir una respuesta de un tipo u otro. Cuando el valor de Q es bastante alto con respecto al de R, lo que se persigue es el conseguir una rápida minimización de los estados, es decir, que el sistema lleve los estados al punto de equilibrio lo antes posible. Si por el contrario, el valor de

R es superior al de Q, se consiguen que, penalizando la rapidez del sistema, los valores de las actuaciones sean lo más pequeños posibles.

Resumiendo esto, el problema de minimización puede enunciarse como

$$\underset{u}{\text{mín}} V(x(0), u). \quad (7.8)$$

La última fase es la correspondiente a la asignación de las primeras variables aplicables al primer tiempo de muestreo obtenidas durante la realización del problema, ya que de todas las obtenidas en cada tiempo de muestreo, obviando por tanto las variables obtenidas para tiempos de muestreo posteriores. De esta forma, el valor a aplicar es

$$u^* = u(1). \quad (7.9)$$

Con esto, se puede concluir una breve explicación acerca del control MPC. Además de lo expuesto anteriormente, hay que considerar que se a la función de minimización se le pueden añadir diversas restricciones, como pueden ser: límites en las variables de control, límites en las salidas del sistema y variaciones máximas y mínimas de las actuaciones entre otras. En el siguiente apartado, todo esto se estudiará un poco más en detalle para el caso concreto de estudio, que es el aplicado al edificio.

### 7.3 Aplicación al caso de estudio

En este caso, los conceptos expuestos anteriormente, se llevarán a la aplicación del sistema concreto del que se dispone, el edificio compuesto de las siete salas. Una de las ventajas que se observan en este controlador es la posibilidad de incorporar restricciones, lo cual cambia un poco la caracterización del sistema utilizado hasta ahora. Se sabe que los modelos de las máquinas obtenidos presentan su rango de operación en un rango de la temperatura del agua que atraviesa la enfriadora entre  $5^{\circ}\text{C}$  y  $15^{\circ}\text{C}$ .

Por esta razón y dadas las circunstancias de que al controlador MPC se puedan añadir restricciones, el nuevo sistema identificado tendrá una nueva salida a controlar que será la temperatura del agua en la salida de la enfriadora ( $T_{wsnext}$ ). De esta forma, la estructura del nuevo sistema identificado es

$$x(k+1) = Ax(k) + Bu(k) \quad (7.10a)$$

$$y(k) = Cx(k), \quad (7.10b)$$

donde  $x \in \mathbb{R}^8$  representa el vector de estados,  $u \in \mathbb{R}^8$  al conjunto de variables de control, por último,  $y \in \mathbb{R}^8$  a las temperaturas de las habitaciones y la temperatura del agua a la salida de la enfriadora.

Por otro lado, la función de coste [11] [9] está compuesta por varios términos, donde todos ellos, representan un fin a conseguir en el cálculo de la secuencia óptima de acciones de control. En dicha función de coste se tienen que establecer una serie de restricciones entre las que caben destacar las siguientes:

- La primera restricción a tener en cuenta es la dinámica que sigue el sistema, es decir el modelo de espacio de estados que ha sido elaborado. Con esta dinámica se calculan algunas variables que son necesaria en la ecuación de predicción.
- Una de las restricciones se encuentran en los valores que pueden tomar las variables de control. En relación a las aperturas de las válvulas, estas pueden variar su apertura entre cero y uno, por lo que los valores calculados deben encontrarse entre ambos valores. Por otro lado, y de forma análoga ocurre con el factor de carga parcial, el cual debe estar comprendido en el mismo rango de valores que las aperturas.
- De forma complementaria se establecen restricciones en las variables de salida. Las temperaturas de las salas están representadas por las primeras variables, mientras que la última de ellas representa la temperatura del agua de salida en la enfriadora, en la cual es realmente importante asegurar su cumplimiento ya que los equipos están diseñados para un rango de valores determinados.
- Por último, en un estado estacionario ficticio, la dinámica del sistema debe seguir cumpliéndose, sabiendo además que los valores de las ecuaciones se mantienen constante y no varían. Esto puede expresarse como que el estado siguiente debe ser igual que el anterior.

La función de coste propuesta, junto con el conjunto de restricciones que debe satisfacer, es

$$\begin{aligned} \min_{x, u, x_s, u_s} \quad & \sum_{k=0}^{N-1} \begin{bmatrix} x(k) - x_s \\ u(k) - u_s \end{bmatrix}^T \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix} \begin{bmatrix} x(k) - x_s \\ u(k) - u_s \end{bmatrix} + [x(N) - x_s]^T P [x(N) - x_s] \\ & + [y_s + d - y_{ref}]^T S [y_s + d - y_{ref}] + [u_s - u_{op}]^T J [u_s - u_{op}] \end{aligned} \quad (7.11a)$$

$$s.a \quad x(k+1) = Ax(k) + Bu(k) \quad (7.11b)$$

$$y(k) = Cx(k) \quad (7.11c)$$

$$x_s = Ax_s + Bu_s \quad (7.11d)$$

$$y_s = Cx_s \quad (7.11e)$$

$$x(0) = \hat{x}_0 \quad (7.11f)$$

$$u \in \mathcal{U} \quad (7.11g)$$

$$u_s \in \mathcal{U} \quad (7.11h)$$

$$y \in \mathcal{Y} \quad (7.11i)$$

$$y_s \in \mathcal{Y} \quad (7.11j)$$

donde,  $u_s \in \mathbb{R}^8$  representa a las variables de control en el estado estacionario,  $y_s \in \mathbb{R}^8$  a las variables de salida medidas en el estado estacionario,  $x_s \in \mathbb{R}^8$  son los estados estacionarios,  $\mathcal{Y}$  representa el conjunto de restricciones aplicables a las salidas del sistema,  $\mathcal{U}$  hacen referencia a las restricciones en las variables de control,  $x_0 \in \mathbb{R}^8$  es el estado inicial del que parte el sistema en cada tiempo de muestreo, al igual que ocurre con los valores de la variable  $d \in \mathbb{R}^8$  que ambos son calculados mediante el observador y, por último,  $y_{ref} \in \mathbb{R}^8$  son el conjunto de referencias establecidas para cada una de las salidas. A continuación, se exponen de una forma más detallada cada uno de los términos que componen la función a minimizar:

- El primero de ellos, representa el seguimiento de los estados que se deben lograr durante el cálculo, es decir, la diferencia entre el estado en cada iteración ( $x(k)$ ) y el estado estacionario al que se debe llegar. Este término es acumulativo, o en otras palabras, debe calcularse y tantos tiempos de muestreo como valga el parámetro del horizonte de predicción ( $N$ ). Por otro lado, la ponderación de este término está marcada por el valor de la matriz diagonal  $Q$ .
- Otro de los términos que se debe calcular y sumar  $N$  veces, representa la diferencia entre el valor de la actuación en cada instante y su valor estacionario. Dicho término cobra mas o menos protagonismo en base al valor de la matriz diagonal  $R$ .
- En relación a los términos que no deben acumularse  $N$  veces, el primero que encontramos, pondera el valor el estado justo en el tiempo de muestreo después del valor de  $N$ . Este término, se conoce como coste terminal y se encuentra ponderado por el valor de la matriz  $P$ , la cual es el resultado de la solución de la ecuación de Riccati en tiempo discreto. El coste terminal surge del control óptimo LQR y garantiza la estabilidad del controlador MPC.

$$A^T P A - P - (A^T P B)(B^T P B + R)^{-1}(B^T P A) + Q = 0. \quad (7.12)$$

- Uno de los términos que asegura el cumplimiento de la referencia en el control, está ponderado por la matriz  $S$ , donde se debe prestar especial atención, puesto que ahora el sistema de estudio está compuesto de una nueva salida, la temperatura del agua, la cual no debe seguir fielmente una referencia si no que debe mantenerse dentro de unos límites establecidos. Para ello el peso de la última componente de la diagonal de la matriz  $S$  (conocida como peso de seguimiento), debe ser bastante inferior al de las otras para aportar un poco de holgura al no cumplimiento de la referencia. Volviendo al término en cuestión, este básicamente evalúa y pondera la diferencia en valor absoluto entre el valor de la referencia y el de la salida en estado estacionario, incluyendo en este las posibles perturbaciones.
- Por último se incluye también un término económico en la función, el cual permite minimizar el factor de carga parcial lo máximo posible, tomando así valores más altos para las aperturas de las válvulas con respecto al fcp. Gracias a esto, se garantizará un menor consumo de energía eléctrica y por tanto un menor coste económico. Este término ni mucho menos tiene por objetivo el aportar una cifra o valor a dicho coste.

A la hora de establecer la sintonización de cada uno de los valores de los parámetros hay que llegar a un compromiso entre rapidez, sobreoscilación y el cumplimiento de la referencia. Para ello, hay que hacer una modificación adecuada en los valores de las matrices de forma que cuando se aumentan los valores de cada una de ellas se obligue a realizar una mayor minimización en dicho término.

En relación a la implementación de dicho bucle de control en MATLAB<sup>®</sup>, el proceso puede dividirse en varios pasos: en primer lugar, se deben leer cada una de las salidas del sistema, una vez se hayan leído, con ayuda del observador y el controlador, se calcula la señal de actuación y, por último, se aplican dichos valores calculados al sistema en cuestión.

Antes de realizar este bucle, se deben realizar una serie de inicializaciones que permiten poder realizar el proceso de forma exitosa. En primer lugar, para lograr la determinación de cada una de las matrices que componen la función de minimización se deben realizar dos acciones. Por un lado, se debe usar la función *dlqr*, con la cual se resuelve la ecuación de Riccati y aporta el valor de la matriz P. El resto de matrices pueden definirse manualmente sabiendo que son matrices diagonales. De forma que el código es:

```
% Calculo de L observador
W = 0.01.*eye(nx+ny);
V = 0.001.*eye(ny);
L = dlqr(A_gorro',C_gorro',W,V);
L = L';

% Calc K, P
Q = 1.*eye(nx);
R = 0.1.*blkdiag(1,1,1,1,1,1,1,10);
[K,P]=dlqr(A,B,Q,R);
K=-K;

% Definicion de S y J
S = 100000.*blkdiag(1,1,1,1,1,1,1,0.01);
J = 0.01.*blkdiag(0.1,0.1,0.1,0.1,0.1,0.1,0.1,1);

% Termino para las restricciones blandas
epsilon = 1000;
```

Como se puede observar en el código, aparece un término *epsilon* ( $\epsilon$ ), el cual es usado para la incorporación de una restricción blanda en las variables de salida medida, pues se producen algunos problemas cuando se introduce restricción en estas variables de forma que se facilita el proceso de iteración y convergencia del algoritmo de optimización introduciendo una pequeña holgura para lograr realizar la simulación. Este término, tiene un peso muy grande para evitar que la holgura calculada sea muy grande y por tanto las restricciones no se cumplan fielmente.

Para la creación del controlador MPC, se debe recurrir a la librería Yalmip [19]. Para ello, primero se deben establecer los puntos de operación sobre los que se han linealizado el sistema de espacio de estados, ya que para la introducción de restricciones estos puntos hay que tenerlos en cuenta. Para la declaración de restricciones se usa una estructura del tipo *params* en la cual se deben almacenar los valores máximos y mínimos que pueden tomar las variables, si estos valores se introducen en términos absolutos (números físicos reales), es cuando hay que restarle los puntos de operación sobre los que se ha linealizado el sistema. Cuando ya se dispone de la estructura, se debe crear unas variables especiales con la función *Polyhedron*, las cuales serán usadas en la definición del controlador. En otras palabras, se tiene que:

```
% Puntos de operacion
params.Uop = [0.3.*ones(nu-1,1);0.1];
params.Yop = [21.4408;21.2162;21.4408;21.4304;21.2619;21.4980;23.1351;12.8903];

% Restricciones establecidas
params.Ylb = [-50.*ones(ny-1,1);5]-params.Yop;
params.Yub = [50.*ones(ny-1,1);15]-params.Yop;
params.Ulb = 0.*ones(nu,1)-params.Uop;
params.Uub = 1.*ones(nu,1)-params.Uop;
```

```

% Creación de los polihedros
FU = [diag(1./params.Uub);diag(1./params.Ulb)];
PU = Polyhedron(FU,ones(2*nu,1));

FY = [diag(1./params.Yub);diag(1./params.Ylb)];
PY = Polyhedron(FY,ones(2*ny,1));

```

Para la creación del objeto del controlador, se deben diferenciar principalmente dos tipos de variables, las llamadas *constr*, son las que representan a las restricciones, mientras las correspondientes a *cost* son las que se aprecian en la función a minimizar. Mediante esto, la definición puede realizarse como:

```

% Restricciones iniciales y estacionarias
constr = constr + [x{1} == x0];
constr = constr + [xs == A*xs + B*us];
constr = constr + [ys == C*xs];

for k = 1:N
    % Funciones costes acumulativas
    cost = cost + [x{k}-xs]' * Q * [x{k}-xs];
    cost = cost + [u{k}-us]' * R * [u{k}-us];
    cost = cost + [epsilon * sum(s{k}).^2];

    % Restricciones dinamicas
    constr = constr + [x{k+1} == A*x{k} + B*u{k}];
    constr = constr + [y{k} == C*x{k}];

    % Restricciones
    constr = constr + [PY.A * y{k} <= PY.b + s{k}];
    constr = constr + [PU.A * u{k} <= PU.b];
    constr = constr + [PY.A * ys <= PY.b + s{k}];
    constr = constr + [PU.A * us <= PU.b];
end

% Resto de costes
cost = cost + [(x{N+1}-xs)'*P*(x{N+1}-xs)];
cost = cost + [(ys-ref+d)'*S*(ys-ref+d)];
cost = cost + [(us-operacion)'*J*(us-operacion)];

% Creación del objeto MPC
options = sdpsettings('solver','quadprog','verbose',1);
ctrl = optimizer(constr,cost,options,vertcat(x0,d,ref),vertcat(u{1},cost));

```

Como se puede ver, este código lo que crea es, básicamente un bloque llamado *ctrl*, al cual se le aportan los valores del estado calculado por el observador, la referencia y el conjunto de estados calculados por el observador referentes a las salidas del sistema (sistema ampliado), el cual está representado por la variable *d*. Lo expuesto aquí, puede verse representado en la Figura 7.1.

Una vez se han realizado estos cálculos y definiciones previas, se puede proceder al primer paso que se debe realizar en el bucle de control, la lectura de las salidas del sistema, es decir, las temperaturas de las salas y la temperatura del agua a la salida de la enfriadora. Para ello se hace uso de la variable llamada *trnInputs*, que es usada para la comunicación entre MATLAB® y TRNSYS. Se debe prestar atención y calcular dichas variables en términos relativos al punto de operación, pues son los puntos donde han sido linealizado el sistema. Dicho esto, se tiene el siguiente fragmento de código.

```

% Lectura de todas las salidas del sistema
Tae = trnInputs(1);
y_actual(1,nStep) = trnInputs(2);

```

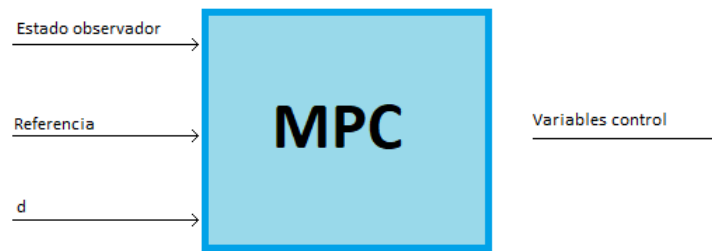


Figura 7.1 Bloque del objeto del controlador (ctrl).

```
y_actual(2,nStep) = trnInputs(3);
y_actual(3,nStep) = trnInputs(4);
y_actual(4,nStep) = trnInputs(5);
y_actual(5,nStep) = trnInputs(6);
y_actual(6,nStep) = trnInputs(7);
y_actual(7,nStep) = trnInputs(8);
y_actual_control(:,nStep-1)=[y_actual(:,nStep);Twsnext(1,nStep-1)]-params.Yop;
```

El segundo paso corresponde al cálculo de la acción de control, pero antes se debe proceder a la estimación del estado actual del sistema, obtenido gracias al observador de estados. Hay que tener en cuenta que se debe usar un sistema ampliado para poder lograr la obtención del parámetro  $d$ . Para ello se emplea un observador de la forma

$$\hat{x}(k+1) = \left( \begin{bmatrix} A & 0 \\ 0 & I \end{bmatrix} - L \begin{bmatrix} C & I \end{bmatrix} \right) \hat{x}(k) + \left( \begin{bmatrix} B \\ 0 \end{bmatrix} + L \right) \begin{bmatrix} u(k) \\ y(k) \end{bmatrix}. \quad (7.13)$$

La estimación calculada es la entrada al bloque de control, junto con la referencia que se quiere conseguir en cada tiempo de muestreo. Para comprobar que el control se produce de forma exitosa, se extrae del bloque de control también el valor del coste se tiene en cada tiempo de muestreo. Esto se puede ver como:

```
% Se obtiene la estimacion de estados
u_observador(:,nStep-1) = [u_actual(:,nStep-1); y_actual_control(:,nStep-1)];
x_observador(:,nStep) = (A_gorro-L*C_gorro)*x_observador(:,nStep-1) + [B_gorro
    L]*u_observador(:,nStep-1);

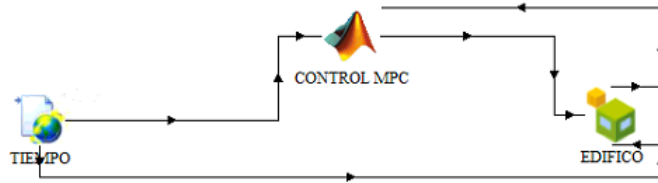
% Se calcula los valores de la variable de control
ctrlsol = ctrl([x_observador(1:nx+ny,nStep);yref(:,nStep)]);
u_actual(:,nStep) = ctrlsol(1:nu);
cost(:,nStep) = ctrlsol(nu+1);
```

Por último, para la elaboración del bucle de control, solo es necesario aplicar las variables obtenidas a cada uno de los elementos físicos que representan cada una. Para ello, solo es necesaria la correcta asignación de cada una de las componentes sus elementos correspondientes.

```
% Se asignan las variables de control
Apertura(:,nStep) = u_actual(1:nu-1,nStep) + params.Uop(1:end-1);
fcp(1,nStep) = u_actual(end,nStep) + params.Uop(end);
```

El código anterior representa el último paso del bucle de control a implementar. Con esto, ya solo es necesario establecer y diseñar el esquema de simulación completo en TRNSYS y, asignar las correspondientes variables entre ellas. De esta forma que el esquema a implementar en dicho programa es que puede visualizarse en la Figura 7.2.

Como se ha visto, el controlador genera unos valores de los parámetros de Apertura y fcp óptimos para realizar un control sobre el sistema en cuestión. Estos valores deben ser aplicados, en primer lugar, sobre los



**Figura 7.2** Esquema de implementación del controlador MPC en TRNSYS.

equipos que generan o calculan la potencia necesaria que se debe aportar sobre cada una de las habitaciones. Para ello, las conexiones entre los bloques deben ser las siguientes:

- Una conexión entre el bloque del tiempo y el del controlador MPC que permita la toma del dato de la temperatura exterior del ambiente, la cual es necesaria para caracterizar el funcionamiento de la enfriadora o chiller.
- Por otro lado, la conexión realizada entre el edificio y el controlador que permite la aportación de cada una de las temperaturas de las habitaciones a dicho controlador. Estas son las entradas del controlador. Para obtener la temperatura del agua a la salida de la enfriadora, se realiza mediante la variable interna disponible en el archivo de MATLAB®.
- De forma análoga, pero en sentido contrario, el bloque de control tiene que aportar cada una de las potencias calculadas a través de los datos de aperturas y fcp del controlador. Estas potencias son las encargadas de regular la temperatura de las habitaciones.
- Por último, se tiene la conexión correspondiente entre el tiempo atmosférico y el edificio, la cual es necesaria para su caracterización y modelado de interacciones con el ambiente, es decir, el modelado de las pérdidas energéticas que se producen.

Con todo esto, ya se puede proceder a realizar las correspondientes simulaciones las cuales van a verificar el correcto funcionamiento del controlador elaborado, además de mostrar las evoluciones que presenten el resto de variables presentes en dicho controlador.

## 7.4 Resultados obtenidos

Para corroborar que la implementación del controlador MPC explicado ha sido correcta, y comprobar así su funcionamiento, se realizará una simulación en TRNSYS con el objetivo de obtener una serie de resultados que puedan ser analizados con el objetivo de extraer una serie de conclusiones.

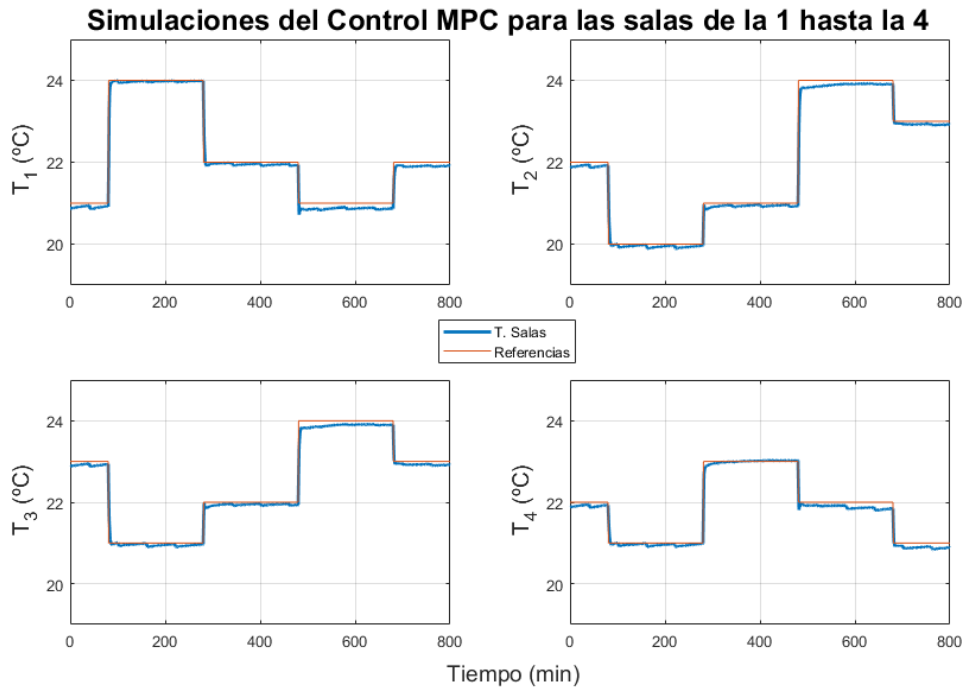
Para dicha simulación se producirán diversos cambios en las temperaturas de las habitaciones para comprobar las respuestas que genera el sistema para llegar a cumplir dicha referencia que se ha establecido previamente. Las simulaciones realizadas tendrán una duración de 800 min, donde se producirán un total de cuatro cambios en las referencias de cada una de las salas, estos cambios se producen cada 200 min. Además, también se tiene como objetivo el comprobar que se satisfacen todas las restricciones que han establecido en la realización del control MPC. En estas simulaciones el control se aplica siempre sobre el sistema real y no sobre el linealizado.

Los resultados de la simulación realizada donde se muestran las evoluciones de las temperaturas que se han obtenido, pueden ser observados en las Figuras 7.3 y 7.4.

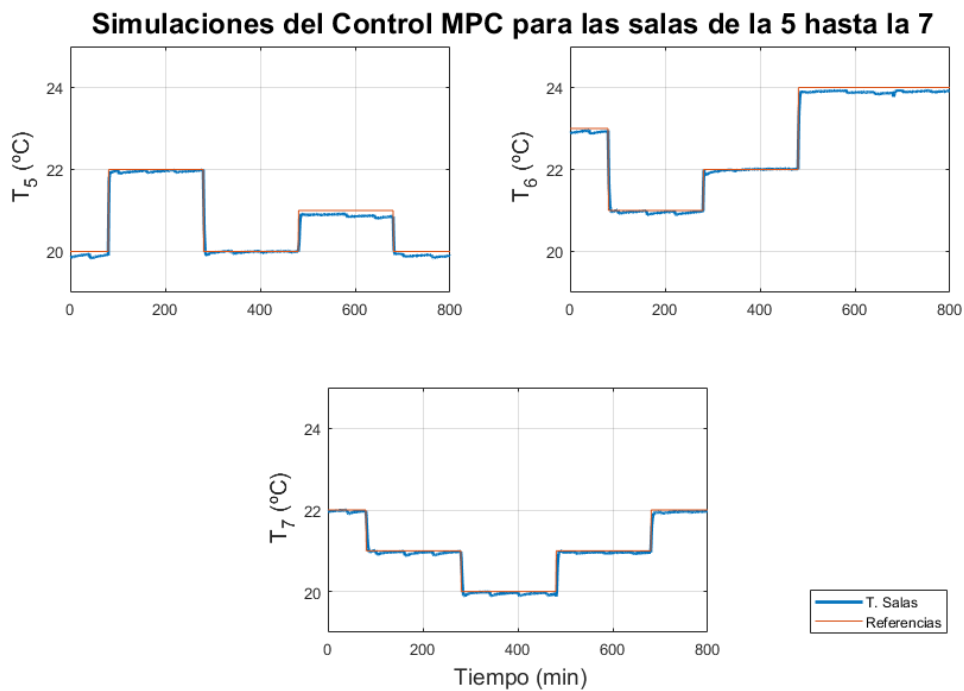
Como se observa, las referencias impuestas son seguidas y mantenidas correctamente y, el tiempo de establecimiento es extremadamente pequeño, por lo que el control es bastante aceptable. Se puede observar que las pequeñas perturbaciones provocadas por las dinámicas que representan las pérdidas por infiltraciones también son superadas con éxito.

En las Figuras 7.5 y 7.6 se puede observar las distintas evoluciones que se producen en las variables de control. Si se observan dichas gráficas, se llega a la conclusión de que la rapidez que se puede observar en el seguimiento de las referencias se puede lograr gracias a los picos que se producen en las actuaciones. Cuando dichos picos se producen hacia un valor superior al que se encuentran (válvula se abre), se observa como la temperatura de la sala correspondiente disminuye, mientras que cuando se producen a un valor inferior (válvula se cierra), las temperaturas aumentan.



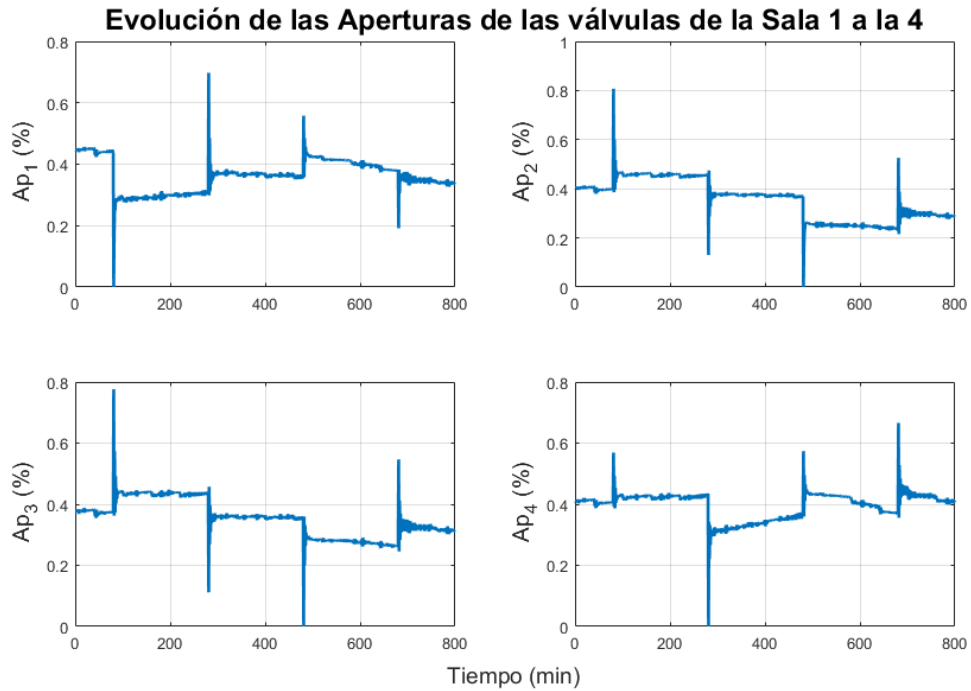


**Figura 7.3** Resultados obtenidos de la implementación del MPC (Salas 1 hasta 4).

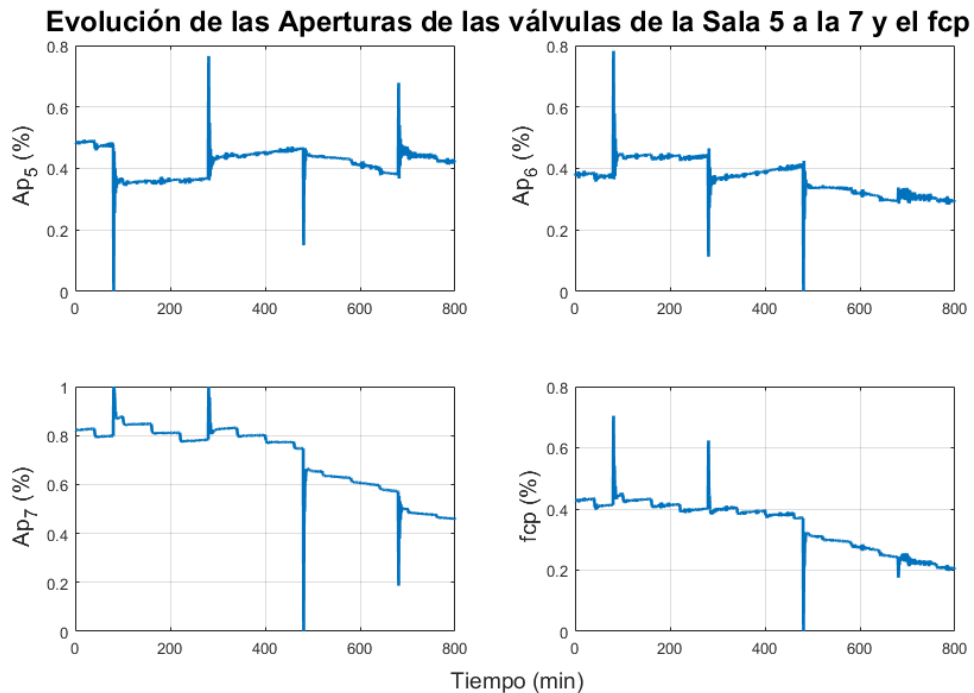


**Figura 7.4** Resultados obtenidos de la implementación del MPC (Salas 5 hasta 7).

De forma algo similar, pero algo más compleja, puesto que influye sobre la temperatura de todas las salas, ocurre con la evolución del fcp. Sin embargo, lo que hay que destacar realmente, es como los valores de las aperturas de las válvulas siempre se encuentran en un valor superior al del fcp, lo que genera siempre un menor consumo que si se produjese en el sentido contrario. Este hecho provoca en la curva de la temperatura del agua algo menos de estabilidad, pues como el fcp siempre va a tender a ser más bajo que las aperturas, el agua a la salida de la enfriadora va a ver como su temperatura se verá incrementada, pero siempre manteniéndose



**Figura 7.5** Resultados obtenidos de la implementación del MPC (Aperturas 1 hasta 4).

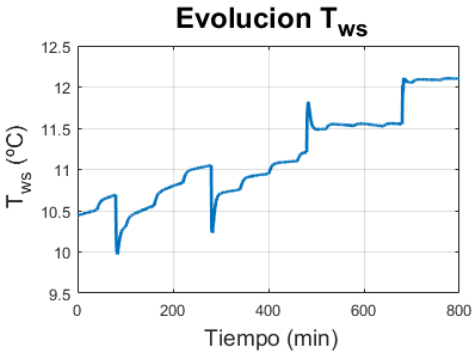


**Figura 7.6** Resultados obtenidos de la implementación del MPC (Aperturas 5 hasta 7 y fcp).

dentro de los límites establecidos. Cuando dicha temperatura se encuentre cerca del límite superior, el fcp debe aumentar para lograr disminuirla.

Todos estos fenómenos se pueden observar en la Figura 7.7, donde la pendiente positiva de la temperatura del agua se produce por la tendencia negativa que se observa en el fcp.

Con estas simulaciones, se llega a la conclusión de que el controlador realiza su función adecuadamente y las referencias alcanzan todos los valores impuestos, y por tanto cumple los objetivos para los que ha sido



**Figura 7.7** Resultados obtenidos de la implementacion del MPC (Temperatura del agua de salida de la enfriadora).

diseñado, es decir el seguimiento de las referencias y el cumplimiento de las restricciones.



## 8 Ampliación del edificio

---

Hasta este momento se ha venido tratando el modelo del edificio como una única planta compuesta por un total de siete salas. Durante el desarrollo de este capítulo, lo que se propone es una ampliación del modelo del sistema, con el fin de conseguir uno mucho más ambicioso.

Es por eso que, en el presente capítulo se va a estudiar el nuevo sistema como se ha venido haciendo con el anterior, definiendo para ello el sistema en TRNSYS con las características necesarias e incorporando posteriormente los modelos de la máquina enfriadora que permita aclimatar el edificio. Cuando ya se disponga de estos modelos, se realizará una identificación de un modelo de espacio de estados que permita obtener un modelo matemático del sistema con el cual se pueda lograr la implementación de controladores como son el LQR y un MPC.

Con dicho capítulo, lo que se pretende es ampliar el conjunto de entradas y salidas del sistema en cuestión de forma que este no suponga ningún impedimento a la hora de realizar la implementación de los controladores y observar así, como toda la teoría desarrollada se puede extrapolar a cualquier número de entradas y salidas.

### 8.1 Definición del sistema

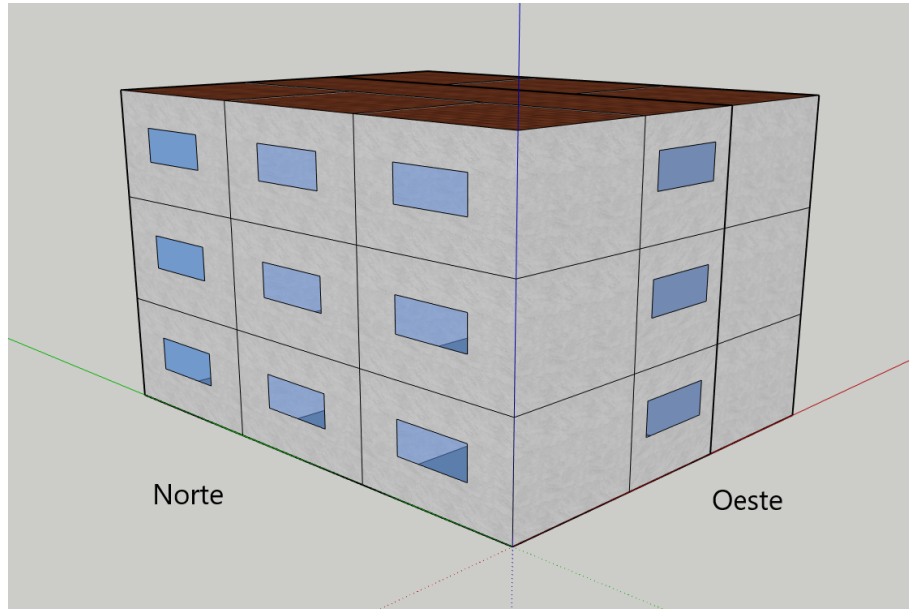
Al igual que se ha realizado con el modelo de una planta, es conveniente establecer y diseñar el modelo que se va a usar para la ampliación. Como se ha visto en el Capítulo 2, el modelo de una planta se basaba en la disposición que presentaban uno de los laboratorios de la Escuela Técnica Superior de Ingeniería de la Universidad de Sevilla, la ampliación que se va a realizar consiste simplemente en un aumento del número de plantas de las que se compone el edificio, presentando todas ellas la misma distribución y dimensiones que la expuesta en el capítulo mencionado.

Al igual que se ha hecho con el modelo de una planta, en la Figura 8.1 se puede observar el diseño del nuevo edificio que se ha realizado con Google SketchUp [16]. Como se puede observar, el edificio está formado por un total de tres plantas donde cada una de ellas está formada por un total de siete salas, siendo una de ellas el pasillo que permite la comunicación entre todas ellas. La superficie de las salas que poseen orientación norte y sur son de unos  $16 m^2$ , mientras que la del pasillo transversal a todas ellas es de  $42 m^2$  (de los cuales  $12 m$  de largo y  $3.5 m$  de ancho). La altura de cada una de las plantas se ha establecido en torno a unos  $3.5 m$ . Adicionalmente se ha añadido una ventana en cada una de las salas la cual permite considerar las interacciones energéticas que se producen entre las salas y el ambiente.

Una vez que se tiene más o menos claro el diseño que sigue el nuevo edificio, hay que centrarse un poco en la nomenclatura se va a usar para cada una de las salas que se han diseñado, de forma que durante el desarrollo del capítulo se puedan referir a ellas de una forma mucho más sencilla. De esta forma se tiene que en base a la Figura 8.1:

- Las salas 1, 8 y 15 son las situadas en la esquina noroeste en las plantas primera, segunda y tercera respectivamente.
- Las número 2, 9, y 16 son las salas que se encuentran contiguas a la sala uno en la cara norte y ubicadas en las plantas primera segunda y tercera.
- En la esquina noreste se pueden encontrar las salas 3, 10 y 17 siguiendo estas la misma distribución en términos de altura que las de los casos anteriores.
- Con respecto a la esquina sudoeste, en ella se pueden encontrar las salas 4, 11 y 18.

- Contiguas a la sala 4 y con una ventana en la cara sur se pueden encontrar las salas 5, 12 y 19.
- Las últimas salas con esta dimensión son las ubicadas en la esquina sudeste con una nomenclatura de 6, 13 y 20.
- Por último, en relación a los pasillos de cada una de las plantas, estos se denotarán como 7, 14 y 21.



**Figura 8.1** Configuración del edificio con tres plantas.

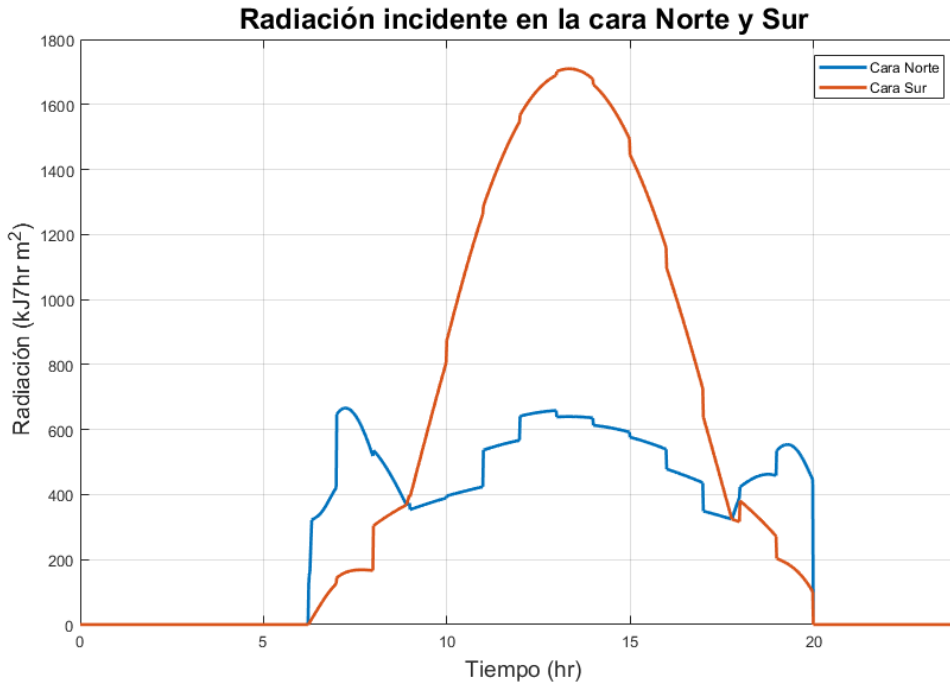
Visto como se ha denotado a cada una de las salas, ya se puede proceder a implementar cada una de ellas en el software TRNSYS, siguiendo para ello las directrices que se han estudiado en el Capítulo 2. Para la implementación, esta se puede llevar a cabo tal y como se realizó el modelo de una planta, con la pequeña salvedad de que lo que en el modelo de una planta eran definidos como techos ("roof"), en el nuevo modelo serán muros adyacentes con las superiores, de forma que los nuevos techos sean los de la tercera y última planta.

Cuando se disponga del modelo del edificio diseñado en TRNBuild, se puede elaborar ya el esquema con el que se van a poder obtener las evoluciones de cada una de las temperaturas de las salas a lo largo de un día con la influencia del tiempo atmosférico sobre el modelo en cuestión. Para ello se deben realizar las conexiones pertinentes entre cada uno de los bloques como son el del edificio y el del tiempo atmosférico, resultando así un esquema como el que se puede observar en la Figura 8.2, en la que el bloque denotado como *Type 155*, no es más que un fragmento de código que permita obtener los datos de la simulación para poder obtener así las distintas gráficas a representar.



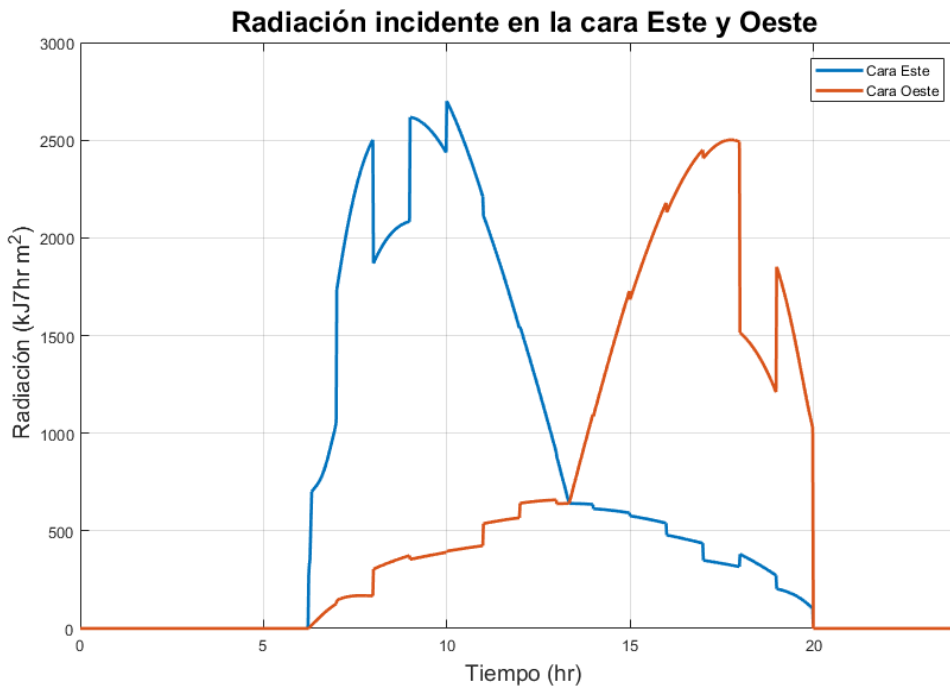
**Figura 8.2** Esquema conexionado para el modelo de tres plantas.

Para comprobar si el diseño y la implementación se han realizado de forma correcta, se va a realizar una simulación en la que se puede observar cómo afecta la influencia de la radiación solar y de la temperatura exterior a cada una de las salas del edificio. Para ello, se lleva a cabo una simulación con una duración de un día. Las radiaciones que incidentes sobre cada una de las caras del edificio, se pueden visualizar en las Figuras 8.3 y 8.4.

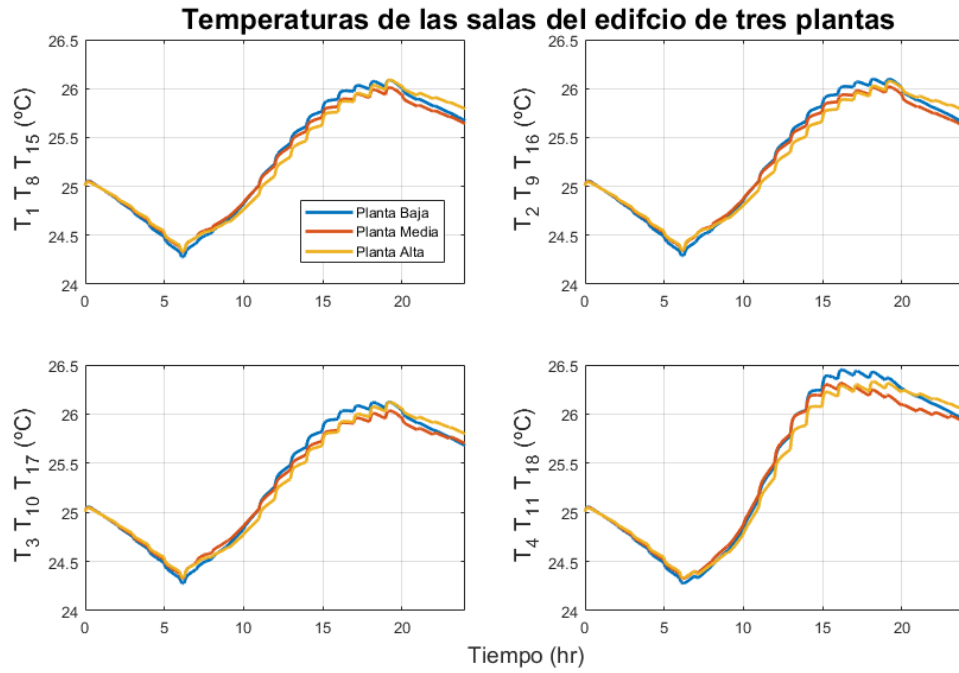


**Figura 8.3** Radiación incidente sobre las caras Norte y Sur del nuevo edificio..

Como se puede observar en las radiaciones incidentes, las conexiones han sido realizadas correctamente ya que, a primera hora del día, como el sol sale por el este, las radiaciones en dicha cara son muy superiores a las demás. De forma contraria ocurre con la cara oeste, ya que es por donde el sol se recoge. Con respecto a las caras norte y sur, en ellas se puede observar que las de la cara norte son más suaves que las de la sur ya que el emplazamiento del edificio que ha sido elegido se encuentra en el hemisferio norte. Con estas radiaciones la evolución de las distintas temperaturas se puede observar en las Figuras 8.5 y 8.6.

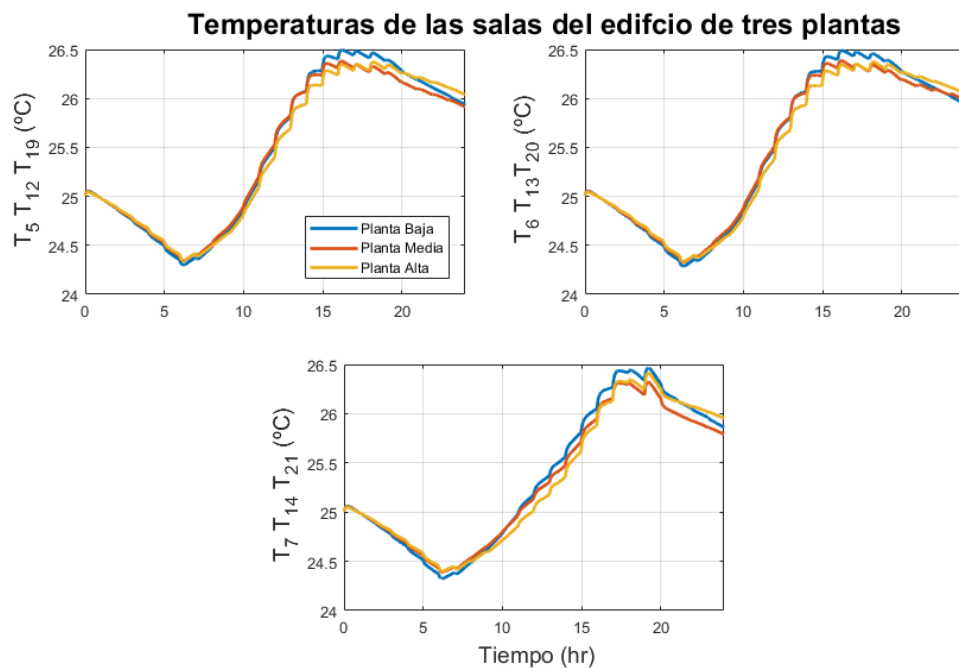


**Figura 8.4** Radiación incidente sobre las caras Este y Oeste del nuevo edificio..



**Figura 8.5** Evolución de las temperaturas en el edificio de tres plantas (Salas 1 a 4).

En los resultados que se obtienen se pueden extraer algunas ideas que verifican que la implementación llevada a cabo ha sido correcta. En primer lugar, se puede observar que las tendencias que siguen de las temperaturas son similares a las que se produce en la temperatura ambiente en un día completo, mostrando un aumento hasta las últimas horas de la tarde y una disminución desde entonces hasta las primeras horas de la mañana cuando se producen los primeros rayos de sol.



**Figura 8.6** Evolución de las temperaturas en el edificio de tres plantas (Salas 5 a 7).



Por otro lado, se puede observar como a última hora de la tarde las temperaturas de las salas situadas en la última planta sufren un notable aumento debido a la propiedad que tiene el aire caliente de dirigirse hacia las zonas superiores del edificio por tener una menor densidad cuando este aumenta su temperatura. De esta forma se puede observar el aumento de temperatura en la última planta, de forma que, el hecho de que no sean muy superiores a las de las otras plantas se puede explicar sabiendo que las condiciones iniciales de las que parte la simulación son idénticas en todas las salas, por lo que esa mayor diferencia de temperaturas entre una sala y otra, podrá ser visualizada si se simula durante bastante más tiempo. Con esto se concluye el diseño e implementación del edificio de estudio y se puede proceder a implementar la nueva máquina enfriadora que va a ser usada para lograr un acondicionamiento de temperatura en la sala.

## 8.2 Incorporación de equipos

Para poder completar el modelado del sistema completo que debe ser analizado, se deben incluir los modelos de los equipos que componen la máquina enfriadora encargada de la refrigeración del edificio. Para ello se puede seguir como referencia el Capítulo 3, si bien habrá que tener en cuenta algunos detalles que serán matizados en este apartado.

El equipo empleado para la refrigeración va a estar formado por los mismos equipos que cuando se trataba el caso de una única planta, lo cuales pueden enumerarse y definirse de una forma algo resumida como:

- Un conjunto de válvulas de tres vías que permiten regular el caudal que es derivado a los distintos fancoils. El parámetro fundamental que modela el funcionamiento de cada una de las válvulas es la apertura, por lo que será una de las variables de control del conjunto de equipos completo (máquina enfriadora). Como el nuevo edificio está constituido por 21 salas, se deben considerar por tanto un total de 21 válvulas de tres vías.
- Otros elementos clave de la máquina enfriadora son los distintos fancoils o intercambiadores de calor que permiten realizar el intercambio energético entre el fluido que los atraviesa y el aire de las salas, absorbiendo el fluido cierta cantidad de calor de las salas para poder disminuir así su temperatura. La máquina enfriadora estará formada por un total de 21 equipos como estos.
- El elemento clave que garantiza el correcto funcionamiento es la enfriadora o chiller cuya función es disminuir la temperatura del agua del equipo completo para poder garantizar así que esta pueda absorber el calor de la sala. Es aquí donde se puede observar una de las modificaciones respecto al caso con una única sala, pues la potencia elegida para dicha máquina ya no es válida para el nuevo sistema puesto que las dimensiones se han triplicado y la potencia demanda por el nuevo edificio es muy superior a la anterior. Por ello se ha elegido un chiller con una capacidad nominal de  $73.5\text{ kW}$ . La variable que permite modelar su funcionamiento es el factor de carga parcial o fcp que determina la cantidad de potencia que se emplea para enfriar el agua que circula por su interior.

Hay que destacar que las ecuaciones y los coeficientes que modelan el comportamiento de cada uno de los equipos se siguen manteniendo ya que el funcionamiento de los equipos no cambia, ya que únicamente se hace uso de más unidades para poder adaptar la máquina al nuevo edificio. El esquema que recoge como se deben conectar todos los equipos y el conjunto de variables que usa cada uno de ellos tampoco presenta ninguna diferencia con el del caso nuevo, dicho diagrama es el que se puede observar en la Figura 3.3.

Con todo lo que se ha enunciado hasta ahora, el código en MATLAB<sup>®</sup> que permite implementar los equipos de una forma correcta sigue el mismo esquema que los casos anteriores realizando una pequeña modificación en parámetros como son el caudal de agua total del equipo que ahora se puede cuantificar en  $210\text{ l/min}$ , de forma que sigan circulando  $10\text{ l/min}$  por cada válvula. Además, el número de veces que hay que llamar a cada una de las válvulas y fancoils pasan de ser 7 a 21, ya que se ha triplicado el número de estos dispositivos. Esto puede observarse en el siguiente fragmento de código:

```
%% Tratamos en primer lugar a la valvula
for i=1:21
    [qsi(i,nStep) qbi(i,nStep)] = Valve(qin(i,1),Apertura(i,nStep));
end

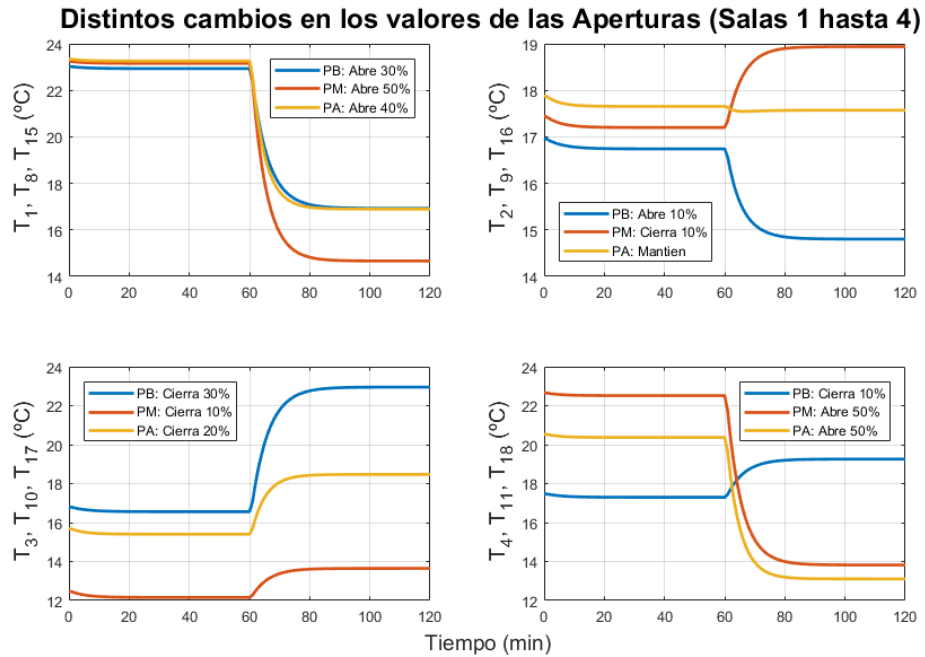
%% Ahora se intercambia calor entre la habitacion y el fancoil
for i=1:21
```

```

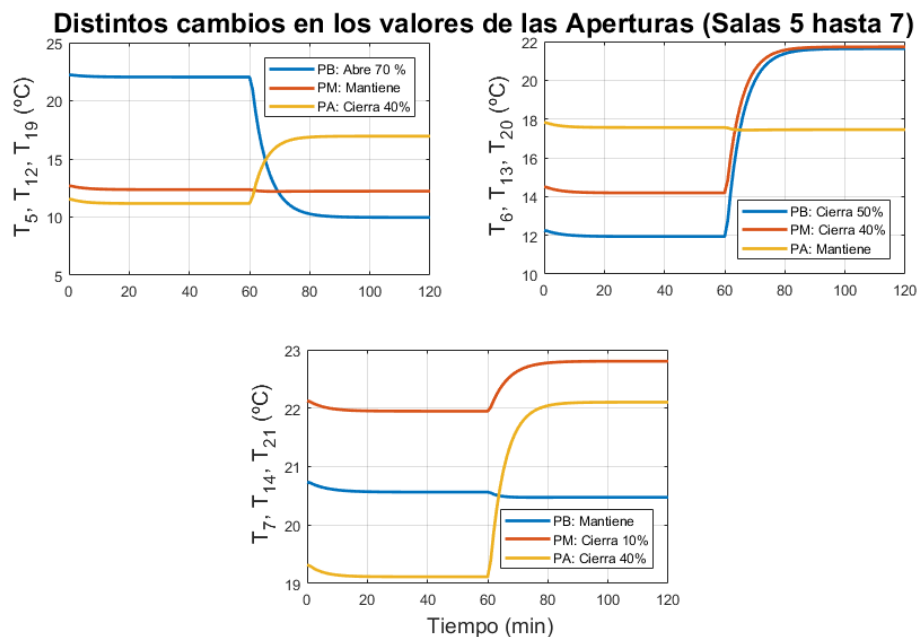
[Potencias(i,nStep) Tout(i,nStep)] = fanCoil(Twsnext(1,nStep-1), qsi(i,nStep
    ));
end

```

Para comprobar que se ha realizado correctamente, se va a realizar una simulación en la que se lleven a cabo distintos cambios en los valores que toman las aperturas de las válvulas y poder observar así la influencia que tiene cada una de las variables (Aperturas y fcp) sobre la temperatura de todas las salas del edificio.



**Figura 8.7** Cambios que se producen por la apertura de válvulas en el nuevo modelo (Salas 1 a 4).



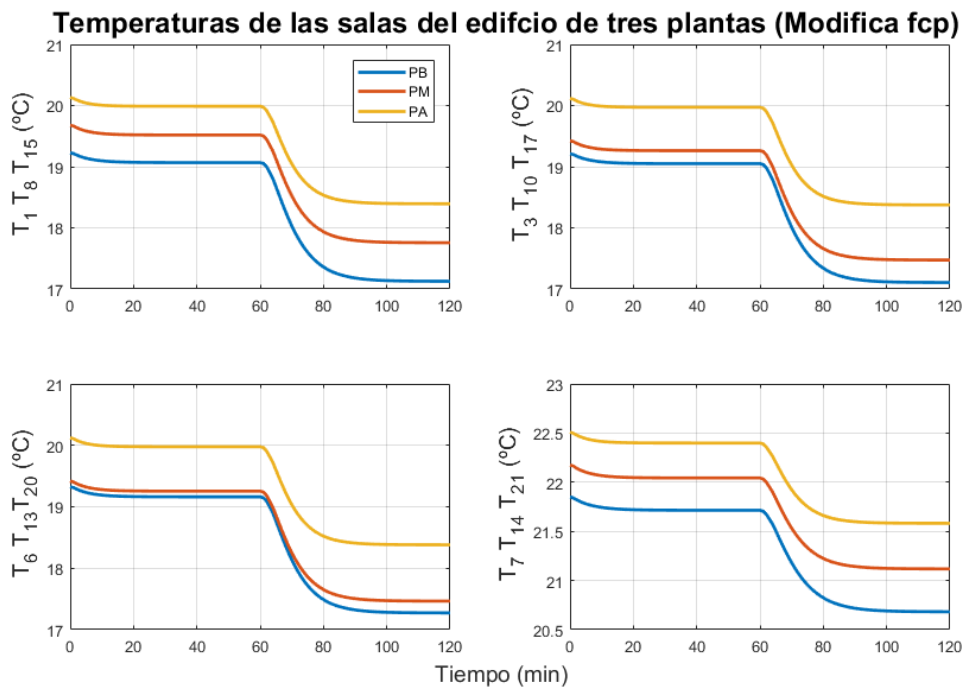
**Figura 8.8** Cambios que se producen por la apertura de válvulas en el nuevo modelo (Salas 5 a 7).

En las Figuras 8.7 y 8.8 aparecen representada cada una de las evoluciones que siguen las temperaturas de las salas existentes. En relación a la nomenclatura de la leyenda hay que puntualizar el significado de PB (Planta Baja), PM (Planta Media) y PA (Planta Alta).

De las figuras pueden extraerse algunas ideas que verifiquen que la implementación del modelo de la máquina enfriadora ha sido el correcto:

- La primera conclusión que se puede obtener con una vista rápida sobre las gráficas, se encuentra en que se puede afirmar que, cuando la apertura de una válvula se modifica hacia un valor superior (se abre más la válvula), la temperatura de la sala o espacio disminuye, mientras que si esta se cierra se produce un aumento de la temperatura. Al igual que ocurriría con el modelo de una planta, este fenómeno se puede explicar de forma física, por lo que se concluye que se ha realizado correctamente.
- En las gráficas se puede observar cómo no todas ellas parten de la misma temperatura, lo cual es lógico pues el valor inicial de las aperturas no es el mismo y por ello estas han variado durante instantes previos a la serie de datos que han sido escogidos para realizar la representación, ya que lo representado no es más que un fragmento de la simulación realizada.

La otra variable que influye en el comportamiento de la máquina enfriadora es el factor de carga parcial, que para observar su influencia sobre la temperatura de las salas se va a realizar una simulación en la que se aumente su valor (de un 0.2 a un 0.3), de forma que la temperatura del agua a la salida del chiller disminuya considerablemente.



**Figura 8.9** Cambios que se producen por la variación de fcp en el nuevo modelo.

Como se puede observar en los resultados de la Figura 8.9, cuando dicho valor se aumenta, se puede ver que las temperaturas de las habitaciones disminuyen, ya que la temperatura a la que sale el agua de la enfriadora disminuye, logrando así, que la potencia absorbida por el agua tras su paso por el fancoil también aumente, generando por tanto una disminución en la temperatura de la sala.

También se puede destacar que el enfriamiento producido en las salas que representan los pasillos es inferior al del resto de salas, ya que el espacio que representa dichas salas es superior al del resto, por lo que este necesita mayor potencia para lograr el mismo enfriamiento.

De esta forma se da por concluida la implementación de los equipos que permiten enfriar las salas, con lo que ya se dispone del modelo del nuevo edificio con los nuevos equipos, que constituyen el nuevo sistema completo. Ahora es necesario obtener un modelo matemático mediante identificación que permita modelar el sistema completo.

### 8.3 Identificación del sistema

Para llevar a cabo la identificación, al igual que se hizo en el caso de una planta, se va a recurrir al toolbox de identificación que está disponible en MATLAB®. Antes de plantear el modelo matemático y definir el conjunto de variables que lo conforman, es importante saber cuáles van a ser las entradas y las salidas del sistema en cuestión.

Al igual que ocurre con el caso de una planta, las entradas del sistema siguen siendo las aperturas de las válvulas y el fcp, mientras que las salidas del sistema son cada una de las temperaturas de las distintas salas existentes. De esta forma se disponen de un total de 22 entradas (21 aperturas de válvulas y el fcp) y de 21 salidas (todas las temperaturas de las salas). De esta forma ya se puede acotar el caso de estudio y poder establecer el modelo de espacio de estados que modelará el comportamiento del sistema completo. Estas ecuaciones pueden establecerse como

$$x(k+1) = Ax(k) + Bu(k) \quad (8.1a)$$

$$y(k) = Cx(k), \quad (8.1b)$$

en el cual ya se pueden caracterizar cada una de las componentes: donde  $x \in \mathbb{R}^{nx}$ ,  $u \in \mathbb{R}^{22}$  e  $y \in \mathbb{R}^{21}$ . Como se puede observar ya solo es necesario conocer el número de estados que definen el sistema.

Para poder determinar el espacio de estados se deben definir el primer lugar el conjunto de pruebas que se deben realizar para poder aplicar el algoritmo de identificación empleado. El objetivo principal de las simulaciones a realizar es poder ilustrar cómo afecta la variación de cada una de las entradas sobre las salidas del sistema, generando para ello una modulación en forma de escalón sobre las entradas de forma que el cambio de la entrada sea instantáneo y de una amplitud determinada, siendo como es lógico el resto de las entradas constantes durante toda la simulación. Adicionalmente, una de las condiciones de las pruebas a realizar es que la temperatura ambiente va a ser considerada constante durante los ensayos, de forma que posteriormente sea considerada como una fuente de perturbaciones que afecta a las salas.

Para realizar las mencionadas pruebas, en los escalones sobre los valores de las aperturas van a producirse variaciones de un 0.3, siendo el aumento en el valor del factor de carga parcial de un 0.1. De esta forma se deben realizar un total de 22 pruebas, una por cada entrada disponible, siendo estas muy similares a las que se realizaban en el Capítulo 4 (más concretamente en la Figura 4.3)

Una vez que se hayan realizado todas las pruebas en cuestión se puede aplicar el código de identificación estudiado que permite determinar las matrices que conforman el modelo de espacio de estados. Tras realizar diversas pruebas con el número de estados a utilizar, se ha llegado a la conclusión de que los mejores resultados se obtienen para un total de 22 estados. Con esto, el parámetro  $nx$  ya queda definido de una forma adecuada.

Si se aplica el código de identificación y se representan los resultados obtenidos, estos se pueden mostrar en las Figuras 8.10 y 8.11. En ellas se puede corroborar que el sistema ha sido identificado de forma adecuada, ya que cuando se produce una modificación en las aperturas de las válvulas se puede observar cómo únicamente afecta a la sala donde se produce esa modificación. Por otro lado, cuando se produce una modificación en el valor del factor de carga parcial, se puede observar como este influye en el valor de todas las temperaturas de las habitaciones. Con esto ya se ha conseguido una identificación que refleja fielmente el comportamiento del sistema y puede hacerse uso de él para poder aplicar los controladores que han sido aplicados en el caso de una planta.

### 8.4 Control LQR

Para comenzar la aplicación de estrategias de control con el objetivo de intentar lograr y mantener en una referencia deseada en las temperaturas de las habitaciones existentes en el edificio, se va a proponer el uso de un controlador LQR (Linear Quadratic Regulator). Para dicho controlador se parte del conocimiento del modelo matemático de espacio de estados que se ha elaborado en el apartado anterior, este modelo ha sido definido como

$$x(k+1) = Ax(k) + Bu(k) \quad (8.2a)$$

$$y(k) = Cx(k), \quad (8.2b)$$

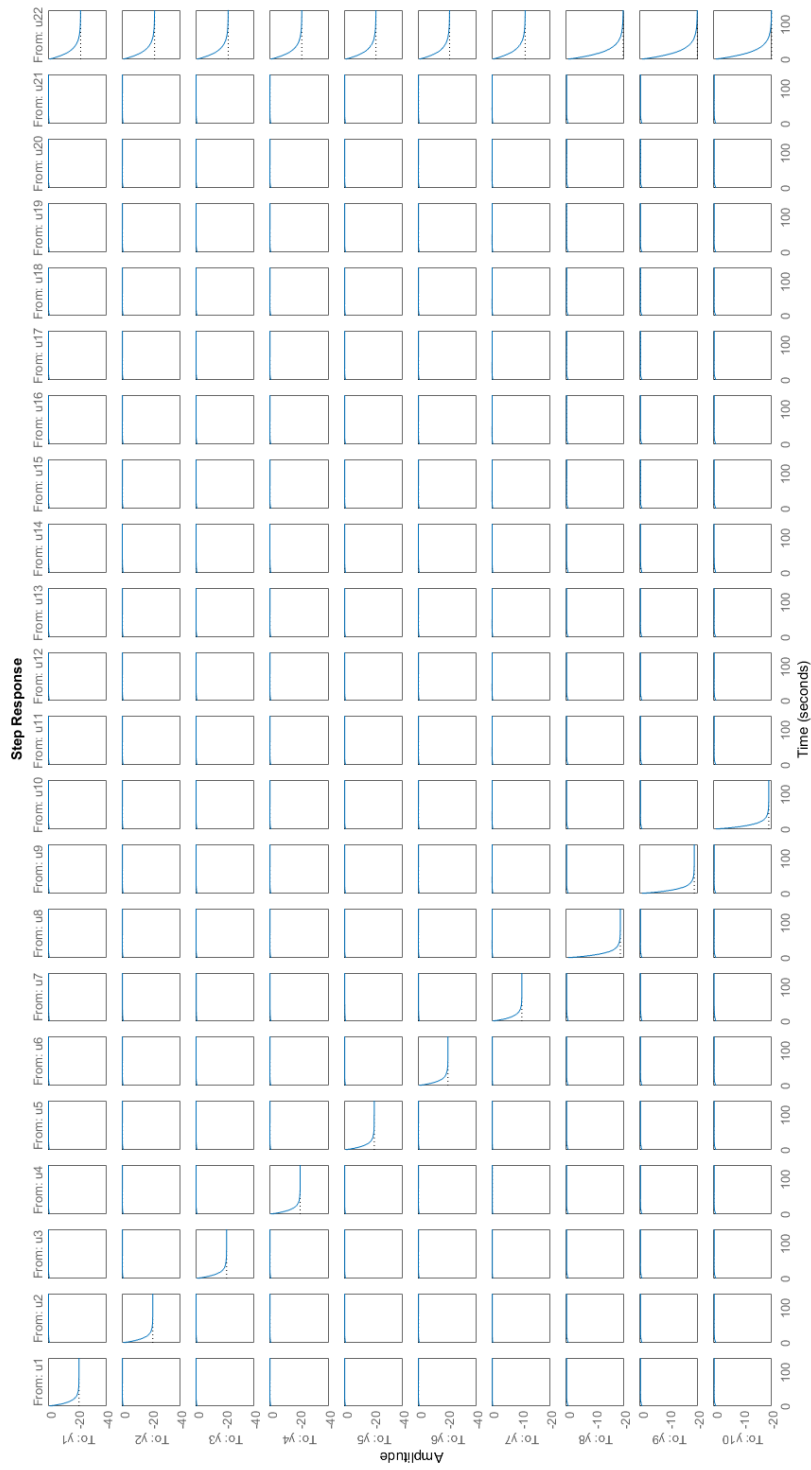


Figura 8.10 Resultados de la identificación realizada (Entradas 1 a 10).

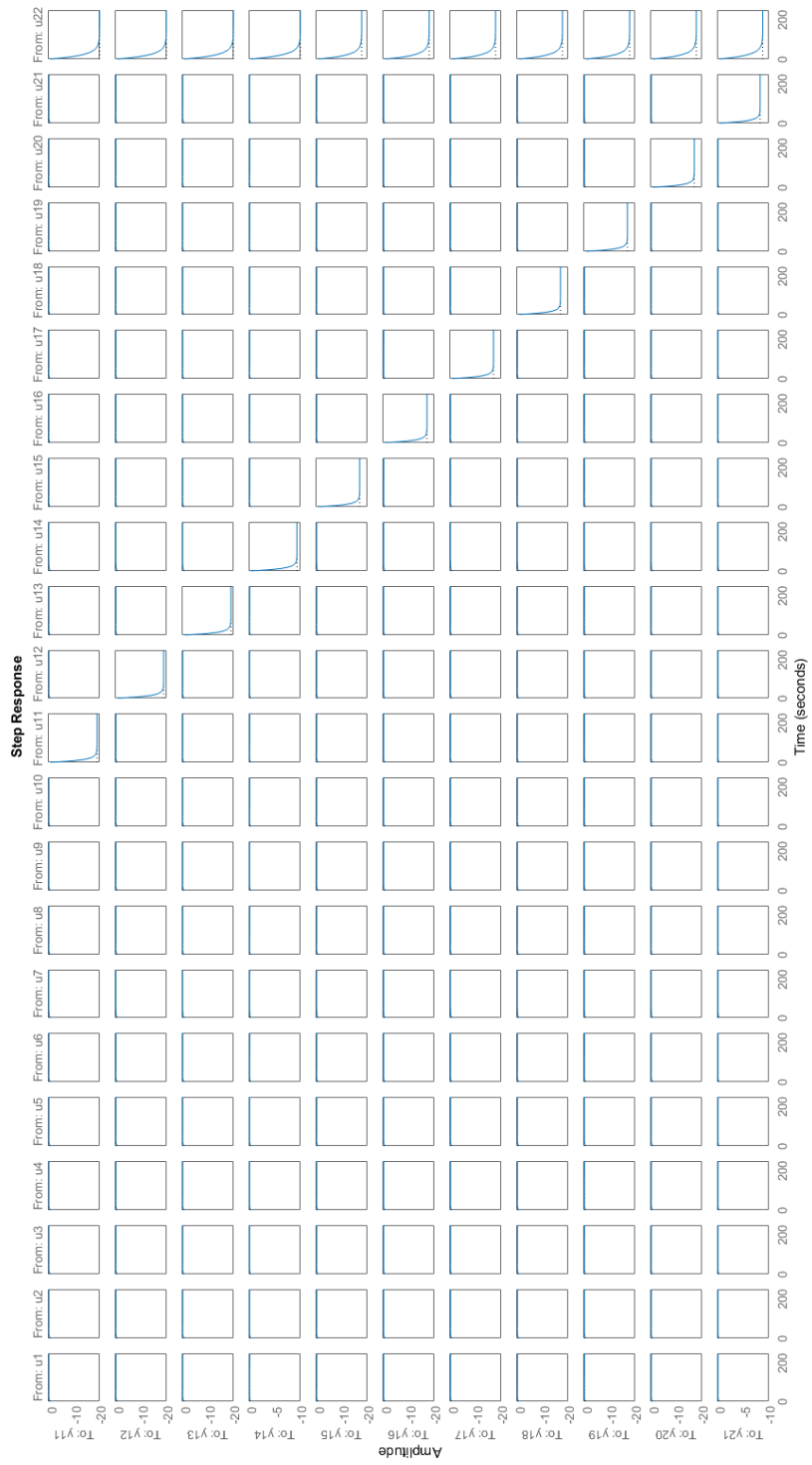


Figura 8.11 Resultados de la identificación realizada (Entradas 11 a 21).

siendo  $x \in \mathbb{R}^{22}$ ,  $u \in \mathbb{R}^{22}$  e  $y \in \mathbb{R}^{21}$ . Para poder estimar los estados reales en los que se encuentra el sistema, se debe emplear un observador de estados que permita obtenerlos a partir de las variables de control y las salidas que estas generan en el sistema. La estructura del observador de estados que se emplea puede enunciarse matemáticamente de la forma

$$\hat{x}(k+1) = (A - LC)\hat{x}(k) + [B \quad L] \mathbf{u}, \quad (8.3)$$

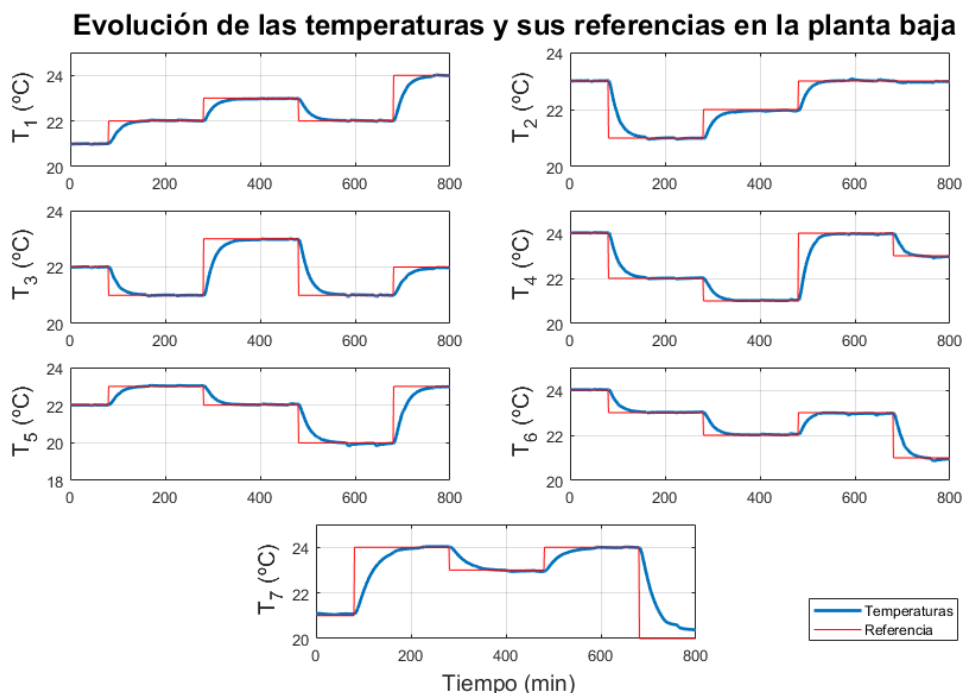
de forma que  $\mathbf{u}$  es un vector en forma de columna que almacene los valores de las actuaciones y las salidas (en ese orden) del instante  $k$ . De esta forma ya se puede estimar los distintos parámetros que en cierta forma definen el estado en el que se encuentra el sistema. La estrategia de control que se emplea está basada en una modificación del típico controlador LQR comúnmente conocido, denotado como control LQR con efecto integral. En dicho controlador, la ley de control que debe aplicarse en cada tiempo de muestreo se puede enunciar como

$$\Delta u(k) = K_z \begin{bmatrix} \Delta x(k) \\ e(k-1) \end{bmatrix} = K_x \Delta x(k) + K_e e(k-1), \quad (8.4)$$

en la que las constantes proporcionales por las que se ven multiplicadas el error y la variación de los estados, resultan de aplicar la ecuación de Ricatti sobre un nuevo sistema ampliado el cual resulta ser

$$\begin{bmatrix} \Delta x(k+1) \\ e(k) \end{bmatrix} = \begin{bmatrix} A & 0 \\ -C & I \end{bmatrix} \begin{bmatrix} \Delta x(k) \\ e(k-1) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} \Delta u(k) \\ \Delta y(k) = C \Delta x(k). \quad (8.5)$$

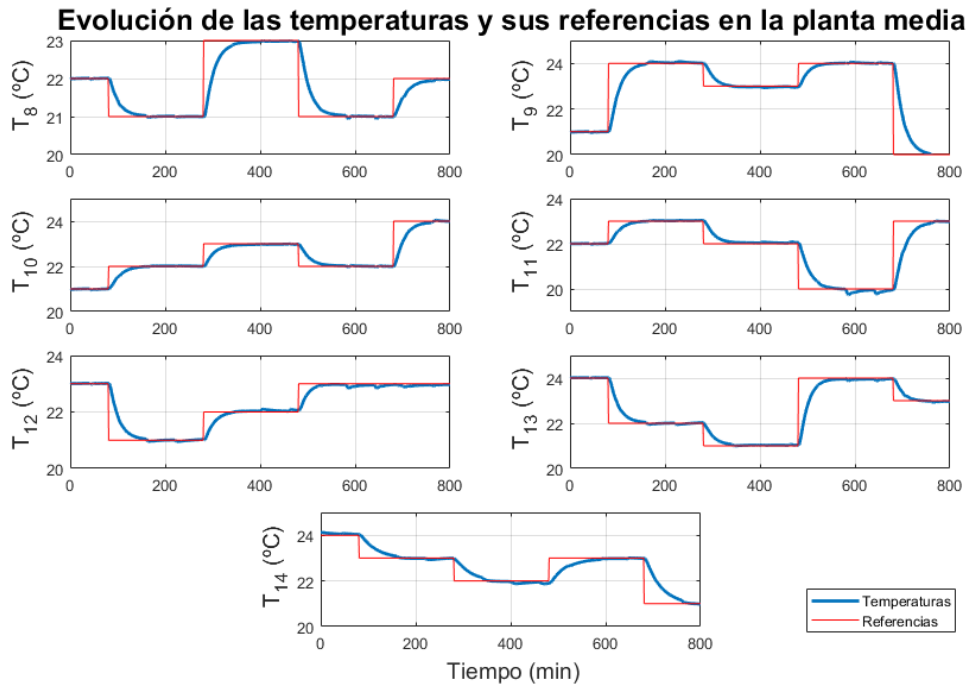
Mediante este modelado, ya se puede proceder a la implementación de los controladores en MATLAB<sup>®</sup> como se ha definido en el Capítulo 6, realizando pequeñas modificaciones de las dimensiones en algunos de los parámetros que modelan en controlador, para poder adaptarlo ahora a las nuevas dimensiones del sistema ampliado.



**Figura 8.12** Temperaturas y Referencias con un controlador LQR (Planta Baja).

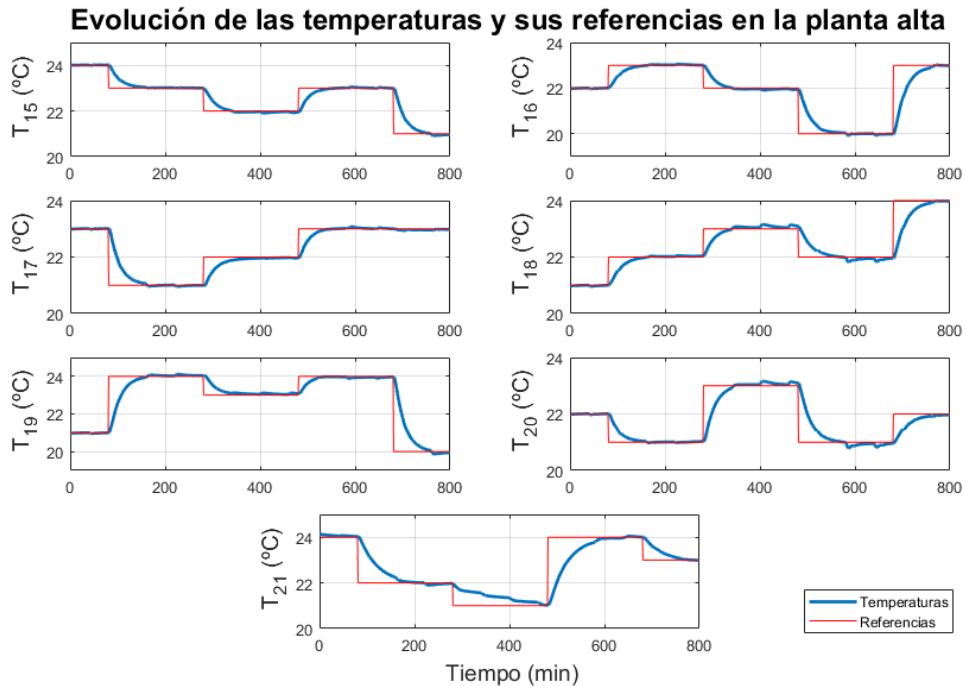
Para comprobar el correcto funcionamiento del controlador LQR implementado, se va a realizar una simulación en la que se producirán un total de cuatro cambios de referencia cada 200 min, de forma que se pueda observar cómo evoluciona cada temperatura llegando a alcanzar el valor deseado. Los resultados que

se obtienen son los que se pueden observar en las Figuras 8.12, 8.13 y 8.14.



**Figura 8.13** Temperaturas y Referencias con un controlador LQR (Planta Media).

Con los resultados obtenidos, se puede afirmar que el controlador ha sido diseñado e implementado correctamente, de forma que otorga a los actuadores los valores que permiten aportar la potencia necesaria para lograr el mantenimiento de las salas a la temperatura deseada.



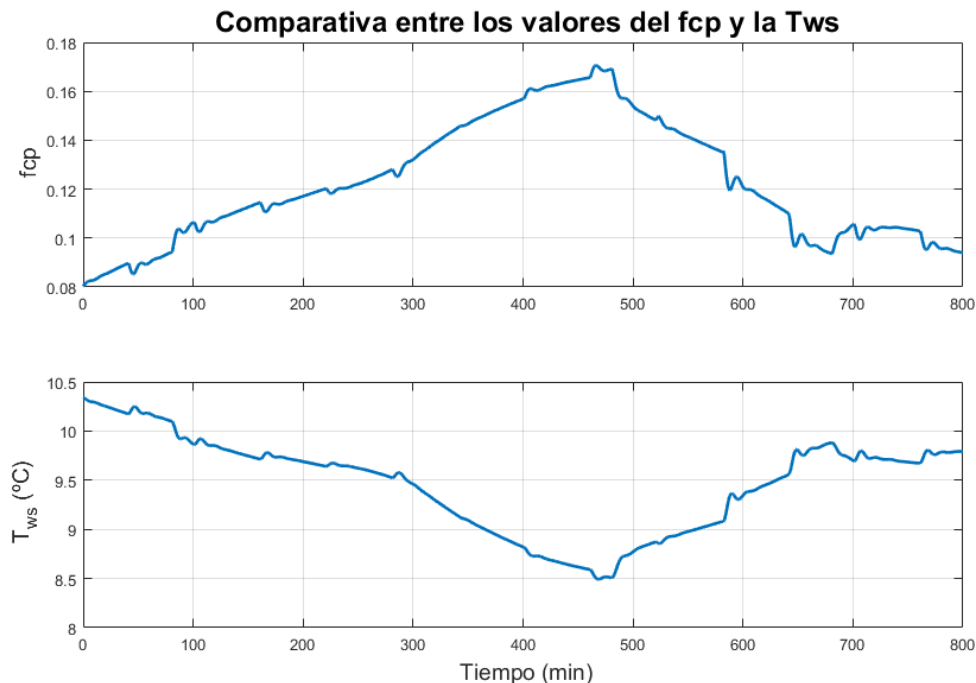
**Figura 8.14** Temperaturas y Referencias con un controlador LQR (Planta Alta).



Adicionalmente, para mostrar la coherencia de los resultados obtenidos se va a representar en la Figura 8.15 la comparación entre los valores que toman el factor de carga parcial (fcp) y la temperatura del agua a la salida de la enfriadora o chiller. En dicha imagen se puede observar que en esos instantes los valores del fcp son muy bajos, lo que no quiere decir que se haya usado una máquina demasiado potente, ya que al principio de la simulación (durante el transitorio que no se ha mostrado), cuando las temperaturas de las salas evolucionan con la temperatura del ambiente y el agua que circula por la máquina refrigeradora todavía no se encuentra acondicionada a la temperatura de trabajo, el esfuerzo que debe realizar el chiller es alto y demanda por tanto mayores valores del fcp que son cercanos a la unidad.

Por otro lado, en la imagen se puede observar como la temperatura del agua se mantiene siempre entre los niveles correctos de funcionamiento, situados estos entre  $5^{\circ}\text{C}$  y  $15^{\circ}\text{C}$ , de forma que cuando la temperatura vaya a sobrepasar tanto el límite inferior como superior, se produce una variación positiva o negativa para evitar este fenómeno. Este hecho justifica la tendencia que siguen ambas curvas representadas y aporta consistencia también a los resultados obtenidos.

Se debe destacar que los valores de las aperturas de cada una de las válvulas existentes no han sido mostrados para evitar así un elevado número de gráficas en este memoria meramente descriptiva, pero si se debe destacar que los valores obtenidos muestran unos valores que concuerdan con un aumento o disminución de sus valores, en función se quieran disminuir o aumentar las temperaturas respectivamente.



**Figura 8.15** Comparativa entre los valores de fcp y de la temperatura a la salida del chiller (Control LQR).

Con estos resultados obtenidos se puede dar por finalizado este apartado relativo a la implementación y diseño de un controlador LQR con efecto integral y se puede ver la importancia de diseñarlo de forma adecuada para poder generar así un enfriamiento de las habitaciones generando el menor gasto energético posible.

## 8.5 Control MPC

Como se ha estudiado en capítulos anteriores, se va a estudiar cómo se comporta y cómo se diseña un controlador predictivo MPC para el sistema en cuestión. En este nuevo objeto de estudio se permite incluir restricciones menos severas en el seguimiento de referencias por parte de las salidas, lo que permite poder establecer unos límites en los valores que toma la temperatura del agua a la salida de chiller, ya que este parámetro es muy crucial a la hora de caracterizar el comportamiento de la enfriadora o chiller. Los valores de la temperatura del agua que marcan el buen funcionamiento de los equipos varían entre  $5^{\circ}\text{C}$  y  $15^{\circ}\text{C}$ .

De esta forma, se puede llevar a cabo una nueva identificación del edificio en la que se contemple la temperatura de dicho agua como una salida más del sistema, de forma que estas salidas estarían compuestas por las temperaturas de las habitaciones y la variable mencionada con anterioridad. De esta forma el modelo de espacio de estados identificado puede enunciarse como

$$x(k+1) = Ax(k) + Bu(k) \quad (8.6a)$$

$$y(k) = Cx(k), \quad (8.6b)$$

donde las dimensiones de las variables que aparecen pueden establecerse como  $x \in \mathbb{R}^{22}$ ,  $u \in \mathbb{R}^{22}$  y  $y \in \mathbb{R}^{22}$ , las cuales son el espacio de estados, las variables de control y las salidas del sistema respectivamente.

Por otro lado, la acción de control obtenida mediante el controlador se puede calcular mediante la minimización de una función de coste que sigue la forma de la que era enunciada en el Capítulo 6. Esta función está compuesta por distintos términos que representan:

- El primero de ellos, corresponde al seguimiento de los estados que se deben lograr durante el cálculo, es decir, la diferencia entre el estado en cada iteración ( $x(k)$ ) y el estado estacionario al que se debe llegar, el cual se encuentra ponderado por la matriz Q.
- El segundo término representa la diferencia entre el valor de la actuación en cada instante y su valor estacionario ponderado también por una matriz R.
- Por otro lado encontramos el que verifica si se ha producido o no el cumplimiento del estado estacionario justo en el tiempo de muestreo después del valor de N. Este término, se conoce como coste terminal y se encuentra ponderado por el valor de la matriz P (solución a la ecuación de Ricatti).
- El término más importante es el que asegura el cumplimiento de la referencia en el control, está ponderado por la matriz S, donde se debe prestar especial atención, puesto que la temperatura del agua no debe seguir una referencia si no que debe mantenerse dentro de unos límites establecidos para lograr un buen funcionamiento de los equipos.
- Por último se incluye también un término económico en la función, el cual permite minimizar factor de carga parcial lo máximo posible para lograr un menor consumo de energía eléctrica y por tanto un menor coste económico.

Todo lo enunciado hasta ahora, puede recogerse en la función siguiente:

$$\begin{aligned} \min_{x, u, x_s, u_s} \quad & \sum_{k=0}^{N-1} \begin{bmatrix} x(k) - x_s \\ u(k) - u_s \end{bmatrix}^T \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix} \begin{bmatrix} x(k) - x_s \\ u(k) - u_s \end{bmatrix} + [x(N) - x_s]^T P [x(N) - x_s] \\ & + [ys + d - y_{ref}]^T S [ys + d - y_{ref}] + [u_s - u_{op}]^T J [u_s - u_{op}] \end{aligned} \quad (8.7a)$$

$$s.a \quad x(k+1) = Ax(k) + Bu(k) \quad (8.7b)$$

$$y(k) = Cx(k) \quad (8.7c)$$

$$x_s = Ax_s + Bu_s \quad (8.7d)$$

$$y_s = Cx_s \quad (8.7e)$$

$$x(0) = \hat{x}_0 \quad (8.7f)$$

$$u \in \mathcal{U} \quad (8.7g)$$

$$u_s \in \mathcal{U} \quad (8.7h)$$

$$y \in \mathcal{Y} \quad (8.7i)$$

$$y_s \in \mathcal{Y} \quad (8.7j)$$

donde  $u_s \in \mathbb{R}^{22}$  representa a las variables de control en el estado estacionario,  $y_s \in \mathbb{R}^{22}$  a las variables de salida medidas en el estado estacionario,  $x_s \in \mathbb{R}^{22}$  son los estados estacionarios,  $\mathcal{Y}$  representa el conjunto de restricciones aplicables a las salidas del sistema,  $\mathcal{U}$  hacen referencia a las restricciones en las variables de control,  $x_0 \in \mathbb{R}^{22}$  es el estado inicial del que parte el sistema en cada tiempo de muestreo, al igual que ocurre con los valores de la variable  $d \in \mathbb{R}^{22}$  que ambos son calculados mediante el observador y, por último,  $y_{ref} \in \mathbb{R}^{22}$  son el conjunto de referencias establecidas para cada una de las salidas. Como se ha podido comprobar algunos de los parámetros deben ser calculados mediante el observador ya que no existe una

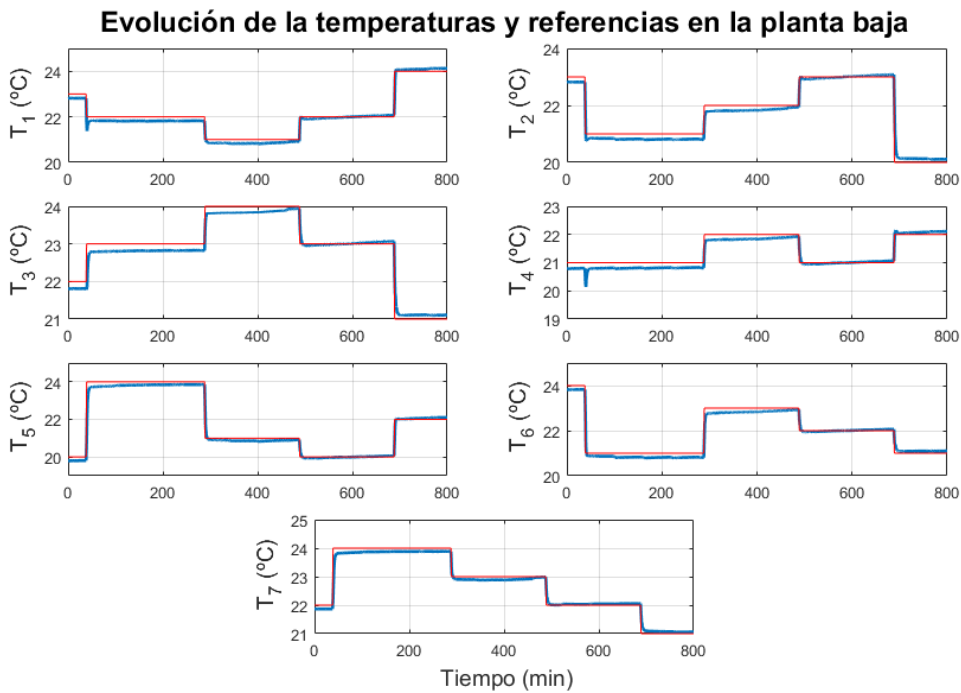
forma directa para poder establecer su valor. La estructura del observador que se emplea presenta una forma como

$$\hat{x}(k+1) = \begin{bmatrix} \hat{x}(k+1) \\ d(k+1) \end{bmatrix} = \left( \begin{bmatrix} A & 0 \\ 0 & I \end{bmatrix} - L \begin{bmatrix} C & I \end{bmatrix} \right) \hat{x}(k) + \begin{bmatrix} B \\ 0 \end{bmatrix} \mathbf{u}, \quad (8.8)$$

donde el vector  $\mathbf{u}$  es una composición de los vectores que agrupan variables de control y las salidas del sistema, es decir, presenta una estructura del tipo

$$\mathbf{u} = \begin{bmatrix} u(k) \\ y(k) \end{bmatrix}. \quad (8.9)$$

De esta forma ya se conoce la estructura del controlador y solo es necesario realizar su implementación haciendo uso para ello del código elaborado para el MPC en el sistema de una única planta. De esta forma se puede proceder a observar su comportamiento realizando para ello una simulación en la que se produzcan distintos cambios de referencia cada 200 min, de forma que se pueda estudiar cómo evolucionan las salidas del sistema ante estos cambios.



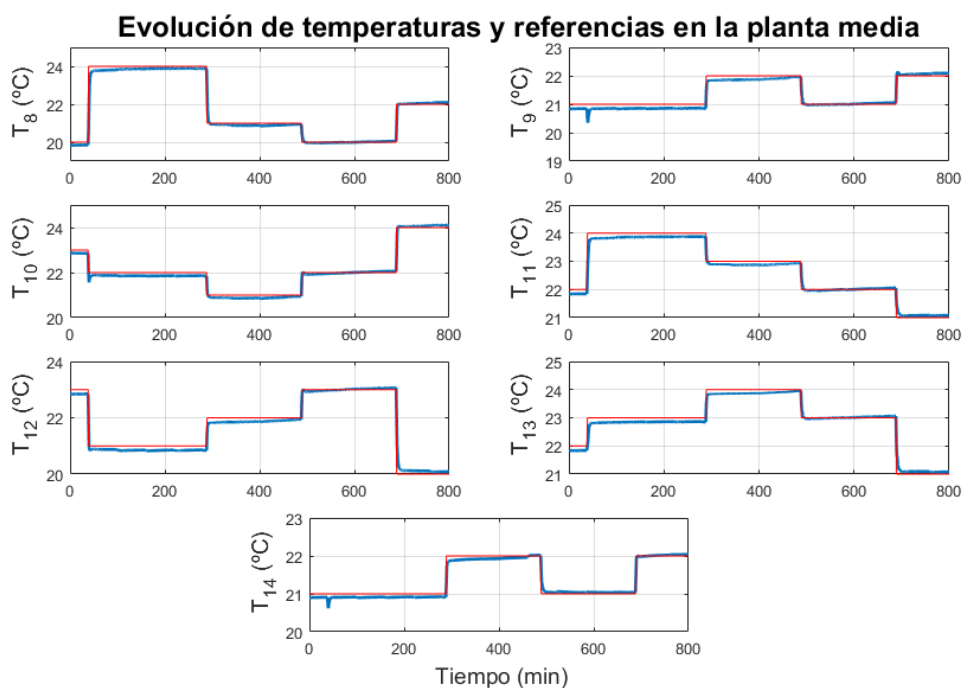
**Figura 8.16** Temperaturas y Referencias con un controlador MPC (Planta Baja).

Los resultados que se obtienen se pueden observar en las Figuras 8.16, 8.17 y 8.18. A la vista de los resultados que se han obtenido se puede afirmar que el controlador MPC se ha implementado y diseñado de forma correcta, ya que se puede observar el seguimiento de las referencias por parte de las temperaturas. De las figuras obtenidas se puede observar también las interacciones que se producen al modificar las referencias en las otras salas, aunque estas son corregidas inmediatamente y llevadas a valor deseado.

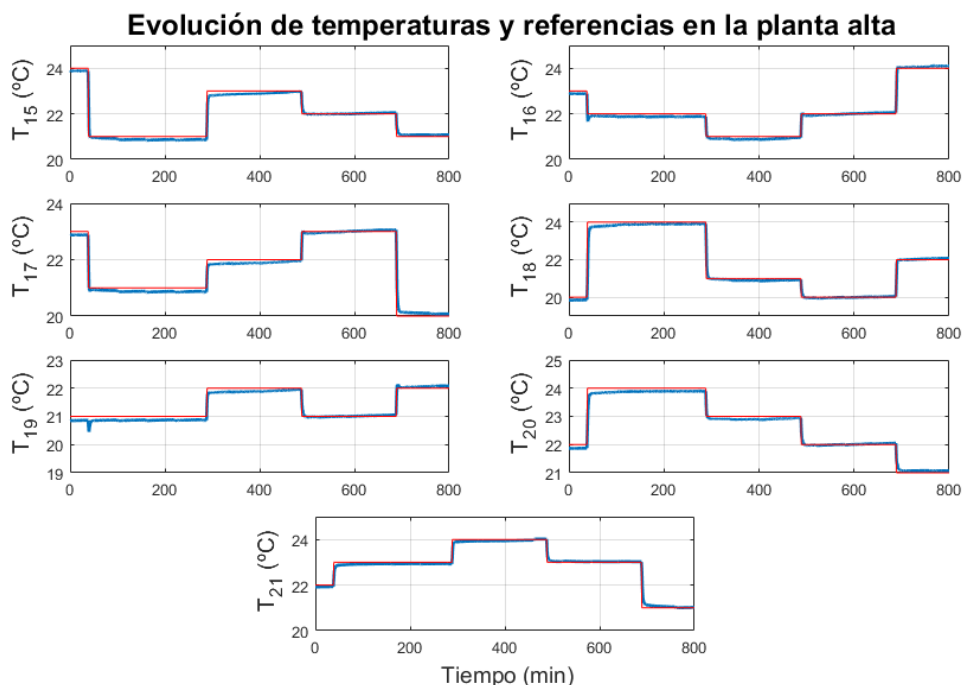
Si se compara los resultados del MPC con los del LQR, se puede observar que el MPC permite una mayor rapidez de control y genera una salida del sistema mucho mejor que la del caso anterior, además de otorgar al control disponer de una gestión de restricciones sobre distintas variables. Sin embargo, también presenta algunos inconvenientes como puede ser el alto coste computacional que conlleva la implementación de un control de este tipo ya que se debe minimizar una función de coste muy compleja en menos de lo que dura un tiempo de muestreo.

La evolución que se produce en los valores que presentan las válvulas, se pueden ver unos resultados muy similares a los que se observaban en el 6, de forma que lo que se consigue es una minimización de los valores que toma el fcp usando para ello mayores aperturas de las válvulas. De esta forma se consigue

una minimización de la variable que produce tanto un gasto eléctrico como económico por lo que interesa reducirla al máximo posible. Estas representaciones de las válvulas no se adjuntan en esta memoria descriptiva para evitar así una sobrecarga de gráficas y permitir una mayor facilidad de lectura.



**Figura 8.17** Temperaturas y Referencias con un controlador MPC (Planta Media).

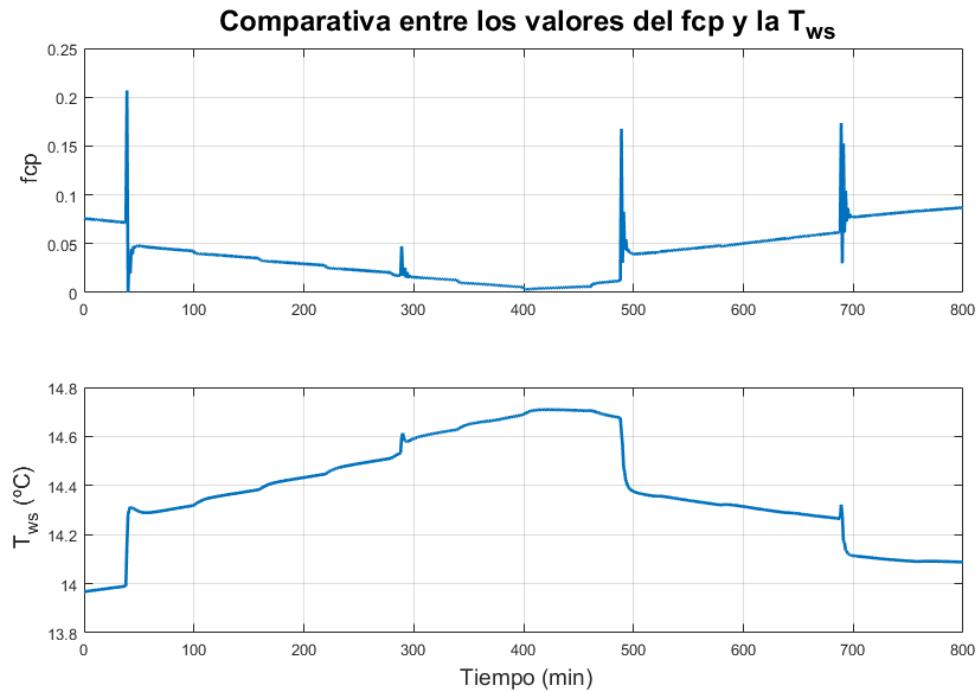


**Figura 8.18** Temperaturas y Referencias con un controlador MPC (Planta Alta).

Adicionalmente, como se ha establecido una nueva salida del sistema que representa la temperatura del agua a la salida del chiller, se desea comprobar como esta se mantiene entre las restricciones establecidas,

pero no sigue una referencia determinada. Para ello se ha elaborado la Figura 8.19, en la que se puede observar cómo evolucionan los valores del fcp y de la temperatura del agua.

Con estos resultados se puede demostrar como el controlador cumple su función manteniendo la temperatura del agua siempre por debajo de unos  $15^{\circ}\text{C}$  y cuando esta aumenta un poco más es cuando el fcp aumenta para poder enfriarla. De esta forma se puede concluir con esta breve descripción del controlador MPC y su comprobación de funcionamiento.



**Figura 8.19** Comparativa entre los valores de fcp y de la temperatura a la salida del chiller (Control MPC).



## 9 Conclusiones y Ampliaciones

---

A lo largo de la realización de este trabajo, se ha elaborado un estudio en profundidad del comportamiento térmico de una instalación basada en la construcción y disposición de uno de los laboratorios de la Escuela Técnica Superior de Ingeniería de la Universidad de Sevilla. Para ello se ha realizado un diseño del sistema mediante la herramienta TRNSYS y posteriormente se ha identificado un modelo de espacio de estados con la ayuda de MATLAB® para poder aplicar una serie de controladores sobre el sistema. En la memoria del proyecto se han recogido los ensayos que se han considerado relevantes para poder realizar un análisis sobre ellos, de forma que a través de todos estos meses de trabajo se ha podido extraer una serie de conclusiones que serán expuestas a continuación.

Durante las primeras semanas de trabajo se ha dado mucha importancia a la elaboración de un buen modelo del sistema que permita aportar realismo al mismo, si bien algunos aspectos no han podido ser matizados de forma profunda debido a la situación excepcional que se ha producido durante el tiempo de trabajo. Todo este trabajo, está recogido en los Capítulos 2 y 3, en los cuales se han recogido todos los conocimientos de cómo se ha llevado la implementación en TRNSYS y cómo ha surgido la necesidad de estudiar la conexión con la herramienta MATLAB® de dicho software.

Para poder llevar a cabo la identificación de unas ecuaciones matemáticas que describan el sistema se ha elegido un modelo de espacio de estados ya que su manejo para grandes cantidades de entradas y salidas es mucho más sencillo que realizar una identificación mediante funciones de transferencia, ya que conllevaría un coste computacional mucho más grande y una mayor complicación en todos los aspectos. También se debe reconocer que las técnicas de control para las funciones de transferencia pueden ser más sencillas pero no son utilizadas en el proyecto debido a que el problema que se aborda, posee un tiempo de muestreo determinado y no puede considerarse como un sistema continuo.

En relación a las técnicas de control empleadas y más concretamente al control LQR con efecto integral, se ha podido observar que es un controlador muy sencillo y establece unos errores en régimen permanente muy cercanos a cero. Como la dinámica de este sistema no es muy rápida esta estrategia de control permite obtener buenos resultados en el seguimiento de referencia, gracias en parte también al efecto integral que se ha añadido en el funcionamiento del controlador.

En relación al controlador MPC que se ha implementado permite además tener acotada entre los puntos de funcionamiento los valores de la temperatura del agua a la salida de la máquina enfriadora o chiller, de forma que permita funcionar adecuadamente a dicha máquina. Uno de los inconvenientes que presenta este tipo de controladores es el coste computacional que conlleva, pero que en sistemas con un tiempo de muestro relativamente alto pueden abordarse sin ningún tipo de problema.

A la vista de los resultados obtenidos, se puede afirmar que el objetivo inicial propuesto se ha cumplido de forma notoria, incluso llegando a realizar una ampliación al edificio inicial y pudiendo obtener técnicas que permiten un control de la temperatura en ambos casos.





# Bibliografía

---

- [1] A. Cardona, D. Limón, J.F. Coronel, and L. Pérez-Lombard, *Creación de casos de estudio para la gestión energética de sistemas HVAC en edificios basados en trnsys*, 2019.
- [2] Limón D., *Diapositivas de la asignatura Control de Procesos Industriales, 4º GIERM*, Escuela Técnica Superior de Ingeniería, Universidad de Sevilla.
- [3] Rodríguez D., *Diapositivas de la asignatura Complementos de Control, 4º GITI*, Escuela Técnica Superior de Ingeniería, Universidad de Sevilla.
- [4] A.M. Foster, M.J. Swain, R. Barrett, and S.J. James, *Experimental verification of analytical and CFD predictions of infiltration through cold store entrances*.
- [5] E. Hendricks, O. Jannerup, and P. Sørensen, *Linear systems control deterministic and stochastic methods*, 1º ed., 2008.
- [6] Ogata K., *Sistemas de control en tiempo discreto*, 2º ed., 1996.
- [7] Ogata K., *Ingeniería de control moderna*, 5º ed., 2010.
- [8] S.A. Klein, W.A. Beckman, J.W. Mitchell, J.A. Duffie, N.A. Duffie, J.C. T.L. Freeman, Mitchell, J.E. Braun, B.L. Evans, J.P. Kummer, R.E. Urban, A. Fiksel, J.W. Thornton, N.J. Blair, P.M. Williams, D.E. Bradley, T.P. McDowell, M. Kummert, and D.A. Arias, *Trnsys 16 a transient system simulation program. Volume 1, Getting Started.*, 5º ed., 2007.
- [9] Daniel Limón, Ignacio Alvarado, Teodoro Alamo, and Eduardo F Camacho, *Mpc for tracking piecewise constant references for constrained linear systems*, *Automatica* **44** (2008), no. 9, 2382–2387.
- [10] Lennart Ljung, *System identification*, Wiley Encyclopedia of Electrical and Electronics Engineering (1999), 1–19.
- [11] J.G. Ordóñez, J.M. Nadales, D.Limón, Luis Perez-Lombard, Juan F. Coronel, and A. Cardona, *Modelo de sistema de refrigeración de edificio orientado al control y gestión de la eficiencia energética*, 2020.
- [12] José L. Vázquez Otero, *Análisis de pérdidas energéticas por infiltración de aire*, IES Universidad Laboral de Culleredo, 2016.
- [13] Luis Pérez-Lombard, José Ortiz, and Christine Pout, *A review on buildings energy consumption information*, *Energy and buildings* **40** (2008), no. 3, 394–398.
- [14] A Picard, R S Davis, M Glaser, and K Fujii, *Revised formula for the density of moist air (CIPM-2007)*, 2008.
- [15] Rawlings and James B., *Model predictive control: Theory and design / by james b. rawlings and david q. mayne.*, 2013.
- [16] Página web de Google SketchUp, <https://www.sketchup.com/es/plans-and-pricing/sketchup-free>.
- [17] Página web de la International Energy Agency, <https://www.iea.org/>, 2020.

- [18] Página web del distribidor de TRNSYS español, <https://aiguasol.coop/es/software-energia/trnsys-18-simulacion-energetica/>.
- [19] Página web Multi Parametric Toolbox (MPT3), <https://yalmip.github.io/solver/mpt/>.
- [20] University of Wisconsin-Madison © 2005 by the Solar Energy Laboratory, *Trnsys 17 a transient system simulation program. Volume 5, Multizone Building modeling with Type56 and TRNBuild*, 7<sup>o</sup> ed., 2012.
- [21] University of Wisconsin Madison © 2005 by the Solar Energy Laboratory, *Trnsys 17 a transient system simulation program. volume 3, standard component library overview*, 8<sup>o</sup> ed., 2014.
- [22] University of Wisconsin-Madison © 2005 by the Solar Energy Laboratory, *Trnsys 17 a transient system simulation program. Volume 4, Mathematical Reference*, 10<sup>o</sup> ed., 2014.