

Escola Universitària
Politécnica de Mataró

Enginyeria Tècnica Industrial: Especialitat Electrònica Industrial

CONTROL I SIMULACIÓ EN ENTORN LABVIEW

AUTOR: ROGER PEYCASAT ARISA
PONENT: JOAN TRIADÓ AYMERICH

PRIMAVERA 2009





Agraïments

La realització d'aquest treball no ha estat fàcil.

L'entrada a l'empresa ha suposat un gran repte per a la quantitat de coses noves que he hagut d'aprendre. El programa LabVIEW és un programa complex de fer funcionar i entendre al principi. He passat moltes hores pensant la solució a errors que m'he anat trobant al llarg de tot aquest temps i aprenent multitud de mètodes per a crear estructures que ressolessin els problemes amb els programes.

També s'ha de dir que he gaudit molt d'entendre aquest treball ja que ha estat un tema que m'ha agradat molt des del primer moment.

Així doncs, moltíssimes gràcies a tot el personal de la secció d'enginyeria d'HVAC de l'empresa ALTE, especialment a l'enginyer responsable del Departament de Programació i Control, Marc Abad. Gràcies també a tota la família, amics i companys que m'han ajudat i animat en tot moment per a poder fer el treball i acabar amb aquest la carrera d'Enginyeria Industrial: Electrònica Industrial a l'EUPMT. També agrair tot el professorat que durant aquests anys m'ha ensenyat una carrera que valoro moltíssim.

GRÀCIES.

ROGER PEYCASAT ARISA





Resum

El treball de final de carrera que es presenta a continuació consisteix en realitzar una simulació de la temperatura del conducte d'un aire condicionat d'un vagó de tren i a més a més es dissenya un programa per a poder fer controls de funcionament i manteniment del sistema.

La part inicial és una introducció al programa LabVIEW, que serà la base de tot el desenvolupament del treball. També s'expliquen algunes nocions bàsiques sobre el funcionament dels aires condicionats de l'empresa ALTE.

A continuació entra en escena tot el procés per a aconseguir una simulació amb el programa. Un cop aconseguida, s'explica com funciona el programa i s'introdueix el programa de test (Test HVAC), que serà capaç de realitzar operacions de manteniment i rutina de funcionament amb l'autòmat.



Resumen

El trabajo de fin de carrera que se presenta a continuación consiste en realizar una simulación de la temperatura del conducto de un aire acondicionado de un vagón de tren y además se diseña un programa para poder hacer controles de funcionamiento y mantenimiento del sistema.

La parte inicial es una introducción al programa LabVIEW, que será la base de todo el desarrollo del trabajo. También se explican algunas nociones básicas sobre el funcionamiento de los aires acondicionados de la empresa ALTE.

A continuación, entra en escena todo el proceso para conseguir una simulación con el programa. Una vez conseguida, se explica cómo funciona el programa y se introduce el programa de test (Test HVAC), que será capaz de realizar operaciones de mantenimiento y rutina de funcionamiento del autómatas.



Abstract

The final thesis work that follows consists into perform an air-conditioned duct temperature simulation of a train wagon and also a system maintenance and function control program is designed.

The first part is an introduction to the LabVIEW program, which will be the basis for all development work. Then it explains some basics of the air-conditioned work of ALTE Company.

Then comes on the scene all the process to get a simulation with the program. Once achieved, it is explained how the program works and the test program (Test HVAC) is introduced, and it will be able to perform the automaton function routine and maintenance operations.





MEMÒRIA



Índex

1. Objecte del projecte	pàg. 11
2. Justificació del projecte	pàg. 12
3. Especificacions i abast del projecte	pàg. 13
4. Antecedents del projecte	pàg. 14
5. Introducció al LabVIEW	pàg. 15
6. L'aire condicionat	pàg. 19
7. Solució al control de temperatura	pàg. 29
8. Controladors alternatius	pàg. 31
9. Obtenció del model de segon ordre	pàg. 32
10. El programa de control en C	pàg. 37
11. Model PID en LabVIEW	pàg. 42
12. Obtenció de l'equació en diferències	pàg. 47
13. Control i regulació de l'aire condicionat	pàg. 50
14. Traducció de C a llenguatge gràfic LabVIEW	pàg. 54
15. Funcionament del programa de simulació	pàg. 67
16. Presentació de l'autòmat	pàg. 81
17. Test HVAC	pàg. 83
18. Bootloader	pàg. 96
19. Arxius d'ajuda	pàg. 97
20. Millores	pàg. 99
21. Avantatges per a l'empresa	pàg. 100
22. Pressupostos i estudi econòmic	pàg. 102
23. Conclusions	pàg. 103
24. Bibliografia	pàg. 104



1. Objecte del projecte:

Aquest projecte tracta de la realització d'un programa informàtic amb LabVIEW capaç de simular una màquina d'aire condicionat en un vagó de tren o metro per a l'empresa ALTE, situada al polígon industrial Congost, Granollers.



2. Justificació del projecte

El que busca l'empresa a l'hora de realitzar aquest treball és disposar d'un programa que li estalviï realitzar proves reals d'aires condicionats en vagons de tren, amb multitud de condicions de temperatures internes i externes.

L'objectiu que ha de complir el projecte és el d'estalviar un gran cost com és el de practicar proves de funcionament massives per a comprovar que el producte funciona correctament en múltiples situacions. Si fem una prova per trobar la funció de transferència de la màquina, podem simular totes les condicions que vulguem. A més a més, serem lliures de controlar els paràmetres de funcionament, així com totes les característiques addicionals que comentarem en el treball. A més, lògicament, haurà de reproduir amb fidelitat la resposta real del sistema.



3. Especificacions i abast del projecte

El projecte inclou bàsicament la cerca d'un model per a permetre el seu us com a substitut, en els assajos, del procés de simulació d'un aire condicionat en un vagó de tren. A més a més, aquesta simulació es realitzarà amb la incorporació d'un autòmat que realitzarà el control del sistema. Serà un control a través de PID, que és el sistema més eficaç, senzill i efectiu. La simulació es realitzarà amb el programa LabVIEW i mostrarà amb una aplicació tot el procés de control i l'evolució del sistema del conducte, així com serà capaç de modificar variables importants com mostrar un gràfic de l'evolució de la temperatura. A més a més durem a terme un altre programa que serà capaç de treballar amb l'autòmat i fer operacions de manteniment, preparar esdeveniments per després simular-los amb el programa, connectat o no a l'autòmat.



4. Antecedents del projecte

El projecte no parteix de cap situació prèvia, sinó de la idea de l'empresa d'estalviar costos a partir de simulacions. A més a més, és una aplicació cada cop més interessant per als clients, ja que demanen poder fer ells mateixos proves per a comprovar el funcionament dels equips entregats i manipular-los sense utilitzar els equips reals.



5. Introducció al LabVIEW

Què es LabVIEW?

LabVIEW (Laboratory Virtual Instrument Engineering Workbench, de National Instruments) és un llenguatge de programació gràfic pel disseny de sistemes d'adquisició de dades, instrumentació i control. Labview permet dissenyar interfícies d'usuari mitjançant un programa executable final. Es pot dissenyar especificant el seu sistema funcional, el seu diagrama de blocs o un programa en llenguatge informàtic. Labview és alhora compatible amb eines de desenvolupament similars i pot treballar amb programes d'altres àrees d'aplicació, com per exemple Matlab. Té l'avantatge de que permet una fàcil integració amb hardware, específicament amb targetes de medicació, adquisició y processament de dades (incloent adquisició d'imatges).

Aplicacions de LabVIEW

Labview té la seva major aplicació en sistemes de medicació, com el monitoratge de processos i aplicacions de control. Labview és molt utilitzat en el processament digital de senyals (wavelets, FFT, Total Distorsion Harmonic TDH), processament en temps real d'aplicacions biomèdiques, manipulació d'imatges i audio, automatització, disseny de filtres digitals, generació de senyals, entre d'altres.

Programació gràfica amb Labview

Quan un dissenya programes amb Labview està treballant sempre sota el que es denomina VI, és a dir, un instrument virtual. Es pot crear un VI a partir d'especificacions funcionals. Aquest VI es pot utilitzar en qualsevol altra aplicació com una subfunció o subprograma dins d'un programa general. Els VI's es caracteritzen per:

Ser una figura quadrada amb el seu respectiu símbol relacionat amb la seva funcionalitat.

Establir una relació amb l'usuari a través d'una interfície.



Disposar d'entrades amb el seu color identificatiu de la dada (booleà, enter, matriu...).

Tenir una o varies sortides.

Ser reutilitzable.

Nocions bàsiques de l'estructura del programa

En l'ambient de treball de Labview existeixen dos panells, el panell frontal y el panel de programació o diagrama de blocs. En el panell frontal es dissenya la interfície amb l'usuari i en el panell de programació es relacionen els elements utilitzats en la interfície mitjançant operacions que determinen en sí com funciona el programa o el sistema. Exactament és la part on es realitzen les especificacions funcionals, és a dir, on el llenguatge escrit pren forma de manera gràfica.

En el panell de programació es pot dissenyar de manera gràfica i com si fos un diagrama de blocs el funcionament del sistema. La programació gràfica es basa en la realització d'operacions mitjançant l'assignació d'icones que representen les dades numèriques i icones que representen els procediments que s'han de realitzar (VI's), amb aquestes icones i mitjançant una connexió simple (una línia) s'enllacen icones per determinar una operació i/o una funció.

Al dissenyar el programa de forma gràfica, es fa visible una programació orientada al flux de dades, on es té una interpretació de les dades també de forma gràfica, per exemple, una dada booleana es caracteritza per ser una connexió verda. Cada tipus de dades s'identifica amb un color diferent. També és necessari tenir en compte que quan es realitza una connexió a un VI aquesta connexió s'identifica per un tipus de dada específica, que ha de coincidir amb el tipus de dada de l'entrada del VI (encara que a vegades el programa fa adaptacions lògiques, com per exemple relacionar un enter de 16 bits amb un de 32) permetent una concordança en el flux de dades.

El flux de dades va d'esquerra a dreta en el panell de programació i està determinat per les operacions o funcions que processen les dades. Es fàcil observar en el panell de programació com es computen les dades en cada part del programa quan es realitza una execució del programa pas a pas. En Labview les variables es representen mitjançant



una figura, tant en el panell frontal com en el panell de programació, d'aquesta forma es pot observar la seva resposta en la interfície de l'usuari i en el flux de dades del codi del programa.

Entorn LabVIEW

La programació gràfica de Labview consta d'un panell frontal i un panell de codi, tal i com hem mencionat abans. En el panell frontal és on es dissenya la interfície d'usuari i s'ubiquen els controls i indicadors. En el panell de codi es troben les funcions. Cada control que s'utilitza en la interfície té una representació en el panell de codi. Igualment els indicadors necessaris per entregar la informació processada a l'usuari tenen una icona que els identifica en el panell de codi o de programació. Els controls poden ser booleans, numèrics, caràcters, matrius, vectors o una combinació de tots els anteriors anomenada cluster. El cluster pot agrupar qualsevol quantitat i tipus de dada. Cada dada es pot llegir o escriure.

Las funciones poden ser VI's predissenyats i poden ser reutilitzats en qualsevol aplicació. A més a més, es poden crear subprogrames a partir d'una sèrie d'elements. Per exemple, es pot fer un subprograma que converteixi les hores, minuts i segons a segons. Aquest subprograma tindria tres entrades (hh.mm.ss) i una sortida (ss.).

Tal i com hem dit, aquests blocs funcionals consten d'entrades i sortides, igual que en un llenguatge de programació estàndard. Les funcions processen les entrades i entreguen una o varies sortides. Aquests VI poden també estar conformats d'altres subVI's (o com hem dit abans, subprogrames) i així successivament. D'aquesta forma es pot representar com un arbre genealògic on un VI es relaciona o depèn de varis SubVI's. És la jerarquia del programa.

Labview té VI's d'adquisició de dades i imatges, de comunicacions, de processament digital de senyals, de funcions matemàtiques simples, fins i tot funcions que utilitzen altres programes com Matlab o HiQ per resoldre problemes.

Labview també es pot utilitzar per a fer gràfics en tres dimensions, en coordenades polars i cartesianes, té aplicacions per audio y es pot comunicar amb la targeta de so de l'ordinador per treballar conjuntament. Entre les seves moltes funcions especials es troben les de processament d'imatges, com capturar una imatge a través d'una targeta



d'adquisició, analitzar-la y entregar respostes que difícilment altres sistemes realitzarien. En la nostra aplicació, LabVIEW prendrà i enviarà dades a un autòmat.

Flux de Dades

Una altra característica es troba en el flux de dades, que mostra l'execució seqüencial del programa, és a dir, una feina no s'inicia fins no tenir en totes les seves variables d'entrada informació, o que les feines predecessores s'hagin acabat d'executar. Degut al llenguatge gràfic el compilador que té Labview és més versàtil ja que sobre el mateix codi de programació es pot veure fàcilment el flux de dades, així com el seu contingut. A més a més, en tot moment es mostra si el programa és executable, i si no ho és, quins són els possibles errors que provoquen que no funcioni.

Com utilitzar l'ajuda?

Per a visualitzar el quadre d'ajuda, simplement ens dirigim a Help»Show Context Help o pressionant <Ctrl-h>. En aquest quadre es mostra la informació de l'objecte sobre el qual ens situem amb el punter del ratolí. També es pot veure que en el quadre d'ajuda s'observa un Link blau, que si hi cliquem ens porta al Labview help, on trobarem una informació més completa de l'element, amb fins i tot, exemples d'aplicació i altres ajudes d'elements relacionats.



6. L'aire condicionat

Principis de l'aire:

L'aire atmosfèric està format per aire humit, que alhora es pot separar en aire sec i vapor d'aigua. A més a més està format per micropartícules de fum, pols, pol·len, gasos industrials, etc...

La composició de l'aire sec és de nitrogen en un 78%, Oxigen en un 21% i l'1% restant són altre gasos com l'Argó, l'Anhídric carbònic, el Neó, l'Heli, el Metà o l'Hidrogen.

L'aire condicionat es basa en controlar aquests tres elements de manera que l'ésser humà tingui les millors condicions per a respirar aquest aire. Per a controlar l'aire humit (aire sec + vapor d'aigua) s'ha de poder controlar la temperatura de l'aire i la seva humitat. A més a més, com menys micropartícules perjudicials hi hagi millor per als passatgers. Així doncs s'ha de realitzar un filtrat per a minimitzar les molèsties i perjudicis que puguin ocasionar.

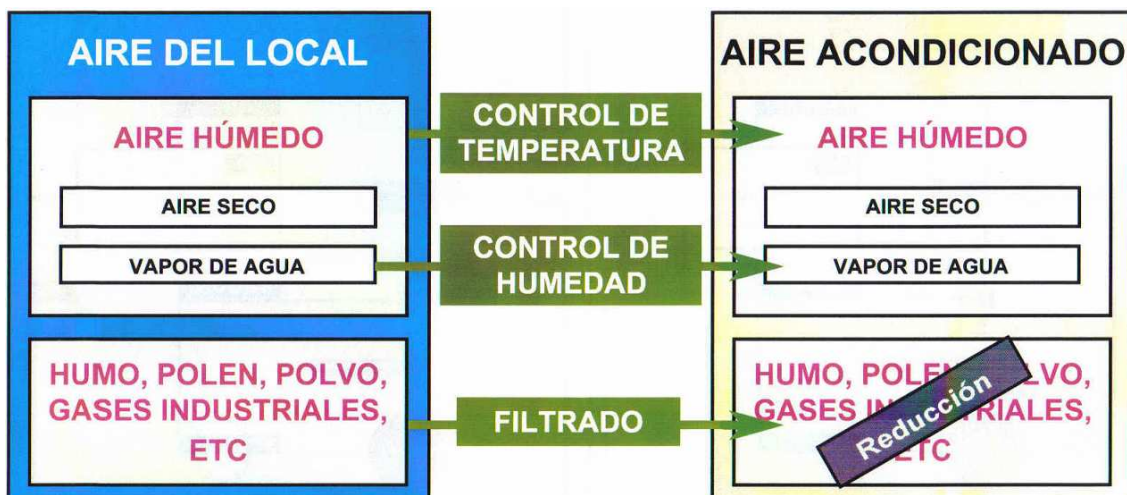


Figura 1 – Descripció del control de l'aire



El sistema per a regular la temperatura pot treballar per a escalfar o refrigerar l'aire. Es tracta de la calefacció o la refrigeració.

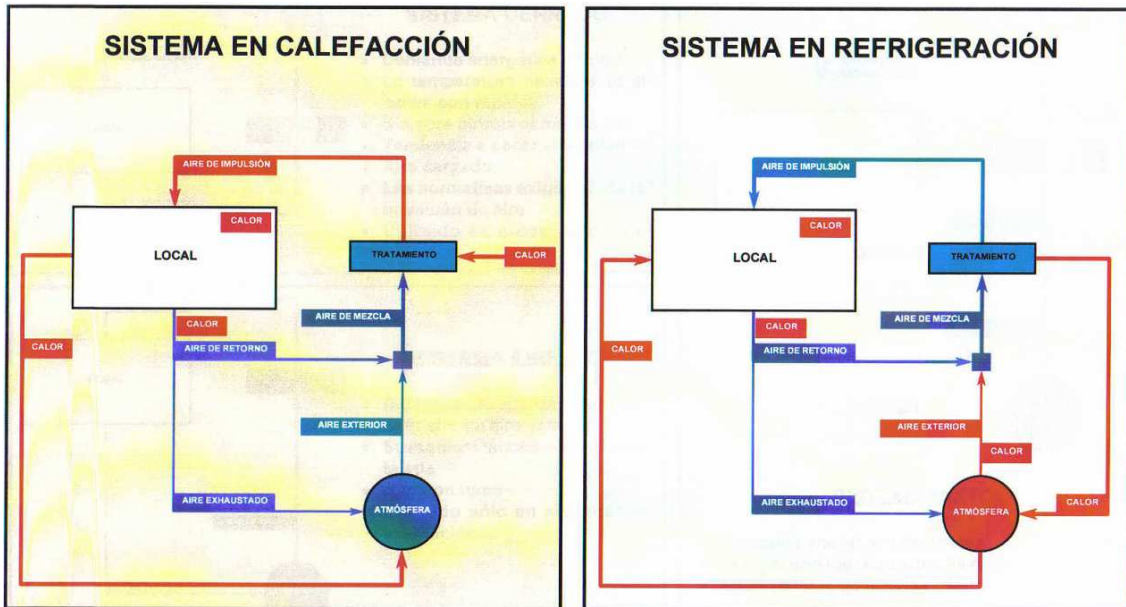


Figura 2 - Sistemes de calefacció i refrigeració del local

A més a més, l'aire que es tracta pot proveir d'un sistema tancat, d'un sistema obert o d'un sistema mixt.

En un sistema tancat la demanda energètica és mínima. La temperatura desitjada s'aconsegueix amb rapidesa. Però sempre circula el mateix aire, té tendència a assecar l'ambient i l'aire es carrega. Per això les normes ferroviàries exigeixen una certa renovació de l'aire. Aquest mètode només s'utilitza per a processos de pre-condicionament.

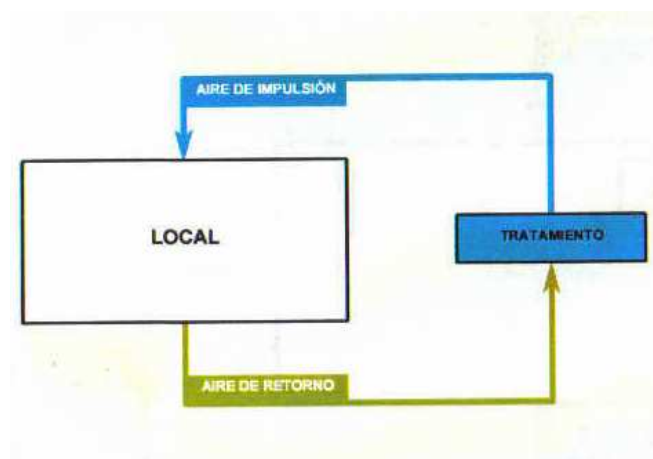


Figura 3 – Sistema tancat



En un sistema obert la demanda d'energia és màxima. Només circula aire exterior i això vol dir que es renova constantment. Per contra, es requereix una potència instal·lada molt alta i per tant el consum és molt alt. Només s'utilitza en sistemes de ventilació, on no es requereix escalfar ni refredar l'aire.

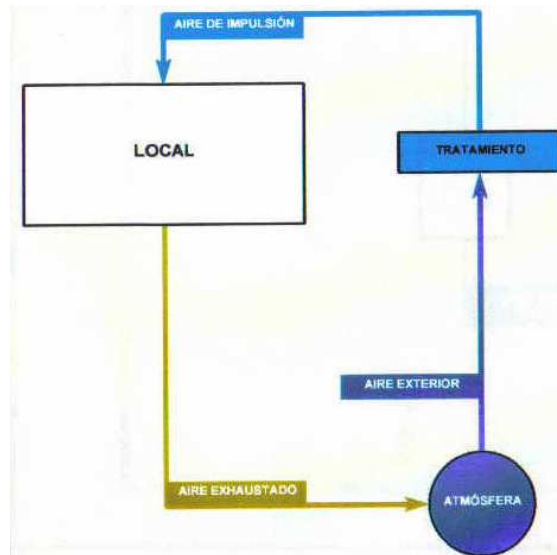


Figura 4 – Sistema obert

Finalment en el sistema mixt es combinen els dos sistemes anteriors. Així es té una demanda energètica raonable i es compleix amb els requisits ferroviaris de renovació de l'aire. És el sistema habitual de funcionament.

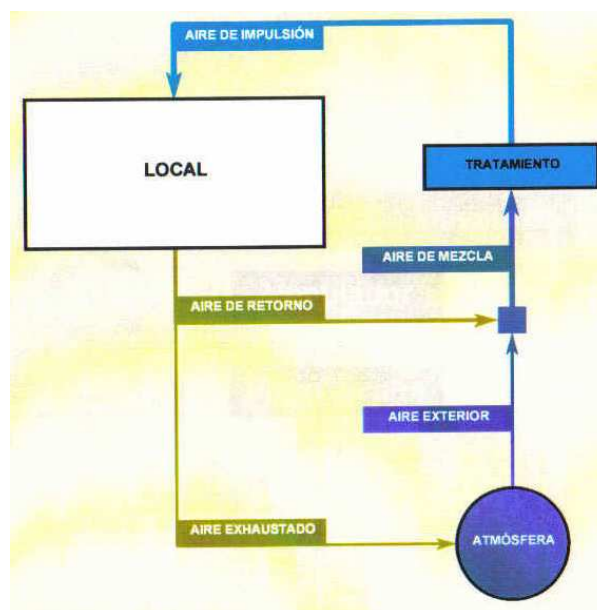


Figura 5 – Sistema mixt



La climatització:

La climatització d'un local consisteix en aconseguir que les característiques de l'aire del local siguin les adequades per als ocupants. El conjunt d'aquestes característiques adequades s'anomenen condicions de confort.

La pròpia naturalesa de l'aire proporciona les característiques sobre les que actua un equip de climatització:

- **Puresa de l'aire:** Eliminació dels contaminants. S'aconsegueix mitjançant filtres d'aire que retenen les partícules indesitjables.
- **Grau d'humitat:** Quantitat de vapor d'aigua contingut en l'aire. El grau d'humitat no sol controlar-se en sistemes de climatització ferroviaris ja que mai es tendeix a ressecar l'ambient quan hi ha passatgers dins (el condicionament sempre es realitza sense passatgers, abans d'arrencar el tren). No obstant el sistema de refrigeració manté un grau d'humitat confortable.
- **Temperatura:** Ha de ser l'adequada per al millor confort dels ocupants. Ens cas de que es necessiti augmentar la temperatura del local s'utilitzarà el sistema de calefacció i si es precisa d'una disminució de la temperatura, s'utilitzarà un sistema de refrigeració.

L'aire sobre el que es realitzen totes aquestes operacions per controlar la seva puresa, grau d'humitat i temperatura s'anomena aire tractat. Les condicions de confort d'un local s'aconsegueixen mitjançant la impulsió, a l'interior del mateix, d'una certa quantitat d'aire tractat. Com que la pressió del local es manté aproximadament constant i en general lleugerament superior a l'atmosfèrica, la introducció d'aire al local origina la sortida d'aire des del local cap a fora o cap a la màquina de tractament d'aire. Aquesta sortia igualment, pot ser natural o forçada.

Així doncs existeixen diferents tipus d'aire involucrats en tot el procés de climatització del local:

- **Aire d'impulsió:** És l'aire tractat que s'introdueix al local.
- **Aire de mescla:** L'aire tractat és en general una mescla de l'aire provinent de l'ambient (o aire exterior) i el del propi local que s'està climatitzant (o aire de retorn).



- **Aire exhaust:** és l'aire que s'extreu del local com a conseqüència de la introducció d'aire exterior.

El sistema de climatització es pot dividir com ja s'ha introduït anteriorment en tres subsistemes bàsics:

- **Circulació d'aire:** Conjunt d'elements (ventiladors) que permeten la circulació de l'aire en el local.
- **Calefacció:** Conjunt d'elements capaços d'aportar calor al local.
- **Refrigeració:** Conjunt d'elements capaços d'extreure calor del local. Els sistemes clàssics de refrigeració també extreuen humitat.

El sistema de climatització, per a sales i cabines és semblant i està format pel sistema de ventilació-calefacció i el sistema de refrigeració. Disposa de dos equips, un de 6kW i l'altre de 12kW. La combinació d'aquests dos proporcionen la potència suficient com per escalfar o refredar el local. A continuació s'explicaran algunes característiques i components de cadascun.

Ventilació-Calefacció:

A part de l'estructura que sosté tots els elements, la unitat de ventilació-calefacció està formada pels següents elements:

- Sis motoventiladors d'impulsió (un grup de 2 i un altre de 4).
- Una etapa de calefacció dividida en tres conjunts de 8 resistències elèctriques cadascun (1 conjunt per al grup de 2 ventiladors i 2 conjunts pel grup de 4 ventiladors).
- Dos filtres d'aire exterior.
- Un filtre d'aire de retorn.
- Dues comportes d'aire exterior amb els seus actuadors elèctrics.
- Una comporta d'aire de retorn amb el seu actuator elèctric.
- Dos termòstats de seguretat.
- Dos termòstats fusibles de seguretat.
- Una comporta de regulació manual exclusiva per a la cabina.



- Dos extractors d'aire.
- Un regulador de velocitat de gir per a cada ventilador.

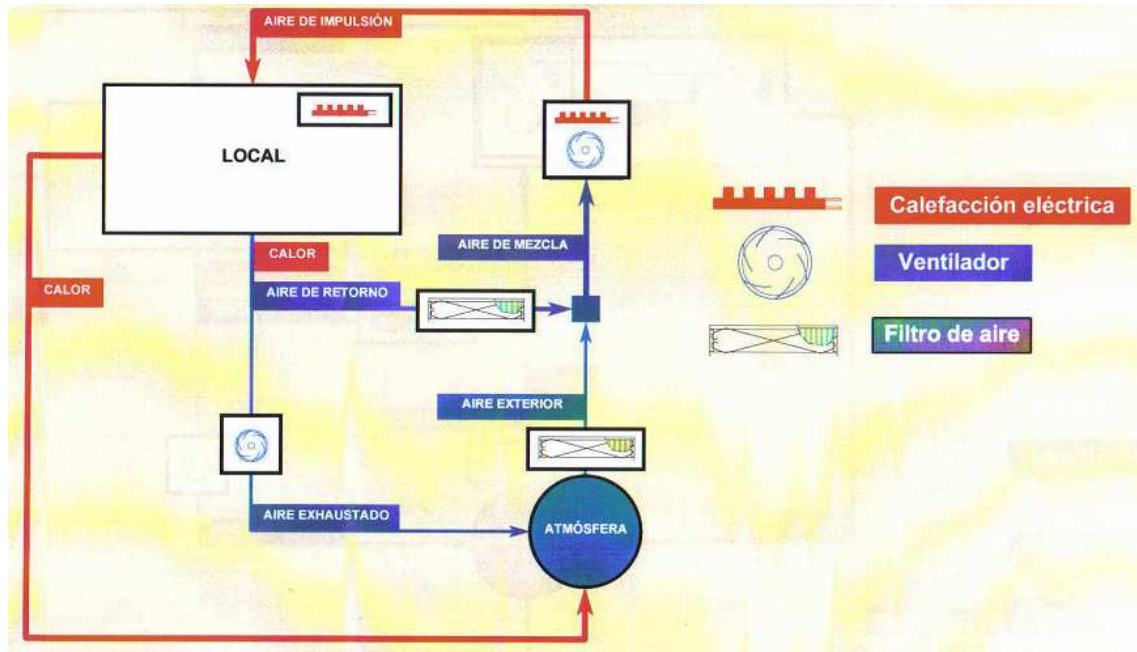


Figura 6 – Esquema de calefacció

La situació de la unitat de calefacció de sala en els vagons de trens és al sostre.

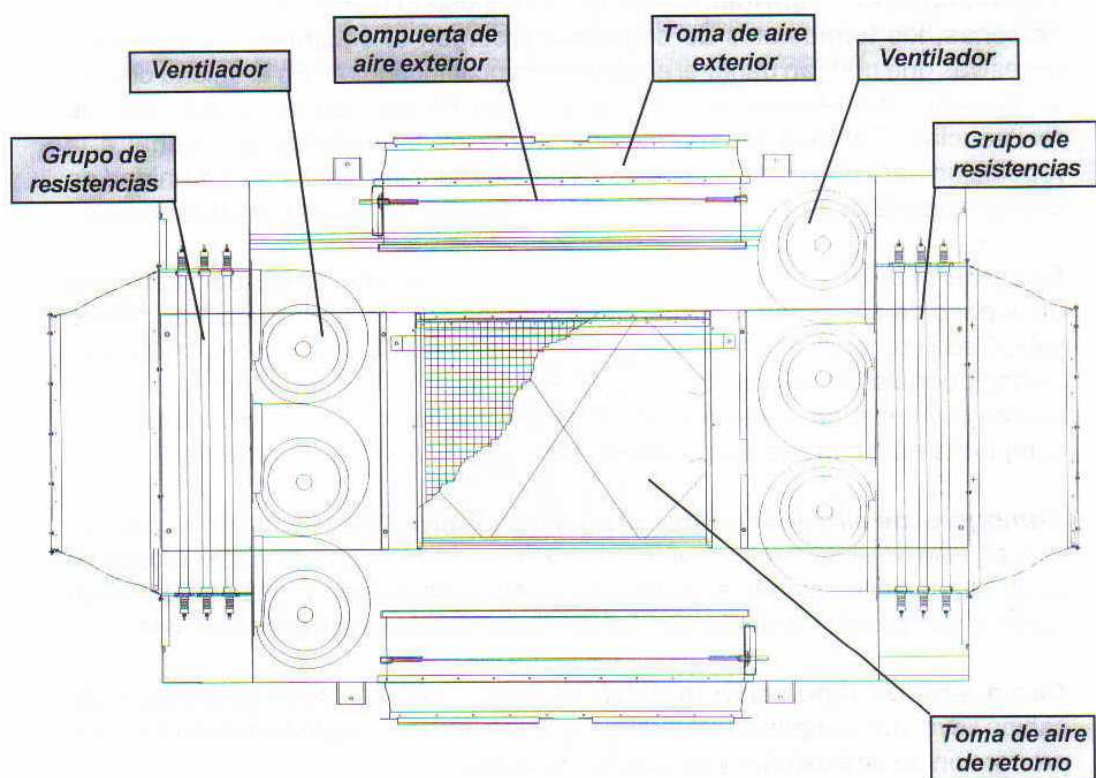


Figura 7 – Vista general de l'equip de calefacció



Refrigeració:

A l'igual que en el cas anterior, a més de l'estructura, els elements de la unitat refrigeradora són els següents:

- Dos compressors.
- Una bateria condensadora.
- Un ventilador de condensació.
- Un dipòsit de líquid.
- Un dipòsit de separació de líquid.
- Una vàlvula solenoide de líquid.
- Una vàlvula de tancament manual.
- Un visor de líquid.
- Un filtre deshidratador.
- Una vàlvula d'expansió termostàtica.
- Dues bateries evaporadores.
- Un ventilador d'impulsió per cada evaporador.
- Una vàlvula de control de capacitat by-pass.
- Una vàlvula solenoide de by-pass.
- Una vàlvula d'expansió termostàtica.
- Una vàlvula d'alta pressió.
- Una vàlvula de baixa pressió.
- Un manòmetre d'alta pressió.
- Un manòmetre de baixa pressió.
- Una comporta de refrigeració.
- Un filtre d'aire exterior.

A continuació hi ha un petit document gràfic amb nocions bàsiques de funcionament de la refrigeració dels aires condicionats de l'empresa.

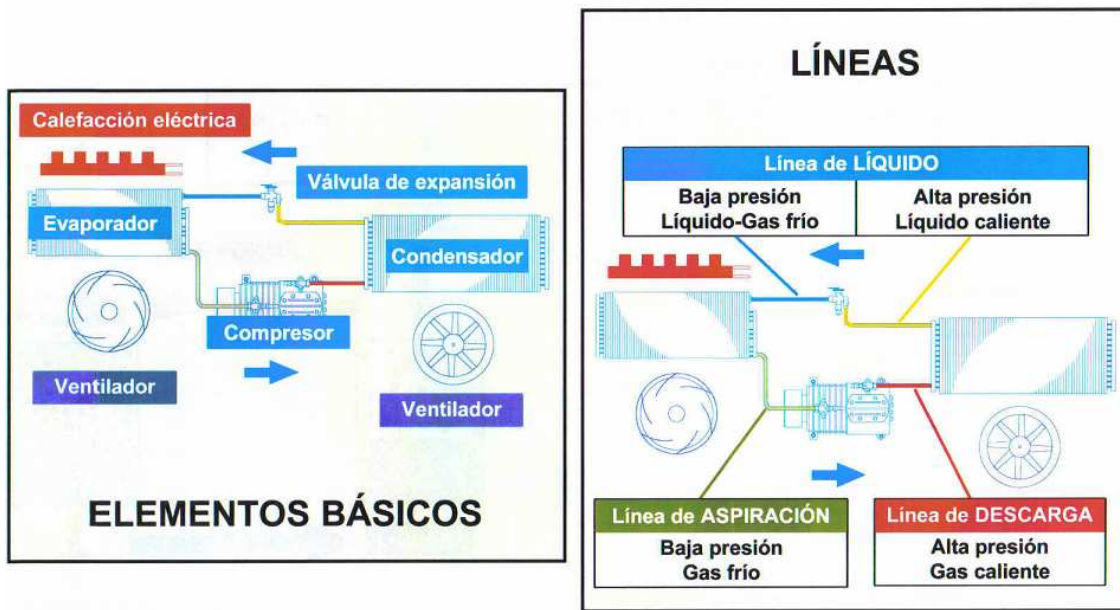


Figura 8 – Esquema bàsic de refrigeració

A continuació es mostra el diagrama de Mollier del cicle del refrigerant. Són les quatre etapes de compressió, condensació, expansió i evaporació.

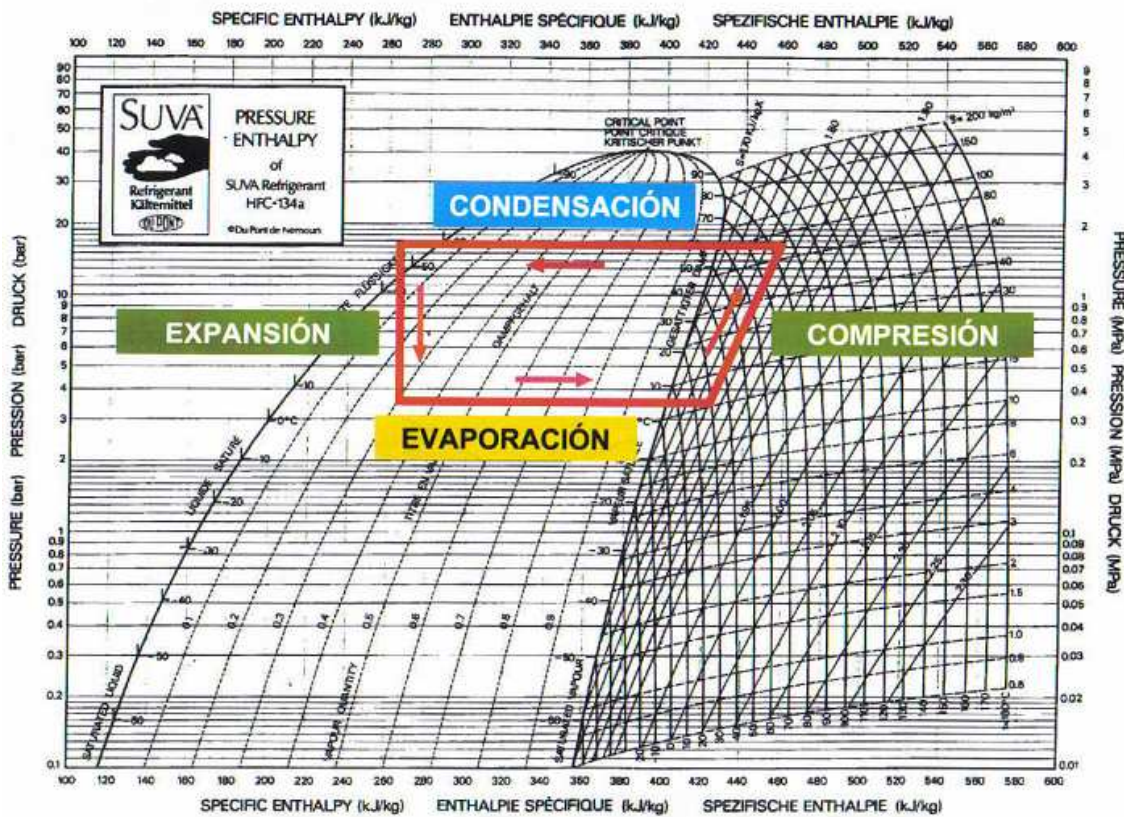


Figura 9 – Diagrama de Mollier del cicle refrigerant



Tot seguit un petit esquema amb els elements descrits anteriorment.

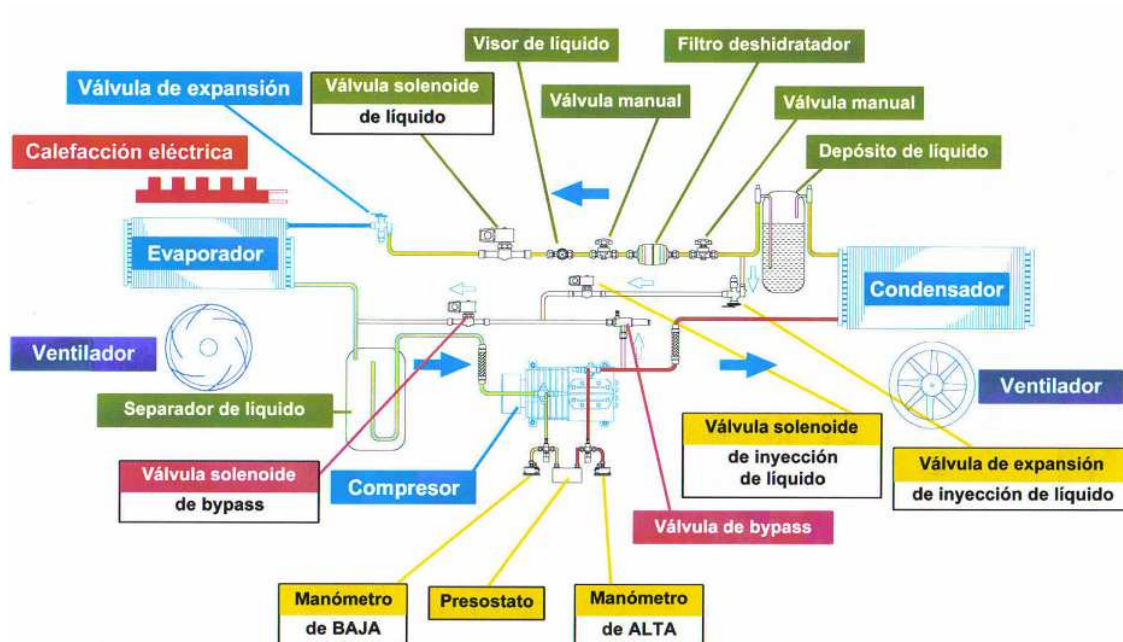


Figura 10 – Esquema general del circuit de refrigeració

Finalment existeixen tot un seguit de normes de manteniment preventiu. A continuació se n'anomenen algunes:

- Inspecció del quadre elèctric.
- Comprovació de la posada en marxa i selecció de temperatures.
- Regulació electrònica i sondes de temperatura.
- Comprovació del funcionament de les comportes d'aire.
- Comprovació del funcionament dels motoventiladors d'impulsió i del condensador.
- Comprovació del funcionament de les vàlvules de líquid i by-pass.
- Localització de fugues.



Finalment es mostren uns esquemes de situació dels equips d'aire condicionat, calefacció i refrigeració en els vagons dels trens.

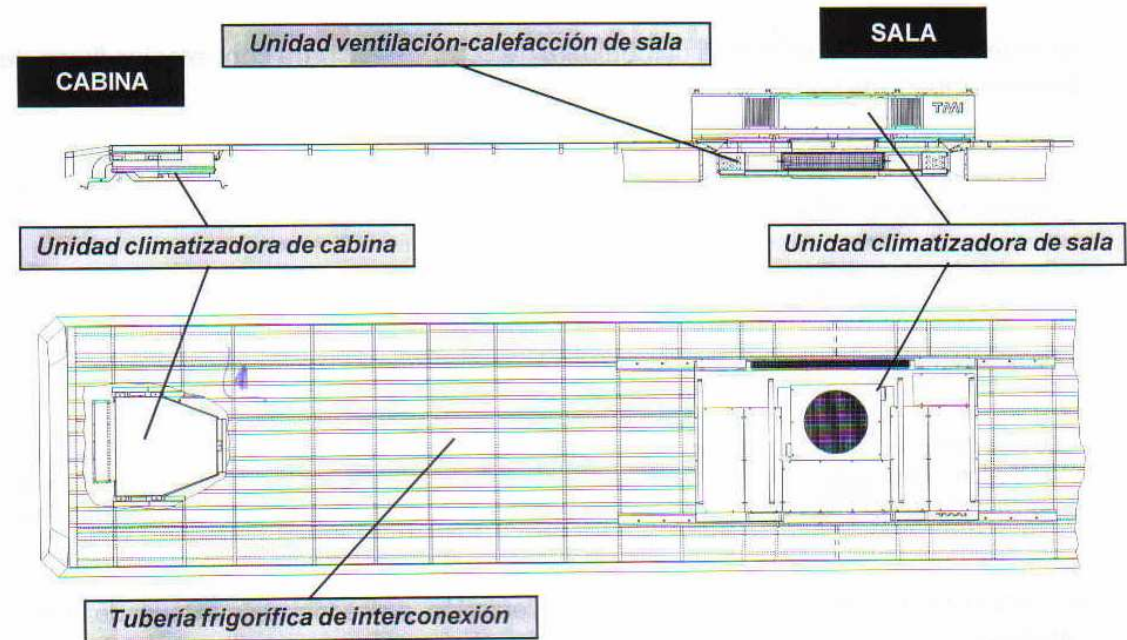


Figura 3. Disposición general del sistema

Figura 11 – Situació dels equips en el sostre del vagó



7. Solució al control de temperatura

Per a poder realitzar el control de temperatura d'una manera eficaç i satisfactòria per a l'estat dels passatgers (sense grans variacions de temperatura en poc espai de temps) es recorre a la utilització d'un sistema de regulació.

Des dels inicis, l'empresa s'ha decantat cap a la utilització d'un control PID amb un diagrama de control concret.

Es realitzen dos controls de temperatura perquè el sistema sigui més eficaç. Es controla la temperatura de la sala i la temperatura del conducte, en cascada, ja que una depèn directament de l'altra.

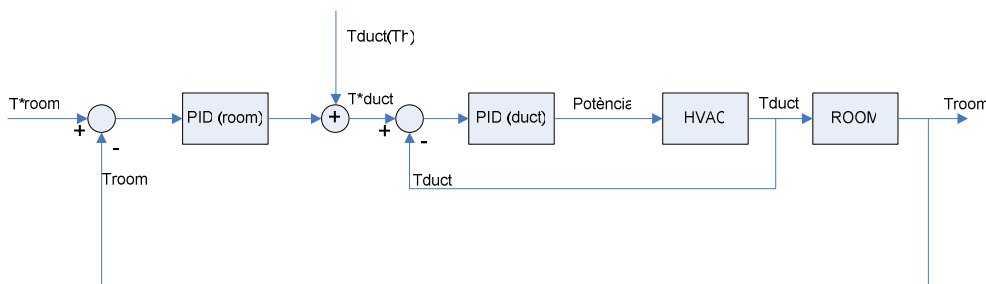


Figura 12 – Diagrama de blocs del sistema

Aquest és el diagrama de blocs corresponent al control del vagó.

El valor $T_{duct}(T_h)$ (la temperatura de conducte que garanteix una temperatura de sala de 22°C) correspon a la temperatura teòrica del conducte. La sortida del PID de la sala també dona una altra temperatura de conducte.

El signe de suma encerclat simbolitza la mitjana entre aquestes dues temperatures i genera la temperatura de consigna del conducte, que serà la que s'introduirà com a dada en la simulació.

La part del control de la temperatura de la sala no es realitza en aquest projecte. Es comporta de la mateixa manera podem dir que l'evolució de la temperatura de conducte regula la temperatura de la sala. No s'ha pogut realitzar la simulació de la sala ja que



l'empresa encara no disposa d'una evolució de temperatures de la sala quan els motors s'activen a plena potència. El mètode per a resoldre aquesta part serà el mateix que el que es mostrarà per a resoldre la part del conducte i el seu PID.

Així doncs, el diagrama de blocs que ocuparà la simulació que es realitzarà és el següent.

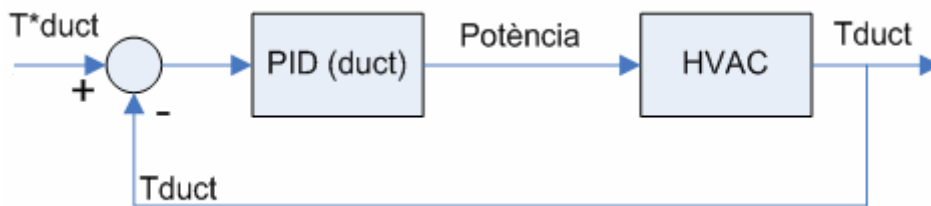


Figura 13 – Diagrama de blocs del conducte

Es disposa d'una temperatura de conducte de consigna o setpoint. Amb la temperatura real del conducte es calcula l'error. Aquest entra al PID, que dóna una senyal de sortida en forma de senyal PWM. Aquesta senyal activa durant un cert temps les dues bandes de calefacció i proporciona una nova temperatura. El temps de durada d'aquest cicle és de 210 segons.

Per a escalfar un vagó que està a una temperatura inicial de 0°C fins a 22°C pot passar tranquil·lament una hora.



8. Controladors alternatius

Ràpidament s'ha comentat que el controlador escollit és el PID, però també es podien haver triat d'altres controladors. Perquè s'han descartat?

El **controlador de temps mínim** s'ha descartat perquè el control de temperatura és un tipus de control que s'ha de tenir molta cura. No pot ser la resposta més ràpida la més adequada ja que la potència consumida augmentaria molt i l'aire experimentaria una evolució massa sobtada de la temperatura. A més a més, no és flexible i només admet una resposta temporal.

Al controlador per **cancel·lació de pols** li passa més o menys el mateix. No és prou flexible com per a poder ser modificat, a l'igual que el **controlador per assignació de pols**.

També existeixen els **controladors de retorn d'estat** i els **controladors d'avanç i/o retard**.

Els primers són útils si es treballa amb models de variable d'estat, però no és el cas.

Els segons són molt populars com a controladors continus implementats amb circuiteria elèctrica i electrònica però no són els òptims per a un control de temperatura.

Així doncs els controladors PID són els escollits perquè són els més flexibles i modelables. A més a més són fàcils d'implementar electrònicament i són els més utilitzats per a la majoria de controls. Aquest és un fet a tenir en comte ja que solen ser també els més econòmics.



9. Obtenció del model de segon ordre

Per a poder realitzar el control de l'aire condicionat, primer de tot hem de saber com es comporta. La manera més fàcil de fer-ho és mitjançant una aproximació a través d'un sistema de segon ordre.

L'únic que necessitem és saber la resposta de la temperatura quan s'engega a plena potència. Aquesta és la única prova real que farem. A partir d'aquí tot es resoldrà fora del taller.

A continuació observem com evoluciona la temperatura del conducte d'aire condicionat a una sèrie d'activacions dels motors dels aires condicionats:

Prenem com a vàlida la part on s'activa l'aire condicionat, estant apagat fins a plena potència i obtenim mostres de la temperatura del conducte cada 5 segons.

La temperatura inicial és de 25.5°C i la temperatura final és de 57.1°C. En un document adjunt d'Excel, podrem veure el valor de cada dada en cada instant.



A continuació, veiem la part on s'activen els 18 kW de potència total del sistema.

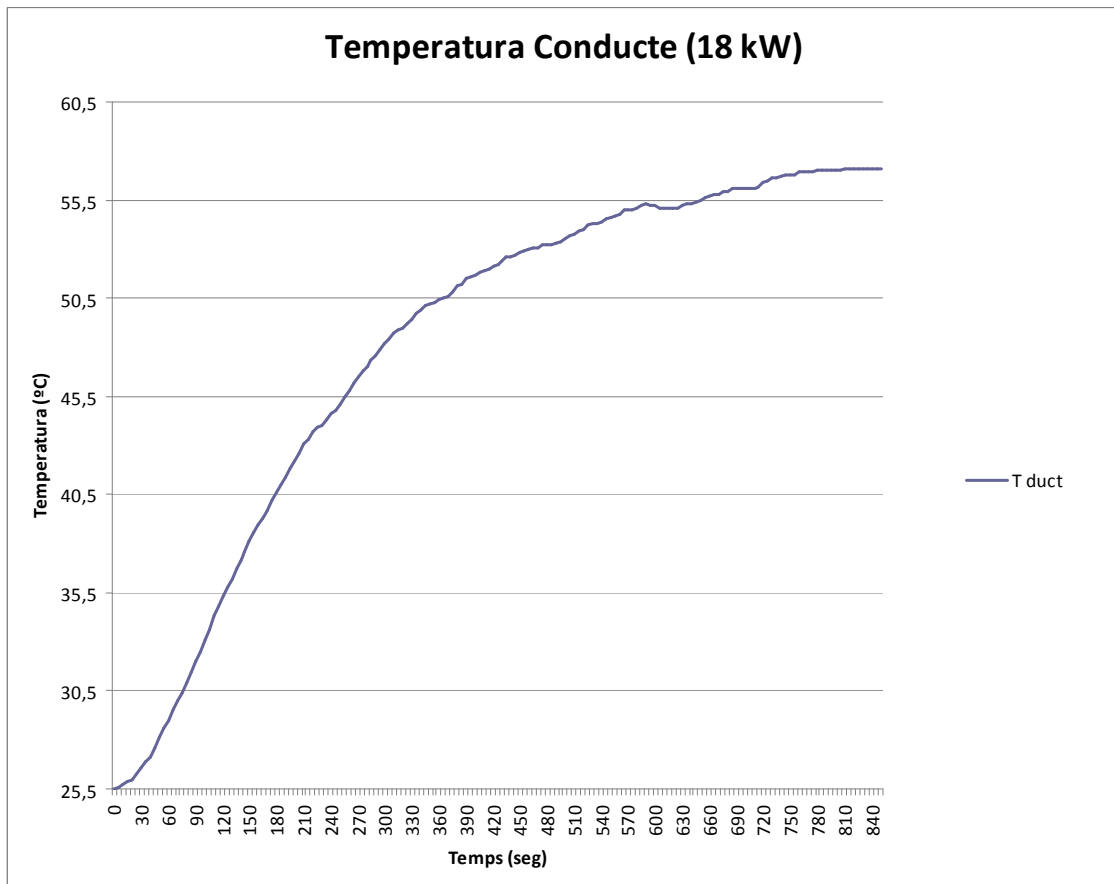


Figura 14 – Gràfic de l'evolució de la temperatura del conducte en condicions de 0°C de temperatura exterior i pressió atmosfèrica

A partir de las mesures de temperatura en cada instant de temps busquem una sèrie de paràmetres que ens determinaran l'equació de segon grau que simularà el procés del sistema. Aquests paràmetres son T_s , ω_n y ξ .

El temps d'estabilització (T_s) s'ha establert en 724 segons, ja que és el temps que tarda el sistema en assolir el 98% de la temperatura final.

$$T_s = \frac{6}{\xi \cdot \omega_n} \rightarrow \xi \cdot \omega_n = \frac{6}{724} = 0.0082873$$

Un paràmetre característic de respostes temporals es τ . En el nostre cas,

$$\tau = \frac{1}{\xi \cdot \omega_n} = 120.67 \text{ seg.}$$



$\xi = 1$, ja que és un sistema sense sobreimpuls i s'estabilitza a un cert valor.

Per tant, $\omega_n = 0.0082873$ y $\omega_n^2 = 0.0000686792$.

Una cop determinats els paràmetres, ja els podem introduir a l'equació de segon grau de la següent manera:

$$FT = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad \text{On FT és la funció de transferència del sistema.}$$

Per tant, substituint els valors:

$$FT = \frac{0.0000686792}{s^2 + 0.0165746s + 0.0000686792}$$

Aquesta equació s'introdueix en el programa MATLAB de la següent manera per a obtenir la seva resposta temporal:

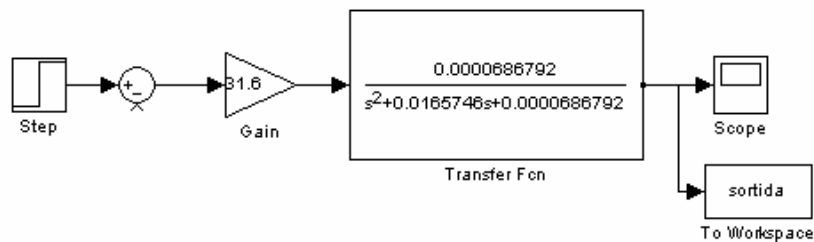


Figura 15 – Diagrama de blocs en MATLAB

Amb un esglaió d'altura 1, el guany de 31.6 és per a igualar el valor final de la temperatura real, ja que aquesta passava d'una temperatura inicial de 25.5 °C fins a 57.1°C (la diferència es 31.6). A més a més el sistema es mostreja cada 5 segons.



La resposta vista per el Scope té la següent forma:

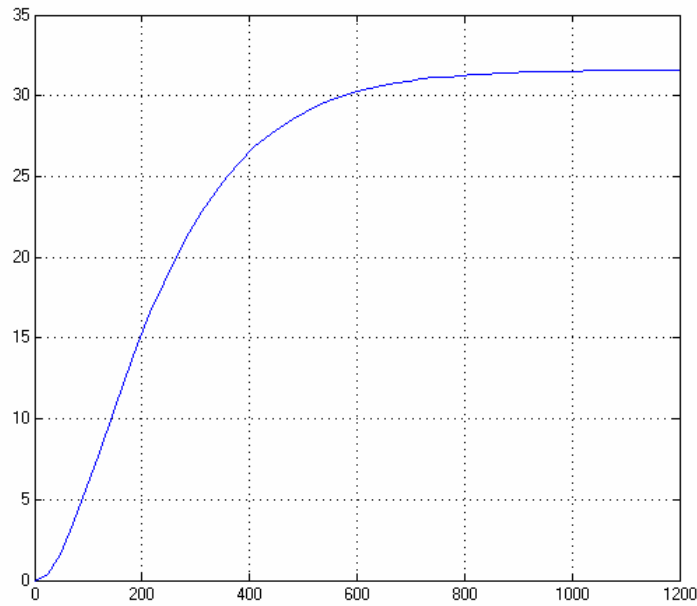


Figura 16 – Gràfic de la temperatura de conducte simulada

Així doncs, per a un temps de simulació de 840 segons, observem que la resposta real y la resposta simulada tenen la següent forma (començant amb $T = 25.5^{\circ}\text{C}$):

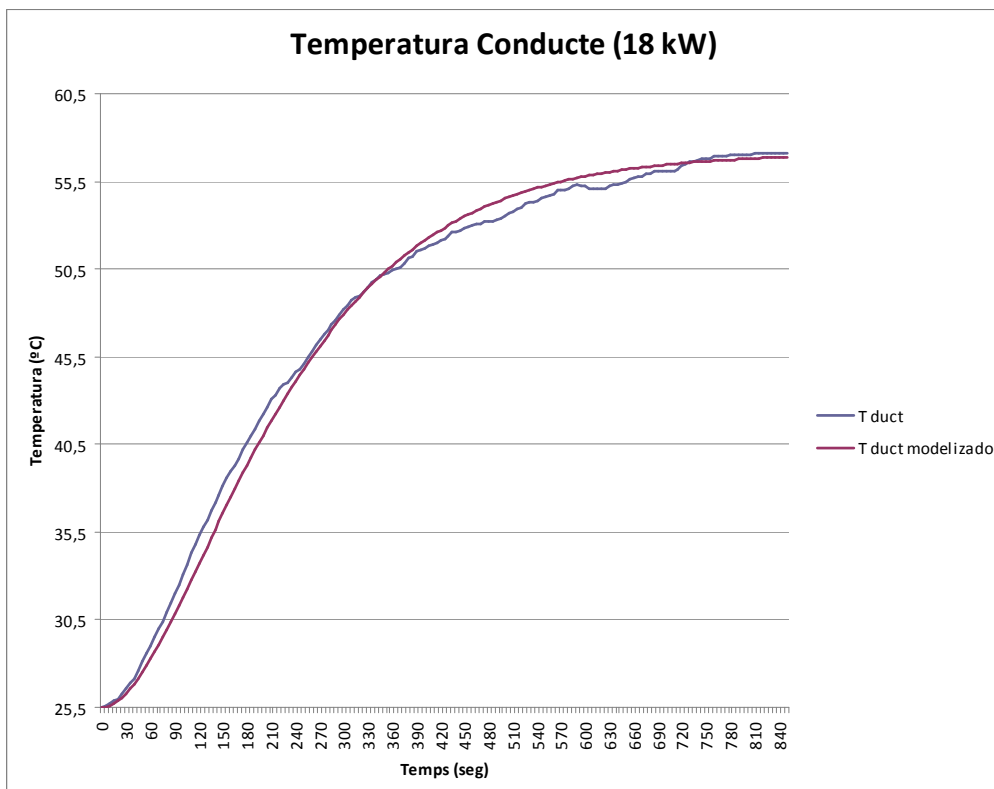


Figura 17 – Superposició dels gràfics de temperatura de conducte real i simulada

Podem observar com el model s'aproxima bastant a la resposta real.



De fet, es pot calcular l'error en cada instant de mostreig (també adjunt en el mateix Excel), fent la resta de la temperatura real i la temperatura de la resposta del sistema de segon ordre. Si ho fem, podrem veure com l'error màxim de temperatura és de poc més d'un grau.

Per tant, acceptem aquesta equació com a vàlida per iniciar la nostra simulació.

$$FT = \frac{0.0000686792}{s^2 + 0.0169746 \cdot s + 0.0000686792}$$



10. El programa de control en C

És la base de tota la simulació.

L'empresa ha dissenyat un programa de control completíssim, línies i línies de codi molt exhaustiu, amb tota la gestió dels paràmetres que estan involucrats amb l'autòmat, des de l'activació d'alarmes per algun mal funcionament fins a la declaració de tots els valors dels paràmetres que intervenen.

Com és lògic, el programa no es pot mostrar ja que és un secret d'empresa, però se'n pot parlar una mica.

Primer de tot s'ha de comentar que la base del programa és més o menys sempre la mateixa, però que per cada client existeixen variacions, degut a diferents peticions de funcionament que donen, diferents tipus de vagons destinats a vàries utilitats, com pot ser passatgers, sales per a nens, vagons-restaurant, cabines del maquinista, etc... .

Per exemple, per a un client noruec és important prestar atenció a les temperatures extremadament baixes que s'assoleixen i establir un sistema de control especial per al sistema de calefacció. O un client algerià, on demana un tractament especial per a la refrigeració, ja que les temperatures són més altes, i a més a més, reclama una alarma per a detectar si hi ha sorra acumulada en els ventiladors, a part d'altres infinites peticions per a un correcte i òptim funcionament desitjat.

El programa bàsicament està separat en quatre parts:

- El programa principal.
- Subrutines comunes o llibreria.
- Capçalera, declaració de constants, estructures, variables...
- Característiques pròpies de la obra i diferent per cada client.



A continuació es veurà simbòlicament l'estructura de la part de la programació del funcionament del PID. Tot el que estigui marcat amb `/**` significa un comentari explicatiu afegit per a aquest treball.

```

short PID(
    struct CONTROL_LOOP      *Loop,
    int                      Feedback,
    char                     OnlyProp,
    char                     IntegralFreezing)
{
    int PID_out, PID_aux;
    int PD_out;
    float AuxFloat, NewIntegTerm;
    int IntegAux;
    short s_aux;

    /**Inicialització de variables

    // Error limitation
    LimitShort(
        &(Loop->Error),
        MAX_POS_PID_ERROR,
        MAX_NEG_PID_ERROR);

    /**Limitació màxima i mínima de l'error. LimitShort és un subprograma que limita per sobre i per sota un valor.

    // Proportional Term
    Loop->PropTerm = (Loop->KpX10) * (Loop->Error);
    Loop->PropTerm /= 10;

    /** Càlcul del terme proporcional (P)

    if ( !OnlyProp)
    {
        // Integral Term
        if ( !IntegralFreezing && ( Loop->TiX1 > 0 ) )
        {
            AuxFloat = (float) (200 * (Loop->TiX1));
            IntegAux = (Loop->KpX10) * (Loop->hX10) * ((Loop->Error) + (Loop->ErrorPrev));
            Loop->IntegrTerm = (float) IntegAux;
            Loop->IntegrTerm /= AuxFloat;
            Loop->IntegrTerm += Loop->IntegrTermPrev;
        }
        else
            Loop->IntegrTerm = Loop->IntegrTermPrev;

    /**Càlcul de terme integral (I)

```



```
// Derivative Term
if ( Loop->ND > 0 )
{
    s_aux = Loop->ND;
    while ( s_aux > 0 )
    {
        Loop->FeedbackArray[s_aux] = Loop->FeedbackArray[s_aux - 1];
        s_aux--;
    }
    Loop->FeedbackArray[0] = Feedback;

    Loop->DerivTerm = (Loop->KpX10) * (Loop->TdX10);
    Loop->DerivTerm *= (Loop->FeedbackArray[Loop->ND] - Loop->FeedbackArray[0]);
    Loop->DerivTerm /= Loop->ND;
    Loop->DerivTerm /= 10;
    Loop->DerivTerm /= Loop->hX10;
}
else
    Loop->DerivTerm = 0;

//Càlcul del terme derivatiu (D)

// PD output
PD_out = (Loop->PropTerm) + (Loop->DerivTerm);

// PID output
PID_out = PD_out + (int) (Loop->IntegrTerm);

//***Càlcul de la sortida del PID

PID_aux = PID_out;

// PID output limiting
s_aux = (short) PID_out;

LimitShort(
    &(s_aux),
    Loop->MaxPosPID_out,
    Loop->MaxNegPID_out);

//***Limitació de la sortida del PID.

PID_out = s_aux;

// Anti-Wind-up algorithm
if (PID_out != PID_aux)
// If the PID is saturated
{
    NewIntegTerm = (float) (PID_out - PD_out);
```



```

    if ( ( (Loop->IntegrTerm >= 0) && (NewIntegTerm >= 0) ) || ( (Loop->IntegrTerm <=
0) && (NewIntegTerm <= 0) ) )
        // if the signus hasn't changed, get the new reduced Integral Term
        Loop->IntegrTerm = NewIntegTerm;
    else
        // The anti-Wind-up algorithm cannot change the signus of the Integral Term
        Loop->IntegrTerm = 0;
    }

    // Update auxiliar variables
    Loop->ErrorPrev = Loop->Error;
    Loop->IntegrTermPrev = Loop->IntegrTerm;
}
else
{
    s_aux = (Loop->PropTerm);

    // PID output limiting
    LimitShort(
        &(s_aux),
        Loop->MaxPosPID_out,
        Loop->MaxNegPID_out);

    PID_out = s_aux;

    // Reset of IntegrTerm and DerivTerm
    Loop->ErrorPrev = 0;
    Loop->IntegrTermPrev = 0;

    for ( s_aux = 0; s_aux <= Loop->ND; s_aux++)
        Loop->FeedbackArray[s_aux] = Feedback;
}

return ( (short) PID_out );
}
/* NON USED
void ResetControlLoop_OLD(
    struct CONTROL_LOOP          *ControlLoop,
    int      ResetTimer)
{
    ControlLoop->ErrorPrev = 0;
    ControlLoop->IntegrTermPrev = 0;
    ControlLoop->PID_out = 0;
    if (ResetTimer)
        ControlLoop->TiDown = 0;
}
*/
void ResetControlLoop(
    struct CONTROL_LOOP          *ControlLoop,
    int      Feedback,

```




```
int ResetTimer)
{
    short s_aux;

    ControlLoop->ErrorPrev = 0;
    ControlLoop->IntegrTermPrev = 0;
    ControlLoop->PID_out = 0;

    for ( s_aux = 0; s_aux <= ControlLoop->ND; s_aux++)
        ControlLoop->FeedbackArray[s_aux] = Feedback;

    if (ResetTimer)
        ControlLoop->TiDown = 0;
}

void ResetPowerLoop(
    struct POWER_LOOP *PowerLoop)
{
    PowerLoop->TiUp = PowerLoop->hX10;
}

/**Actualitzacions de temps i valors de constants per a un nou cicle.
```

Així es gestiona el PID. Prenent aquesta programació com a guia, es pot dissenyar ja el primer subVI, el que simularà l'acció del control PID.



11. Model PID en LabVIEW

Per començar a treballar amb LabVIEW és important que es controlin tots els recursos que s'han d'utilitzar. Com que és una subrutina, no interessa l'aspecte final que tingui el panell frontal, per tant es treballarà en el panell de codi o "block diagram". Si es fa click amb el botó dret, es veu com s'obre el menú d'objectes. Es poden trobar els enters, booleans, operacions, vectors, estructures i moltes coses més. Així doncs, a partir d'aquí es comença a seguir el que ens dicta el codi en C.

Es pot observar com hi ha variables que apareixen dins d'altres grups que no s'han mencionat. Això és perquè en la declaració de variables s'agrupen i se'ls li dóna un nom al grup. Simplement, s'ha de fer l'operació inversa per trobar la variable que es desitja.

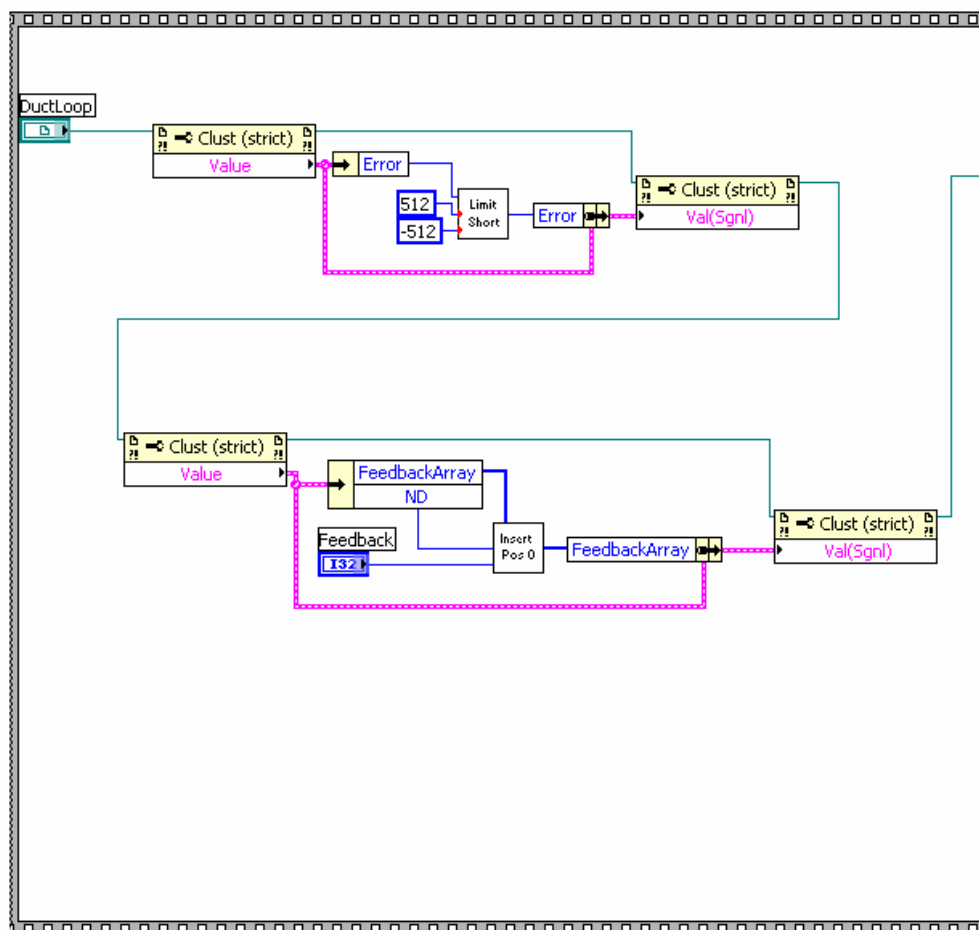


Figura 18 – Part del programa on s'aplica la funció LimitShort



```
// Error limitation
LimitShort(
    &(Loop->Error),
    MAX_POS_PID_ERROR,
    MAX_NEG_PID_ERROR);
```

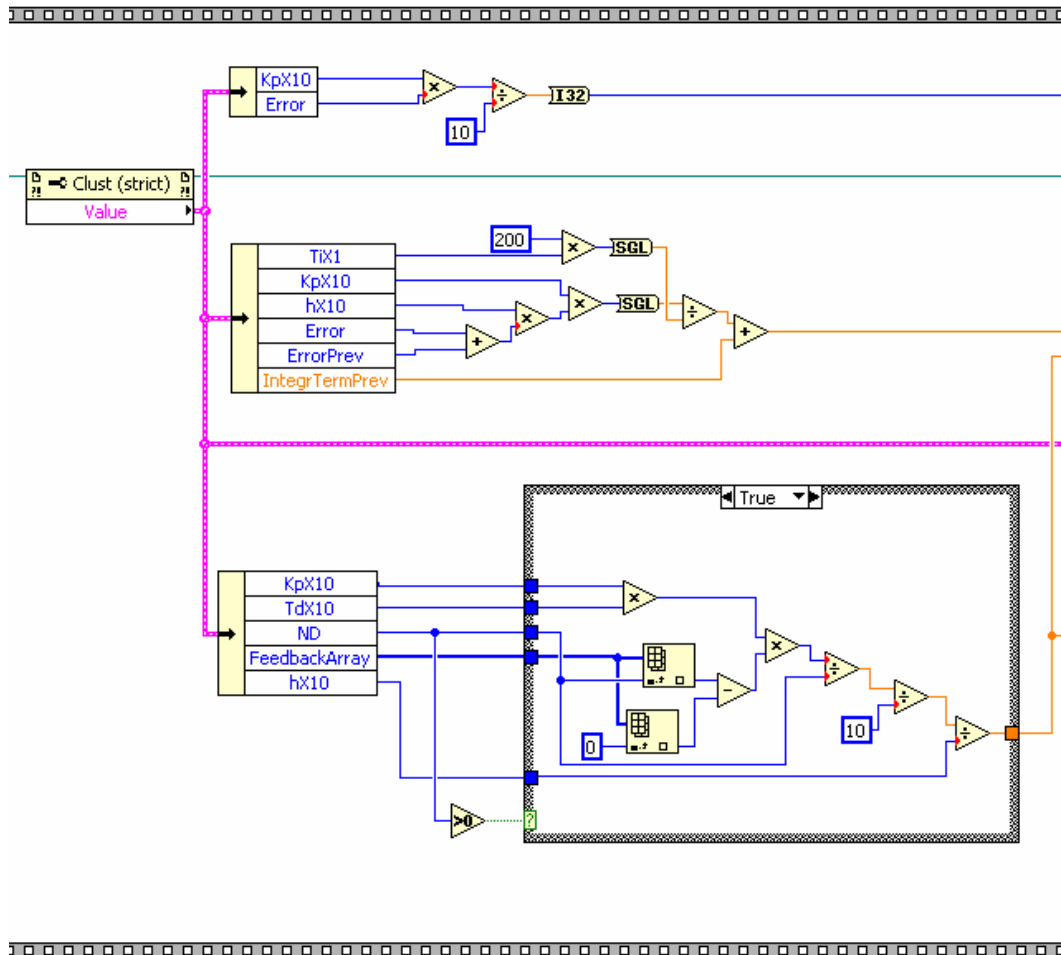


Figura 19 – Part del programa on s'aplica el control PID

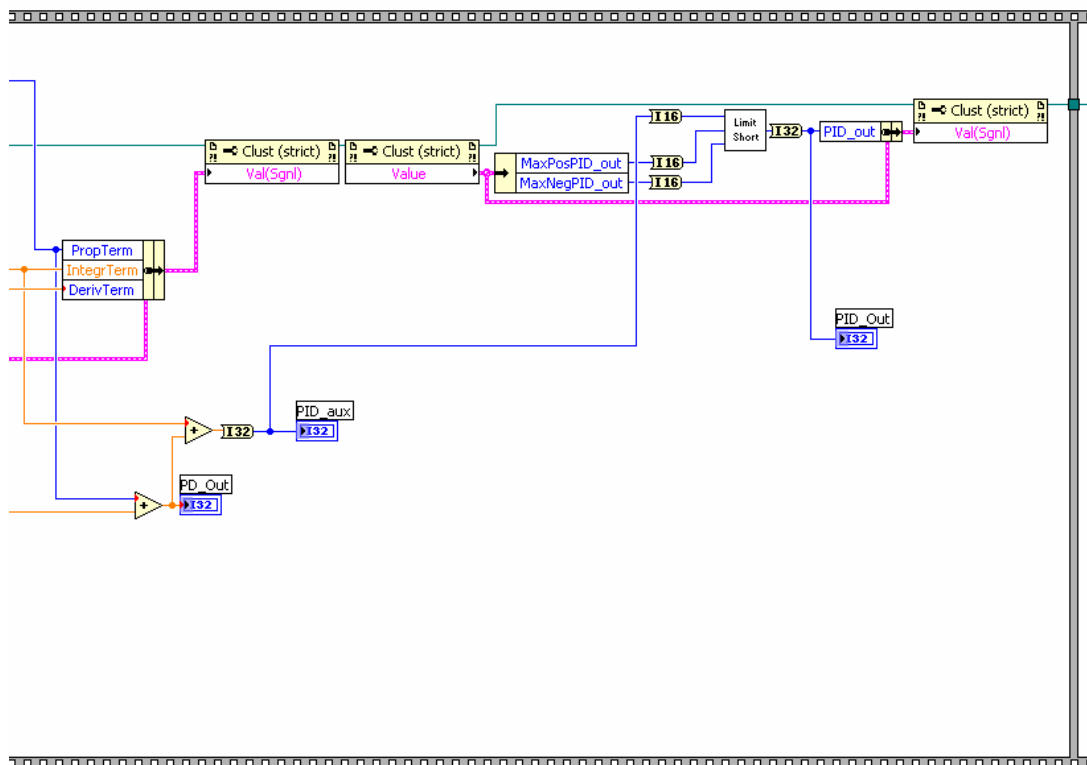


Figura 20 – Part del programa on s'aplica el control PID

```
// Derivative Term
if ( Loop->ND > 0 )
{
    s_aux = Loop->ND;
    while ( s_aux > 0 )
    {
        Loop->FeedbackArray[s_aux] = Loop->FeedbackArray[s_aux - 1];
        s_aux--;
    }
    Loop->FeedbackArray[0] = Feedback;

    Loop->DerivTerm = (Loop->KpX10) * (Loop->TdX10);
    Loop->DerivTerm *= (Loop->FeedbackArray[Loop->ND] - Loop->FeedbackArray[0]);
    Loop->DerivTerm /= Loop->ND;
    Loop->DerivTerm /= 10;
    Loop->DerivTerm /= Loop->hX10;
}
else
    Loop->DerivTerm = 0;
if ( !OnlyProp )
{
    // Integral Term
    if ( !IntegralFreezing && ( Loop->TiX1 > 0 ) )
    {
        AuxFloat = (float) (200 * (Loop->TiX1));
        IntegAux = (Loop->KpX10) * (Loop->hX10) * ((Loop->Error) + (Loop->ErrorPrev));
        Loop->IntegrTerm = (float) IntegAux;
    }
}
```



```

Loop->IntegrTerm /= AuxFloat;
Loop->IntegrTerm += Loop->IntegrTermPrev;
}
else
    Loop->IntegrTerm = Loop->IntegrTermPrev;
// Proportional Term
Loop->PropTerm = (Loop->KpX10) * (Loop->Error);
Loop->PropTerm /= 10;

// PD output
PD_out = (Loop->PropTerm) + (Loop->DerivTerm);

// PID output
PID_out = PD_out + (int) (Loop->IntegrTerm);
PID_aux = PID_out;

// PID output limiting
s_aux = (short) PID_out;

LimitShort(
    &(s_aux),
    Loop->MaxPosPID_out,
    Loop->MaxNegPID_out);
PID_out = s_aux;

```

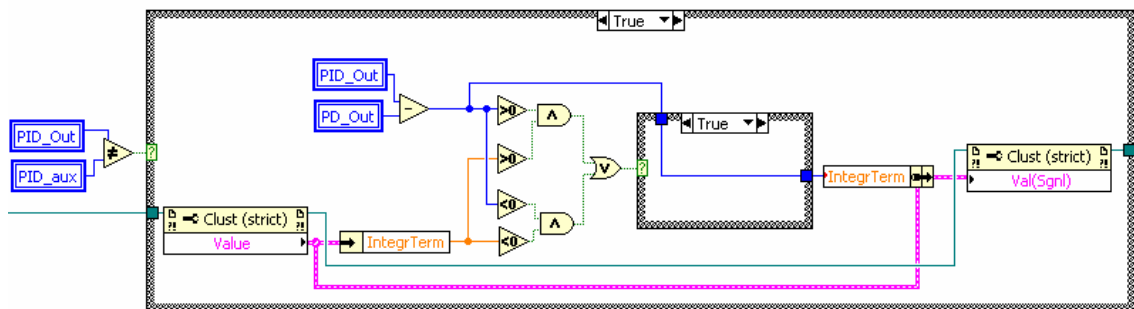


Figura 20 – Part del programa on s'aplica el tractament de sortida del PID

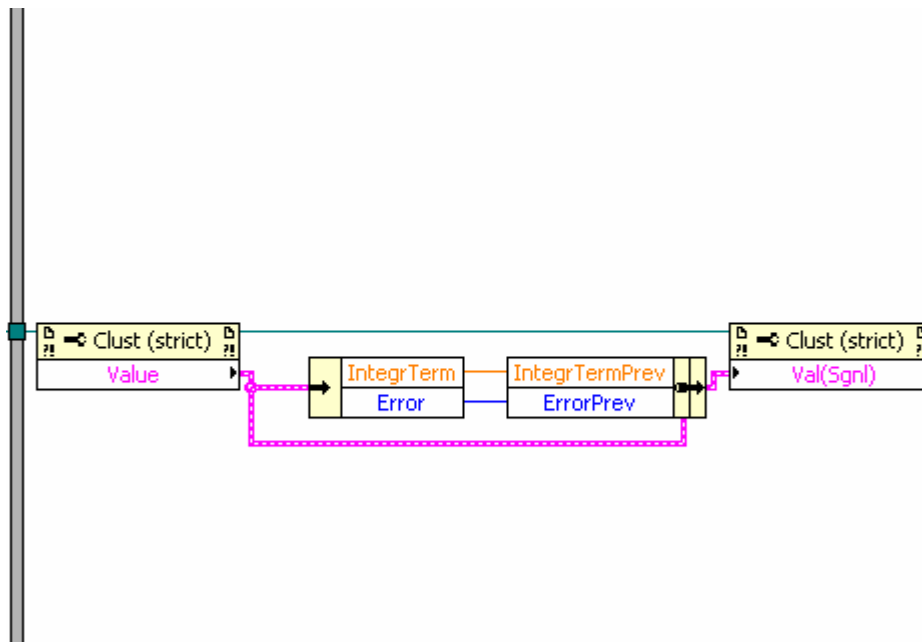


Figura 21 – Part del programa on s'actualitzen els termes integrals i l'error

Es poden notar totes les variables amb els noms que s'utilitzen al programa en C. Per exemple, es pot trobar K_pX_{10} , T_iX_{10} , T_dX_{10} , PID_OUT , $Error$, etc...

A més a més, es veu com intervenen d'altres subVI's com el `LimitShort` que s'ha comentat abans, amb uns límits superior i inferior de 512 i -512 en un cas, o del valor `MaxPosPID_OUT` i `MaxNegPID_OUT`.

També s'observa com es desagrupen els grups de variables per seleccionar-ne una o varies, o com es guarden aquestes un altre cop dins el grup. Aquests grups s'anomenen clusters.

Al final, s'actualitzen els valors previs, que en el següent Scan, seran els actuals.



12. Obtenció de l'equació en diferències

Per a poder operar l'equació de segon grau de la simulació, s'ha de passar a equació en diferències per així poder treballar amb el LabVIEW més còmodament.

Així doncs, es partim de l'equació que s'ha obtingut anteriorment:

$$FT = \frac{0.0000686792}{s^2 + 0.0165746 \cdot s + 0.0000686792}$$

Es passa l'equació en pla S a pla Z, amb un temps de mostreig de 0.11 segons, que és un temps correcte i òptim per a mantenir la funció semblant sense que perdi precisió pel fet de passar-la a digital i a més a més amb un retenidor d'ordre zero (r.o.z.). Així doncs, es té:

$$FT(z) = \frac{0.0000004153 \cdot (z + 0.9994)}{(z - 0.9991)^2}$$

Es pot escriure la funció de transferència com el quocient del valor de la sortida entre el valor de l'entrada i desenvolupar-lo:

$$\frac{Y(z)}{X(z)} = \frac{0.0000004153 \cdot (z + 0.9994)}{(z - 0.9991)^2} = \frac{0.0000004153 \cdot z + 0.00000041505082}{z^2 - 1.9982 \cdot z + 0.99820081}$$

Un cop fet això, es pot multiplicar cada factor per obtenir una equació més còmode:

$$Y(z) \cdot [z^2 - 1.9982 \cdot z + 0.99820081] = X(z) \cdot (4.153 \cdot 10^{-7} \cdot z + 4.1505082 \cdot 10^{-7})$$

Si es multiplica a cada banda per z^{-2} es tindrà l'equació encara més senzilla, ja que per passar a equació en diferències es necessiten exponents negatius perquè significaran al cap i a la fi, els valors dels estats anteriors. Si fossin valors positius s'estaria parlant d'estats que encara no han passat i això físicament és impossible.



$$Y(z) \cdot [1 - 1.9982 \cdot z^{-1} + 0.99820081 \cdot z^{-2}] = \\ = X(z) \cdot [4.153 \cdot 10^{-7} \cdot z^{-1} + 4.1505082 \cdot 10^{-7} \cdot z^{-2}]$$

Si s'aïlla la incògnita desitjada, que és la sortida de la simulació, es té:

$$Y(z) = [1.9982 \cdot z^{-1} \cdot Y(z) - 0.99820081 \cdot z^{-2} \cdot Y(z)] \\ + [4.153 \cdot 10^{-7} \cdot z^{-1} + 4.1505082 \cdot 10^{-7} \cdot z^{-2}] \cdot X(z)$$

Finalment es passa ja a equació en diferències:

$$y(k) = [1.9982 \cdot y(k-1) - 0.99820081 \cdot y(k-2)] + [4.1505082 \cdot 10^{-7} \cdot x(k-1) \\ + 4.1505082 \cdot 10^{-7} \cdot x(k-2)]$$

Es pot observar com la sortida depèn en certa proporció de l'estat anterior i el dues vegades anterior de la mateixa sortida, i depèn també proporcionalment de l'estat anterior de l'entrada.

Per a traslladar aquest resultat final a LabVIEW i seguir amb la simulació, ja es pot exposar quina forma tindrà aquesta equació per a ser operativa. Si es presta atenció, es pot assimilar que la manera més senzilla és separar en dos vectors diferents els valors que multipliquen l'entrada i la sortida. Per tant crearem dos vectors que anomenarem Array_A_Conducto (per a la sortida) i Array_B_Conducto (per a l'entrada).

També es farà una petita adaptació escalar. Les entrades s'hauran d'adaptar per a les temperatures d'exemple que hem agafat per a fer el sistema de segon ordre. Així doncs, s'haurà de partir dels 25.5°C i arribar a pujar 31.6°C, fins als 57.1°C. Com que es treballa en dècimes de grau, es multipliquen els valors del vector de l'entrada per 255 i 316.



D'aquesta manera, els vectors en el programa queden de la següent manera:

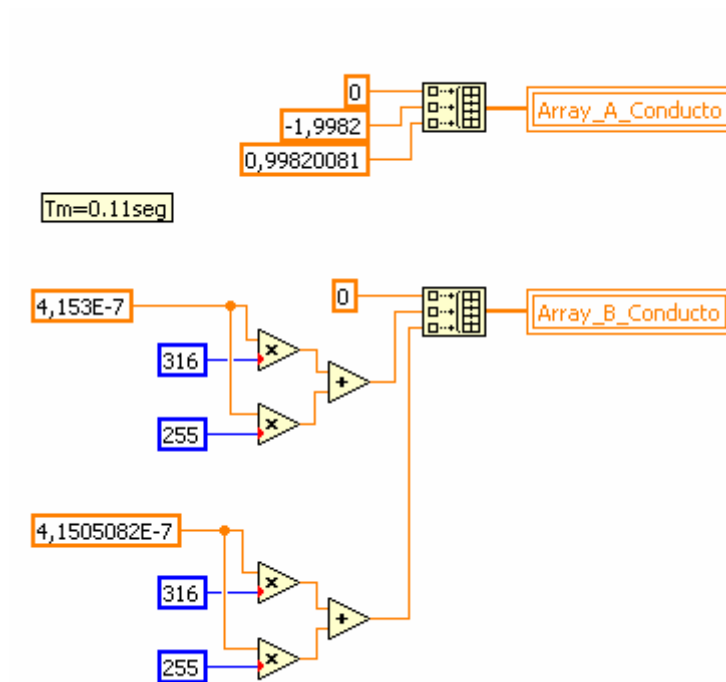


Figura 22 – Inicialització dels vectors de les equacions en diferències

Es pot veure com es crea un vector. Les tres posicions del vector en l'ordre desitjat s'introdueixen al símbol de inicialitzar vector. La sortida és una línia més gruixuda, que simbolitza que tenim un vector, i es guarda en la variable desitjada.



13. Control i regulació de l'aire condicionat

El valor de sortida del PID indica en quin estat ha d'estar el sistema i la velocitat de canvi del valor del PID indica el temps de permanència en una o altra etapa. Com que el valor de la sortida del PID és continu és necessari acotar rangs de valors. S'establiran cinc etapes de funcionament diferents:

- Pre-Escalfament (Pre-Heating): Quan s'engeguen els motors i l'aire del vagó és més fred que la temperatura desitjada (normalment seran 22°C), s'entra en aquesta fase. Consisteix en estar-se un cert temps (depenent de la temperatura del vagó) amb els motors activats al 100% en mode de calefacció per a assolir una temperatura prou raonable com perquè després entri en funcionament el control de l'autòmat.
- Escalfament (Heating): Un cop s'ha pre-escalfat la sala l'autòmat entra en funcionament i comença a regular-ne la temperatura. Aquesta és una de les fases que es simularà i correspon a un valor positiu de la sortida del PID. Segons la temperatura de la sala, l'autòmat donarà una consigna de temperatura de conducte. A partir d'aquí i durant cada cicle de 210 segons, l'aire condicionat s'activarà durant un cert temps, amb un o dos motors funcionant i escalfarà el conducte a la temperatura adequada i dictada pel programa de l'autòmat.
- Ventilació (Ventilation): Quan s'està dins d'un marge de $\pm 1^\circ\text{C}$ de la temperatura de consigna el motor entra en una fase de ventilació. Simplement, ni s'escalfa ni es refreda la sala, sinó que l'aire recircula pel vagó. Pot ser que recirculi tot l'aire, només una part i l'altra part provinent de l'exterior o tot l'aire es renovi amb aire nou de l'exterior.
- Pre-Refredament (Pre-Cooling): Tal com passa amb el pre-escalfament, quan es té una temperatura inicial a l'hora d'activar el motor molt alta, s'engeguen els motors a plena potència en mode de refrigeració durant uns minuts. Quan



s'assoleix una temperatura més baixa entrarà en funcionament el control de l'autòmat.

- Refredament (Cooling): Un cop s'ha pre-refredat el vagó l'autòmat entra en funcionament i comença a regular-ne la temperatura. Aquesta fase també es pot simular i correspon a un valor negatiu de la sortida del PID. Segons la temperatura del vagó, l'autòmat donarà una certa sortida pel control de l'aire condicionat. L'activació d'un o dels dos motors i el temps que s'estaran activats durant els 210 segons que dura un cicle.

Així doncs, es pot fer un esquema molt representatiu del funcionament de l'aire condicionat segons l'etapa on ens trobem i des de cada una cap a quina etapa ens podem dirigir:

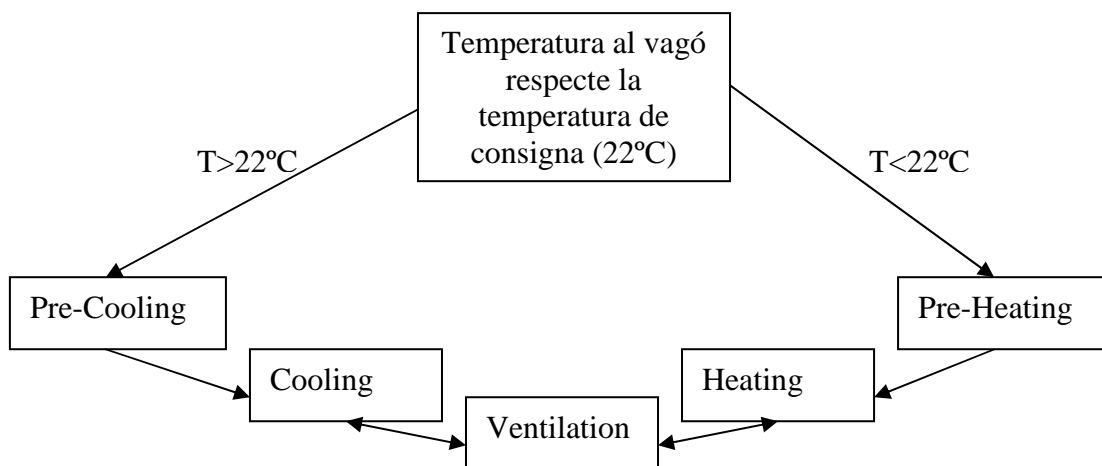


Figura 23 - Esquema de modes de funcionament

Els modes Pre-Heating i Heating corresponen a l'estat de calefacció i els modes Pre-Cooling i Cooling corresponen a l'estat de refrigeració. Ventilation és un estat on els motors ni escalfen ni refreden. Només fan recircular l'aire pel vagó o cabina d'una manera o altra tal com s'ha explicat.

Per a establir l'etapa, ja sigui a l'inici (activació dels motors) o en qualsevol moment s'han d'observar les sondes de temperatura del vagó. Són dues i es dona a la targeta



d'adquisició de dades la mitjana d'elles. Segons la temperatura demandada pel control, es decidirà en quina etapa de funcionament ha d'operar l'equip.

El temps mínim de permanència en una etapa és de 15 segons, excepte quan s'ha de parar el compressor, llavors l'equip s'espera 5 minuts abans de poder tornar a arrencar. Un cop en marxa de nou, l'equip estarà funcionant almenys 2 minuts abans de tornar-se a parar el compressor. En calefacció, com que hi ha dos motors s'efectua una combinació de temps i activació de cadascun d'ells per a aconseguir l'escalfament desitjat. Sempre s'activa primer el motor de 6 kW i fins que no és necessària més potència (i el motor de 6 kW està funcionant a ple rendiment i durant tot el temps de cicle), no s'activa el motor de 12 kW.

Altres requisits ben lògics i que estan inclosos en el programa en C de control són per exemple que la calefacció de la sala no entrarà en funcionament si la temperatura exterior és superior a 22 °C. El temps mínim de marxa és d'1 minut. En el sistema de cabina també s'executa el mateix funcionament.

Tal com s'ha comentat abans, la temperatura de la sala es calcula com a promig de les lectures dels dos sensors de temperatura. En cas de que una de les sondes quedés fora de rang, és a dir oberta o en curtcircuit, el programa rebutjarà aquesta lectura i només considerarà correcta la lectura de la sonda que funciona. Si la lectura de la sonda de temperatura exterior és errònia se li assigna el valor de 21°C.

La refrigeració també té diferents requeriments especials. La sonda de temperatura del conducte s'utilitza per activar les inhibicions per baixa temperatura en els conductes. Si la temperatura de l'aire impulsat és menor de 5°C s'activa la vàlvula del 50% i l'equip de refrigeració actua al 50% de la seva potència màxima. Si la temperatura d'impulsió baixa per sota dels 3°C s'activen les vàlvules del 50% i el by-pass conjuntament i l'equip de refrigeració no aporta fred.

També existeixen limitacions segons la temperatura que registri la sonda exterior. La posada en marxa del sistema de refrigeració s'anul·la si la temperatura exterior és menor de 15°C i s'inhibeix si la temperatura exterior és major o igual a 16°C. Igualment



s'inhibeix la posada en marxa de la calefacció si la temperatura exterior és igual o està per sobre dels 22°C i es permet la seva actuació si la temperatura és menor de 21°C.

Tal i com s'ha explicat, en l'arrencada el sistema de climatització pot funcionar, en refrigeració o calefacció, a plena potència fins que s'aconsegueix una temperatura pròxima a la consigna (setpoint). Un cop arribat a aquest punt el sistema funciona normalment i sense tenir en comte l'error acumulat entre temperatura mitjana i setpoint durant el procés. Les inhibicions i limitacions es mantenen operatives durant tot el procés de pre-acondicionament.

L'equip de climatització té la possibilitat de realitzar un test. En un altre programa amb LabVIEW que s'explicarà més endavant es veurà com es pot dur a terme un manteniment del sistema, generar situacions per a crear els estats de calefacció i refrigeració, estats on no funciona (es deshabilita) algun component, etc.



14. Traducció de C a llenguatge gràfic LabVIEW

Ja s'ha vist un exemple (PID) de com es tradueix el llenguatge C a LabVIEW, però no s'ha explicat ben bé cada operació com es realitza. Existeixen molts objectes que necessiten una explicació. Tot seguit es comentaran:

- **Operacions aritmètiques:** Són les sumes, restes, multiplicacions, divisions, residus, potències, etc. Es poden realitzar sobre qualsevol nombre de qualsevol mida, ja sigui sol o en un vector o matriu, un paràmetre o una variable ja anomenada, etc.. Permet realitzar totes les operacions vàlides i si hi ha una relació errònia (com per exemple multiplicar dos vectors de mida diferent) el programa mostra l'error. Són operacions fàcils de connectar ja que disposen normalment de dues entrades i d'una sortida i aquesta es pot guardar segons el tipus de variable que és, ja sigui enter, vector, matriu.... Per exemple, aquí tenim algunes mostres.

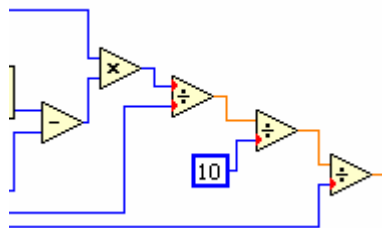


Figura 24 – Símbols d'operacions aritmètiques

És molt representatiu que en aquest exemple el que estem fent és una resta de dos elements seguit d'una multiplicació i tres divisions. S'observa com a l'inici els connectors són de color blau. Això és perquè el programa els tracta com si fossin enters de 16 bits o menys. De fet, segur que ho són. Però a l'arribar una divisió el resultat probablement ja no serà enter i per tant es converteix en real de 64 bits. El connector taronja dóna mostra d'això. Si un es fixa per exemple en el número 10, s'observa com just a l'entrar al símbol de divisió hi ha un punt vermell. Aquest punt significa que es tradueix l'enter a real.

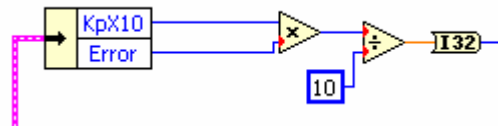


Figura 25 – Símbols d'operacions aritmètiques

Aquest és un altre exemple curiós. Es multipliquen dos elements que són variables que estan agrupades en un cluster (línia rosa), es divideixen i dona un resultat real. Llavors hi ha un símbol que ens transforma el connector taronja en blau. Simplement acota el número que es té a la sortida de la divisió a 32 bits.

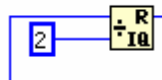


Figura 26 – Símbols d'operacions aritmètiques

L'element que s'observa ja necessitaria l'activació de l'ajuda. És una divisió però que com a sortida dona el residu (R) i el quocient (IQ). Així doncs, el fet de dividir entre 2 és un bon mètode per a saber si es té un nombre parell o senar. A més a més, aquesta operació es pot fer tant amb nombres enters com amb reals.

- **Operacions booleanes:** Un altre tipus de variable amb la que es treballa en LabVIEW són les variables booleanes, que es representen amb el color verd. Aquestes requereixen de certes operacions lògiques que són exclusives com pot ser la negació, la funció OR, la funció AND i altres. Així doncs, per exemple podem tenir:

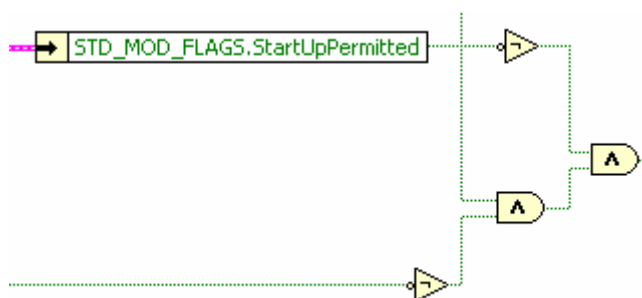


Figura 27 – Símbols d'operacions booleanes



En aquest exemple de programa es veu com d'un cluster es treu una variable booleana. Es nega i es combina amb una porta AND. L'altra entrada de la porta AND també prové d'una port AND, amb dues entrades, una negada. Com es pot comprovar, és molt gràfic i llegible quina funció pertany a cada símbol.

- **Operacions amb caràcters:** Són operacions per a llegir, escriure, detectar caràcters o "strings", obrir arxius de text o XML i d'altres. El més important són les eines per a treballar amb arxius XML, ja que són arxius que guarden dades sobre les variables que es poden utilitzar en programes i simplement llegint aquest arxiu XML i enregistrar tots els seus caràcters en variables de "strings", i llavors ja es tenen carregades les dades en el programa. També és útil per a crear arxius d'històrics.
- **Operacions de gestió de fitxers:** Són icones que tracten amb operacions bàsiques com són obrir i triar un fitxer d'una carpeta, guardar o fins i tot crear-lo. Són les operacions que s'utilitzaran a l'hora d'obrir un arxiu XML, per exemple.
- **Dialog&UserInterface:** El més interessant d'aquest apartat és la icona de l'error. Totes aquelles icones susceptibles de crear un error (Open file, Read XML File, etc...) tenen una sortida que es diu error. Si es llegeix l'error amb la icona esmentada, podem detectar-lo i fer amb ell certes operacions. Es pot obviar, es pot preguntar per si es vol continuar executant el programa tot i l'error i d'altres opcions.
- **Operacions de comparació:** Són operacions on entren dos nombres reals o enters i surt un booleà. És el cas d'operacions com més gran que, més petit que, igual... També hi ha operacions com si un nombre és igual o diferent de 0. En aquest cas només hi ha una entrada. També entren en aquest grup, però sense ser operacions de comparació els increments i decrements. A continuació es poden veure alguns exemples:

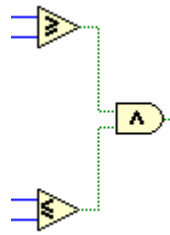


Figura 28 – Símbols d'operacions de comparació

S'observa com hi ha un símbol de major o igual i un altre de menor o igual. En els dos casos entren dues variables enteres i surt una variable booleana, que en aquest cas després d'això, es realitza una operació AND.



Figura 29 – Símbols d'operacions de comparació

En aquest dos casos es pot veure com tots els símbols són molt gràfics. És lògic com es pot veure que si s'està comparant un nombre amb si és més gran o no que zero, el símbol només tingui una entrada. En canvi, si s'està comparant la igualtat de dos nombres, evidentment hi haurà dues entrades.

La majoria de vegades aquestes comparacions serveixen per entrar en estructures condicionals: Si el nombre és més gran que 0, fer això, sinó, fer allò.

A continuació es veurà com s'organitza aquesta estructura i d'altres.

- **Estructures:** Són les estructures de iteració, condició, ... (if, while, for...) i tenen una forma molt clara. Un bloc tancat, on dins s'hi pot escriure el que es desitgi. Cadascun té les seves característiques com per exemple que en una iteració una de les entrades serà el nombre de repeticions i a dins hi haurà un comptador on mostrarà el nombre de repeticions que es porten. O una condició tindrà com a entrada un booleà que determinarà la condició certa o falsa i es permetrà escriure



dins el bloc la condició certa o falsa simplement fent click en una fletxa a la part superior. A continuació alguns exemples:

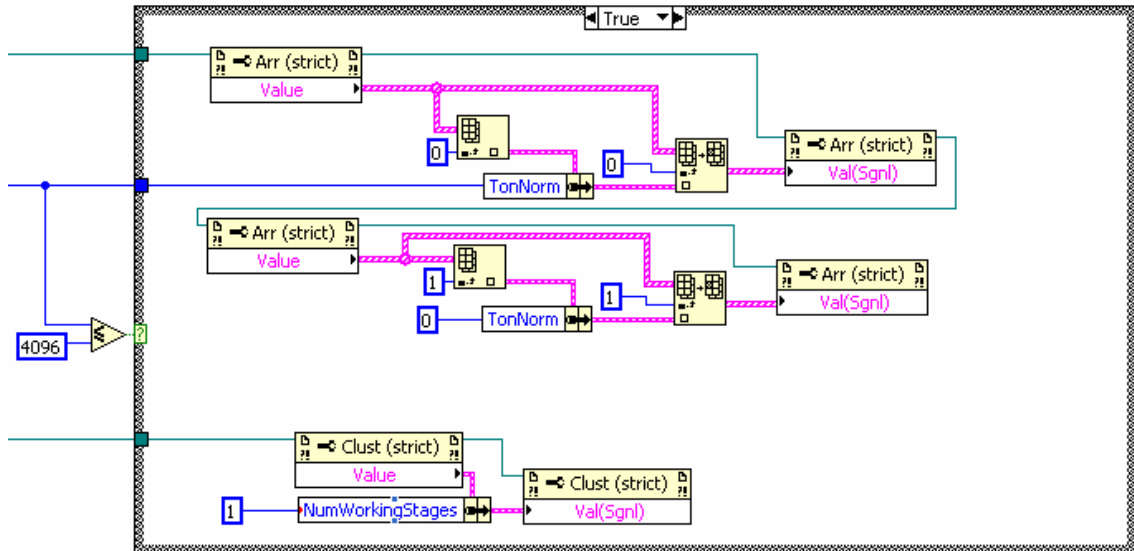


Figura 30 – Estructura condicional certa

Aquí hi ha una estructura condicional que mostra la condició certa i el que es realitza dins. En aquest cas, s'entrarà a l'estructura quan el valor de l'entrada sigui més petit o igual que 4096. Si l'entrada és més gran que 4096 s'entrarà en el cas fals i es tindrà el següent.

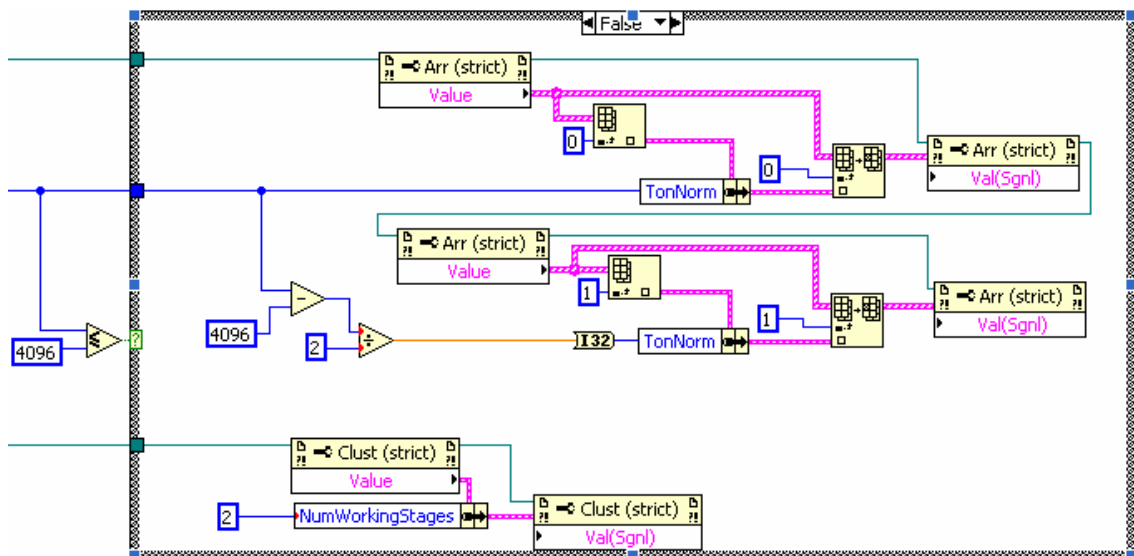


Figura 31 – Estructura condicional falsa



També podem trobar estructures condicionals més senzilles i amb entrades provinents d'un cluster com pot ser la següent captura.

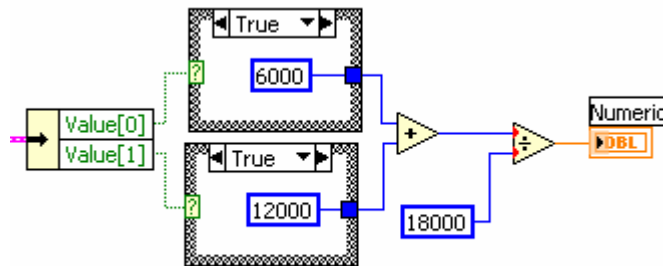


Figura 32 – Estructures condicionals certes

Dos valors booleans provinents d'un cluster condicionen l'entrada a l'estructura condicional. A més a més es pot veure com al final de tota la operació el nombre resultant es guarda en la variable "Numeric". De fet, la part de programa plasmada és de la subrutina per a obtenir la potència unitària. Els dos booleans són l'activació dels dos motors, el de 6 kW i el de 12 kW. El resultat es suma i es divideix entre 18 kW, que és la potència total. D'aquesta manera s'obté el valor unitari de la potència total. Aquesta pot tenir un valor de 0 (motors apagats), 1 (motors engegats) o 0.33 (motor de 6 kW engegat, ja que és l'únic que pot funcionar sol).

Una estructura "for" pot ser per exemple aquesta.

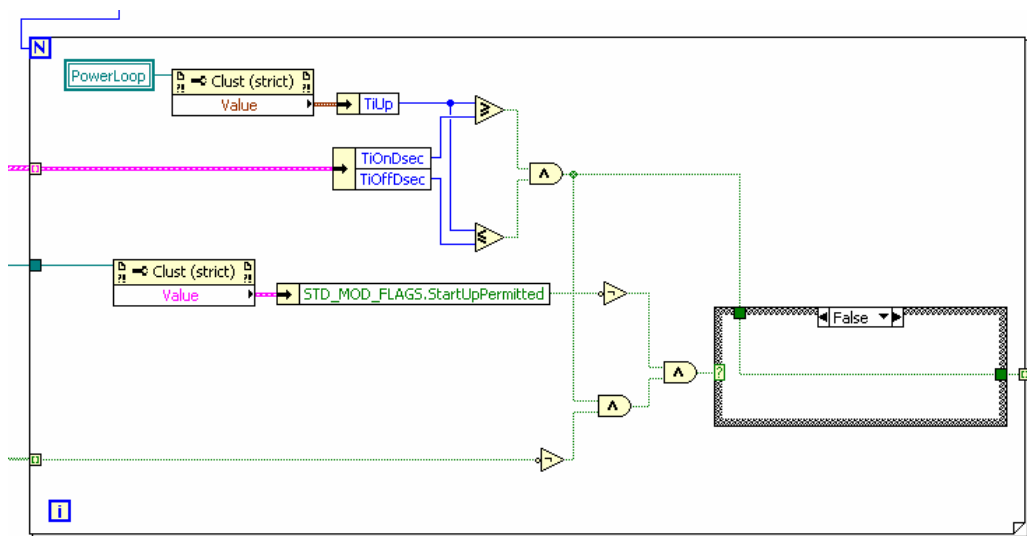


Figura 33 – Estructura iterativa



Es pot veure com a la part superior esquerra hi ha una entrada amb una “N”. Aquest serà el nombre de repeticions del bucle. A la part inferior esquerra es pot veure un petit símbol amb una “i” que representa la repetició que s’està duent a terme. De vegades pot ser útil aprofitar aquest valor per a treballar amb alguna variable dins de l’estructura. A més a més es pot veure com si a l’entrada tenim un vector (com és el cas, un s’ha de fixar en el connector verd de sota), a l’entrar dins de l’estructura s’agafa la i-èsima posició d’aquest vector. Això es veu clar perquè abans d’entrar a l’estructura el connector és més gruixut (simbolitzant així que és un vector) i a l’entrar aquest es converteix en un connector més prim. A la sortida passa el contrari, surt un vector.

Una estructura molt utilitzada és la “Flat Sequence”. Es tracta d’una estructura que separa diferents parts del programa per fer-lo més estructurat, ordenat i entenedor. El seu perfil és semblant al d’un rodet de pel·lícula.

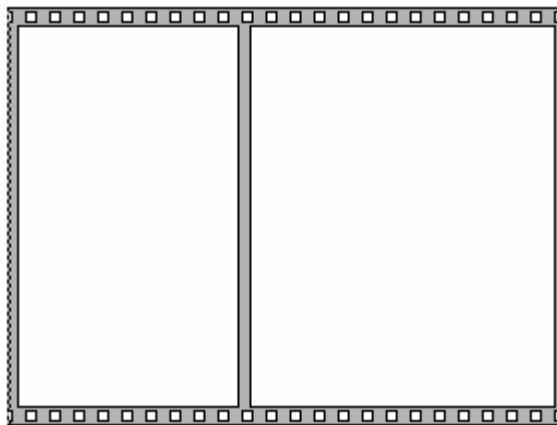


Figura 34 – Estructura “Flat Sequence”

També existeix l’estructura while, que mentre no es compleixi una condició, el bucle s’executarà infinitament. En el nostre cas aquesta estructura es fa servir per a executar el programa de simulació. Mentre no s’activi la parada s’ha d’anar executant.

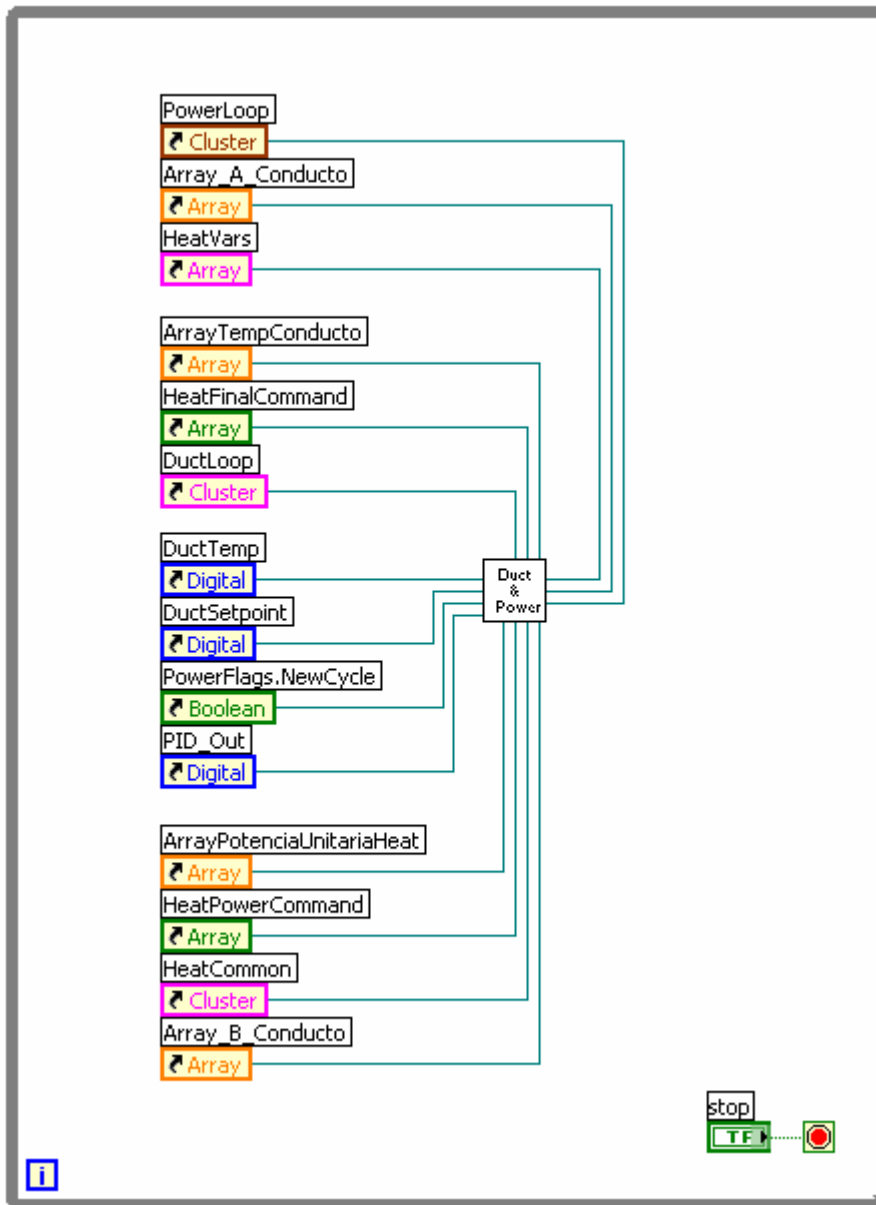


Figura 35 – Estructura “while”

També disposa de la variable que compta en quina i-èsima repetició s'està. Com es pot observar, una variable booleana controla la parada. Quan sigui cert que hem pitjat el botó de STOP (ja es veurà quan palrem del panell frontal del programa de simulació) s'acabarà el bucle.

- **Vectors:** Fins ara s'han pogut veure alguns símbols que encara no s'han explicat. Alguns d'aquests són les diverses operacions que es poden fer amb



vectors, com inicialitzar, insertar elements, eliminar parts del vectors, trobar la seva mida, eliminar parts, etc.

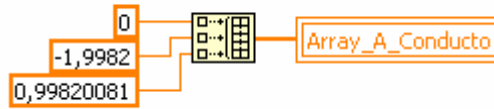


Figura 36 – Símbol d'inicialització de vectors

El símbol que s'observa crea un vector. Pot tenir les entrades que un vulgui simplement fent-lo més gran aniran apareixent entrades. En aquest cas són tres entrades reals que generen un vector que es guarda en la variable que es veu. Com ja s'ha comentat, el connector que transporta una dada de vector és més gruixut que el que transporta una sola dada, sigui real, booleana, entera...

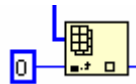


Figura 37 – Símbol de selecció de posició en un vector

Aquest és un altre símbol molt utilitzat. Es tracta de seleccionar la posició del vector de la qual es vol obtenir el seu valor. En aquest cas, volem obtenir la primera posició. Es pot veure com l'entrada superior és el vector, que és més gruixut i la sortida és un enter.

- **Clusters:** Aquesta és l'estructura més complicada de manejar de LabVIEW. Bàsicament un cluster és una agrupació de variables de qualsevol tipus. Per a crear-lo s'han d'inicialitzar totes les variables i agrupar-les amb una eina anomenada "Bundle" o "Bundle by name". Llavors es crea una variable cluster que es guarda. Per a seleccionar qualsevol variable dins del cluster s'ha de fer la operació inversa "Unbundle" o "Unbundle by name". La diferència entre aquestes operacions és que una no et mostra el nom de la variable que has inicialitzat i l'altra sí.

Aquests clusters s'han de tractar de manera especial si es volen utilitzar a diferents VI's. S'han de passar per referència. Per a poder fer això s'ha de crear



una referència. Així doncs, és molt típic que veiem una referència que se n'extreu el valor i aquest és un cluster. Poden ser altres coses com es veurà a continuació. El color de connector i variable que representa el cluster és el rosa mentre que el que es fa servir per a la referència és el color verd fosc. A continuació alguns exemples.

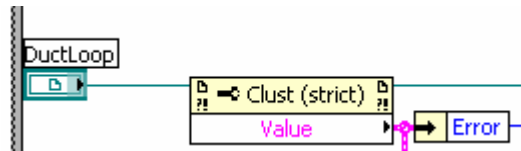


Figura 38 – Operació d'extracció d'un element d'un cluster

El símbol de DuctLoop és una referència, per tant, podem deduir que es mostra una part d'un subVI. Amb el símbol que hi ha a continuació s'extreu el seu valor. Aquest símbol és un node de propietats. Conté totes les característiques que pot tenir una referència. A partir d'aquí ja es té el cluster seleccionat i ja només cal destriar la variable que es desitja, en aquest cas, l'error. Es pot veure més clar tot això si es mostren les inicialitzacions del programa principal i més concretament, la inicialització del cluster DuctLoop.

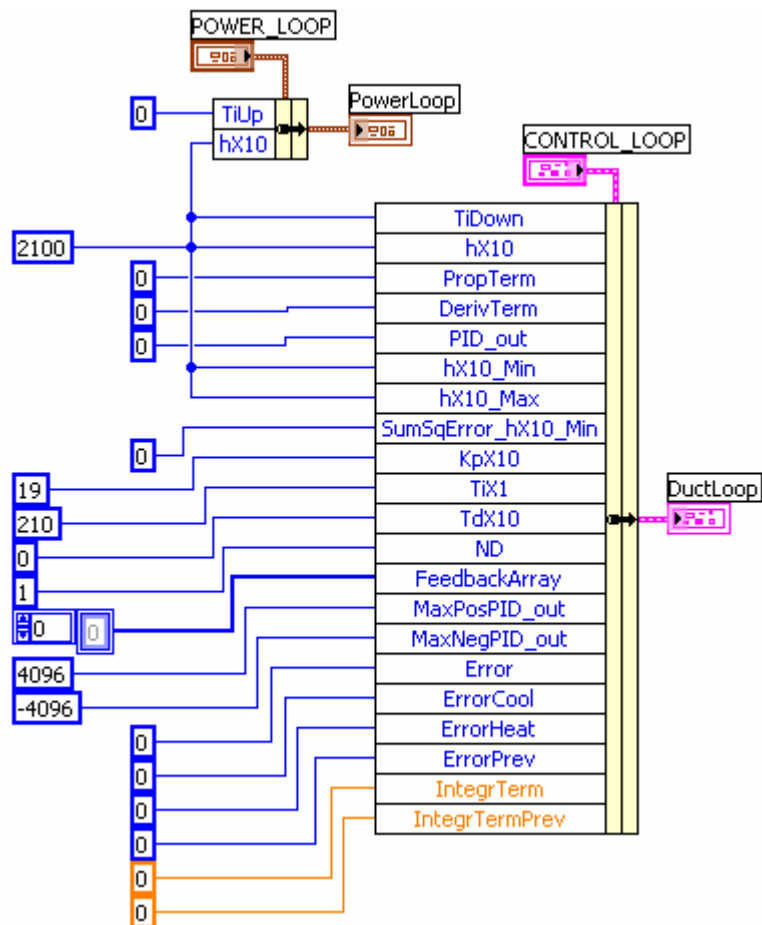


Figura 39 – Inicialització d'un cluster

Es pot veure com una de les variables inicialitzades és l'error. I tal com s'ha comentat, s'ha fet servir un "Bundle by Name" per a inicialitzar el cluster DuctLoop. El símbol que apareix a dalt com a CONTROL_LOOP és el mateix que el DuctLoop, l'únic que de la manera que es treballa a l'empresa se li ha donat un nom diferent per a tenir més clares les variables. Es pot observar com hi ha diferents tipus de variables com vectors, enters i reals. A cadascun se li pot assignar un nom per a tenir més clar quin element es tria quan es selecciona un cluster.

Després d'això es crea la referència de DuctLoop per a poder passar tota aquesta informació a un subVI.

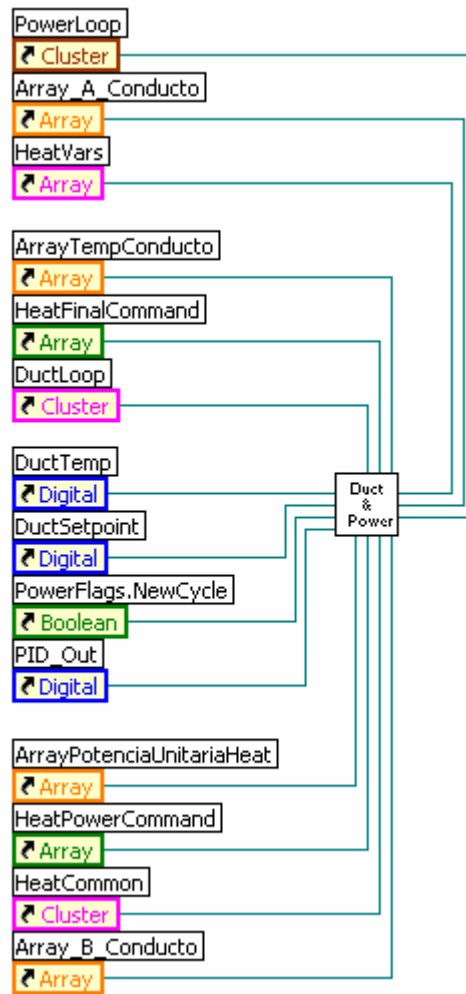


Figura 40 – Elements d’entrada a un subVI

Tot el que entra en el subVI són referències. Els diferents colors simbolitzen el tipus de variables que contenen.

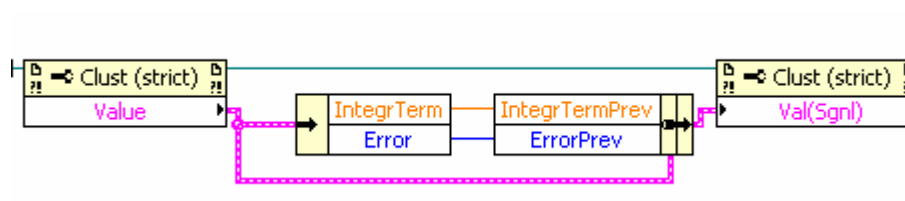


Figura 41 – Obrir i guardar elements en un cluster

En aquest cas es pot veure com el node de propietat també serveix per a guardar (Val(Sgnl)) elements del cluster. En el cas del “Bundle by Name” es pot observar com se li passa la referència del cluster (connector rosa de sota) perquè així sap de quin cluster està tractant i on es guarden les variables que té connectades.



Quan volem utilitzar més d'una vegada un element en un mateix VI o subVI s'han de crear variables locals. Simplement és una referència a una variable. La diferència entre una variable local i el seu símbol originari és aquesta.

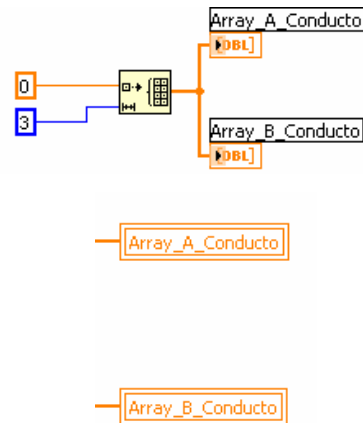


Figura 42 - Creació d'una variable local

En aquest cas es veu com s'inicialitzen els vectors. Després, quan es volen tornar a utilitzar utilitzem una variable local que es refereix al nom que conté. Aquesta variable local pot ser llegida o es pot escriure a sobre, quedant guardat el valor que se li assigna. En la figura superior s'escriu sobre una variable local.

- **Els subprogrames (subVI):** Són símbols que expressen la utilització d'un subprograma. Són uns quadres que es poden editar per escriure el nom del subprograma i que contenen entrades i sortides. En aquest subprograma s'enllaça a la icona els elements que es volen que siguin entrades i els que es volen que siguin sortides. Així doncs quan s'utilitza aquest subVI en un altre programa, s'enllacen com a símbols de lectura les entrades i com a símbols d'escriptura les sortides. Es pren com a exemple aquesta multiplicació entre dos vectors:

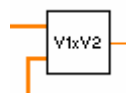


Figura 43 – Símbol de subVI

Hi ha dues entrades que són els dos vectors (més gruixuts) i una sortida que és el resultat de la multiplicació.



15. Funcionament del programa de simulació

Un cop s'ha escrit tot el programa en LabVIEW es pot procedir a crear l'arxiu executable. És un procés senzill que no cal explicar. Simplement es crea una carpeta amb una sèrie d'arxius on un d'ells és el programa executable, en qualsevol ordinador, sense necessitat de tenir el software que es fa servir per a fer córrer el programa (mode run).

L'aspecte general del programa un cop es comença a executar és el següent:

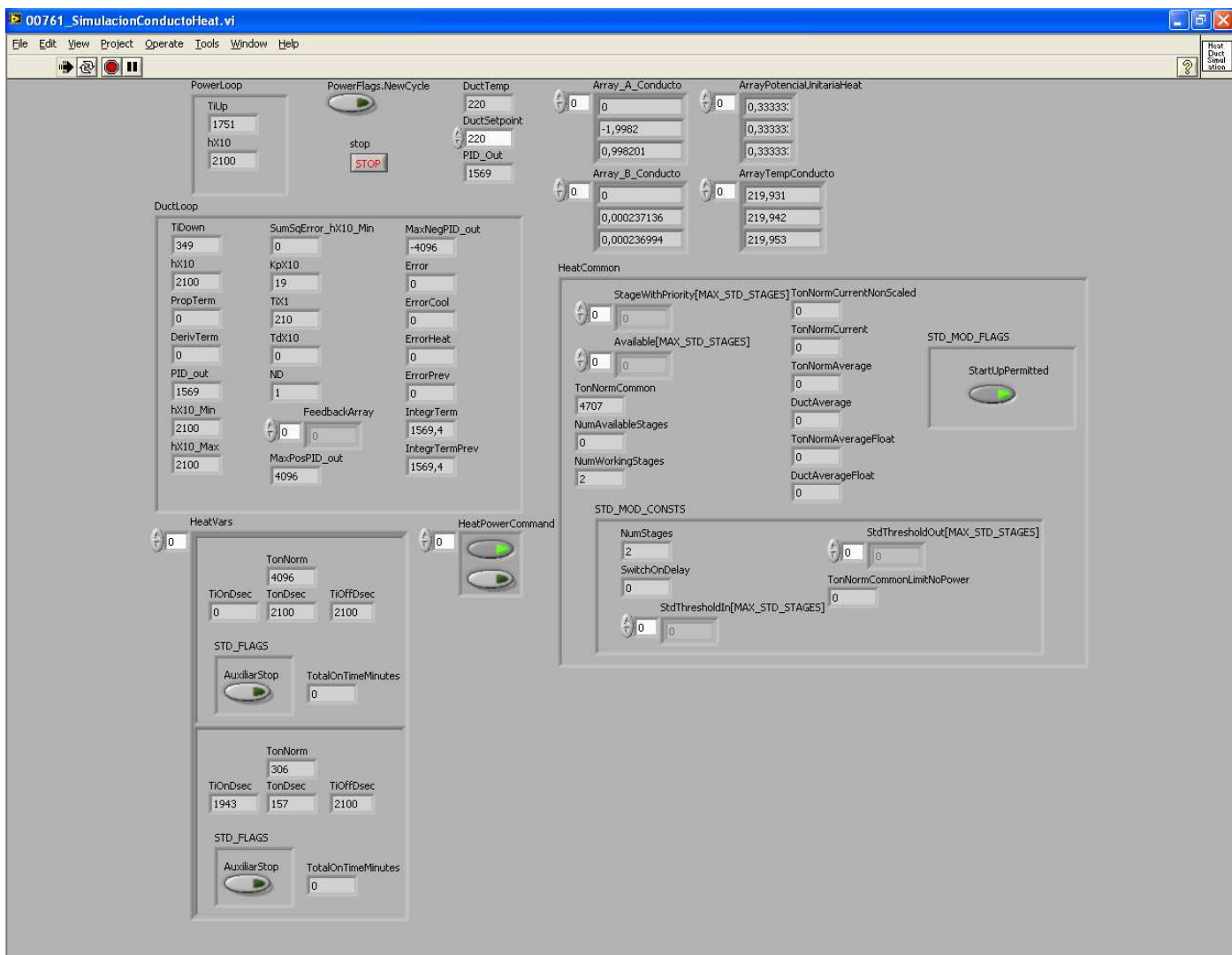


Figura 44 – Aspecte general del programa de simulació

A continuació s'explicarà cada component que es veu per a tenir més coneixement de les parts que intervenen en la simulació.



- PowerLoop:

Es tracta d'una petita finestra situada a la part superior esquerra que indica en quin instant de temps es troba la simulació actualment (compta dècimes de segon des de 0 i incrementant) i quin és el temps total de cicle. És un cluster.



Figura 45 – Estructura PowerLoop

- PowerFlags.NewCycle i STOP:

Són dos polsadors que estan just a la dreta del PowerLoop.

El PowerFlags.NewCycle indica quan s'inicia un nou cicle, activant-se en la primera dècima de segon. La simulació és tan ràpida que la majoria de cops no s'arriba a apreciar com s'encén la llumeta.

El botó de STOP tal com indica el seu nom atura l'execució del programa. Per a tornar-lo a fer córrer s'ha de prémer una botó amb una fletxa que simbolitza l'acció de RUN.



Figura 46 – Elements de control STOP i nou cicle

- Temperatures de conducte i valor de sortida del PID:

Seguint cap a la dreta es troben tres quadres. Contenen el valor de la temperatura actual de simulació del conducte, la temperatura de consigna i el valor de la sortida del PID. L'únic element que podem modificar és la temperatura de



consigna. La temperatura actual és nova per a cada dècima de segon del cicle i la sortida del PID s'actualitza a cada inici de cicle.

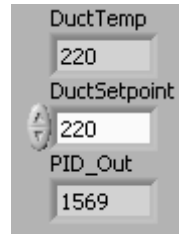


Figura 47 – Temperatura real, de consigna i sortida del PID

La temperatura que es mostra està en dècimes de grau. La temperatura del conducte real a l'inici de la simulació és de 255 (25.5°C)

- Vectors de constants:

Són les constants que s'han trobat mitjançant l'aproximació de segon ordre. Són els paràmetres de les equacions en diferències.

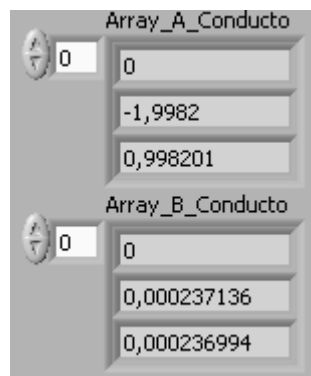


Figura 48 – Vectors de les constants de les equacions en diferències

- Vectors de potència unitària i temperatures actuals:

Finalment a la part superior dreta es troba el vector de la potència unitària que hi ha en cada instant de cicle. Si els motors estan apagats és 0. Si només funciona el de 6 kW és 0.33333 i si funcionen els dos és 1. Aquests valors aniran canviant a mesura que variï la temperatura de consigna i provoqui l'activació dels motors corresponents.



El vector de sota correspon a les temperatures actuals del conducte i la potència unitària.

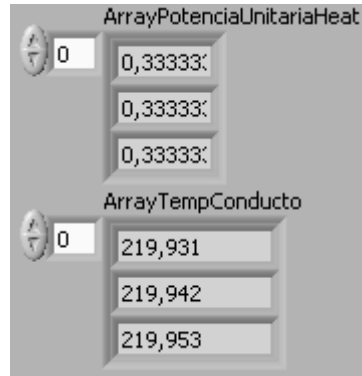


Figura 49 – Vectors de potència unitària i temperatura real

- DuctLoop:

És un altre cluster. Conté valors més específics sobre el PID i els temps de cicle. Per exemple hi podem trobar els valors de les constants proporcionals, integrals i derivatives del PID (K_pX10 , T_iX1 i T_dX10). Hi ha d'altres elements com són els rangs de l'error màxim, el temps de cicle total, el temps de cicle actual en mode de compta enrere, etc...

El més interessant és observar com varia l'error i com repercuteix això en el valor de la sortida del PID i el terme integral.

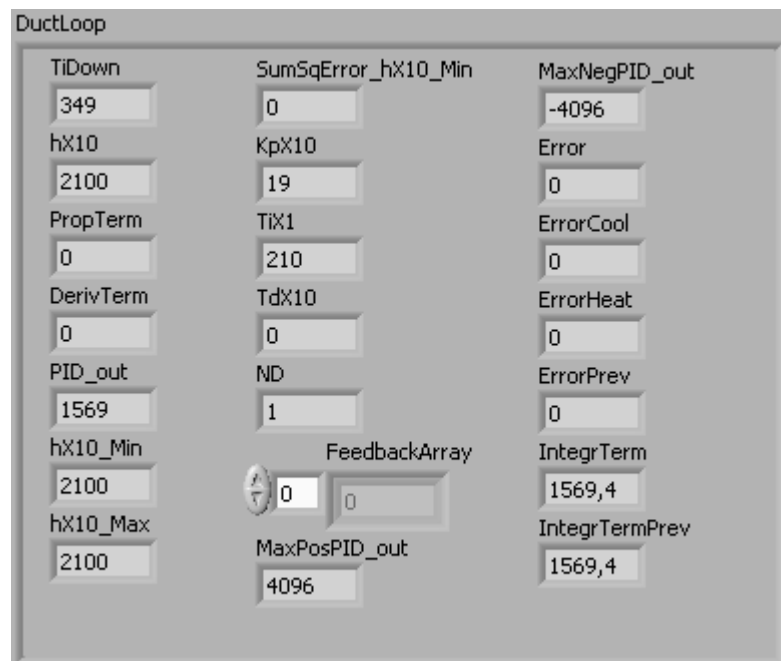


Figura 50 – Estructura DuctLoop



- HeatVars i HeatPowerCommand:

Són dos clusters que estan a la part inferior esquerra.

HeatVars representa els temps d'activació de cada banda de motors. La part superior es refereix a la banda de 6 kW i al part inferior a la banda de 12 kW. Surten expressats els temps d'activació de cada banda (TonDsec), el temps de no activació de cada banda (TiOnDsec), el temps total de cycle (TiOffDsec) i d'altres components amb menys importància, com per exemple TonNorm, que és una representació del temps total d'activació de les dues bandes respecte del total del cycle.

HeatPowerCommand és un cluster que mostra els moments d'activació de cada motor. El switch superior es refereix a la banda del motor de 6 kW i el switch inferior a la banda del motor de 12 kW.

Aquests dos interruptors es van activant a mesura que la sortida del PID requereix més o menys potència segons l'error que hi ha entre la temperatura real i la consigna.

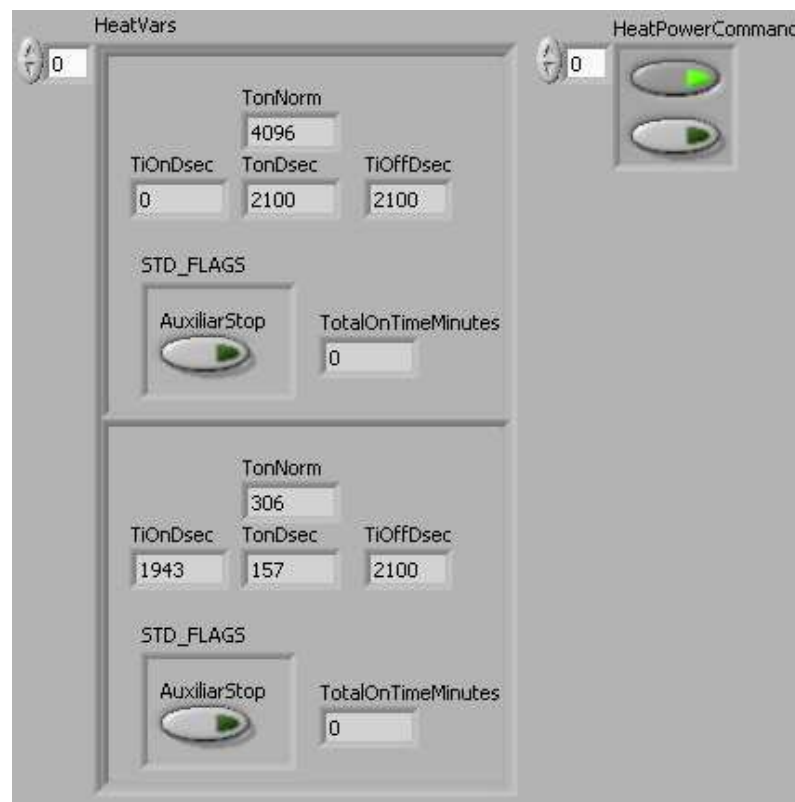


Figura 51 – Estructura HeatVars i activacions de motors

En aquest cas, la banda de 6 kW està activada i durant 157 dècimes de segon de les 2100 del cycle la banda de 12 kW estarà activada.



En el següent cas s'observa com les dues bandes estan activades.

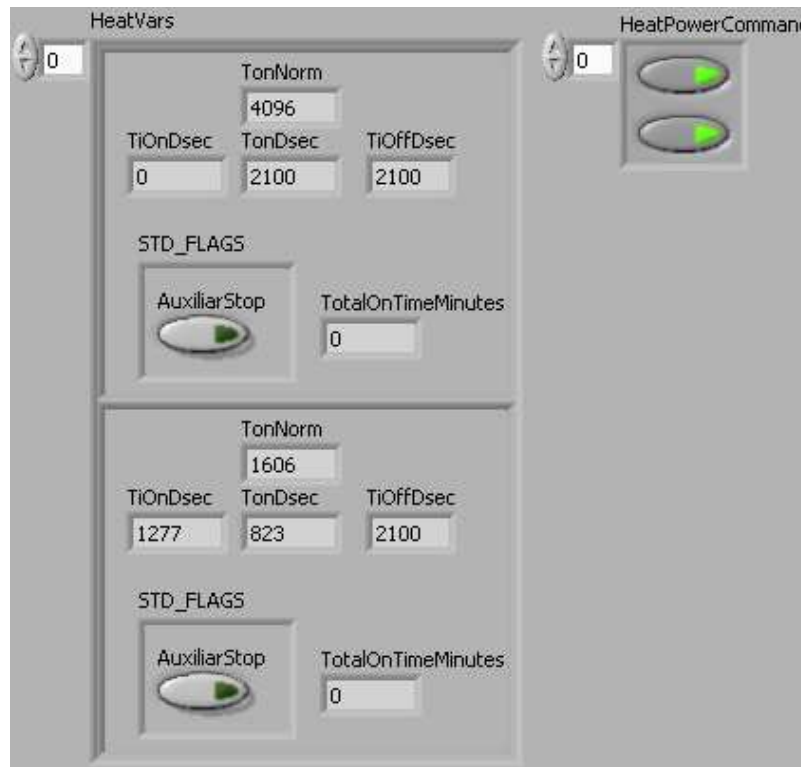


Figura 52 – Estructura HeatVars i activacions de motors

- HeatCommon:

És l'últim element que queda per descriure i està situat a la part central a la dreta. Està compost per diferents variables internes del programa. D'aquí només interessa el número total de bandes disponibles (NumWorkingStages), que en el cas de la simulació és 2, i el temps total de funcionament (TonNormCommon).

Aquesta última variable és un pèl complexa d'explicar. El temps total de cicle és de 2100 segons. A això se li fa correspondre un nombre binari, el 4096. Com que hi ha dues bandes de potència, una el doble que l'altra, aquest nombre total és 12288. Si els motors estan apagats, TonNormCommon és 0. Si estan tot el cicle a plena potència, TonNormCommon és 12288. Entre aquests dos valors hi ha tot el marge per a fer correspondre aquest valor a un cert temps d'activació de cada banda. Si el valor és superior a 4096 llavors la primera banda està sempre activa i la segona s'activa durant una certa quantitat de temps.

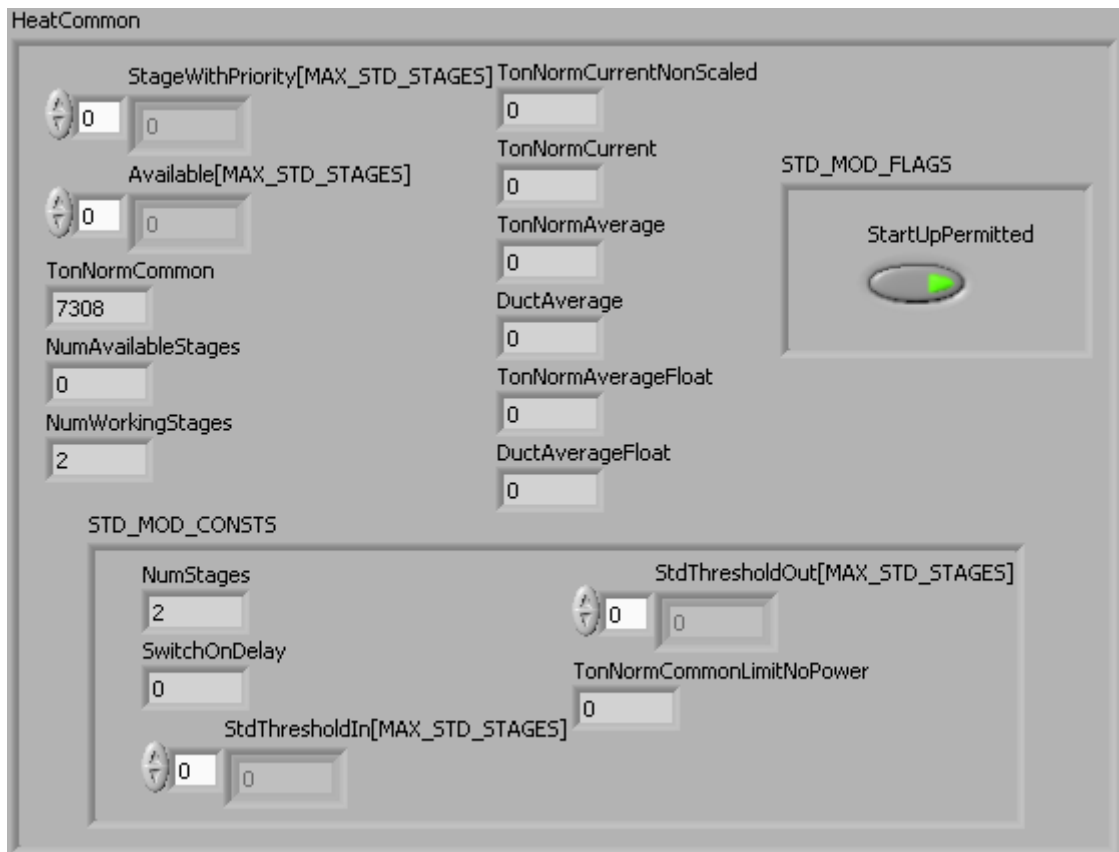


Figura 53 – Estructura HeatCommon

Característiques generals i especificacions de la simulació:

El programa és ben senzill d'utilitzar. Un cop es fa doble Click en l'arxiu executable el programa ja comença en mode RUN. A partir d'aquí es selecciona la temperatura de conducte que es desitja.

Els sistema identifica quin és l'error entre la temperatura actual i la de consigna i el PID actua. La seva senyal de sortida és interpretada pel programa com una activació dels motors durant un cert període de temps. El programa va realitzant aquestes operacions cada inici de cicle fins a assolir la temperatura de consigna.

Si un es fixa en l'evolució de la temperatura, es pot observar com aquesta disminueix sensiblement quan les bandes no estan activades o quan la temperatura és més alta i només hi ha activada una banda. Això és degut a que quan no estan activades les resistències calefactores, el motor ventilador continua girant i per tant, la temperatura disminueix.



Una altra especificació és que quan el temps d'activació de les resistències és menor a 50 dècimes de segons, aquesta activació s'anul·la i no es produeix. Com es pot veure en el següent exemple, el temps d'activació de la segona banda és 44 i no s'activa la banda de 12 kW en cap moment.



Figura 54 – Estructura HeatVars i activacions de motors

Es pot comprovar com el temps d'activació és 44 i el temps de no activació és de 2100. Això vol dir que el motor està sempre apagat. En canvi, s'observa com la primera banda està activada tot el cicle.

Si es deixa el sistema a màxima potència es pot comprovar com s'arriba a la temperatura màxima que s'havia vist a l'inici de tot amb la prova real. En aquell cas s'arriba als 57.1°C. En la simulació s'arriba a una temperatura una mica superior, concretament a 58.5°C.

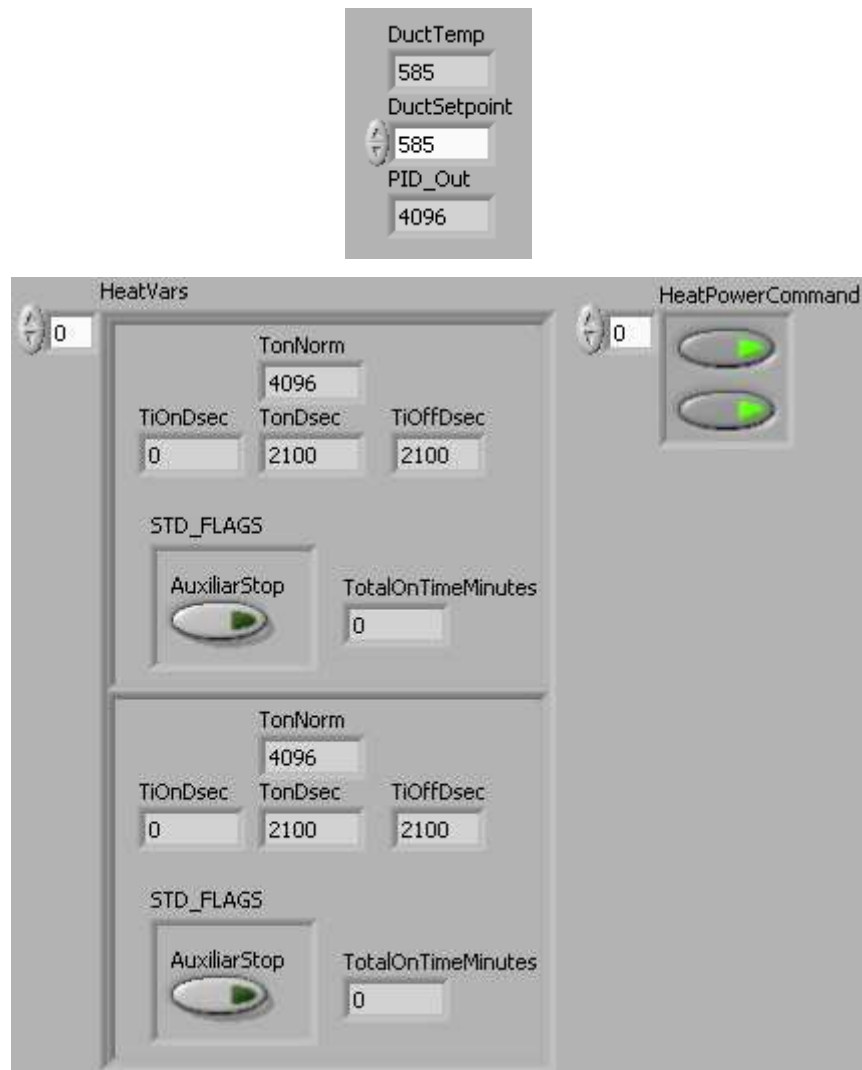


Figura 55 – Estructura HeatVars i activacions de motors i temperatura real, de consigna i sortida del PID

Les dues bandes estan activades al màxim i la temperatura del conducte és de 58.5°C. La sortida del PID és la màxima permesa pel programa.

Aquesta és una altra demostració de que la simulació és acurada i precisa.

Per a tenir una concepció més clara de l'estructura del programa a continuació es presenta l'esquema de la jerarquia de subprogrames dels que està compost.



Simulació del conducte:

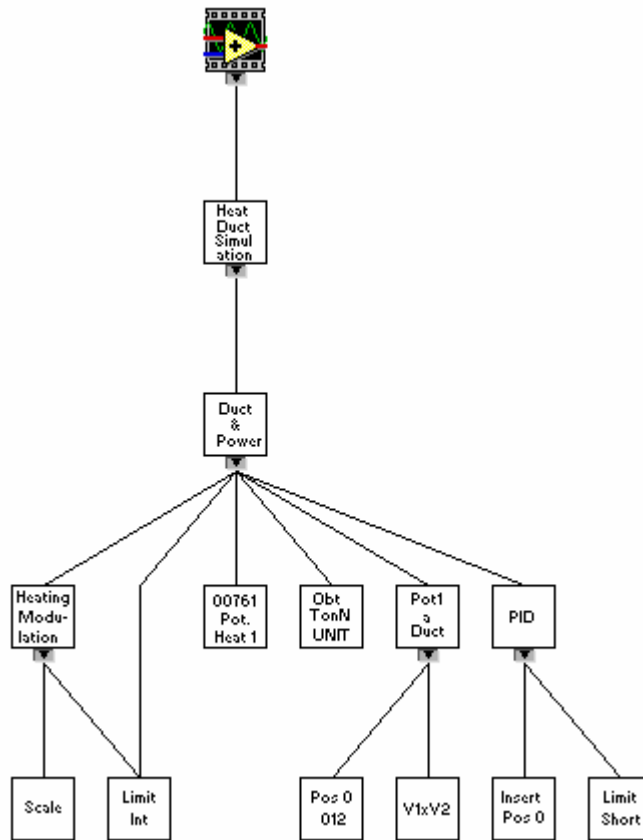


Figura 56 – Jerarquia del programa de simulació

- 00761_SimulacionConductoHeat
 - o 00761_Duct&Power
 - PID
 - LimitShort
 - InsertarPos0_Array
 - ObtNewTonNorm
 - LimitInt
 - 00761_HeatingModulation_UNIT
 - Scale
 - LimitInt
 - 00761_PotenciaUnitariaHeat
 - ModeloConductoOrden2
 - Multiplicar2vectores
 - InsertarPos0_Array012



Exemple de simulació:

Per a obtenir una temperatura de conducte desitjada només s'ha d'introduir el valor de consigna i el programa anirà progressant fins assolir aquesta temperatura.

En qualsevol instant d'execució, un es pot fixar en els temps assignats a l'activació de cada motor. Aquests temps provenen de la potència unitària assignada al motor que dona la sortida del control.

Així doncs en un moment determinat pot ser que la sortida sol·liciti una activació del motor de 6 kW de tot el temps de cicle (2100 dècimes de segon) i una activació de 1050 dècimes de segon (la meitat del període). Per tant, la senyal d'activació per a cada motor serà.

Motor de 6 kW:

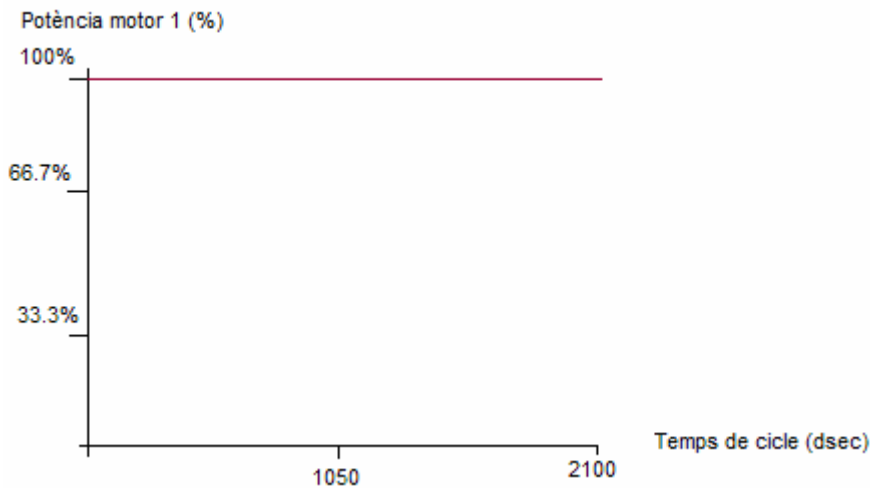


Figura 57 – Cicle del motor de 6 kW



Motor de 12 kW:

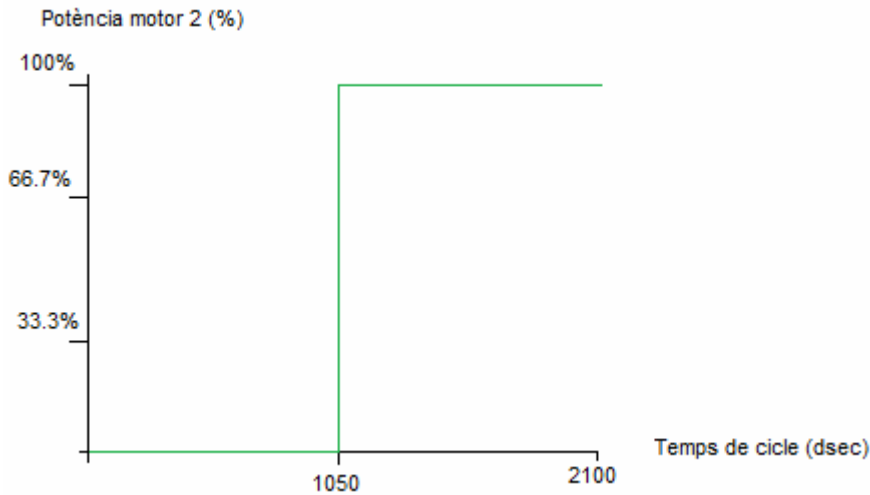


Figura 58 – Cicle del motor de 12 kW

I la combinació dels dos donarà com a resultat la senyal final.

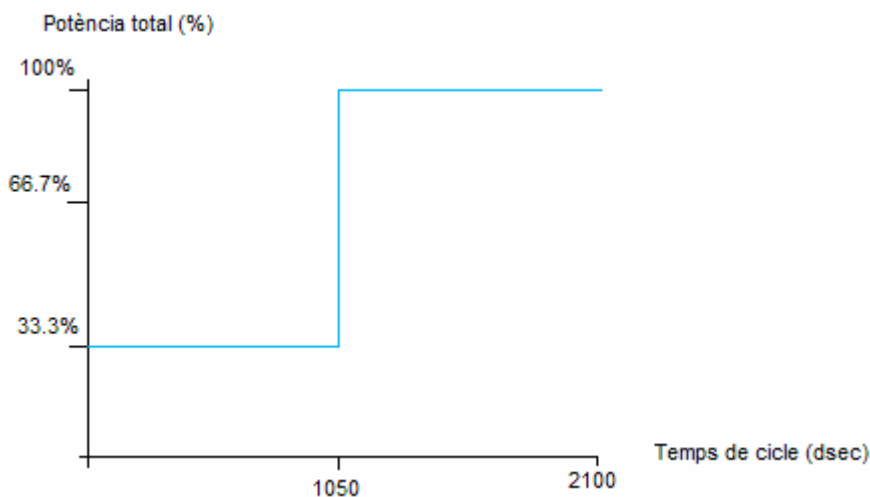


Figura 59 – Cicle dels dos motors

Si s'imagina que aquesta combinació genera una temperatura de conducte que és la de consigna, simplement s'aniria repetint aquesta seqüència indefinidament.

De fet, aquesta senyal prové de la potència unitària. Aquesta es destria entre les dues potències dels motors i se li assigna un temps. Si es calculés la mitjana del gràfic superior es veuria com aquest nombre (entre 0 i 1) correspon a la potència unitària designada pel control.



Resultat de la simulació:

Per a veure els resultats de la simulació és necessari veure l'evolució de la temperatura en tot moment, ja que no s'ha pogut dissenyar un gràfic per a que mostrés el progrés de la temperatura del conducte real i la temperatura de consigna.

Per això anteriorment s'ha explicat el tipus de senyal que es genera en un determinat cicle.

Si s'obris el programa i s'introduís una temperatura de consigna per exemple de 300 (30°C), el primer cicle el motor encara no estaria encès i per tant, la temperatura inicial (25.5°C) començaria a baixar, ja que la temperatura exterior representa que és de 0°C. A la que passessin els 210 segons es llegiria la temperatura actual (que pot haver baixat perfectament 15°C) i el programa generaria una senyal de sortida per activar els dos motors.

A partir d'aquí la temperatura aniria pujant i els motors s'anirien activant segons l'error entre la temperatura de consigna i la temperatura real del conducte. A mesura que passen els cicles, l'error es va fent més petit, ja que els motors arriben progressivament a l'estat on generen una potència suficient com per tenir la temperatura de consigna desitjada a l'inici de cada cicle.

Com que els motors es van encenent i apagant (ja que el software així ho determina) és possible que quan un motor estigui apagat la temperatura de consigna disminueixi una mica i que durant el moment que estiguin activats els dos motors la temperatura pugi per sobre de la temperatura de consigna. Aquesta variació pot ser de 2 o 3 graus i no influiran gaire en el que realment es busca, la bona regulació de la temperatura de la sala.



Referit a la refrigeració, aquest programa no la pot simular, ja que només simula calefacció. Es necessitaria l'equació en diferències de la refrigeració real, de la qual l'empresa no disposa encara les dades.

Simplement funcionaria de la mateixa manera que la calefacció amb l'única diferència de que la potència unitària de sortida seria negativa. Al ser negativa vol dir que s'està refrigerant i per tant, la temperatura del conducte disminuiria.

Així doncs, l'evolució de la temperatura seria que cada cicle té una fase de pujada de temperatura. En algun moment comença a baixar la temperatura perquè els motors estan apagats o només hi ha activat el motor de 6 kW i no dóna prou potència com per escalfar més el conducte. La temperatura a l'inici de cada cicle aniria pujant fins que s'aconseguís la temperatura de consigna. Un cop arribat a aquest punt, la temperatura aniria oscil·lant entre 2 i 3°C al voltant de la temperatura de consigna.

16. Presentació de l'autòmat

ALTE té un proveïdor d'autòmats anomenat DICOEL. Aquesta empresa subministra l'autòmat que es fa servir per al control de l'aire condicionat a través del software dissenyat a l'empresa. És un autòmat dissenyat especialment per l'ús ferroviari i està homologat i disposa de les aprovacions de la normativa ferroviària.

L'autòmat que s'utilitza és l'anomenat THEC06. A l'annex es disposa d'un manual complet amb tota la informació sobre aquest element, però a continuació s'explicaran algunes característiques bàsiques d'aquest autòmat.

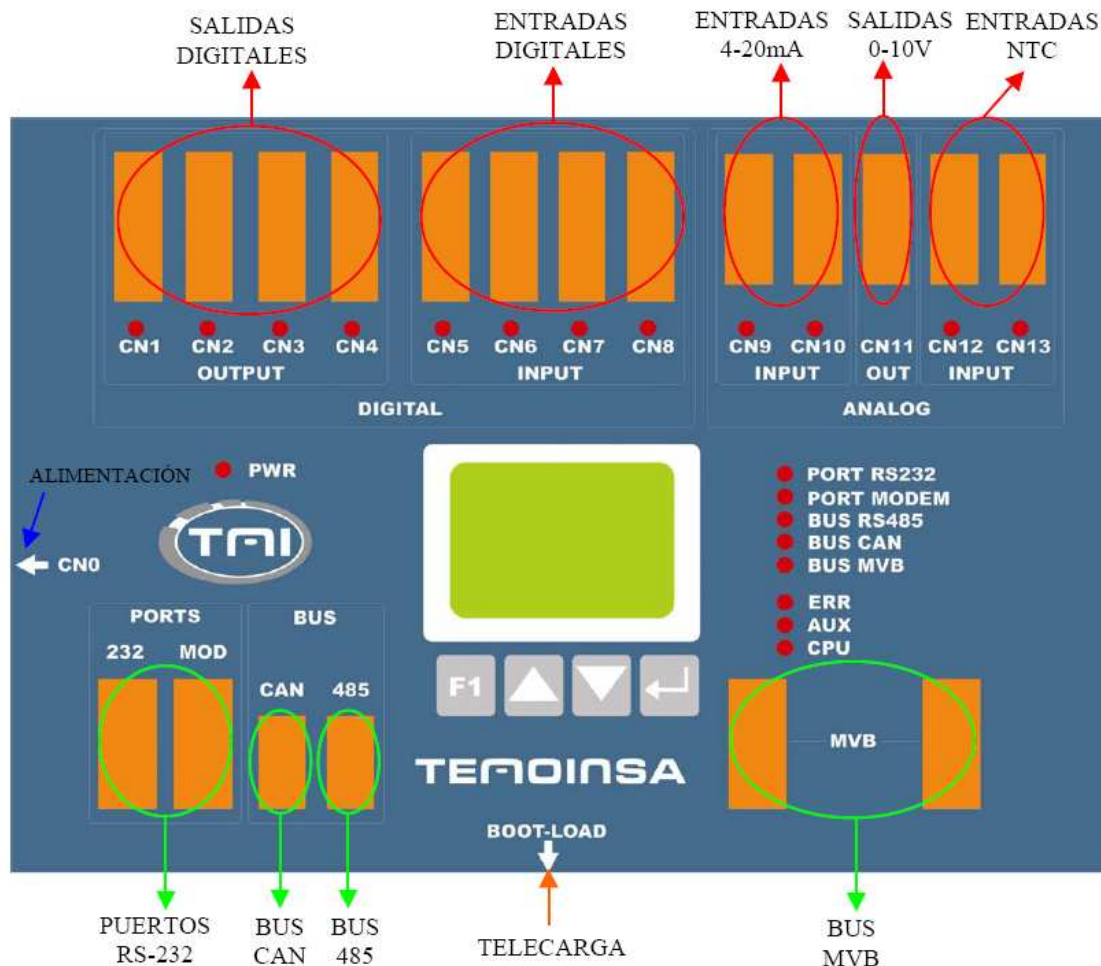


Figura 60 – Vista frontal de l'autòmat THEC06, panell de comandament i connectors

Aquest és l'aspecte general del panell de comandament de l'autòmat i a més a més es mostren els elements característics que s'explicaran a continuació.



El THEC06 disposa de:

- 4 entrades digitals de 8 bits cadascuna (CN5-CN8).
- 4 sortides digitals de 8 bits cadascuna (CN1- CN4).
- 4 entrades analògiques de 8 bits separades en dos blocs de dos connectors. L'un està adaptat per a sensors de temperatura NTC (CN12-CN13) i l'altre està adaptat per a sensors de pressió de 4-20mA (CN9-CN10).
- 1 sortida analògica de 0-10V.
- El connector d'alimentació a 24V CN0.
- 2 ports de connexió RS-232
- 1 BUS CAN
- 1 BUS 485
- 2 BUS MVB
- Una entrada per a la càrrega del software.
- 1 pantalla de control
- 4 tecles per a manejar la pantalla. Es pot dissenyar un software per a controlar les accions i combinacions de les tecles.
- Un LED per a cada un dels components explicats (menys la pantalla i les tecles) que indiquen el bon funcionament de cada connector i port.



17. Test HVAC

Nota: Tota la creació del software que es veurà a continuació ha estat creada per l'autor del treball. No s'explica amb detall com s'ha realitzat tot el procés per a crear el programa i que es vegin totes les icones, títols, leds, pestanyes... ja que no és el més important i transcendent. Són moltes hores que s'han dedicat per a poder arribar a mostrar un software clar, entenedor, senzill i explicatiu. L'interessant d'aquest treball és saber com funciona el software dissenyat i quines utilitats i avantatges té per a l'empresa.

Un cop realitzada la simulació l'empresa va considerar oportuna la inclusió d'un programa de Test de l'aire condicionat (HVAC) per a l'autòmat TEHC 06. Té el nom de "HVAC TestAndSimulation V10".

Aquest programa té l'objectiu de poder realitzar tasques de manteniment i prova de funcionament de sensors i actuadors des de l'ordinador, generar esdeveniments de diferents possibles funcionaments de l'HVAC com pot ser per exemple l'estat de calefacció, l'estat de repòs o d'altres i finalment provar-los de dues maneres diferents. Establint connexió amb l'autòmat i la màquina o fent servir el programa de simulació en comtes de la màquina. A continuació s'explica pas a pas els diferents modes de funcionament.



Inici de l'execució:

El programa executable que es pot trobar en el CD d'annexos comença amb la següent pantalla.

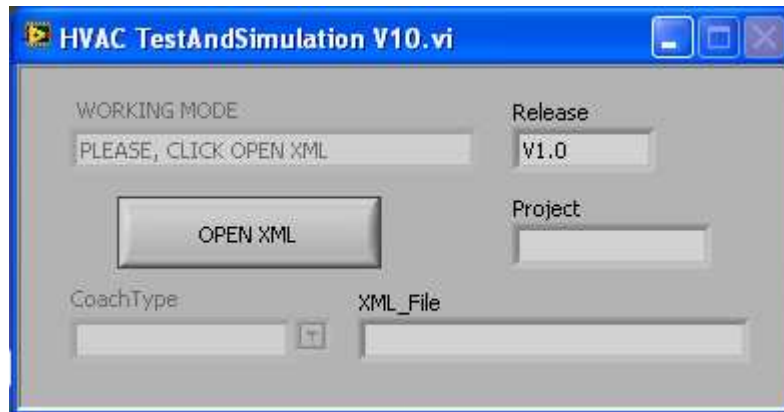


Figura 61 – Finestra d'inici

Es pot observar com d'inici només es permet la opció d'obrir un XML. Així doncs, fem click en el polsador i ens surt el següent menú per seleccionar i obrir un arxiu XML.

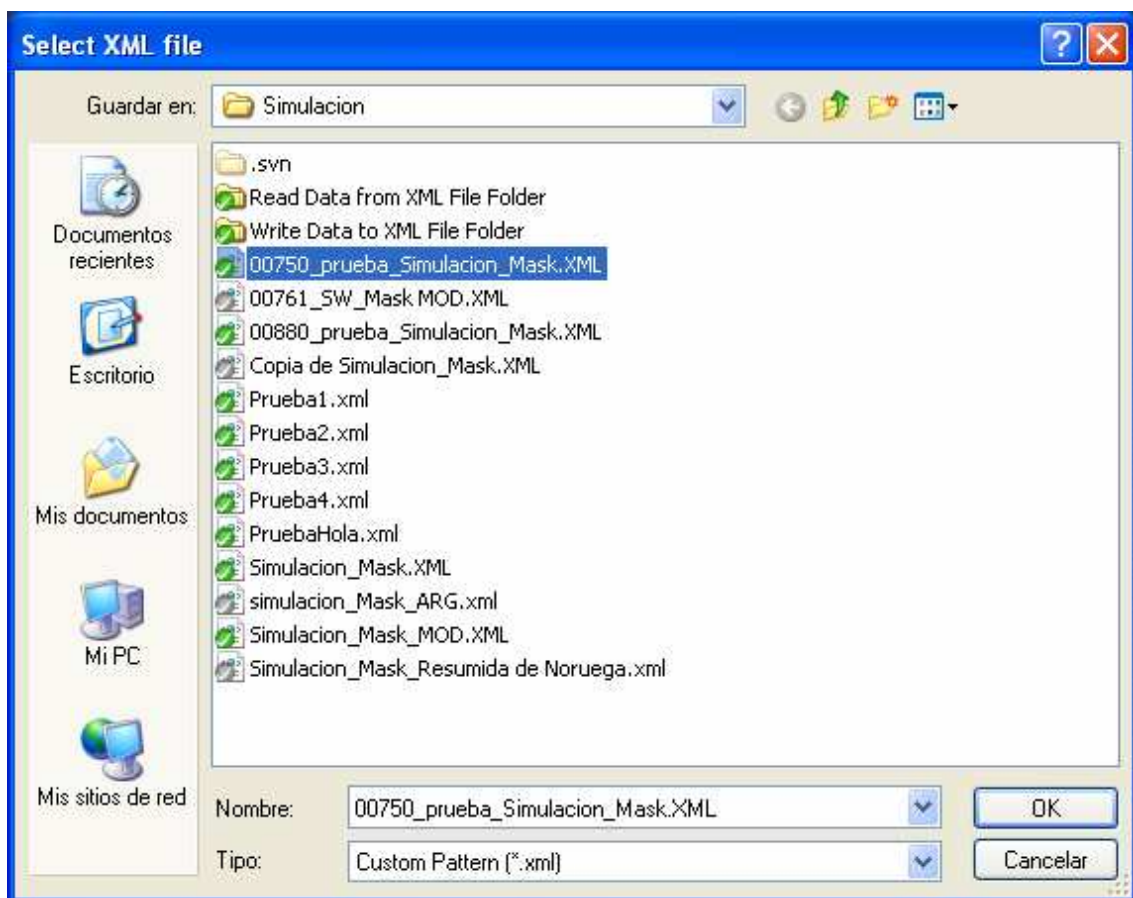


Figura 62 – Finestra per obrir un arxiu XML



S'ha de seleccionar un arxiu XML creat expressament per a cada obra. Les obres en aquesta empresa es seleccionen per números. Així doncs, en aquest carpeta hi ha tres obres diferents: la 00750, la 00761 i la 00880.

Aquests arxius XML contenen els noms de totes les variables que es carregaran el programa de TEST: Noms de connectors (CN), noms d'alarmes, codis, entrades i sortides digitals i analògiques, etc. Per a dissenyar-los només s'ha hagut de seguir un ordre estricte de preferències amb etiquetes i dades per a cada una d'elles. Per exemple, si per a cada grup de connectors hi ha vuit dades, l'etiqueta principal conté el nom del connector, per exemple, CN6 i dins d'aquest, hi ha el nom de cada dada. El programa té uns subVI's que destrien aquesta informació i la situen en les caselles que els hi correspon.

Seguim endavant seleccionant la obra 00750 i ens torna a aparèixer el menú principal de nou.

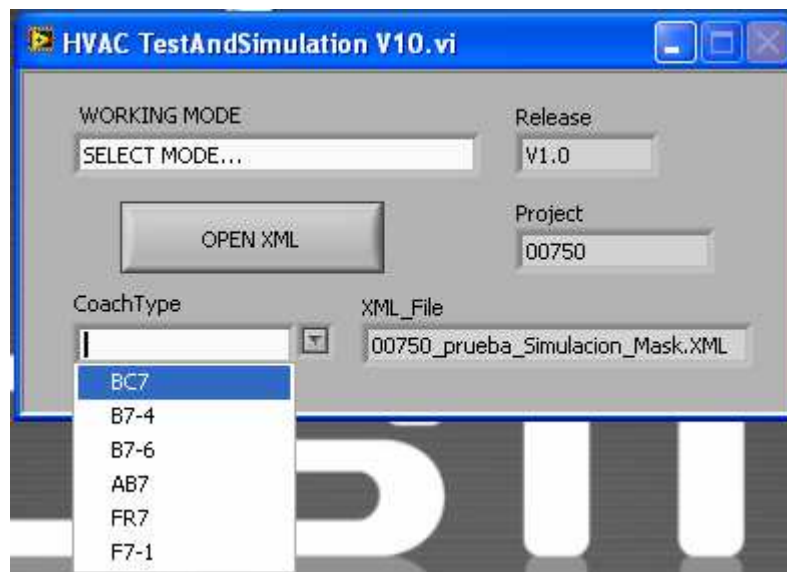


Figura 63 – Selecció de vagó

Al llegir l'arxiu s'omplen alguns espais com per exemple el nom de l'arxiu que s'ha seleccionat i el número d'obra. A més a més s'activen dues pestanyes noves. La de sota a l'esquerra ens permet seleccionar el tipus de vagó amb el que volem treballar.

Cada obra disposa de vagons diferents ja que hi poden haver vagons restaurant, vagons de passatgers de diferents classes, vagons amb zones lúdiques, cabines i d'altres i a cadascun se li assigna un nom. Així doncs, els elements que conformen cada vagó



poden variar una mica i per tant, els noms de les dades que omplirien el TEST poden ser diferents. Es selecciona el tipus de vagó BC7, que correspon a un vagó de passatgers. Finalment queda seleccionar el tipus de mode de treball.

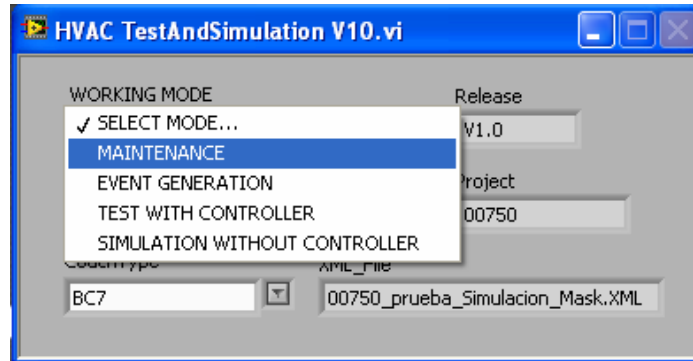


Figura 64 – Selecció de mode de funcionament

Es pot escollir entre manteniment, generació d'esdeveniments, realitzar un test amb l'autòmat o realitzar un test sense ell, és a dir, amb el programa de simulació que s'ha dissenyat. A continuació s'explicarà cadascun d'aquests modes de funcionament.

Manteniment:

Un cop seleccionada la opció MAINTENANCE apareix la finestra del menú principal.

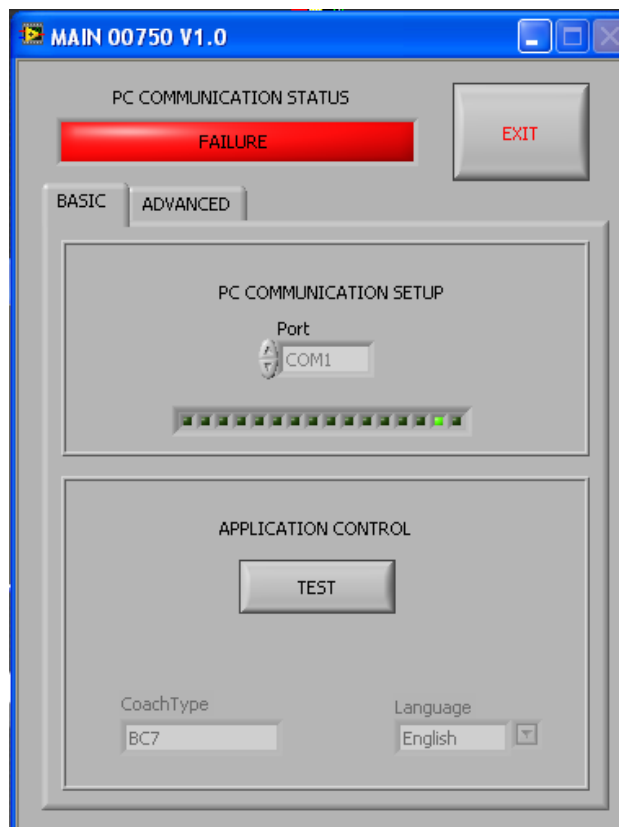


Figura 65 – Menú principal



Aquest menú té a la part superior l'estat de comunicació amb l'autòmat. Si és de color vermell vol dir que no hi ha connexió o existeix un error de connexió. En el cas de que estigués verd significaria que s'ha establert comunicació. Just al costat hi ha el botó de sortida de l'aplicació. En altres programes semblants existeix un altre botó que permet accedir a un PDF d'ajuda. És el cas de l'aplicació que instal·la el programa en C a l'autòmat.

Hi ha una altra pestanya a part de la que es mostra que permet accedir a opcions avançades, però són útils per a accions internes de l'empresa.

També es pot veure a la part superior el tipus de vagó que s'ha seleccionat i que no es pot modificar i l'idioma en què s'està treballant. De moment només existeix l'anglès però està previst fer els programes també en francès, italià i castellà.

Finalment hi ha el botó de TEST, que permet accedir a la pantalla on es gestionaran totes les operacions. Si la connexió amb l'autòmat és correcta i fem click s'obre la següent pantalla.

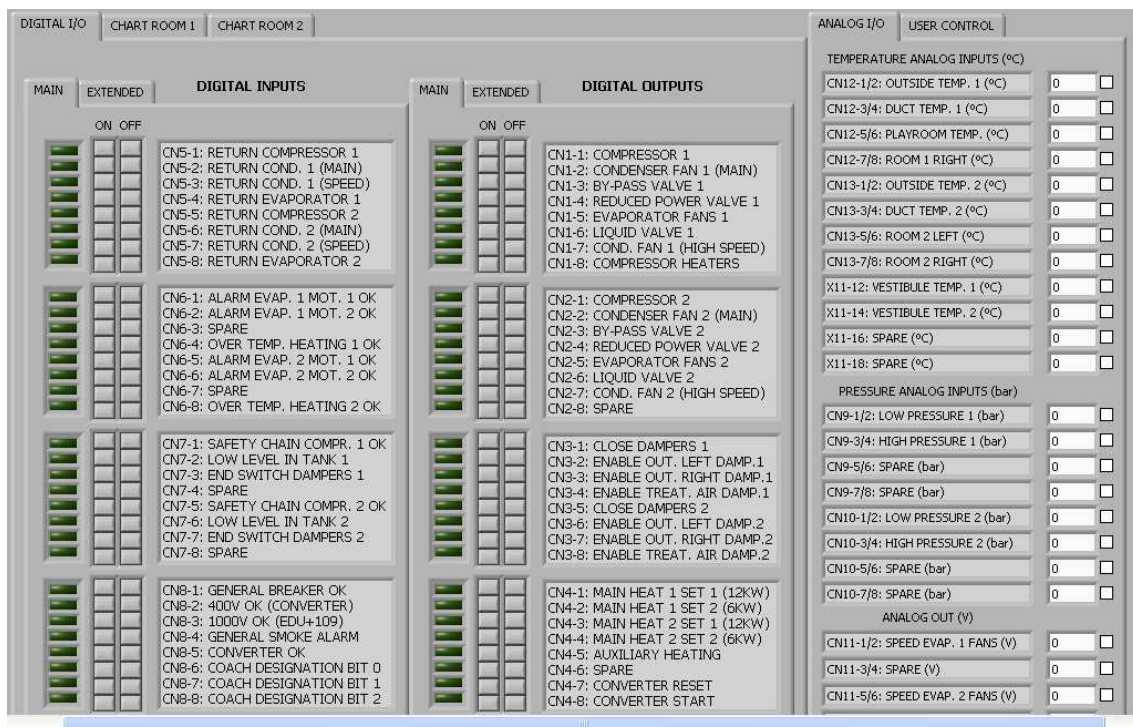


Figura 66 – Pantalla de test del mode de manteniment



Tots els textos que apareixen han estat extrets de l'arxiu XML que s'ha obert anteriorment. La funció del manteniment és activar o desactivar connectors per a veure si responen correctament. Les digital inputs són les entrades que li arriben a l'autòmat, és a dir, totes les dades dels sensors i actuadors del sistema d'aire condicionat. I les digital outputs són les sortides de l'autòmat cap a l'HVAC.

Així doncs es poden modificar totes aquestes variables per a veure la reacció de les altres i què provoca.

Generació d'esdeveniments:

Es parteix des del mateix punt que en el cas anterior, la finestra MAIN. En aquest cas també s'ha de polsar el botó de TEST i apareixerà la mateixa pantalla que abans però amb una finestra de menú superior. L'única diferència fins aquí és que la generació d'esdeveniments es pot realitzar sense estar establerta la connexió ja que és un programa que genera esdeveniments i per això no es necessita connexió amb l'autòmat.

Figura 67 – Pantalla de test del mode de generació d'esdeveniments



La pestanya inferior és exactament igual que en el cas de manteniment. En canvi apareix una pestanya superior que permetrà gestionar els esdeveniments creats.

Per a crear un esdeveniment s'han de prémer els polsadors de ON o OFF de les entrades digitals. Això significaria que a l'autòmat se li estan entrant dades com si procedissin de la màquina d'HVAC.

Per exemple, en la imatge anterior s'ha creat l'estat de calefacció inicial. Quan aquest esdeveniment es faci servir connectat a l'autòmat, aquest donarà respostes a través de la seva sortida digital. A més a més es poden modificar (forçant els seus valors) els valors de les entrades analògiques com són el control de les temperatures i pressions i es poden veure els resultats de les sortides, com són els voltatges dels evaporadors. Si no es forcen els valors l'autòmat dona les dades reals que correspondrien a l'esdeveniment generat.

Un cop s'han seleccionat tots els polsadors per a generar l'esdeveniment s'han d'acabar d'omplir una sèrie de pestanyes. Primer de tot se li ha d'assignar un temps de durada. A més a més, se li donarà un número per establir l'ordre d'execució i se li donarà un nom. A part, es pot afegir una petita explicació si es desitja. Quan es dugui a terme l'execució d'un esdeveniment apareixerà a més a més l'adreça on s'ha guardat aquest.

El mateix succeeix amb els estats. Són processos semblants als esdeveniments amb l'única diferència que en els modes de TEST que vindran a continuació explicats no s'utilitzen. Simplement són indicadors.

Les opcions que es poden dur a terme amb un esdeveniment són les de guardar-lo o eliminar-lo. Aquests quedaran ordenats segons l'índex que se'ls hi doni i s'executaran en aquest ordre. Llavors amb tot el conjunt d'esdeveniments que es creïn es pot guardar en un arxiu XML que crea directament el programa LabVIEW, guardar com un altre arxiu i carregar un d'aquests arxius.



Test amb controlador:

Igual que en els casos anteriors es continua amb la finestra de MAIN. En aquest cas es necessita de la connexió amb l'autòmat ja que aquest i el seu programa de control carregat seran els que processin les dades.

Quan s'obre la finestra de TEST apareix una altra finestra demanant que s'obri un arxiu XML. Està demanant que s'obri algun arxiu que s'ha creat anteriorment amb el generador d'esdeveniments.

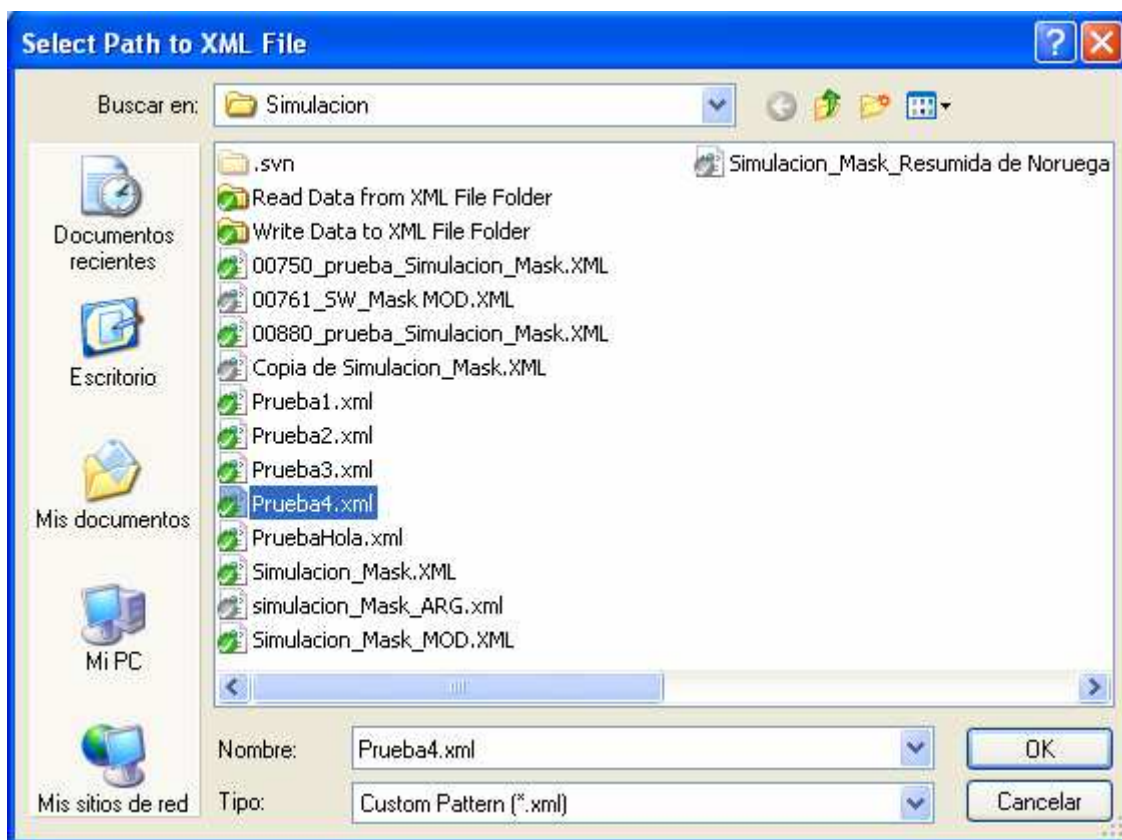


Figura 68 – Finestra per obrir XML en mode de test

En aquest cas s'ha de recordar el nom amb què s'ha guardat l'arxiu on s'han creat els esdeveniments desitjats. En el cas que es mostra a continuació és Prueba4.xml

El problema pot sorgir quan es tria un XML que no ha estat dissenyat pel generador d'esdeveniments. El programa contempla aquesta opció, dóna error i es torna a mostrar la finestra de MAIN.



Si s'ha seleccionat un XML correcte es passa a la següent finestra.

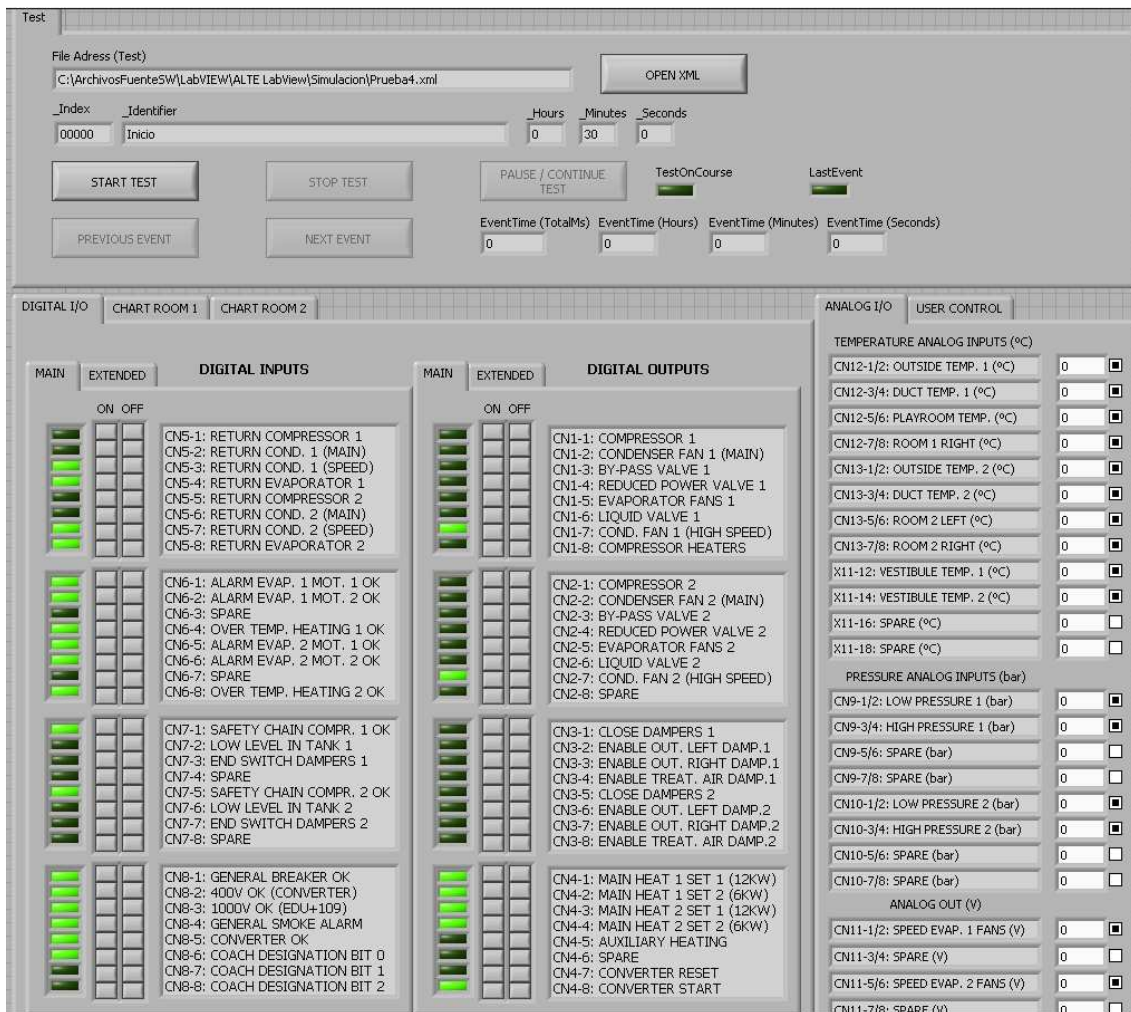


Figura 69 – Pantalla de test del mode de test

La primera diferència que s'observa és que en lloc de la finestra d'esdeveniments ha aparegut la finestra de gestió de test.

Aquesta finestra està composta per les dades bàsiques del XML obert com són la seva adreça, l'índex, l'identificador i el temps total de l'esdeveniment actual. A més a més es mostra el temps total que porta transcorregut el test, a la part inferior esquerra. Només quan el test està aturat es pot fer click en el botó d'obrir XML. Es desplegaria un altre cop la finestra anterior per poder seleccionar un altre arxiu XML generat amb el gestor d'esdeveniments. Dos leds mostren si el test està realitzant el seu últim esdeveniment i si està en marxa. Si aquest led està encès permanentment està en marxa, si fa pampallugues està en pausa i si no està activat vol dir que el test no està en marxa.



Finalment hi ha els quatre polsadors de control d'esdeveniments. Inicialment només està disponible el botó de START. Un cop polsat el test comença a funcionar. S'activen les entrades corresponents al primer esdeveniment i l'autòmat, segons el funcionament del programa, va generant diferents sortides. En el cas de la imatge anterior, el mode de calefacció genera senyals d'activació dels ventiladors i les resistències (MAIN HEAT 1 SET 1 (12KW), per exemple) i l'activació d'altres senyals com els condensadors d'alta velocitat. De fet, una característica dels vagons és que aquests condensadors han d'estar sempre activats.

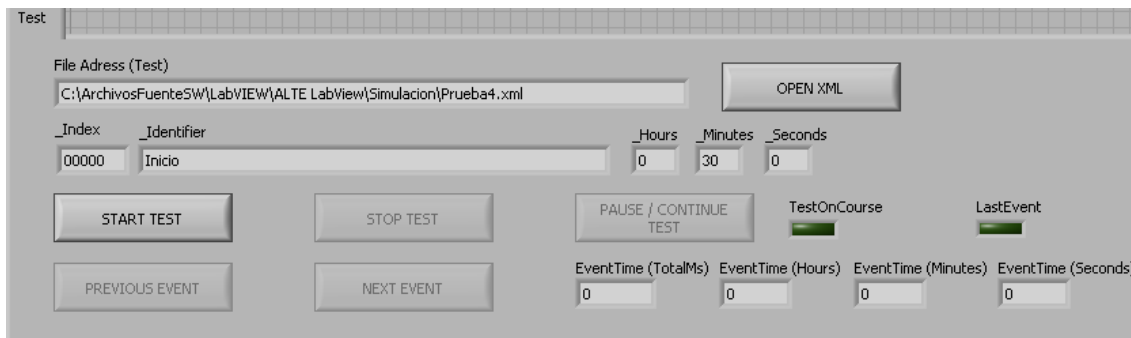


Figura 70 – Pestanya de gestió del test

A més a més estan disponibles els polsadors de següent esdeveniment i parada de test i el polsador d'inici de test es desactiva.. Si es fa click en el polsador de següent esdeveniment automàticament es permetrà accedir a l'esdeveniment anterior. Cada cop que es faci un canvi d'aquests, el temps total es reinicia. Cada canvi d'esdeveniment comporta el canvi de l'índex, l'identificador i el temps total en hores minuts i segons de l'esdeveniment. Finalment quan s'arriba a l'últim esdeveniment el botó d'esdeveniment següent es desactiva i el led d'últim esdeveniment s'encén.

Els polsadors de les entrades i sortides analògiques poden estar forçats a un valor desitjat o estar no forçats i llavors el valor serà el que doni l'autòmat segons la gestió que faci de totes les dades que li entren.



Test sense controlador:

Funciona de la mateixa manera que el cas anterior, l'únic que pot córrer sense estar connectat al controlador. Com que s'utilitza el programa de simulació creat anteriorment, les respostes que es generaran provindran de les dades de sortida que doni la simulació.

Un aspecte comú entre els dos tipus de test que s'han comentat (amb i sense controlador) és que disposen d'un gràfic per a veure les evolucions respecte el temps de les temperatures exteriors, de setpoint, de conducte...

També disposen d'una finestra, a la dreta, en una pestanya al costat de les entrades i sortides analògiques que conté altres dades interessants com poden ser el nombre i tipus d'alarmes actives, algun conjunt de dades internes interessants per a l'anàlisi a l'empresa, el cotxe seleccionat (que es pot modificar, ja que algunes de les entrades a l'autòmat són els bits que identifiquen el vagó i si es modifiquen, es canvia de tipus de vagó) i finalment a la part inferior, una pestanya on es pot introduir una contrasenya que se li proporciona al client. Si s'introdueix la clau correcta es disposa de llibertat total i sense restriccions de programa per activar qualsevol connector o introduir qualsevol valor de temperatura o pressió.

Per exemple, el que es podria fer, però que no tindria cap lògica és activar alhora els compressors i els evaporadors. Aquesta és una limitació que fixa el programa, però que amb aquesta opció comentada, queda eliminada.

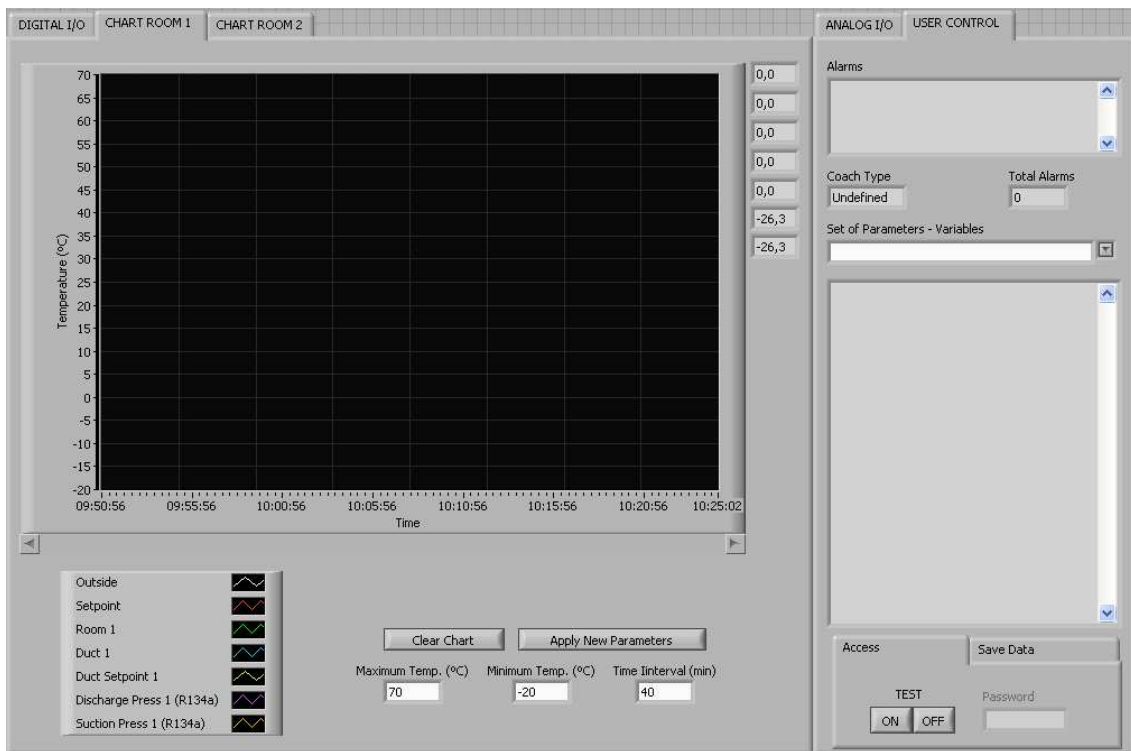


Figura 71 – Finestra de gràfic i d'alarmes

Finalment i al costat d'aquesta pestanya hi ha una altra pestanya, SAVE DATA. Si s'hi accedeix i es fa click en el botó de START, totes les operacions que es registrin durant el test quedaran guardades en un arxiu XML fins que es polsi el botó de STOP. Hi ha una barra lluminosa a sota que s'encén quan s'enregistren les dades.

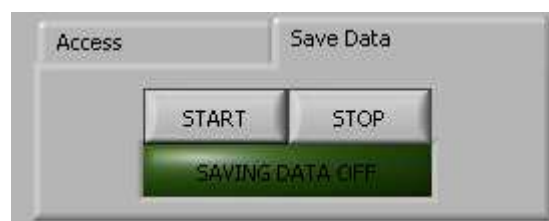


Figura 72 – Pestanya amb la opció de guardar el test en un arxiu XML



A continuació també es mostra la jerarquia del programa i els seus subprogrames.

Test HVAC (High Ventilation Air Conditioned)

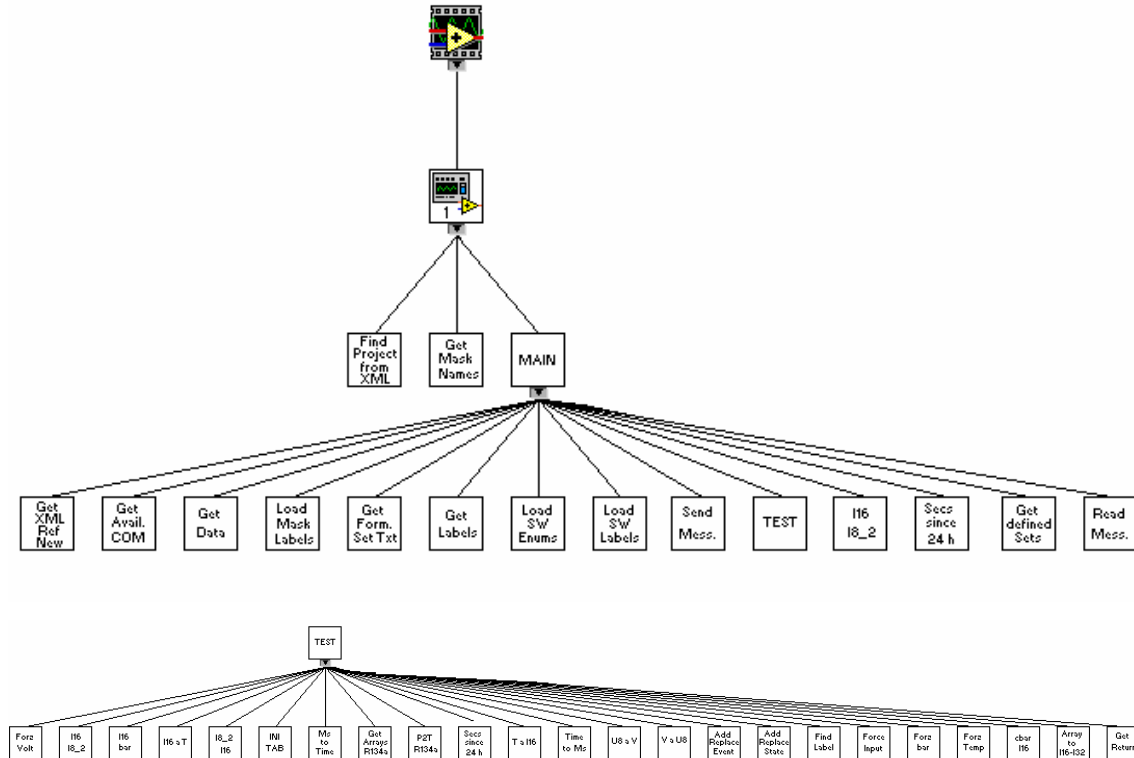


Figura 73 – Jerarquia del programa de test HVAC

- HVAC TestAndSimulation V10
 - o FindProjectFromXML
 - o GetMaskNames
 - o MAIN
 - TEST

Els subVI's MAIN i TEST tenen multitud de subprogrames interns que no trascendeix la seva identitat, ja que són programes de gestió de vectors, càrrega de dades i d'altres menys importants. Realment són aquests dos subVI's juntament amb el VI principal (HVAC TestAndSimulation V10) els programes que s'han dissenyat i que es poden obrir i treballar amb ells.



18. Bootloader

El Bootloader és el programa que s'utilitza per a carregar el programa de control en C dins de l'autòmat.

El procés de càrrega consta en obrir els arxius compilats en C, llegir-los i programar-los a l'autòmat.

Hi ha diverses opcions com per exemple triar el port de comunicació, encara que en aquesta empresa es fa servir un adaptador USB per a connectar-lo amb el port RS-232. Les altres opcions són més específiques per al desenvolupament i ús intern o sigui que bàsicament és així com es carrega l'autòmat amb el programa de control.

La finestra de History mostra l'evolució de tot el procés.

L'aspecte del programa és el següent.



Figura 74 – Finestra principal del BootLoader



19. Arxius d'ajuda

Una forma molt interessant de prestar ajuda en qualsevol moment d'una operació amb un programa de software l'empresa és incloure l'enllaç a un arxiu d'ajuda des del menú principal del programa.

Es crea un PDF amb totes les explicacions necessàries per a poder manejar el programa en qüestió. Amb una operació amb LabVIEW es pot crear un botó que permet obrir un arxiu situat en un fitxer. Aquest arxiu ha de ser inamovible ja que l'adreça està fixada en el programa.

A més a més es pot aprofitar que pot existir la selecció d'idioma perquè s'obri el PDF amb l'idioma corresponent, si és que existeix.

En els programes que s'han mostrat no hi ha aquesta opció, excepte el de BootLoader. En aquest programa existeix un botó anomenat PDF DESCRIPTION que et permet accedir a l'ajuda per a utilitzar aquest programa. Aquest PDF està inclòs en els annexos del treball.

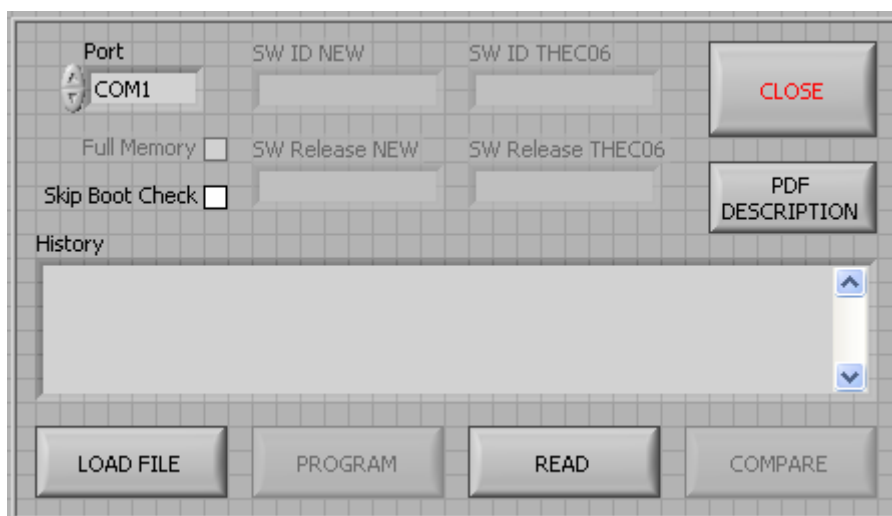


Figura 75 – Finestra del BootLoader amb la opció d'obrir el PDF d'ajuda



També hi ha un altre programa on existeix aquesta ajuda. És un programa com el de TEST però sense tota la gestió d'esdeveniments. Només hi ha interacció amb l'autòmat. El PDF està inclòs en el menú principal MAIN. És per a un client italià i inclou un botó (DESCRIZIONE PDF) per a obrir un arxiu d'ajuda que s'ha escrit a l'empresa i que lògicament estarà en italià.



Figura 76 – Finestra de test d'una altra obra amb la opció d'obrir també un PDF d'ajuda

Es pot observar com pràcticament té la mateixa forma i contingut que l'arxiu de MAIN que s'ha mostrat anteriorment.



20. Millores

Els programes que es desenvolupen sempre estan sotmesos a possibles millores. Els programes que s'han realitzat per aquest treball són tot just la primera versió, V1.0.

A mesura que es vagin introduint aquests programes en el software per als clients, aniran arribant sol·licituds o reclamacions a l'empresa de modificacions de software per a realitzar alguna operació que no s'havia comentat en els contractes.

Així doncs, per exemple, els clients podrien demanar a l'empresa que la generació dels esdeveniments i la seva activació en un test es puguin fer des de la mateixa pantalla.

Una altra millora podria ser que el programa de simulació inclogui la generació d'un gràfic d'evolució de temperatures i per exemple, que es puguin guardar les dades de temperatura en un arxiu XML.

També seria molt interessant poder generar el gràfic de la senyal PWM que es genera per a regular la potència del sistema.

En definitiva, és un treball que té molt marge de millora ja que ha estat la primera vegada que l'empresa ha decidit fer un projecte d'aquest tipus i tot just s'ha desenvolupat la primera versió. Un cop s'hagi provat en diferents clients i s'obtingui la seva valoració es podrà seguir amb una millora continua cap a un software eficaç, econòmic, útil i pràctic.



21. Avantatges per a l'empresa

L'objectiu d'aquest projecte és estalviar els costos que suposen fer proves de funcionament en condicions reals. Per això es realitza la simulació i el control d'esdeveniments. A part d'això, existeixen altres proves que si que es poden realitzar a l'empresa, però només es realitzen en les condicions ambientals que es donin en aquell moment determinat i no es poden variar.

Com que aquestes proves s'haurien de fer en una instal·lació especialitzada, on es sotmetria el vagó a diferents condicions de temperatura, tant interiors com exteriors, diferents condicions de pressió, humitat... Aquesta és una operació necessària per a afegir un grau de garantia de qualitat, però resulta molt car, ja que per començar, el lloc on es fan aquestes proves és a Milà, Itàlia.

La solució a aquesta gran despesa ha estat simular aquest procés. Els avantatges econòmics es poden separar en avantatges logístics, temporals i de personal i d'altres.

Els avantatges logístics en aquest cas són ben clars. El fet de transportar tot el material per a realitzar una prova en un altre país i tornar-lo a la seu de l'empresa és un cos que no existirà.

Els avantatges temporals són una mica més complicats de veure. El fet de realitzar aquesta simulació i tot el programa de test ha suposat setmanes i mesos de feina. Segur que anar a fer aquesta simulació no hauria costat tant, però el gran avantatge és que un cop creat el programa, qualsevol altre projecte que es realitzi només haurà d'invertir molt poca quantitat de temps ja que només s'haurà de trobar l'equació de la simulació i realitzar només alguns petits canvis que sol·liciti el client. És doncs clar que un cop fet aquest programa el temps que es gastaria fent les proves a Itàlia seria molt més gran que realitzar les simulacions i proves amb l'autòmat.

Els avantatges de personal també són evidents. Mentre que per a realitzar els programes que es mostren només s'ha destinat el becari i autor del projecte, amb l'ajuda puntual de



l'enginyer responsable del software i control, per a realitzar les proves a Itàlia s'haurien hagut d'enviar per a cada prova a dues persones representants de l'empresa amb el corresponent cost del viatge i allotjament.

D'altres avantatges són per exemple, la retribució econòmica per entregar al client un software d'ajuda per al manteniment i prova del funcionament dels equips. Augmentar la qualitat del producte entregat al client amb un potent software.



22. Pressupostos i estudi econòmic

Per a realitzar aquest treball final de carrera no s'ha hagut de comprar material. Degut a que ha estat un treball realitzat a l'empresa ALTE, tot el material ha estat subministrat per aquesta.

S'ha necessitat d'un ordinador amb el programa LabVIEW 8.2 instal·lat, a part de tot el software bàsic tipus Office. S'ha hagut de disposar d'un autòmat THEC06 per a realitzar les proves de funcionament amb el Test HVAC i tots els cables i adaptadors necessaris per a realitzar la connexió amb el PC.

Així doncs es pot dir que el **pressupost** per a fer aquest treball ha estat 0.

Si s'hagués de realitzar aquest treball sense l'ajuda de l'empresa els costos haurien estat els de tot el material anomenat anteriorment.

Per a realitzar l'**estudi econòmic** del treball per aprofitar-lo com a producte de mercat un s'ha de fixar primer de tot en si existeixen productes semblants. En principi no s'ha trobat cap empresa que realitzi específicament simulacions per a d'altres companyies, però si que existeixen algunes empreses que també realitzen simulacions dels seus processos per a benefici propi intern.

El producte que es vendria seria el de poder fer simulacions a partir d'un simple gràfic d'evolució respecte el temps del paràmetre que es vulgui simular.

Amb l'objectiu de substituir la idea de benefici propi intern aquest projecte podria desenvolupar-se per al servei a d'altres empreses, encara que seria un procés totalment innovador, sense cap experiència de mercat coneguda.



23. Conclusions

Un cop finalitzat el treball es pot arribar a la conclusió de que la simulació és un mercat nou, amb molta projecció de futur i en constant evolució.

La realització d'una simulació per al control de la temperatura del conducte d'un vagó de tren pot ser només el principi de moltes aplicacions en aquest camp que tot just es comença a utilitzar en processos industrials.

A més a més, la inclusió d'idees com crear un software de control i manteniment d'aplicacions suposa una passa endavant cap a la consagració de la qualitat d'un producte tan comú com un aire condicionat, afegint aplicacions per a la millora contínua d'un producte d'alta demanda com és aquest.



24. Bibliografia

[1] FEVE (Ferrocarriles de Vía Estrecha)

Manual de TEMOINSA

[2] **Antoni Manuel Lázaro i Joaquín del Río Fernández**

LabVIEW 7.1

Ed. THOMPSON, 2007

[3] MANUAL DE FUNCIONAMIENTO THEC06

Manual de DICOEL

[4] Tutorials de funcionament de LabVIEW www.ni.com/labview/

[5] www.wikipedia.org

[6] Diversos manuals de funcionament de l'autòmat i software de l'empresa ALTE.



ANNEXOS



Documents annexos

Es disposen de 2 CD's idèntics amb el següent material annex:

- Contingut del treball.
- Programa de simulació (carpeta Duct Simulation).
- Programa TEST HVAC (carpeta TestandSimulation)
- Manual en PDF del funcionament de l'autòmat THEC06.
- Manual de Bootloader.
- XML de la obra 00750.
- Carpetes dels codis font de la simulació, el programa de test i el codi en C.
- XML de l'arxiu d'esdeveniments (Prueba4)
- Manual de funcionament del software de control (SW de Control Unificat THEC06).
- Resum en els tres idiomes del projecte.



FI