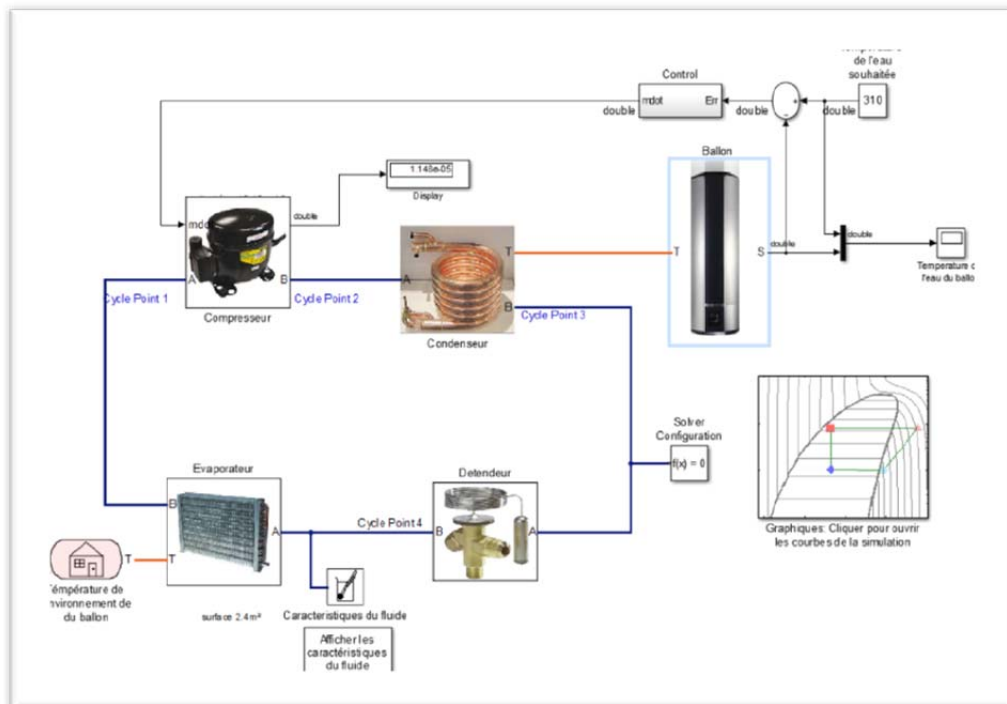
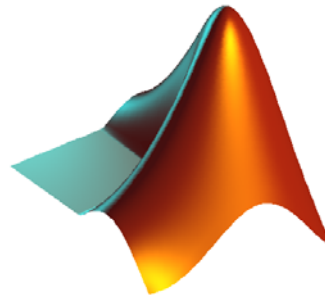


Modélisation multi physique

ESSAI DE MODELISATION DE MACHINE THERMODYNAMIQUE



Sommaire

Essai de modélisation multi-physique d'un ballon thermodynamique	2
Avant-propos.....	2
1 Modélisation.....	2
1.1 Le modèle de départ.....	2
1.2 L'adaptation pour le ballon thermodynamique	3
1.3 Les résultats	6
2 Acquisition	8
2.1 Le GUI	8
2.2 Bilan de ce dispositif de mesure :	9
2.3 Analyse de l'écart entre la mesure et la simulation	10
3 Limite du modèle :	10
Annexes :.....	11
1 Script Matlab pilotage.m	11
2 Les capteurs	14
2.1 Capteur de température : Danfoss AKS-11	14
2.2 Capteur de pression : Danfoss AKS 32R.....	15

ESSAI DE MODELISATION MULTI-PHYSIQUE

D'UN BALLON THERMODYNAMIQUE

Avant-propos

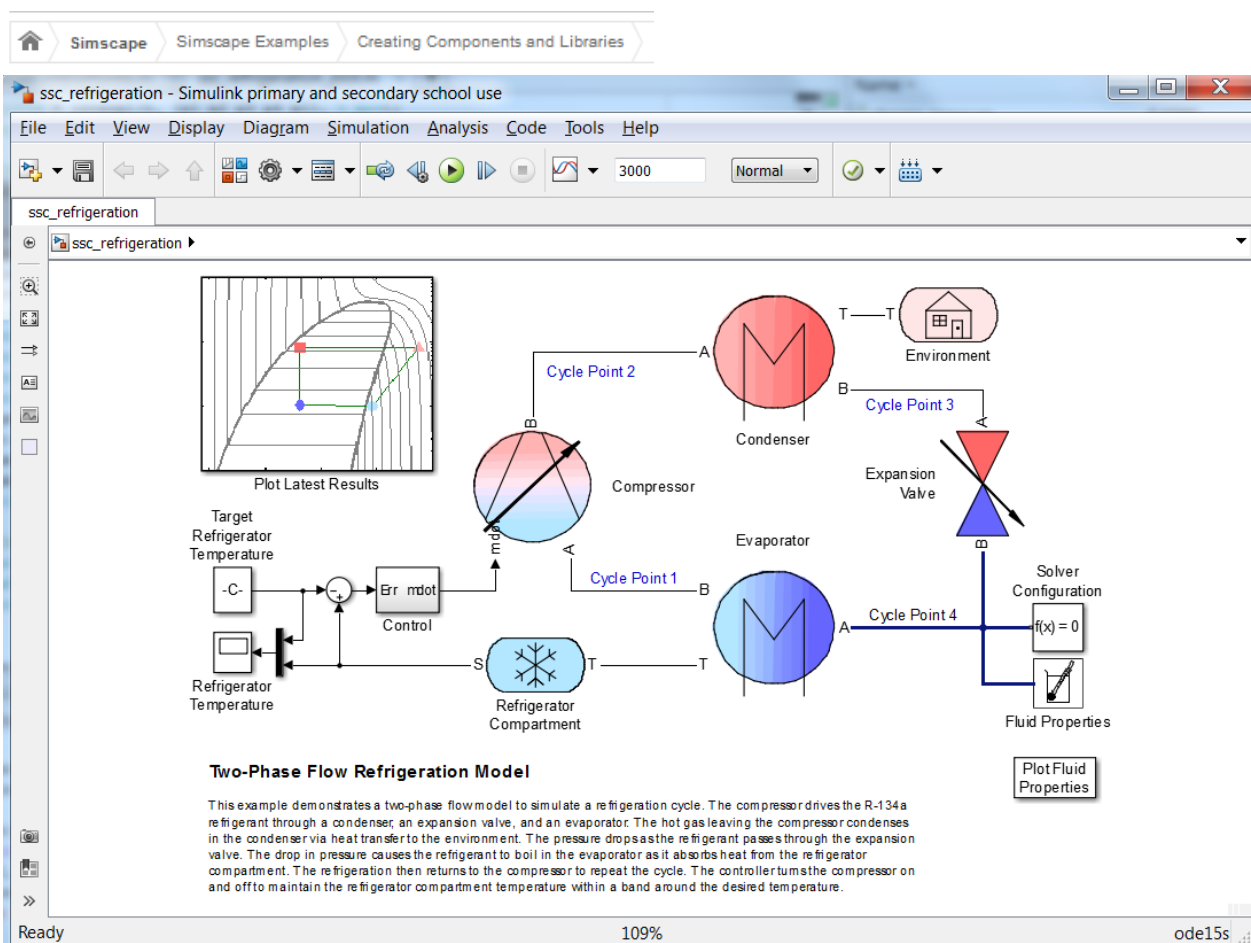
Depuis la version 2014a de Matlab, une démonstration d'utilisation des circuits thermo fluidiques est proposée dans les exemples de Simscape. L'étude d'un réfrigérateur permet d'observer le comportement du fluide dans différents points du circuit. C'est à partir de cette démo que j'ai tenté d'adapter le modèle en pompe à chaleur pour un ballon thermodynamique.

La montée en température du ballon est observable de façon fiable. Les différentes informations sur le fluide sont affichées en fin de simulation, avec un diagramme de Mollier animé.

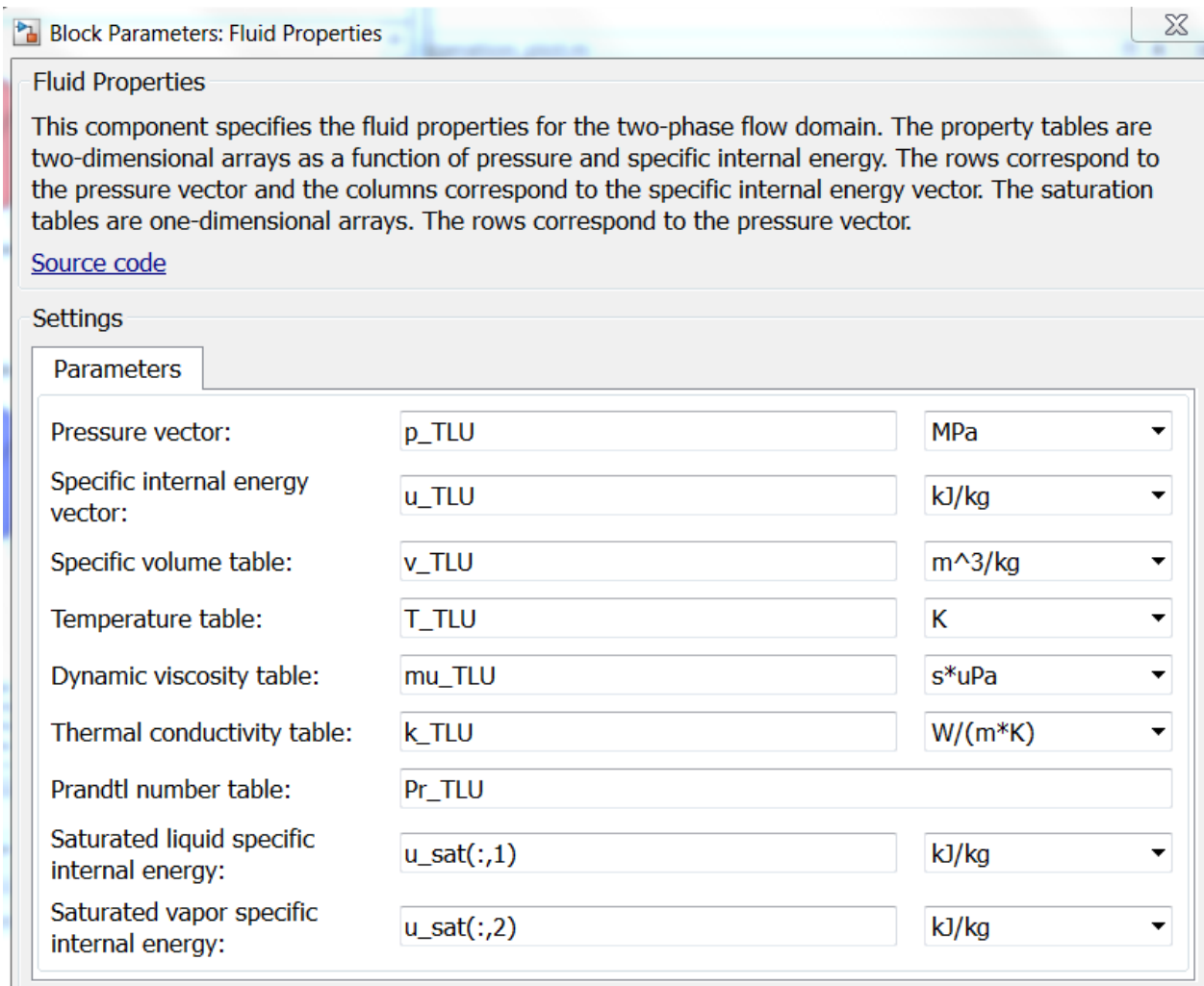
1 Modélisation

1.1 Le modèle de départ

L'étude des machines thermo fluidiques est maintenant disponible avec l'intégration des propriétés des fluides dans les circuits. Un exemple d'utilisation sur un réfrigérateur est proposé dans les exemples de Simscape.

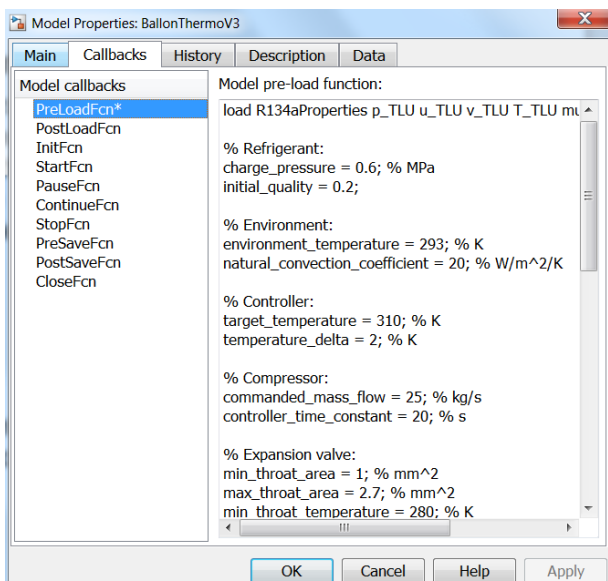


Les caractéristiques du fluide, R-134a, sont enregistrées dans un tableau de valeurs.

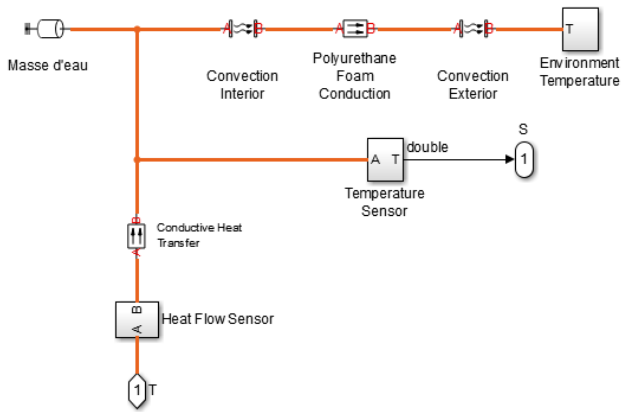


Plusieurs résultats sont enregistrés pendant la simulation pour observer l'évolution des paramètres. La montée en température du chauffe-eau est observée dans un scope.

1.2 L'adaptation pour le ballon thermodynamique

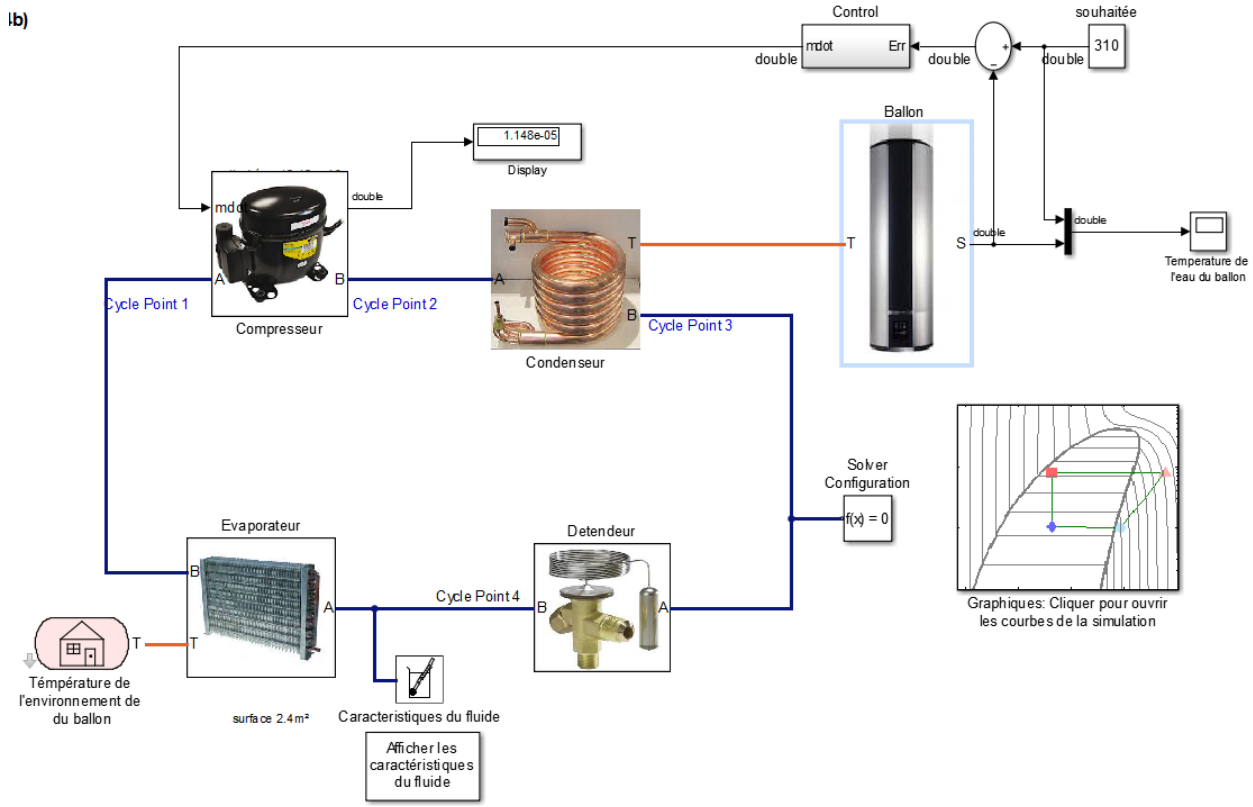


Pour notre étude du ballon thermodynamique, plusieurs adaptations du modèle exemple ont été nécessaires. De nombreux paramètres sur les caractéristiques des composants restent incertains, les surfaces des échangeurs, les caractéristiques de la vanne de détente, etc... Ces paramètres pour l'étude d'un équipement spécifique seront à rechercher et à personnaliser dans les variables du modèle. Ce réglage est à modifier dans les propriétés du modèle dans la fonction « Pré_Loadfcn ».



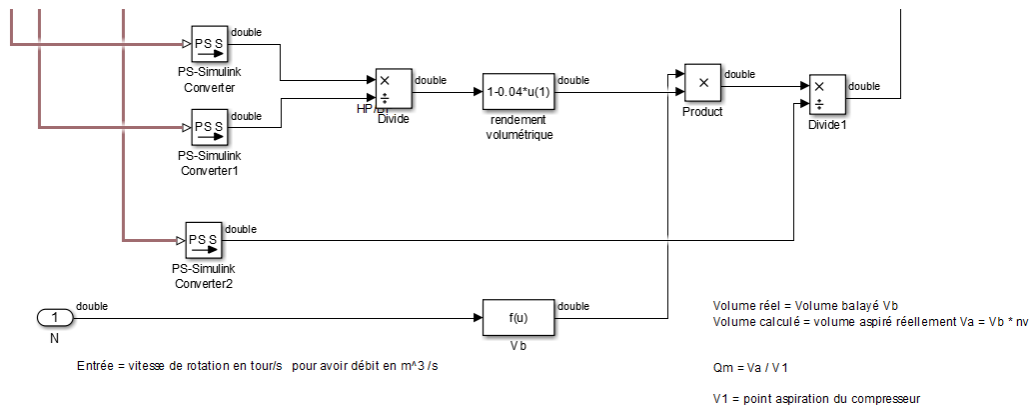
Dans cette modélisation, le système thermodynamique capte la chaleur pour l'apporter au ballon d'eau chaude sanitaire.

La masse d'eau à chauffer va également perdre des calories avec le milieu environnant.



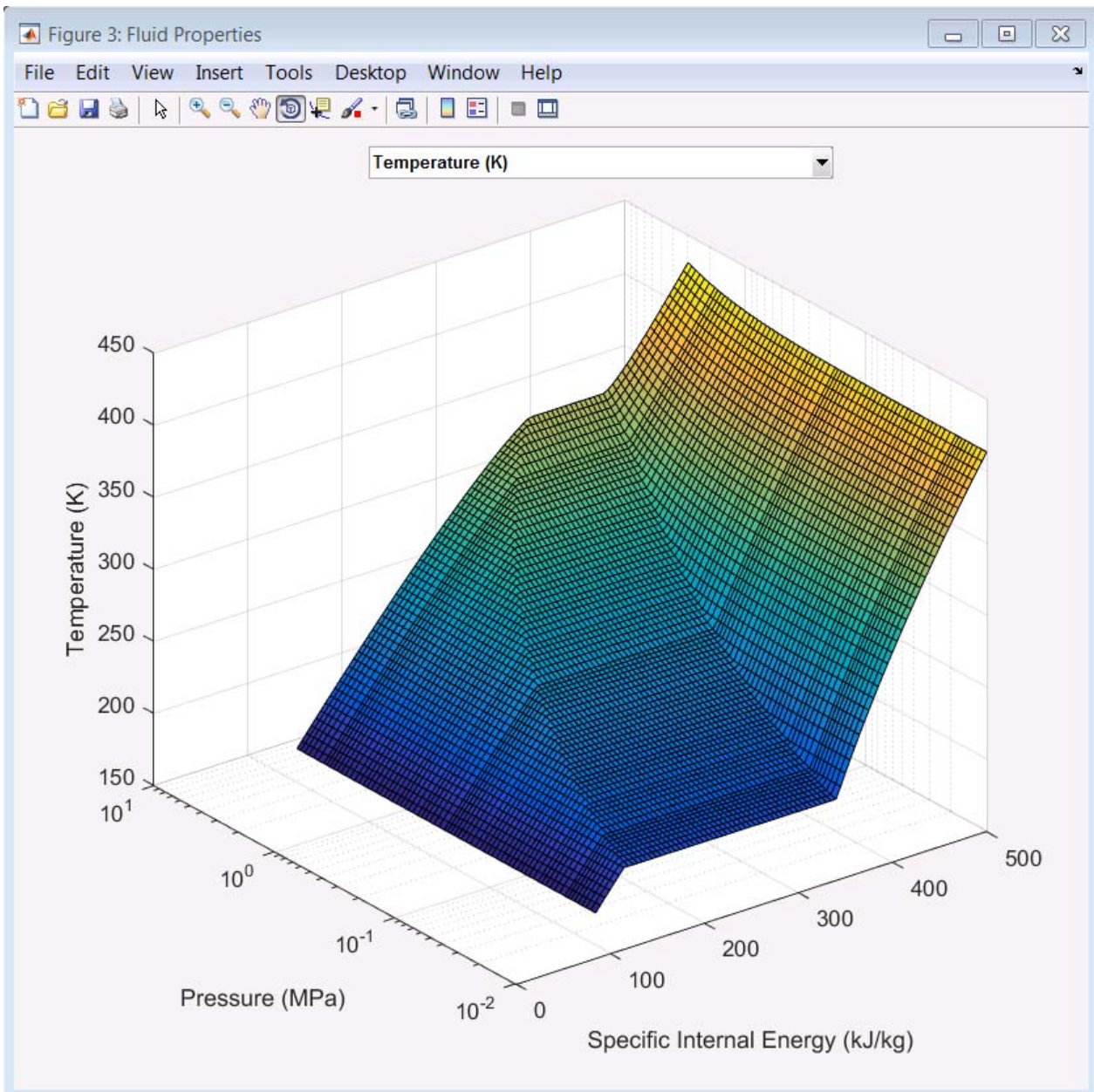
- Pilotage du Compresseur

Le pilotage du compresseur est fait en intégrant le rendement volumétrique, de façon à être proche des pratiques professionnelles du domaine des frigoristes.



Le pilotage est donc fait en vitesse de rotation du moteur du compresseur.

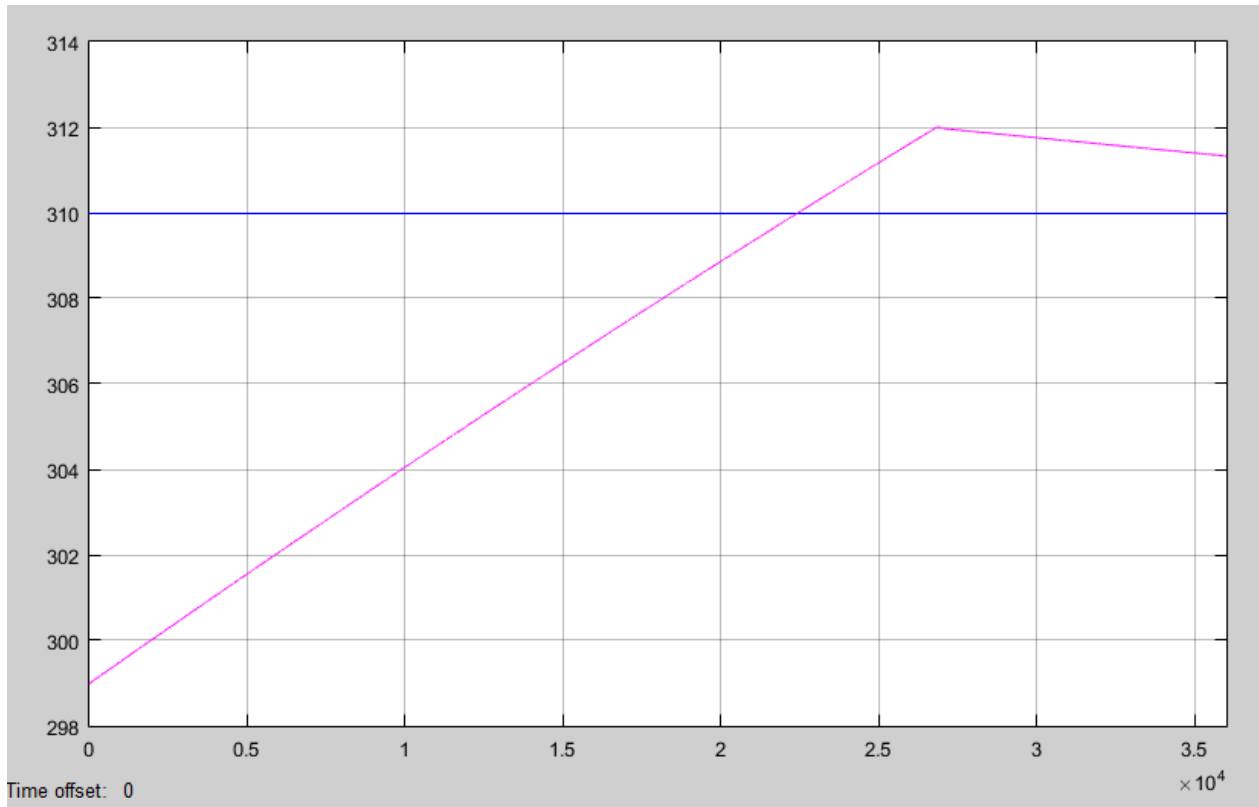
Le fluide utilisé est du R134a.



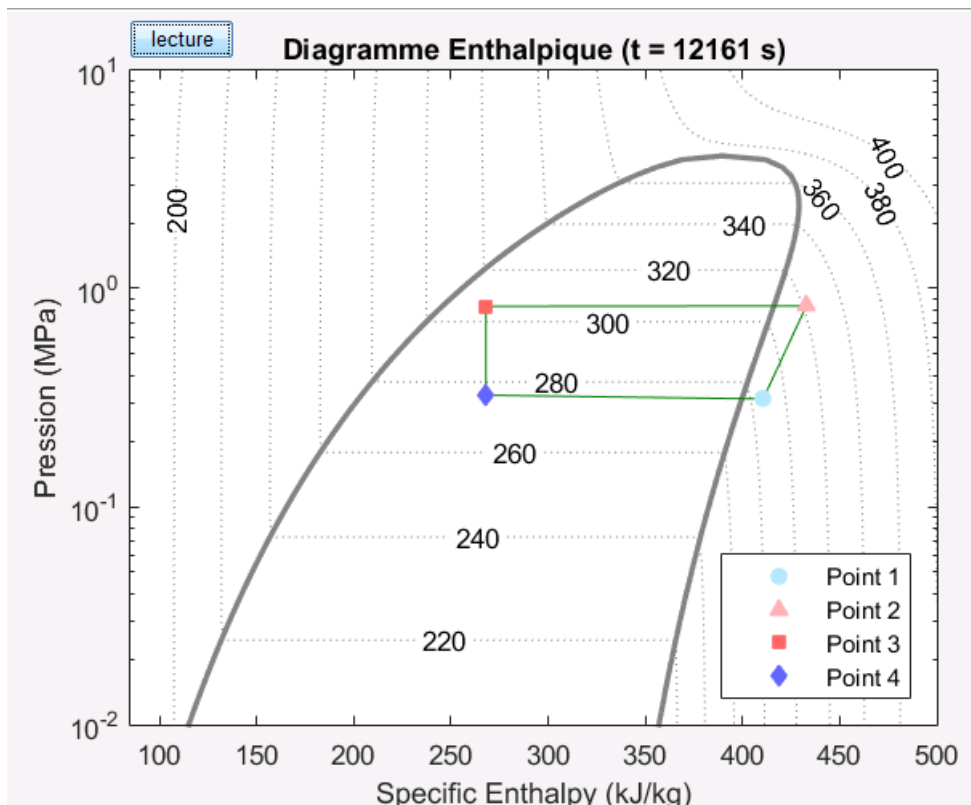
Visualisation des caractéristiques du fluide (tableau de valeurs pour le R134-a).

1.3 Les résultats

Avec les paramètres sélectionnés, nous obtenons la simulation de montée en température suivante.

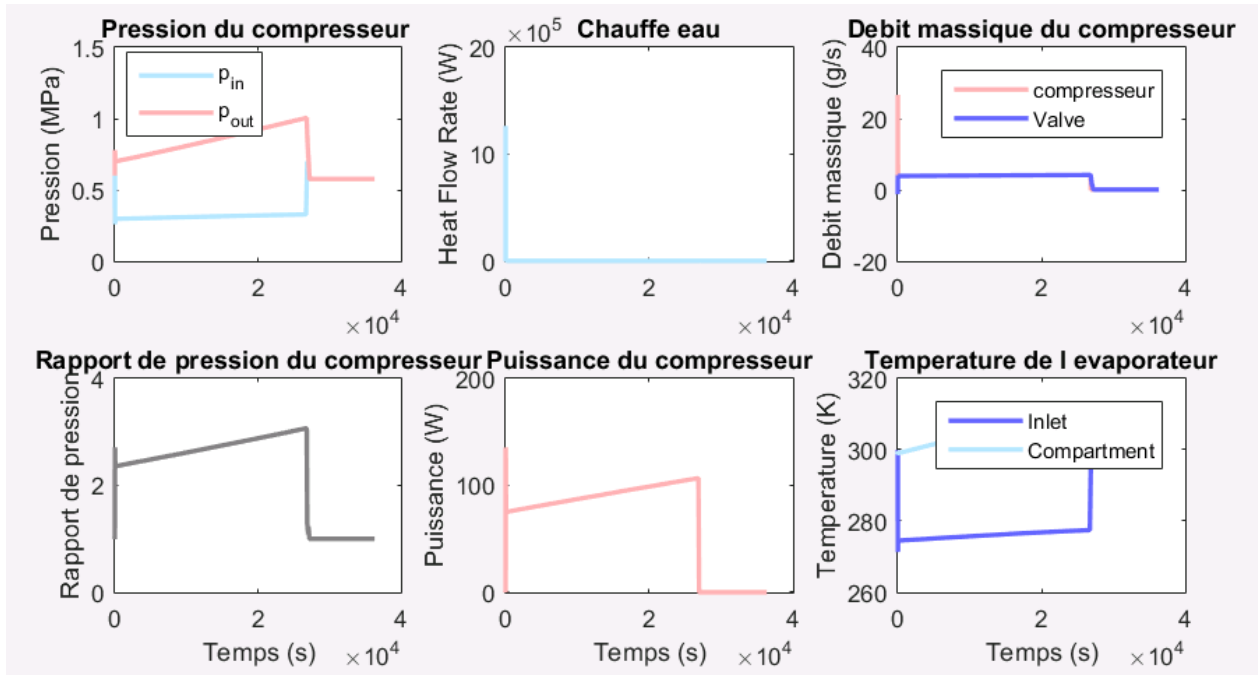


Nous pouvons également observer l'animation du diagramme de Mollier en fonction du temps. Le graphe dynamique est généré à la fin de la simulation.



Les quatre points de mesure sont issus des capteurs placés dans le modèle.

Plusieurs autres grandeurs physiques sont mises en forme pour observer les performances de la machine.

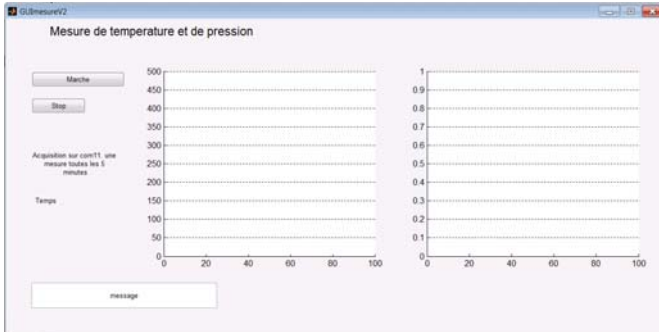


2 Acquisition

Pour la mesure des paramètres dans le circuit, nous utilisons une carte Arduino Uno connectée à l'ordinateur. Les capteurs sont pilotés par un script Matlab.

L'acquisition des températures et pressions dans le circuit est faite à intervalle régulier.

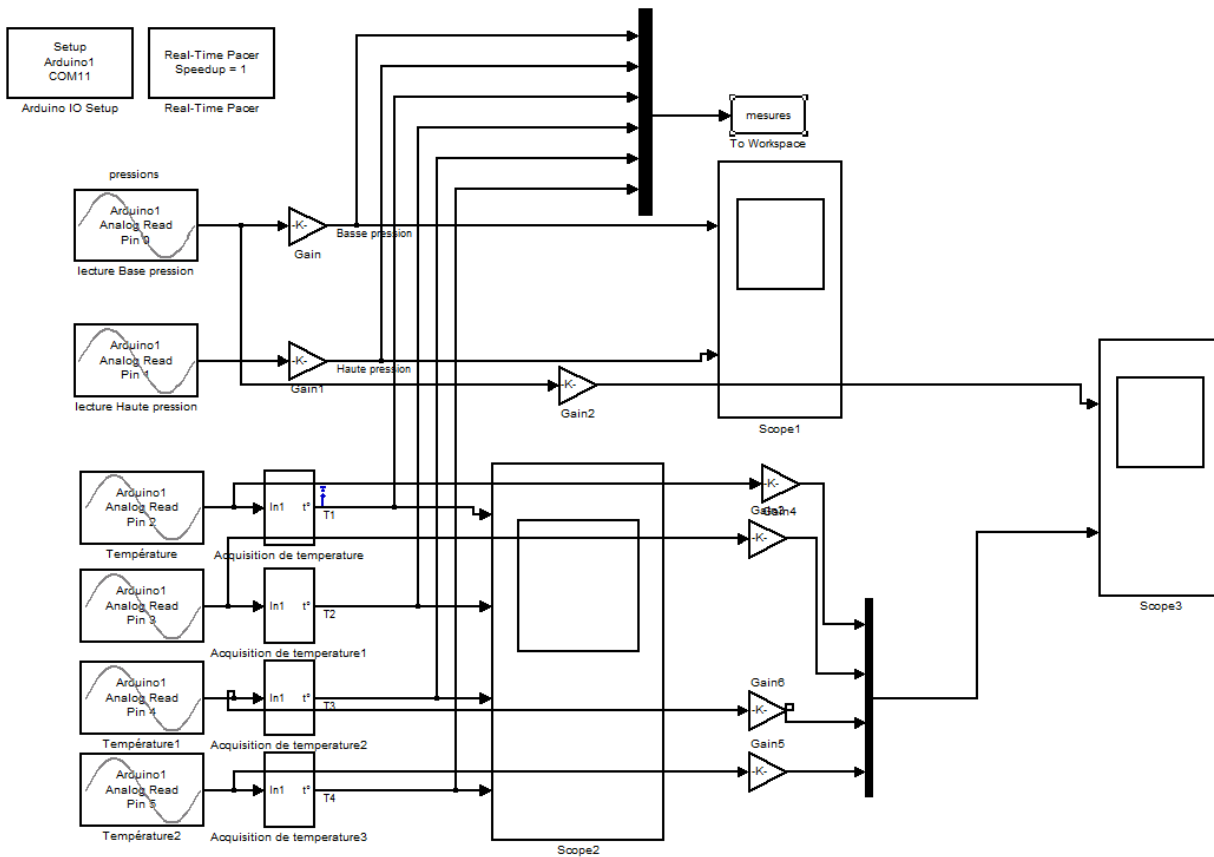
2.1 Le GUI



Le pilotage de l'acquisition est fait par une interface graphique permettant le lancement des mesures. Les points sont positionnés à chaque acquisition, toutes les cinq minutes. Le programme associé au GUI est un script matlab. L'acquisition est faite par le script « pilotage.m », exécuté tant que l'utilisateur n'arrête pas le traitement par une action sur le bouton « stop ».

Le programme enregistre les données de température et de pression dans les points stratégiques du circuit fluide, les deux pressions, haute et basse et les quatre températures à la sortie des composants. Le fichier est de type CSV pour être importé dans un tableur ou directement dans matlab pour être analysé.

Après une action de l'utilisateur sur le bouton marche, une boucle réalise une acquisition en lançant une simulation du modèle de mesure réalisé avec la bibliothèque Arduino IO. (matlab R2012).



Les scopes sont utilisés uniquement pour la mise au point du modèle. L'ensemble des acquisitions est stocké dans la variable « mesure » dans le workspace de matlab, sous la forme d'une matrice. L'exploitation des valeurs à chaque mesure est la lecture de la première ligne de la matrice. C'est avec cette ligne de valeur que les graphiques sont actualisés et mis à l'échelle.

Les différentes fonctions sont commentées dans le code matlab en annexe.

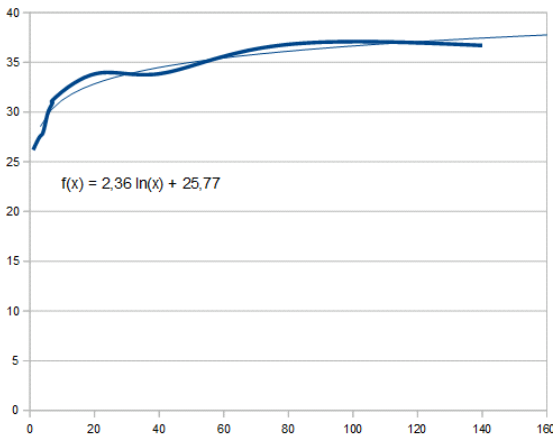
Les capteurs utilisés sont analogiques. Les données sont converties en valeurs directement exploitables.

2.2 Bilan de ce dispositif de mesure :

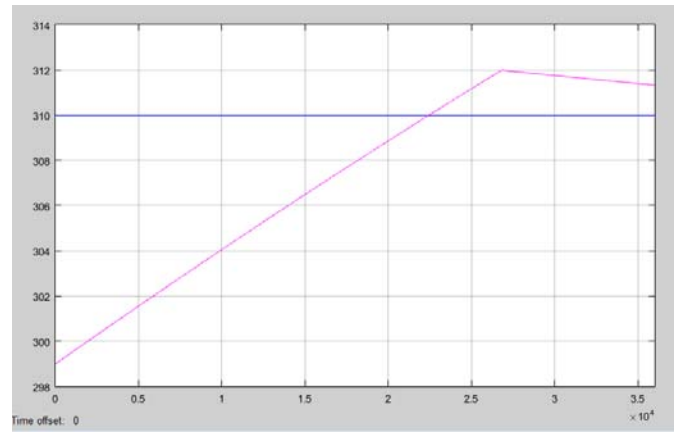
Avec une carte à bas prix, il est possible d'instrumenter un dispositif pour une longue période d'enregistrement avec un monitoring en temps réel par l'actualisation des courbes sur le PC. La sauvegarde des données dans un fichier permet de mener une analyse des écarts avec une simulation dans une seconde phase de l'étude.

Le modèle utilisé peut être adapté à d'autres systèmes. Les morceaux de script sont réutilisables. Il faut rester vigilant sur la vitesse de traitement de la chaîne d'acquisition. La liaison série par le port USB ne permettra pas une acquisition fiable sur des systèmes électroniques rapides.

2.3 Analyse de l'écart entre la mesure et la simulation

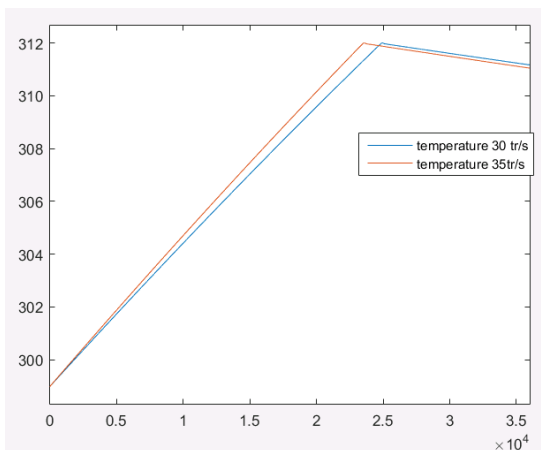


Montée en température mesurée



Montée en température simulée

La simulation est bien trop lente avec une durée d'environ 5 heures pour monter de 26° à 35° alors que la mesure sur le ballon réel est d'environ une heure. Le paramétrage du modèle nécessite un réglage complexe par la multiplicité des paramètres de l'installation. L'allure de la courbe de température simulée est trop linéaire.



Influence du débit massique sur la montée en température

L'intérêt pédagogique de la simulation résidera dans l'observation de l'influence de certains paramètres comme la masse d'eau à chauffer ou l'influence du débit de fluide dans le compresseur par exemple.

La visualisation du diagramme de Mollier animé est très parlante pour observer la baisse de performance du système en fonction de la température de l'eau du ballon.

3 Limite du modèle :

Certains paramètres de l'installation à modéliser sont inconnus. Les valeurs sont restées inchangées depuis l'exemple de départ. La courbe de montée en température est donc bien loin des mesures effectuées sur le ballon étudié.

La lenteur de la montée en température du modèle est certainement due à un débit massique beaucoup trop faible. En fonctionnement normal, le compresseur devrait débiter dans les 40 g.s^{-1} . Ici, le débit est 10 fois inférieur. Je ne suis pas parvenu à faire fonctionner le modèle avec un pilotage à 48 tour.s^{-1} , vitesse à laquelle le moteur du compresseur devrait normalement tourner. La simulation ne fonctionne plus avec une commande supérieure à 35 tr.s^{-1} .

Ce modèle très complexe sera rendu plus simple dans les futures versions de Simulink puisque les équipes de développement de Matworks travaillent sur une bibliothèque pour le domaine thermofluidique qui sera intégrée dans les outils de Simscape.

Annexes :

1 Script Matlab pilotage.m

```
%open up datalogging file
clock_now = clock; %gets a 6-element vector of the current date and
time in decimal form [year month day hour minute seconds]
clock_now(6) = round(clock_now(6)); %round to the nearest second
date_and_time_string = ''; %initialize

for i = 1:1:6 %build up the time into a string
    date_and_time_string =
[date_and_time_string,num2str(clock_now(i),'%02d')];
    if i==3 || i==5 %after the day, and after the minute, add an
underscore in the file name
        date_and_time_string = [date_and_time_string,'_']; %add an
underscore (_)
    end
end
%create the "data" directory in the current location, if necessary
if exist('data')~=7 %if "data" does not yet exist
    mkdir('data'); %creat it
end
file_name = ['data/data_',date_and_time_string,'.csv'];
fid = fopen(file_name,'w');

%reception des mesures du système
%lancement du temps
tic;

global keep_going;
keep_going = true;
loop_i = 1; %loop counter (which loop # are we on?)

%format du graphe pour la pression : pression
%dans le GUI : graphique
valeurMaxiPression=5;
ylim(handles.pression,[0,valeurMaxiPression])
title(handles.pression,'Mesure des pressions')
xlabel(handles.pression,'temps')
ylabel(handles.pression,'pression en bar')
grid(handles.pression,'on')
%format du graphe pour les températures : graphique
%dans le GUI : temperature
valeurMaxiTemperature=5;
ylim(handles.temperature,[0,valeurMaxiTemperature])
title(handles.temperature,'Mesure des températures')
xlabel(handles.temperature,'temps')
ylabel(handles.temperature,'températures en °c')
grid(handles.temperature,'on')
```

```


```

%preparation des graphiques
num_itns2plot = 100; %number of iterations to plot.
plot_handles = NaN(num_itns2plot,7); %create an array of NaNs
handle_i = 1; %a counter (handle index) to know where to place the
next value
%in the above arrays

while keep_going
 %see if the stop button has been pressed (to stop the loop)
 % stop_btn_data = get(handles.stop_btn,'UserData'); %grab the
stop_btn user data
 %keep_going = ~stop_btn_data.stop; %when stop button is pressed,
its stop_btn_data.stop value is changed from 0 to 1, so ~1 will be 0,
making keep_going false, thereby
 %forcing the while loop to stop

 %lancer le modèle
sim('mesurePressionDanfossAKS.mdl');
temps = toc;% enregistrement du temps écoulé depuis de début de
l'acquisition
toto=[mesures(1,1),mesures(1,2),mesures(1,3),mesures(1,4),mesures(1,5)
,mesures(1,6),temps];
%plot_handles(handle_i,1)=loop_i;
%plot_handles(handle_i,2)=toto(1);
%recherche valeur maximale
for i=1:1:2;
 if toto(i)>valeurMaxiPression
 valeurMaxiPression=toto(i)+5;
 end
end
for i=3:1:6;
 if toto(i)>valeurMaxiTemperature
 valeurMaxiTemperature=toto(i)+5;
 end
end
% plot data

% limite des graphiques

 xlim(handles.temperature,[0,loop_i]);
 ylim(handles.temperature,[0,valeurMaxiTemperature]);
 xlim(handles.pression,[0,loop_i]);
 ylim(handles.pression,[0,valeurMaxiPression]);
% end
%plot mesures
plot(handles.temperature,loop_i,toto(3),'*b',loop_i,toto(4),'+red',loo
p_i,toto(5),'*black',loop_i,toto(6),'+b');
plot(handles.pression,loop_i,toto(1),'*b',loop_i,toto(2),'+r');

set(handles.textTime,'String',num2str(temps));

handle_i=handle_i+1;
%fin graphique

```


```

```

fomat_donnees='';%initialisation de la mise en forme des mesures dans
le fichier
  %Première boucle: initialisation de l'acquisition des mesures
  if loop_i==1
      %plot the legend

      %Write the header to the datalogging file
      %General Header notes
      fprintf(fid,['Acquisition des températures et pression'...
                  '\n\n']);
      %Column Numbers
      for i=1:1:length(toto)
          fprintf(fid,'%d,',i);
          if i==length(toto)
              fprintf(fid,'\n');
          end
      end
      %Print Column Names
      fprintf(fid,'BP,HP, T°1 , T°2 , T°3 , T°4, temps');
      fprintf(fid,'\n');
      %one-time format string creation
      format_string = ''; %initialize string
      for i = 1:1:length(toto);
          format_string = strcat(format_string,'%3.2f'); %append the
proper # of format strings
          if i < length(toto)
              format_string = strcat(format_string,','); %add in the
commas for all but the last one
          else %i = length(data_to_log)
              format_string = strcat(format_string,'\n'); %add in
the new line command
          end
      end
      end
      %end of if loop_i==1
      %Actually log the data into the file (once per flight controller
loop!)
      fprintf(fid,format_string,toto);

      loop_i = loop_i + 1; %incrementation du compteur de boucle
      %attente entre deux mesures
      pause(60*1);
      %if toc>50
      %   keep_going=false;
      %end
end
fclose(fid); %close the datalogging file
delete(instrfind({'Port'},{'COM11'}));% fermeture de la liaison COM11

```

2 Les capteurs

2.1 Capteur de température : Danfoss AKS-11

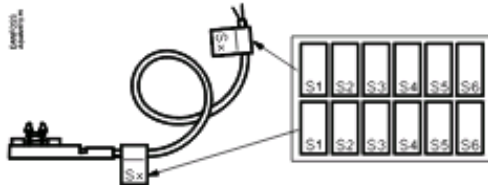
Refrigeration Controls

INSTRUCTIONS



AKS 11
Pt 1000 ohm

DB 4R9690



ohm	°C	°F	ohm	°C	°F	ohm	°C	°F
803	-50	-58	918	-21	-6	1031	8	46
807	-49	-56	922	-20	-4	1035	9	48
811	-48	-54	926	-19	-2	1039	10	50
815	-47	-53	929	-18	0	1043	11	52
819	-46	-51	933	-17	1	1047	12	54
823	-45	-49	937	-16	3	1051	13	55
827	-44	-47	941	-15	5	1055	14	57
831	-43	-45	945	-14	7	1058	15	59
835	-42	-44	949	-13	9	1062	16	61
839	-41	-42	953	-12	10	1066	17	63
843	-40	-40	957	-11	12	1070	18	64
847	-39	-38	961	-10	14	1074	19	66
851	-38	-36	965	-9	16	1078	20	68
855	-37	-35	969	-8	18	1082	21	70
859	-36	-33	973	-7	19	1086	22	72
862	-35	-31	977	-6	21	1090	23	73
866	-34	-29	980	-5	23	1093	24	75
870	-33	-27	984	-4	25	1097	25	77
874	-32	-26	988	-3	27	1101	26	79
878	-31	-24	992	-2	28	1105	27	81
882	-30	-22	996	-1	30	1109	28	82
886	-29	-20	1000	0	32	1113	29	84
890	-28	-18	1004	1	34	1117	30	86
894	-27	-17	1008	2	36	1121	31	88
898	-26	-15	1012	3	37	1124	32	90
902	-25	-13	1016	4	39	1128	33	91
906	-24	-11	1020	5	41	1132	34	93
910	-23	-9	1023	6	43	1136	35	95
914	-22	-8	1027	7	45			

Approx. 4 ohm/°C Approx. 2.25 ohm/°F

RI.14.H4.00 09-1990

2.2 Capteur de pression : Danfoss AKS 32R

Technical brochure

Transmetteur de pression avec signal de sortie ratiométrique Type AKS 32R, AKS 2050



Caractéristiques générales

- Sonde "High Tech" d'où une excellente précision
- Compensation de température sélective
- Compatible avec tous les réfrigérants, y compris l'ammoniac et le CO₂
- Régulateur de tension intégré
- Protection efficace contre l'humidité
- Conception robuste
- Grande résistance aux contraintes mécaniques telles que coups, vibrations et surpressions.
- Protection contre le bruit électromagnétique (CEM) selon la directive CEM de l'UE (marqué CE).
- Sorties à polarité protégée
- Signal de sortie spécialement adapté aux convertisseurs A/D ratiométriques raccordés.
- Principe de mesure, capteur de type scellé (référence de pression = 1013 mbar).
- Homologation UL.