

Modellering af køleanlæg

Axel Wright Larsen

Morten Juel Skovrup

Version 1.50. 2009-09-24

PSO projekt nummer 337-064.

Indholdsfortegnelse

Indledning	3
Anlæg	4
Model.....	5
Belastninger	5
Isvandstank	5
Kølerum	5
Stempelkompressor	5
Skruekompressorer.....	6
Separatorer	6
Kondensator side.....	6
Regulering	6
Resultater	8
Konklusion og anbefalinger	10
Referencer	11
Appendiks A. Kodestumper	12
Isvandstank	12
Kølerum.....	14
Separatorer	15
Sætpunktregulator.....	16

Indledning

Nærværende rapport dokumenterer modelleringen af køleanlægget ved Christiansfeld Mejericenter. Hensigten med modellen er at kunne undersøge reguleringsstrategier, hvor kompressorernes kapacitet ikke styres efter et konstant sugetryk; men mere efter hvilket kølebehov der eksisterer på de enkelte kølesteder.

Ideen er at se på de enkelte kølesteder, og ud fra afvigelsen mellem den ønskede og den reelle lufttemperatur, vurdere om der er behov for et lavere sugetryk eller om sugetrykket omvendt kan hæves. I praksis involverer det at se på hvor temperaturen ligger i det termostat-bånd der er defineret for det enkelte kølested.

Danfoss har tidligere udviklet en tilsvarende regulering (P0-optimering) til brug i supermarkedskøleanlæg – ideen er at udvikle en algoritme der kan benyttes på industrielle distribuerede systemer.

Model og forslag til regulering er udviklet som en del af PSO-projektet: Dynamisk sugetryksregulering i industrielle køleanlæg. Projekt nummer 337-064.

Anlæg

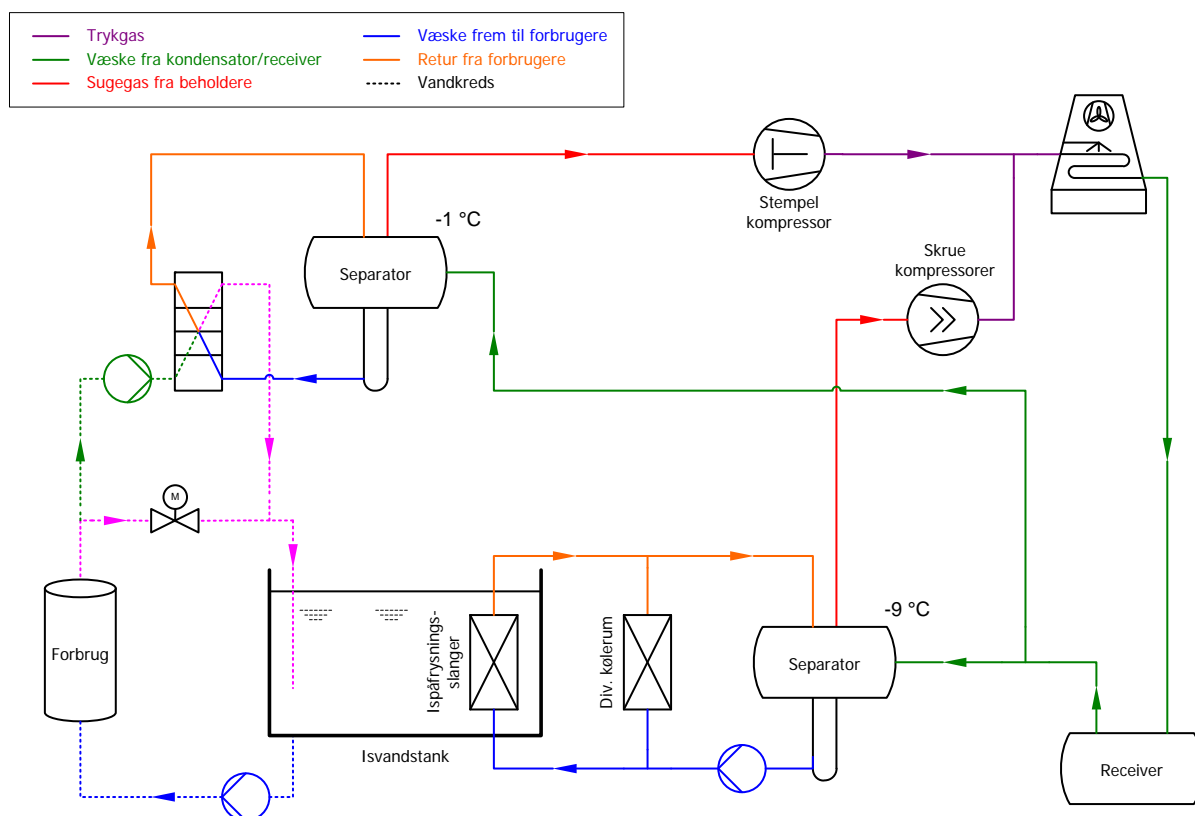
Det betragtede anlæg er beskrevet i detaljer i [1].

Anlægget består i hovedtræk af 2 temperatursystemer:

Et system med pumpecirkulation af R717 med et konstant sætpunkt på $-9\text{ }^{\circ}\text{C}$. Dette system betjener et antal kølerum samt en isvandstank med ispåfrysningsslanger.

Et system med naturlig cirkulation af R717 som hjælper isvandsanlægget, hvis kapaciteten af isvandet genereret med anlæg 1. ikke er tilstrækkelig. Dette anlæg kører med en fordamningstemperatur på $-1\text{ }^{\circ}\text{C}$

Af praktiske årsager, er diagrammet for det virkelige anlæg (kan ses i [1]) reduceret til følgende diagram, som benyttes under modelleringen:



Forbrug – eller belastninger – på køleanlægget er reduceret til:

”Forbrug” som er dækket af isvandsanlægget (mælkekøl) samt af $-1\text{ }^{\circ}\text{C}$ kredsen hvis nødvendigt.

Isvandsanlæg og diverse kølerum, som er dækket af $-9\text{ }^{\circ}\text{C}$ kredsen.

Eneste tilnærmelse i forhold til det virkelige anlæg, er at stempel kompressoren alene kører $-1\text{ }^{\circ}\text{C}$ kredsen. I det virkelige anlæg har skrukompressorerne mulighed for at hjælpe til. Denne forsimpelse anses for at være ubetydelig i forhold til formålet med modellen.

Model

Modellen af anlægget er bygget op i WinDali [2]. Nedenstående beskrives modellen af de enkelte delelementer.

Belastninger

Belastningen fra mælkekølingen er modelleret som to sinuskurver der topper kl 9:00 og 17:00. Denne del af belastningen er i modellen en del af modellen for isvandstanken.

Isvandstank

Isvandstanken modelleres med ét dynamisk element: tykkelsen af islaget på ispåfrysningsslangerne.

Input til modellen er et tidspunkt, fordampningstemperatur for skrue og stempelkompressor, samt om stempelkompressoren kører (se senere). Output fra modellen er istykkelsen samt den nødvendige massestrøm af kølemiddel fra -1 °C separatoren (stempelkompressor) og -9 °C (separatoren) skruekompressorerne. Koden for modellen kan ses i Appendiks A.

Kølerum

Hvert kølerum er modelleret med en termisk kapacitet, dvs. at modellen er noget i retning af:

$$M \cdot c_p \cdot \frac{dT}{d\tau} = \dot{Q}_{\text{belastning}} - \dot{Q}_e$$

Hvor T er temperaturen af rummet. Belastningen på rummet er en sum af varmetransmission og aktivitet i rummet. Output for modellen er nødvendig kølemiddelstrøm (kuledeydelsen \dot{Q}_e beregnes ud fra en givet UA værdi for fordamperen).

Temperaturen i rummet er termostatstyret. Modellen kan ses i Appendiks A.

Stempelkompressor

Stempelkompressoren i anlægget er en Sabroe SMC 112S kompressor. Modellen er baseret på polynomietilnærmelser for kompressorens effektoptag og kuldeydelse for forskellige kapaciteter (kompressoren kan kapacitetsreguleres fra 700 til 1500 o/min ved hjælp af en frekvensomformer). Polynomierne er baseret på data fra Sabroe.

Kompressoren reguleres på følgende måde:

- Hvis tykkelsen af isen på ispåfrysningsslangerne er mindre end en minimum værdi starter kompressoren.
- Hvis tykkelsen af isen på ispåfrysningsslangerne er større end en maksimum værdi og kompressoren har kørt i et vist tidsrum stopper kompressoren.
- Hvis kompressoren kører, reguleres kapaciteten af kompressoren således at temperaturen i separatoren holdes på -1 °C. Kapaciteten reguleres med en standard PI-regulator.

Skruekompressorer

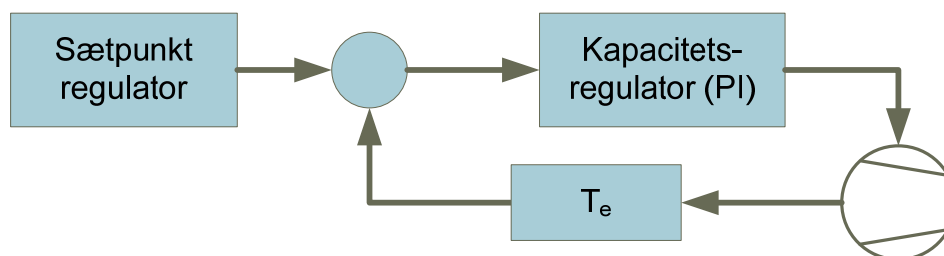
De i alt 3 skruekompressorer er af typen Sabroe SAB 163 H i to forskellige udformninger (to af kompressorerne er med frekvensomformer).

Som for stempelkompressoren er modellen er baseret på polynomietilnærmelser for kompressorernes effektoptag og kuldeydelse for forskellige kapaciteter. Polynomierne er baseret på data fra Sabroe.

Undersøgelsen af kapacitetsreguleringen af skruekompressorerne er hovedformålet med modellen. Reguleringen af sætpunktet for kapacitetsreguleringen kan i modellen sættes til:

- Konstant temperatur på -9 °C i separator (eksisterende regulering).
- Varierende temperatur efter behov (denne regulering beskrives i næste kapitel)

I begge tilfælde reguleres kapaciteten af en PI-regulator:



Separatorer

Hver af de to separatorer er dynamiske elementer i modellen. Det vil sige at massen af kølemiddel – og dermed trykket – i separatoren er dynamisk i modellen:

$$\frac{dM_{separator}}{d\tau} = \sum \dot{m}_{in} - \dot{m}_{kompressor}$$

Modellen er vist i Appendiks A.

Kondensator side

Kondensatorsiden modelleres meget simpelt i modellen, hvor det antages at kondenseringstemperaturen styres efter omgivelserne plus en fast temperaturforskel (som anbefalet i [3]).

Regulering

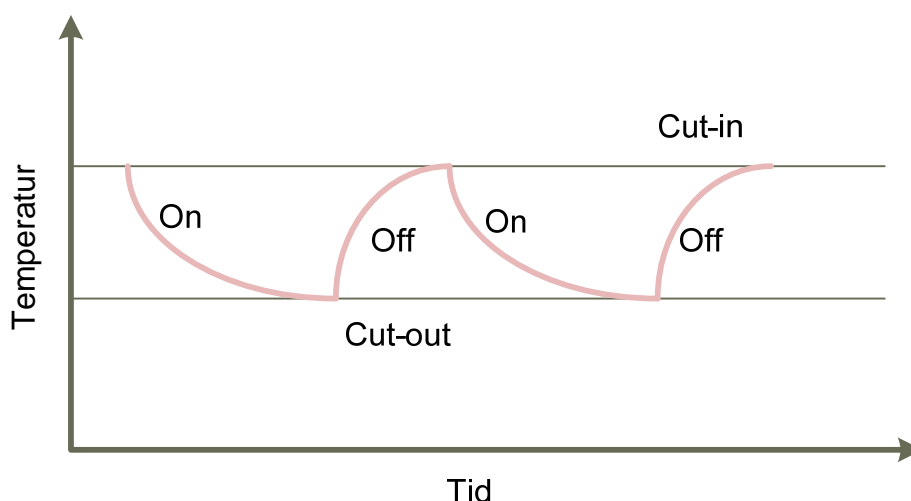
Modellen indeholder følgende regulatorer:

1. Termostatregulering af temperaturen i kølerum
2. PI-regulatorer til regulering af kompressorkapacitet
3. Regulator til regulering af sætpunktet for sugetrykket til skruekompressorerne

Den sidste regulering er den som modellen har til formål at undersøge.

Sætpunkt regulatoren er enten blot en regulator med konstant output (eksisterende regulering) eller også er det en regulering som forsøger at udtrykke, hvor belastet de enkelte kølesteder er. Hvis de er lidt belastede hæves sugetrykket, hvis de er meget belastede sænkes sugetrykket.

Konkret går reguleringen ind og ser på de enkelte kølerum, og hvor temperaturen ligger i forhold til termostatreguleringens cut-in og cut-out værdier:

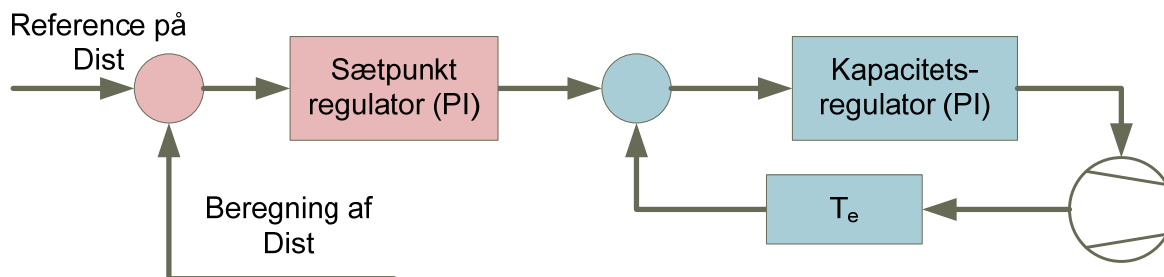


Reguleringen, der er implementeret, gør følgende:

1. Beregner middelværdien af lufttemperaturen over en periode. Beregningen medtager kun temperaturer, hvor køling har været On.
2. Middellufttemperaturen benyttes til at udtrykke en "afstand" fra Cut-in værdien i procent:

$$Dist = \frac{T_{cut-in} - T_{mean}}{T_{cut-in} - T_{cut-out}}$$

- 3.
4. Afstanden Dist beregnes for alle kølerum (som i parentes bemærket kan have forskellige værdier for T_{cut-in} og $T_{cut-out}$)
5. Den minimale værdi af Dist registreres og benyttes som tilbagekobling til en PI regulator som illustreret nedenfor:



Resultater

Nedenstående er vist resultater for et døgn simulering med konstant og variabelt sætpunkt på skruekompressorernes sugetryk.

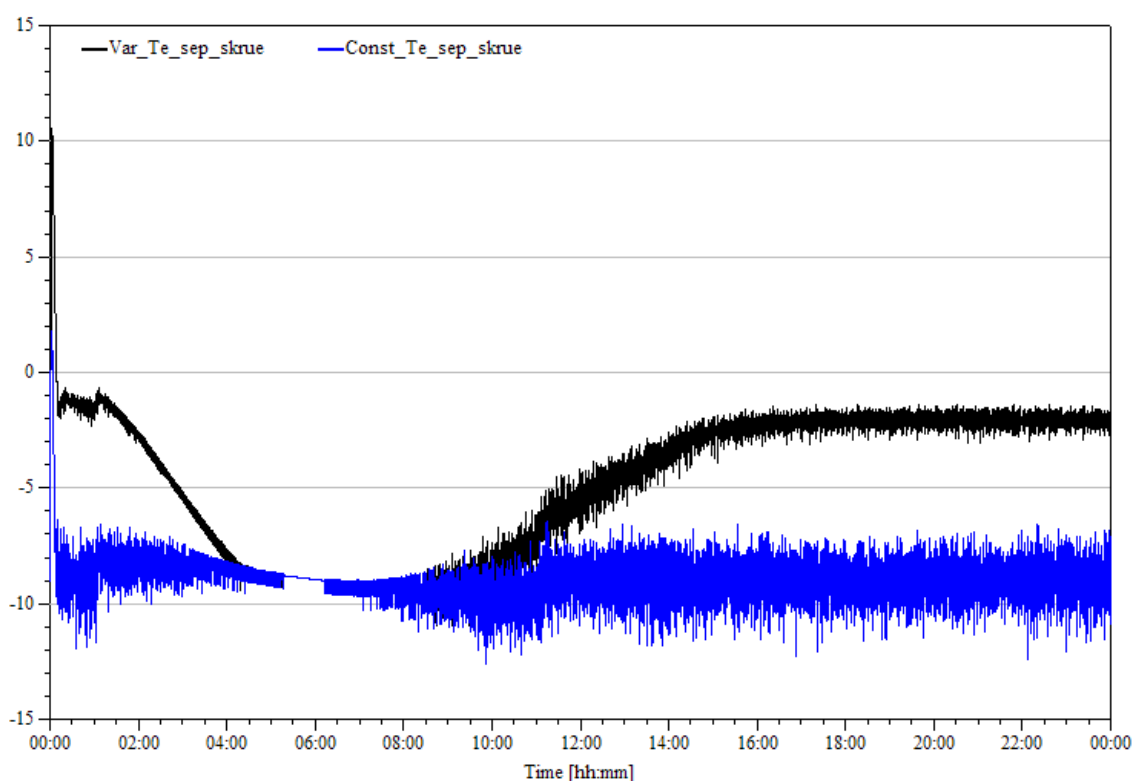
Simuleringen er kørt med 10 kølerum.

Overordnet kan følgende noteres:

	Konstant sætpunkt	Variabelt sætpunkt
Energiforbrug [kWh]	4798	4431

Altså en besparelse på omtrent 8% ved at køre med variabelt sætpunkt på kapacitetsregulatoren til skruekompressorerne.

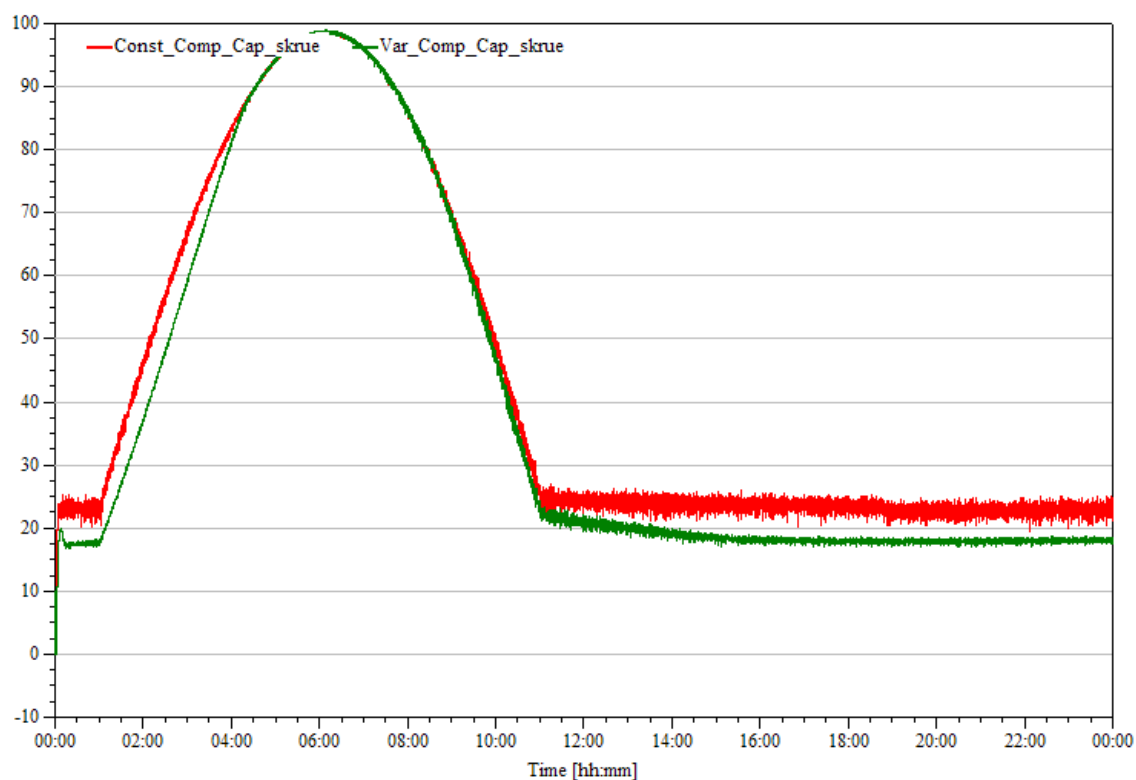
Resultatet kan også illustreres ved at se på sugetrykket i separatoren for skruekompressorerne:



Af plottet ses at regulatorerne (sætpunkt + kapacitet) holder sugetrykket på en værdi væsentligt over de -9°C i store perioder når sætpunktet for sugetrykket er variabelt (den sorte kurve). Sætpunktregulatoren har en nedre begrænsning sat til -9°C .

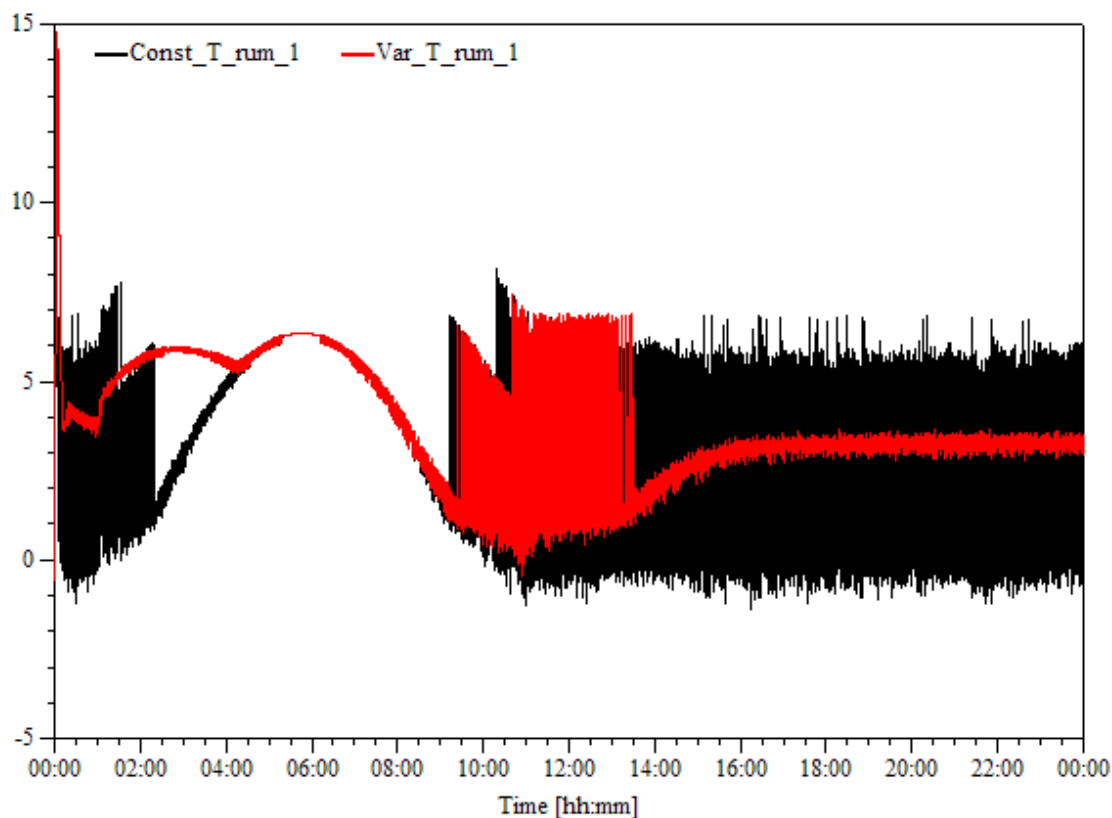
Grunden til at sugetrykket dykker under -9°C er dynamikken (trægheden) i systemet.

For den tilsvarende periode viser næste plot kapaciteten af skruekompressorerne (100% svarer til at alle tre skruer kører fuld kapacitet):



Det ses at den variable sætpunktstyring betyder at kapaciteten på skruekompressorerne i perioder kan sænkes (variabelt sætpunkt svarer til den grønne kurve) – og dermed kan der spares energi. Samtidig ses det at lavlast situationen for det simulerede døgn svarer til en relativt lille belastning (kompressorerne kører med kun ca. 20% af den maksimale kapacitet).

Den følgende figur viser temperaturen i de et af kølerummene ved henholdsvis konstant og variabelt sætpunkt:



Det ses tydeligt at den variable sætpunktregulering har medført at termostatreguleringen i rummet i perioder er sat ud af kraft (sugetrykket er hævet så meget at lufttemperaturen i rummet ikke kommer under cut-out).

Konklusion og anbefalinger

Modellen er indtil videre ikke valideret mod målinger på anlægget i Christiansfeld. På trods af dette vurderes det, at der ligger et besparelsepotentiale i størrelsesordenen 10% på anlægget i Christiansfeld ved at styre sætpunktet for kapaciteten af skruekompressorerne efter belastningen på kølerummene.

Næste skridt i udviklingen af en reguleringsalgoritme bør være at teste algoritmen på et virkeligt anlæg, og registrere om energiforbruget falder i den samme størrelsesorden som forudsagt.

Referencer

- [1] Petersen, Jesper W. – ”Beskrivelse af køleanlægget ved Christiansfeld Mejeri-center”. Notat, Enervision A/S, 18-06-2007.
- [2] Skovrup, Morten Juel – *WinDali. A Modeling and Simulation System for Microsoft Windows, Version 2.10*. Version 1.39 del af Ph.D. rapport.
- [3] Skovrup, Morten Juel; Jakobsen, Arne – *Forslag til energioptimal styring af kondenseringstryk, Carlsberg Case*. Projektrapport ESO. 28-02-2001.

Appendiks A. Kodestumper

Dette appendiks indeholder de vigtigste kodestumper for de enkelte komponentmodeller.

Isvandstank

```

procedure TIceWaterTank.ModelEquations(Klokken,T_jord,T_e_screw,T_e_recip: TFloat; RecipOn : TBoolean; var YDot:
array of TFloat);
var
  M_dot_is : TFloat;      // ændring i ismasse pr. sekund [kg/s]
  r_0       : TFloat;      // øjeblikkelige radius af isrør inkl. islag [m]
  R_isroer  : TFloat;      // isrørets varmeresistivitet: UA = Pi*L/m
  dx_isdt   : TFloat;      // beregnet ændring i istykkelsen [m]
  m_w_load,m_w_HX : TFloat;
begin
  // Beregning af belastning fra Mælkekøling - 2 (positive) sinuskurver:
  // amplitude=FQ_load_max, periode=2 [hrs], topper = kl. t_forb1 og kl. t_forb2
  if (Klokken >= (t_forb1-1)) and (Klokken <= (t_forb1+1)) then      // 1. mælkekøl
    FQ_load := FQ_load_max * sin((Klokken-(t_forb1-1))*PI/2)
  else if (Klokken >= (t_forb2-1)) and (Klokken <= (t_forb2+1)) then // 2. mælkekøl
    FQ_load := FQ_load_max * sin((Klokken-(t_forb2-1))*PI/2)
  else                                          // INGEN mælkekøl på dette tidspunkt
    FQ_load := 0;

  FT_w      := 0;
  m_w_load := FV_w_load/3600*Rho_w;
  FT_HX_in := FQ_load/(m_w_load*Cpw)+FT_w;
  if RecipOn then
    begin
      m_w_HX      := FV_w_HX/3600*Rho_w;
      FT_HX_out   := (FT_HX_in-T_e_recip)/exp(FUA_HX/(m_w_HX*Cpw))+T_e_recip;
      FQ_HX_recip := m_w_HX*Cpw*(FT_HX_in-FT_HX_out);
      Fm_recip    := FQ_HX_recip/FRefrig.RT(T_e_recip);
    end
  else
    begin
      FT_HX_out := FT_HX_in;

```

```

    FQ_HX_recip := 0;
    m_w_HX      := 0;
    Fm_recip    := 0;
end;
FQ_w_load := FQ_load-FQ_HX_recip;

// beregn UA-værdi af isrørene m. islag på
R_isroer := 1/(Falpha_i*Fd_isroer_i) + ln(Fd_isroer_y/Fd_isroer_i)/Flamb_roer +
           ln((Fd_isroer_y + 2*Fx_is)/Fd_isroer_y)/Flamb_is + 1/(Falpha_y*(Fd_isroer_y + 2*Fx_is));
FUA_is   := PI*FL_isroer/R_isroer; // UA-værdi af ÉT isrør

if RecipOn then
    FQ_tubes := Fn_isroer*FUA_is*(FT_isvand-T_e_screw)
else
    FQ_tubes := 0;

// ISVANDS-beregninger
FQ_is_tr := Fk_isvand_jord * FA_isvand_jord * (T_jord - FT_isvand); // kuldetab ved trans. [kW]
M_dot_is := (FQ_tubes - FQ_w_load - FQ_is_tr)/dH_is; // masse'flux' af is [kg/s]
r_0      := Fd_isroer_y/2 + Fx_is; // isradius [m]
Fm_screw := FQ_tubes/FRefrig.RT(T_e_screw);
dx_isdt  := sqrt(Power(r_0,2) + M_dot_is*vs_is/(Fn_isroer*FL_isroer*PI)) - r_0; //dx_is/dt ligning
if (dx_isdt < 0) then
    YDot[IdxFx_is-1] := Max(dx_isdt,-Fx_is) // Fx_is må IKKE komme under 0 [m].
else
    YDot[IdxFx_is-1] := dx_isdt;

end; // END procedure TIceWaterTank.ModelEquations

```

Kølerum

```

function TColdRoom.Controller: TBoolean;
var
    OldOn : TBoolean;
begin
    OldOn := FOn;
    if FOn then
        begin
            if FT < FTMin then
                FOn := False
            end
        else
            begin
                if FT > FTMax then
                    FOn := True
                end;
            end;
        Result := (OldOn <> FOn);
        if FOn then
            FOnVal := 1
        else
            FOnVal := 0;
        end;

procedure TColdRoom.ModelEquations(Hour,TAmb,TGround,Te: TFloat; var YDot: array of TFloat);
var
    Q_kr_A,Q_kr_TR,RT : TFloat;
begin
    // beregn kuldekald fra AKTIVITET inde i kølerummet
    if (Hour >= 8) and (Hour <= 18) then
        Q_kr_A := FQMax * sin((Hour-8)*PI/10)
    else
        Q_kr_A := 0;
    // beregn kuldekald fra VARMELEDNING ind i kølerummet
    Q_kr_TR := DefKAir*FAAir*(TAmb-FT) + DefKGround*FAGround*(TGround-FT);

    if FOn then
        FQ_e := FUA_e*(FT-Te)
    else

```

```

    FQ_e := 0;
    RT   := FRefrig.RT(Te);
    FmGas := FQ_e/RT;
    // beregn TOTALT kuldekald fra kølerummet
    FQ_kr_tot := Q_kr_TR + Q_kr_A + FQInternal;
    YDot[IdxFT-1] := (FQ_KR_tot - FQ_e)/(FMAir*Cp_luft);
end;
```

Separatorer

```

procedure TSeparator.CalcState;
begin
    {Ud fra massen beregnes temperaturen (trykket) i separatoren}
    Fv_g := FV_separator/FM_separator;
    FT_e := FRefrig.TDewV(Fv_g);
end;

procedure TSeparator.ModelEquations(m_in : array of TFloat; m_comp : TFloat;
    var YDot : array of TFloat);
begin
    YDot[FIdxMSep-1] := Sum(m_in)-m_comp;
end;
```

Sætpunktregulator

```

procedure TSetpointController.Control(Rooms: TColdRooms);
var
  i : Integer;
begin
  if FCstSP then
    FOutput := FPID.MinOutput
  else
    begin
      FPV := 100;
      for i:=1 to NColdRooms do
        begin
          if Rooms[i].Dist < FPV then
            begin
              FPV := Rooms[i].Dist;
              FMLCP := i;
            end;
          end;
          FPID.Control(FPV,FOutput);
        end;
      end;
    end;

```

Beregning af *Dist* for et kølerum:

```

procedure TColdRoom.OnSolution;
begin
  FNPoints := FNPoints+1;
  {FSamplesArray husker temperaturer NSamples tilbage. NSamples er parameter
  FNPoints benyttes til at flytte gennem FSamplesArray}
  if FNPoints > FNSamples then
    begin
      {Har nået enden og starter forfra}
      FNPoints := 1;
      if FOn then {Hvis rum er on - dvs. køling - så gem temperatur}
        FSamplesArray[FNPoints-1] := FT
    end

```



```
    else {ellers gem forrige temperatur}
      FSamplesArray[FNPoints-1] := FSamplesArray[FNSamples-1];
      FTMean := Mean(FSamplesArray);
      FInStart := False;
    end
  else
    begin
      if FInStart then
        begin
          {FSamplesArray er ikke fyldt op endnu}
          FTMean := (FNPoints-1)/FNPoints*FTMean+FT/FNPoints;
          FSamplesArray[FNPoints-1] := FTMean;
        end
      else
        begin
          if FOn then
            FSamplesArray[FNPoints-1] := FT
          else
            FSamplesArray[FNPoints-1] := FSamplesArray[FNPoints-2];
            FTMean := Mean(FSamplesArray);
          end;
        end;
      end;
      FDist := (FTMax-FTMean)/(FTMax-FTMin)*100;
    end;
```