

ISSN 0280-5316
ISRN LUTFD2/TFRT--5490--SE

Dynamisk provning av värmeväxlersystem

Magnus Ericsson
Patrik Östberg

Department of Automatic Control
Lund Institute of Technology
December 1993

Department of Automatic Control Lund Institute of Technology P.O. Box 118 S-221 00 Lund Sweden		<i>Document name</i> MASTER THESIS	
		<i>Date of issue</i> December 1993	
		<i>Document Number</i> ISBN LUTFD2/TFRT--5490--SE	
<i>Author(s)</i> Magnus Ericsson Patrik Östberg		<i>Supervisor</i> Gösta Cederin and Sven Erik Mattsson	
		<i>Sponsoring organisation</i> Alfa Laval Thermal, Lund	
<i>Title and subtitle</i> Dynamisk provning av värmväxlersystem (Dynamical Testing of Heat Exchanger Systems)			
<i>Abstract</i> <p>The report discusses mathematical modeling of thermal components, such as heat exchangers and control valves. A flexible model library for simulation has been developed. The object-oriented and equation-based modeling language Omola is used to describe the components in mathematical terms. The library of models can be used to build models of more complex thermal systems, such as heat exchanger units. OmSim which is an integrated interactive environment with tools for development and simulation of Omola models, is used to simulate individual components as well as more complex systems. A simulation model of a heat exchanger unit called TS30 from Alfa Laval Thermal has been developed. The simulation results have been verified against measured data with a very good agreement.</p> <p>The object-oriented model library supports design. It is easy to put together models for different configurations and use simulations to evaluate critical properties and limits. It is also easy to perform more detailed parameter studies. Guidelines for dynamical evaluations of different configurations has been developed.</p>			
<i>Key words</i> Computer simulation; dynamical simulation; modeling; object oriented modeling; model validation; heating and ventilation; heat exchanger; control valve			
<i>Classification system and/or index terms (if any)</i>			
<i>Supplementary bibliographical information</i>			
<i>ISSN and key title</i> 0280-5316			<i>ISBN</i>
<i>Language</i> Swedish	<i>Number of pages</i> 85	<i>Recipient's notes</i>	
<i>Security classification</i>			

Innehåll

Förord	3
Erkännanden	3
1. Inledning	4
1.1 Bakgrund	4
1.2 Syftet	4
1.3 Strategi för arbetet	5
1.4 Rapportens uppläggning	5
2. Matematiska modeller	6
2.1 Modell av en värmeväxlare	6
2.2 Modell av en reglerventil	10
2.3 Modell av temperaturgivare	14
2.4 Sammanfattning	14
3. Verifiering av de matematiska modellerna	15
3.1 Mätningar på en abonnentcentral	15
3.2 Verifiering av värmeväxlarmodellen	16
3.3 Verifiering av reglerventilen AVTQ20	23
3.4 Verifiering av tappvattensidan på TS30	27
3.5 Sammanfattning	30
4. Dynamisk analys	31
4.1 Ett tappvattenprogram	31
4.2 Undersökning av temperaturgivarna	31
4.3 Väggtemperaturer i värmeväxlaren	32
4.4 Jämförelse mellan sommar- och vinterfall	35
4.5 Reglerventilens förstärkning	37
4.6 Slutsatser	38
5. Utprovningsrutin	39
5.1 Förutsättningar för lyckade provresultat	39
5.2 Införandet av ett tappvattenprogram	39
5.3 Konstant tryck över primärsidan	41
5.4 En dynamisk utprovningsrutin	41
5.5 Presentation av mätresultat	43
5.6 Sammanfattning	46
6. Simuleringspaket	47
6.1 Termiskt komponentbibliotek	47
6.2 Initialvärdesproblem vid simuleringar i OmSim	47
7. Sammanfattning	49
8. Referenser	51
A. Enfamiljsmodul TS30	52
B. Programkod	53

Förord

Rapporten presenterar examensarbetet Dynamisk provning av värmeväxlar-system, som utförts i uppdrag åt Alfa Laval Thermal i samarbete med institutionen för Reglerteknik på Lunds Tekniska Högskola under hösten 1993. Målet med arbetet har varit att erhålla en ökad kunskap om de dynamiska egenskaperna i sammansatta system av värmeväxlare och reglerutrustning. Slutprodukten av arbetet är dels ett bibliotek av termiska komponenter som kan användas vid simulering av termiska system, dels en dynamisk utprovning-rutin för nykonstruerade värmeväxlarsystem.

Erkännanden

Vi vill rikta ett stort tack till vår handledare Gösta Cederin på Alfa Laval Thermal som hjälpt oss med teorin rörande värmeledning samt förmedlat värdefulla kontakter.

Vår handledare Sven Erik Mattsson på institutionen för Reglerteknik på Lunds Tekniska Högskola vill vi speciellt tacka för all hjälp och stöd vi har fått genom hela arbetet med bl a implementeringen av modeller i Omola och analysarbetet i simuleringsverktyget OmSim.

Vi är även oändligt tacksamma för alla kommentarer och synpunkter som Gösta och Sven Erik givit på rapportens innehåll och utformning.

Vi är även skyldiga Torsten Bynke på Alfa Laval Thermal ett stort tack för visat förtroende, vilket medfört att vi till stor del har kunnat styra uppläggningsen av arbetet själva. Vi vill även tacka Bengt Göland och Klas Bertilsson på HTC-laboratoriet för all hjälp vad gäller mätningarna på villamodulen TS30. Ett tack riktas även till Göran Hedbys, Stefan Carlström och Maria Nelson för givande diskussioner rörande marknadsmässiga intressen.

Vi vill även passa på att tacka den trevliga personalen på institutionen för Reglerteknik på Lunds Tekniska Högskola för deras osvikliga hjälp och engagemang i vårt arbete.

Lund, december 1993
Magnus Ericsson
Patrik Östberg

1. Inledning

1.1 Bakgrund

Alfa-Laval Thermal har länge konstruerat, producerat och sålt värmeväxlare av olika slag. Värmeväxlare ingår som huvudkomponent i många olika termiska tillämpningar. Abonnentcentraler för uppvärmning av bostadshus och tappvatten är ett exempel, tillämpningar inom processindustrin är ett annat.

Sedan ett par år tillbaka konstruerar företaget även kompletta system, dvs idag tillverkar man inte bara enskilda komponenter utan även mer komplexa system bestående av värmeväxlare, ventiler, pumpar och övrig reglerutrustning. De komplexa systemen kommer härnäst att benämnas för moduler.

Man har hela tiden haft en god kunskap om hur de enskilda komponenterna uppför sig under olika driftförhållanden. När man sedan kopplar ihop komponenterna till ett system blir det mer komplicerat. Det finns därför i dag ett behov av att få en ökad kännedom om hur komponenterna dynamiskt uppför sig tillsammans. Det är bl a viktigt att man kan anpassa reglersystemet och värmeväxlaren till varandra för såväl statiska som dynamiska förhållanden. Exempel på problem som kan uppstå vid en dålig anpassning är följande:

- Övervärmning av värmeöverföringsytor med kalkavlagringar som följd.
- Odämpade temperatursvängningar.
- För lång svarstid för att uppnå en bestämd temperaturnivå.
- För hög primär returvattentemperatur.

Med en ökad kunskap om hur olika komponenter påverkar varandra kan man också på ett mer effektivt sätt ställa krav på leverantörer av reglerutrustning. Man skulle även kunna använda kunskapen vid marknadsföring av företagets produkter.

1.2 Syftet

Examensarbetet syftar till att ge en ökad kunskap om dynamiken i värmeväxlare och moduler. Eftersom det idag inte finns något standardiserat sätt att ange eller mäta dynamiken vill man definiera någon form av nyckeltal för att jämföra de dynamiska egenskaperna i olika typer av värmeväxlare och moduler. Huvudsyftet med arbetet är att definiera dynamiska utprovningrutiner för nykonstruerade moduler.

Följande mål har ställts upp för examensarbetet:

- Ta fram matematiska modeller för ett antal termiska grundkomponenter.
- Bygga upp ett komponentbibliotek för simulering av värmeväxlarsystem.
- Sammanställa intressanta dynamiska egenskaper för fjärrvärmemoduler.
- Definiera jämförbara nyckeltal för olika system.
- Specificera dynamiska utprovningrutiner för nya konstruktioner.

1.3 Strategi för arbetet

För att på ett snabbt och enkelt sätt kunna undersöka uppförandet i en modul har ett modellbibliotek av värmeväxlare och reglerutrustning byggts upp för simulering av fjärrvärmemoduler. Simuleringsmodellerna är skrivna i programspråket Omola och körs under programmet OmSim. En utförlig beskrivning av Omola och OmSim finns bl a i [Mattsson *et al.*, 1993], [Andersson, 1993] och [Nilsson, 1993]. Omola är ett ekvationsbaserat och objektorienterat programspråk framtaget för att förenkla modelleringen av system. OmSim är ett program för att simulera Omola-modeller. Omola och OmSim är utvecklade vid institutionen för Reglerteknik på Lunds Tekniska Högskola där det ingår i ett större forskningsprojekt.

1.4 Rapportens uppläggning

I kapitel 2 presenteras matematiska modeller av några termiska grundkomponenter. Kapitel 3 behandlar verifieringen av de matematiska modellerna. Detta följs av en dynamisk analys av en tappvattenmodul i kapitel 4. Analysen berör en rad intressanta områden som har med regleringen av modulen att göra. En utprovningrutin för nykonstruerade moduler presenteras i kapitel 5, här ges även förslag på hur en presentation av provresultaten kan se ut. I kapitel 6 kommenteras simuleringsbiblioteket som utvecklats och använts under arbetets gång. Slutligen ges en kort sammanfattning av rapporten i kapitel 7.

2. Matematiska modeller

För att kunna beskriva verkligheten vid simulering måste man sätta upp samband som anger hur olika fysikaliska storheter hör ihop. Man kan aldrig få en exakt återgivning av verkligheten, men ofta kan man komma ganska nära den. Det är viktigt att bara ta med det som är relevant för applikationen. Det här avsnittet beskriver hur man återger verkligheten i de olika komponenterna som ingår i en tappvattenmodul.

2.1 Modell av en värmeväxlare

Relationer och ekvationer som används för att modellera värmeväxlare kan hittas i en bra lärobok t ex [Holman, 1972]. Det är främst tryck, flöde och temperatur som beskriver en värmeväxlares termodynamiska uppförande.

Sambandet mellan flödet genom och tryckfallet över primärsidan respektive sekundärsidan beskrivs av ekvation (2.1)

$$p_{in} - p_{ut} = \frac{\rho}{C_v^2} q^2 \quad (2.1)$$

där

- p_{in} Trycket vid inloppet [Pa].
- p_{ut} Trycket vid utloppet [Pa].
- q Flödeshastigheten vid den aktuella sidan [m^3/s].
- ρ Densiteten på transportmediet [kg/m^3].
- C_v Flödeskoefficienten [$\sqrt{kgm^3/(s^2 Pa)}$].

Ekvationerna för att beskriva temperaturförändringarna i en värmeväxlare är lite mer komplicerade. En modell är

$$\frac{\partial T_p}{\partial t} + a_p \frac{\partial T_p}{\partial (x_p/l)} + a_p b_p (T_p - T_s) = 0 \quad (2.2)$$

$$\frac{\partial T_s}{\partial t} + a_s \frac{\partial T_s}{\partial (x_s/l)} + a_s b_s (T_s - T_p) = 0 \quad (2.3)$$

$$a_i = \frac{q_i}{V} \quad i = p, s \quad (2.4)$$

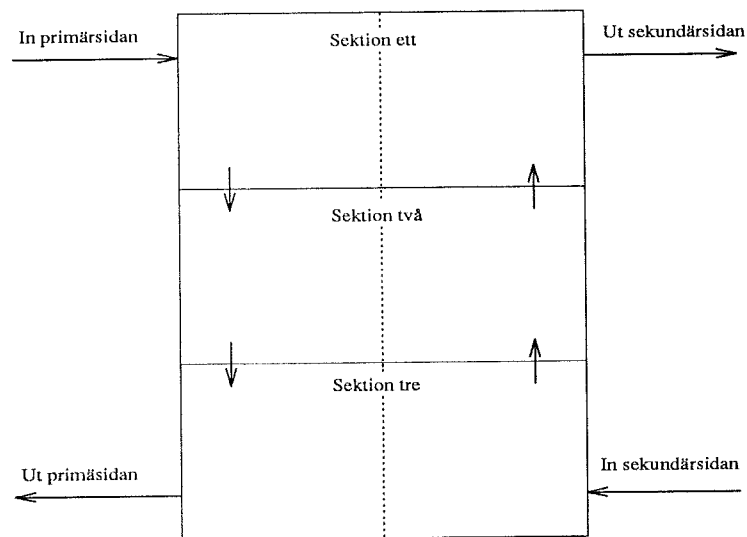
$$b_i = \frac{kA}{q_i \rho c_p} \quad i = p, s \quad (2.5)$$

Indexen p och s står för primär- respektive sekundärsida.

där

- T Temperaturen [$^{\circ}\text{C}$].
- q Flödes hastigheten [m^3/s].
- V Den volym värmeväxlaren rymmer på respektive sida [m^3].
- A Den värmeledande ytan [m^2].
- k Totalt värmegenomgångstal [$\text{W}/(\text{m}^2\text{K})$].
- ρ Densiteten på transportmediet [kg/m^3].
- c_p Specifik värmekapacitet [$\text{J}/(\text{kgK})$].
- x Längdkoordinat [m].
- l Total längd [m].

För att modellen ska bli hanterlig vid simuleringar approximerar man de partiella differentialekvationerna med ett antal ordinära differentialekvationer. Detta gör man genom att diskretisera tillståndsvariablerna i längdled samt att dela upp både primär- och sekundärsidan i ett antal lika stora sektioner, se figur 2.1.



Figur 2.1 En schematisk bild över en värmeväxlare indelad i tre enkla sektioner.

Temperaturförändringen i transportmediet genom en enkel sektion beskrivs av följande ekvationer

$$\frac{dT_{Put}}{dt} = a_p(T_{Pin} - T_{Put} - b_p\Delta T_m) \quad (2.6)$$

$$\frac{dT_{Sut}}{dt} = a_s(-(T_{Sut} - T_{Sin}) + b_s\Delta T_m) \quad (2.7)$$

$$\Delta T_1 = T_{Pin} - T_{Sut} \quad (2.8)$$

$$\Delta T_2 = T_{Put} - T_{Sin} \quad (2.9)$$

$$\Delta T_m = \frac{\Delta T_1 - \Delta T_2}{\ln(\Delta T_1/\Delta T_2)} \quad (2.10)$$

där

ΔT_m Logaritmska medeltemperaturdifferensen [K].

Värmeomgångstalet

Man kan uttrycka värmeomgångstalet för hela värmeväxlaren som en seriekoppling av termiska resistanser

$$\frac{1}{k} = \frac{1}{\alpha_p} + \frac{1}{\alpha_s} + R_w + R_f \quad (2.11)$$

där

- k Värmeomgångstalet [$W/(m^2K)$].
- $\alpha_{p,s}$ Värmeövergångstalet på respektive sida [$W/(m^2K)$].
- R_w Väggs resistivitet [$(m^2K)/W$].
- R_f Resistivitet pga försmutsning [$(m^2K)/W$].

Värmeomgångstalet beror på respektive sidas flöde. Om man inte tar med det här sambandet i modellen utan bara använder sig av ett konstant värmeomgångstal får man ett dåligt uppförande i stationaritet. Vidare beräknas värmeövergångstalet utifrån

$$\alpha = C \left(\frac{q_m}{q_0} \right)^n [1 + a(T - T_0)] \quad (2.12)$$

$$R_w = \frac{\delta}{\lambda_{plåt} Y_F} \quad (2.13)$$

där

- q_m Transportmediets massflöde per kanal [kg/s].
- q_0 Massflöde för normalisering satt till 1 kg/s .
- T Transportmediets medeltemperatur [$^{\circ}C$].
- C, n, a Modellkonstanter identifierade utifrån mätdata.
- δ Plåttjocklek [m].
- $\lambda_{plåt}$ Värmeledningsförmågan för plåten [$W/(mK)$].
- Y_F Ytförstoring pga plåtens vågmönster.
- T_0 En uppskattning av medelvärdet mellan den lägsta och den högsta temperaturen som är aktuella i verkligheten.
I vårt fall satte vi $T_0 = \frac{110-5}{2} = 57.5 \text{ } ^{\circ}C$.
Detta innebär en linjärisering av korrektionstermen för temperaturberoendet $a(T - T_0)$ kring T_0 .

Om man utgår från ekvationerna (2.11) och (2.12) samt kombinerar dem med värmegenomgångsekvationen

$$P = kA\Delta T_m \quad (2.14)$$

där P är värmebelastningen $[W]$.

Då får man efter omskrivning

$$k = \frac{1}{\frac{1}{c q_1^n [1 + a(T_1 - T_0)]} + \frac{1}{c q_2^n [1 + a(T_2 - T_0)]} + R_w + R_f} = \frac{P}{A\Delta T_m} \quad (2.15)$$

och vidare

$$C = \left(\frac{1}{q_1^n [1 + a(T_1 - T_0)]} + \frac{1}{q_2^n [1 + a(T_2 - T_0)]} \right) / \left(\frac{A\Delta T_m}{P} - R_w - R_f \right) \quad (2.16)$$

Beräkning av värmeväxlarens väggtemperatur

Det finns ett intresse av att ha modeller som beskriver hur yttemperaturen varierar genom värmeväxlaren. Anledningen är att vattnet fäller ut kalk på ytor med för hög temperatur. Det kan finnas så mycket kalkavlagringar inne i värmeväxlaren att den förlorar en stor del av sin normala kapacitet. När försmutsningen upptäcks sköljs i regel kanalerna i värmeväxlaren med någon syra, för att den ska återfå sin normala kapacitet. Problemet är att det inte alltid är så lätt att upptäcka försmutsningen, med en dåligt utnyttjad effekt hos värmeväxlaren som följd.

För att beräkna en temperatur mitt inne i den värmeöverförande väggen samt yttemperaturerna på primär- respektive sekundärsidan har vi använt följande ekvationer

$$\frac{dT_{v\ddot{a}gg}}{dt} = \frac{k}{\delta \rho c_p} ((T_{mp} - T_{v\ddot{a}gg}) - (T_{v\ddot{a}gg} - T_{ms})) \quad (2.17)$$

$$T_{Pv\ddot{a}gg} = q_p c_p \Delta T_p \frac{\delta}{2\lambda Y_F A_i} + T_{v\ddot{a}gg} \quad (2.18)$$

$$T_{Sv\ddot{a}gg} = T_{v\ddot{a}gg} - q_s c_p \Delta T_s \frac{\delta}{2\lambda Y_F A_i} \quad (2.19)$$

$$T_{mp} = \frac{T_{Pin} + T_{Put}}{2} \quad (2.20)$$

$$T_{ms} = \frac{T_{Sin} + T_{Sut}}{2} \quad (2.21)$$

$$\Delta T_p = T_{Pin} - T_{Put} \quad (2.22)$$

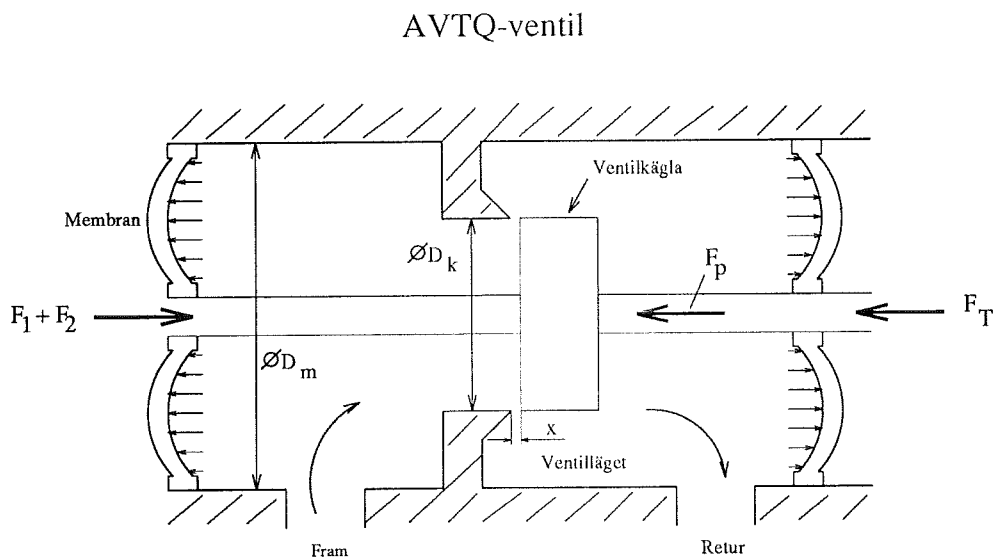
$$\Delta T_s = T_{Sut} - T_{Sin} \quad (2.23)$$

där

$T_{v\ddot{a}gg}$	Temperaturen mitt inne i den värmeöverförande ytan [$^{\circ}C$].
T_{mp}	Transportmediets medeltemperatur på primärsidan [$^{\circ}C$].
T_{ms}	Transportmediets medeltemperatur på sekundärsidan [$^{\circ}C$].
$T_{Pin,Put}$	Inlopps- respektive utloppstemperatur på primärsidan [$^{\circ}C$].
$T_{Sin,Sut}$	Inlopps- respektive utloppstemperatur på sekundärsidan [$^{\circ}C$].
$T_{Pv\ddot{a}gg}$	Väggens yttemperatur på primärsidan [$^{\circ}C$].
$T_{Sv\ddot{a}gg}$	Väggens yttemperatur på sekundärsidan [$^{\circ}C$].
A_i	Sektionens värmeledande yta [m^2].

2.2 Modell av en reglerventil

Eftersom en reglerventil av typ AVTQ20 från Danfoss A/S används i vårt mätobjekt har en modell av denna implementerats. Information om hur ventilen fungerar och är uppbyggd har hämtats från [Andersson *et al.*, 1992] och [Wollerstrand, 1993]. Ventilen fungerar som en proportionell regulator från temperaturen hos tappvattnet till ventilkäglaens öppning x , se figur 2.2. För att förbättra regleregenskaperna har man satt in en framkoppling från flödet på tappvattensidan till ventilkäglaens öppning. Framkopplingen har som funktion att snabba upp regleringen för att förhindra övervärmning vid kraftig minskning av tappvattenflödet. Ventilen består av fyra delar käglaens ventilhus, ett temperaturelement, ett flödesavkännelement och en inställningsfjäder. Hur dessa modelleras beskrivs nedan.



Figur 2.2 Schematisk bild på hur reglerventilen AVTQ är uppbyggd.

Ventilkaraktistik

Förhållandet mellan tryckfallet över och flödet genom en ventil kan beskrivas som

$$q^2 = k_v^2 \Delta p \quad (2.24)$$

där q är flödet genom ventilen och Δp tryckfallet över den. Faktorn k_v ökar med ökande ventilöppning. För en ventil av typ AVTQ20 är k_v maximalt $3.0 \cdot 10^{-6} \text{ m}^3 / (s\sqrt{Pa})$ vid den maximala öppningen $x_{max} = 1.6 \text{ mm}$. Maximala k_v betecknas k_{vs} . Ventilkaraktistiken dvs sambandet mellan ventilläget och k_v för ventilen ser ut enligt följande

$$k_v(x) = \frac{k_{vs0} \frac{x}{x_{max}}}{\sqrt{1 + \left(\frac{k_{vs0}}{k_{vx}} \frac{x}{x_{max}}\right)^2}} \quad (2.25)$$

där

$$\begin{aligned} k_{vs0} &= 4.4 \cdot 10^{-6} \text{ m}^3 / (s\sqrt{Pa}) \\ k_{vx} &= 4.1 \cdot 10^{-6} \text{ m}^3 / (s\sqrt{Pa}) \end{aligned}$$

Om man använder sig av ekvationerna (2.24) och (2.25) erhålles flödet genom ventilen som funktion av ventilläget x och tryckfallet Δp över ventilen.

Temperaturelementet

I reglerventilen sitter ett temperaturelement som omvandlar temperaturskillnaden, mellan mätt temperatur och en referenstemperatur, till en kraft som påverkar ventikägglans läge. Temperaturelementet består av en gasvolym vilken påverkar en kolv. Kolven utgörs av en metallbälg som vid $50 \text{ }^\circ\text{C}$ ger kraften 300N . Gasen ifråga är CO_2 . Temperaturberoendet består i att kol i känselkroppen binder en viss mängd CO_2 . Mängden som binds minskar med ökande temperatur. Kraften som temperaturelementet skapar betecknas F_T . Man kan uttrycka denna som

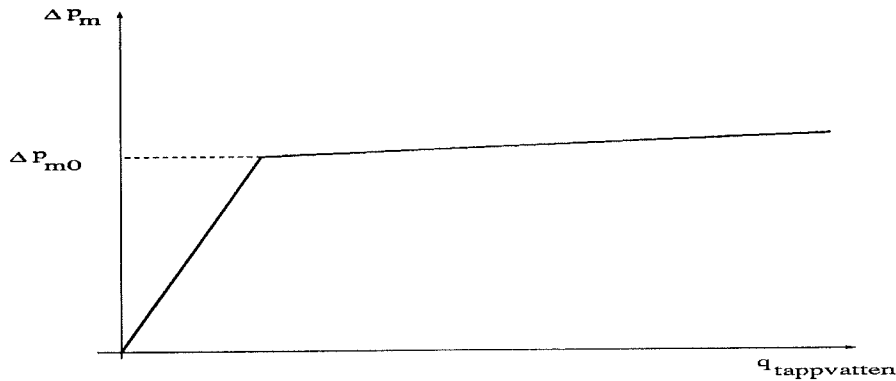
$$F_T = F_0 + K \Delta T_v + c_b x \quad (2.26)$$

där

- ΔT_v Tappvattnets avvikelse från $50 \text{ }^\circ\text{C}$.
- F_0 Kraften då x är noll och mätt temperatur $50 \text{ }^\circ\text{C}$ dvs 300 N .
- K Förstärkningen från tappvattentemperaturen är $0.8 \text{ N}/^\circ\text{C}$.
- c_b Fjäderkonstant på 16400 N/m .

Flödesavkännarelementet

Det flödesavkännande elementet består av ett membran som utvecklar en kraft beroende på tryckskillnaden över en styrventil på tappvattensidan. Styrventilen består av ett litet hål samt en fjäderbelastad kägla som är parallellkopplade. Vid lågt tappvattenflöde täpper käglan till och flödet går genom det lilla hålet. I detta område är tryckfallet proportionellt mot flödet. Käglan öppnar sedan vid ett visst tryckfall som betecknas $\Delta p_{m0} = 0.2 \cdot 10^5$ Pa. Tryckfallet över styrventilen är sedan relativt konstant vid ökat tappvattenflöde, se figur 2.3.



Figur 2.3 Sambandet mellan tryck och flöde för styrventilen på tappvattensidan.

Membranet påverkar en ventilspindel via en fjäder med fjäderkonstanten $c_1 = 13000$ N/m. Fjäderkraften som skapas betecknas F_1 , se figur 2.2. Membranet kan röra sig mellan två ändlägen. Kraften F_1 är 32 N då membranet är vid det nedre läget. Vid det övre är kraften $F_1 = 97$ N. Avståndet mellan övre och nedre läget är 0.005 m. Membranets avstånd från nedre läget betecknas y . Riktningen hos y är samma som riktningen för kägslans läge x .

Anordningen kan befinna sig i tre tillstånd.

I. Membranet i nedre läget, dvs

$$\left\{ y = 0, F_1 = F_{10} - c_1 x = 32 - 13x, \Delta p_m < \frac{F_1}{A_m} \right\} \quad (2.27)$$

II. Membranet i övre läget, dvs

$$\left\{ y = y_{max} = 0.005 \text{ m}, F_1 = F_{10} - c_1 y_{max} - c_1 x = 97 - 13x, \Delta p_m > \frac{F_1}{A_m} \right\} \quad (2.28)$$

III. Membranet rörligt, dvs

$$\left\{ 0 < y < y_{max}, F_1 = A_m \Delta p_m, y = \frac{F_1 - F_{10}}{c_1} + x \right\} \quad (2.29)$$

där A_m är membranets area $= 4.5 \cdot 10^{-3} \text{ m}^2$.

Vid stora tappvattenflöden befinner sig anordningen i tillstånd II, vid små i III. Tillstånd I motsvarar ett tappvattenflöde nära noll.

Inställningsfjäders

Det finns en fjäder som verkar mellan ventilspindeln och ventilhuset. Den skapar en kraft som strävar efter att öppna ventilen. Det är med denna fjäder som man ställer in den önskade tappvattentemperaturen. Kraften betecknas F_2 och är

$$F_2 = F_{20} - c_2 x \quad (2.30)$$

där F_{20} är förspänningen och c_2 betecknar fjäderkonstanten vilken är 5900 N/m. F_{20} justeras med en skruvanordning.

Tryckfallets inverkan på regleringen

Tryckfallet över ventilen påverkar membranen i ventilen, se figur 2.2 på sidan 10. Trycket i framledningsdelen verkar på ringarean mellan diametrarna D_k och D_m . Den resulterande kraften blir i negativ dvs stängande riktning om $D_k < D_m$. Motsvarande sker i returledningsdelen med kraften i positiv riktning om $D_k < D_m$. Den stängande kraften F_p blir

$$F_p = \frac{\pi}{4}(D_m^2 - D_k^2)(p_F - p_R) = A_{mk}(p_F - p_R) \quad (2.31)$$

där

D_m	Diametern på fram- och returledningssidan [m].
D_k	Diametern på kägla [m].
A_{mk}	Resulterande ringarea mellan D_m och D_k [m ²].
$p_F - p_R$	Tryckfallet över ventilkägla [Pa].

Tryckfallet över AVTQ-ventilen betecknas med Δp . Sambandet mellan tryckfallet över ventilkägla och AVTQ-ventilen kan skrivas som

$$p_F - p_R = \Delta p \frac{1}{1 + \left(\frac{k_{v\theta 0} x}{k_{vx} x_{max}}\right)^2} \quad (2.32)$$

Sammanställning av krafterna

Jämvikt för spindeln innebär enligt figur 2.2 att

$$F_1 + F_2 = F_T + F_p \quad (2.33)$$

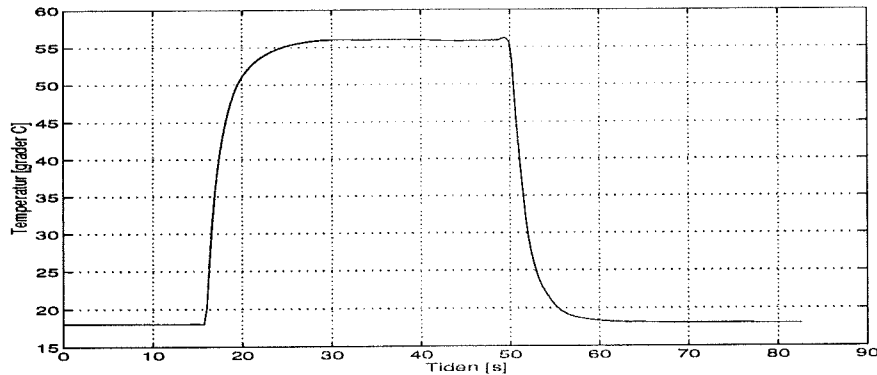
Om man använder sig av ekvationerna (2.26), (2.30), (2.31) och (2.32) erhålles följande samband.

$$F_1 + F_{20} - c_2 x = F_0 + K \Delta T_w + c_b x + A_{mk} \frac{\Delta p}{1 + \left(\frac{k_{v\theta 0} x}{k_{vx} x_{max}}\right)^2} \quad (2.34)$$

Beroende på vilket tillstånd som det flödesavkännande elementet befinner sig i använder man sig av en av ekvationerna (2.27), (2.28) eller (2.29). Då erhålles en ekvation från vilken man kan beräkna ventilläget x .

2.3 Modell av temperaturgivare

En temperaturgivare uppför sig som figur 2.4 visar vid en stegändring av temperaturen.



Figur 2.4 Temperaturgivarens karakteristik.

Den uppför sig som ett typiskt första ordningens system. Temperaturgivaren har därför modellerats enligt ekvation (2.35).

$$T_k \frac{dT_{uppmätt}}{dt} = -T_{uppmätt} + KT_{verklig} \quad (2.35)$$

där

$T_{uppmätt}$	Temperaturen ut från temperaturgivaren [$^{\circ}C$].
$T_{verklig}$	Verklig temperatur [$^{\circ}C$].
T_k	Tidskonstanten i modellen.
K	Statisk förstärkning i modellen.

2.4 Sammanfattning

I kapitlet beskrivs hur en modell av en värmeväxlare kan byggas upp. Modellen som beskrivs kan förbättras ytterligare genom att ta med väggens dynamik. Det skulle ha speciellt stor inverkan vid simulerade uppstarter av stora moduler då uppvärmning av värmeväxlaren, rör och infästningar sker. Väggens dynamik skulle även påverka resultatet vid tjockare plåtar i värmeväxlaren.

En modell av reglerventilen AVTQ20 presenteras också. I modellen har inte friktion och glapp modellerats. En implementering av de båda begreppen skulle vara önskvärd för att få en ännu bättre modell.

Temperaturgivarna uppför sig som första ordningens system och är relativt lätta att modellera.

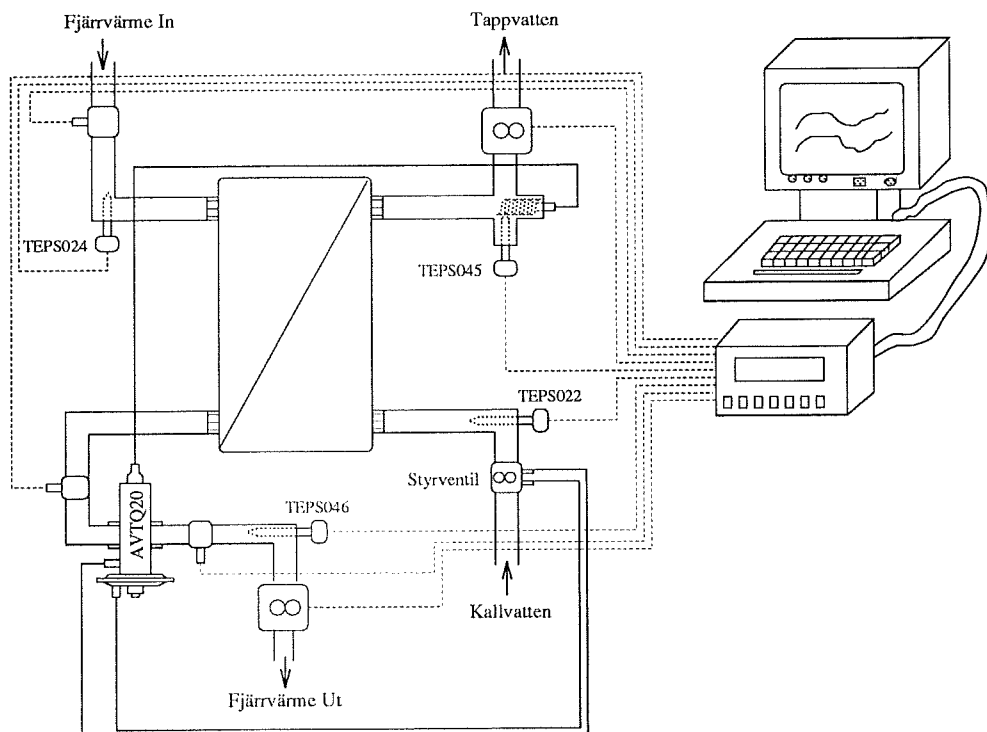
3. Verifiering av de matematiska modellerna

För att undersöka hur de framtagna modellerna stämmer överens med verkligheten har modellerna verifierats på en enfamiljsmodul. Som verifieringsmetod har direkta jämförelser mellan uppmätta och simulerade värden använts. En matematisk analys av mätresultaten har gjorts för att validera strukturen på uppsatta modeller.

3.1 Mätningar på en abonnentcentral

Mätningarna har utförts på en abonnentcentral avsedd för en villa. Enfamiljsmodulen finns uppkopplad i laboratoriet på Alfa-Laval Thermal. I modulen finns en radiatorsida och en tappvattensida. Vi har bara gjort mätningar på tappvattensidan. Mer utförlig dokumentation om modulen finns i appendix A.

En schematisk bild över mätupställningen vid tappvattensidan visas i figur 3.1. Det finns fyra temperaturgivare inkopplade, en på in- och utloppet på både primär- och sekundärsidan. Vidare finns det två flödesmätare, en för fjärrvärmevattnet och en för tappvattnet. Det finns även en differenstryckmätare, med vilken trycket över olika komponenter kan mätas. En nackdel är att endast trycket över en komponent per mätserie kan mätas.



Figur 3.1 En schematisk bild över mätupställningen i Alfa Lavals laboratorium.

Mätningarna har utförts med hjälp av en HP-mät dator som är kopplad till en styrdator via ett GPIB-system. Ett problem som uppkommit är för långa samplingsintervall. Om man tittar på hur snabbt systemet är skulle det vara önskvärt med en samplings tid på ungefär 0.25 sekunder. Efter byte av temperaturgivare och optimering av mätprogrammet kom vi fram till att det blir svårt att komma ner till den önskade samplings tiden om man samtidigt ska ha visuell kontroll över alla mätsignalerna på styrdatorns skärm.

Vid de mätningar som genomförts har ett samplingsintervall på en sekund använts. Alternativet för att få ännu snabbare mätningar är att släppa på kravet att se signalerna samtidigt som mätningen pågår. Då skulle man troligen komma ner i den önskade samplingshastigheten. Vi har emellertid gjort den bedömningen att den nuvarande mät hastigheten är tillräcklig för våra syften.

Efter att ha genomfört den första serien av mätningar uppstod ett problem vid verifieringen av värmeväxlarens modell. Det visade sig att temperaturförändringen genom värmeväxlaren stämde mycket bra på primärsidan medan överensstämmelsen var ganska dålig på sekundärsidan. Efter noggrann felsökning i modeller, mätprogram, data över mätobjekt och mätupställningen kom vi fram till att det var temperaturgivaren på tappvattensidans utgång som hade ett för långsamt insvängningsförlopp och dessutom var felplacerad. Den hade hela tiden suttit en bit ovanför det vattenflöde den ska mäta temperaturen på, varför den har visat ett mer långsamt insvängningsförlopp av temperaturen än vad det i verkligheten var. Efter byte av temperaturgivaren och korrigering av dess placering blev överensstämmelsen mellan de simulerade temperaturerna och de uppmätta avsevärt bättre.

3.2 Verifiering av värmeväxlarmodellen

Verifieringen av värmeväxlarmodellen gjordes i tre steg. Den första är en undersökning av hur många sektioner som behövs i värmeväxlarmodellen. I de två övriga delarna verifieras modellerna av tryckfallet över värmeväxlaren och temperaturförändringen genom den.

Inverkan av antalet sektioner i värmeväxlarmodellen

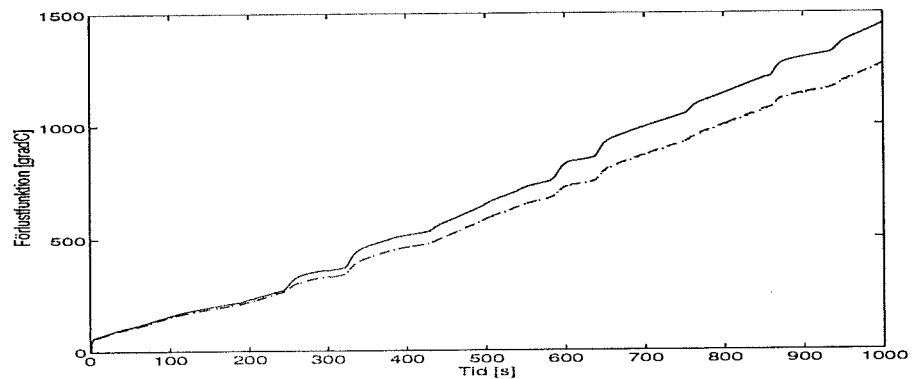
För att undersöka vilken inverkan antalet sektioner i modellen av värmeväxlaren har på simuleringsresultatet, har modeller med upp till sex sektioner simulerats. Vid varje simulering jämfördes de simulerade temperaturerna med de uppmätta. För att få ett rättvist mått på hur bra respektive implementering är kan man införa en förlustfunktion enligt ekvation (3.1). Förlustfunktionen kan tolkas som summan av felen mellan de uppmätta och simulerade temperaturerna.

$$Loss = \int_0^t \sqrt{(T_{Pmätt} - T_{Psim})^2 + (T_{Smätt} - T_{Ssim})^2} dt \quad (3.1)$$

där

$Loss$	Förlustfunktion [$^{\circ}C$].
t	Tid [s].
$T_{Pmätt}$	Uppmätt returvattentemperatur på primärsidan [$^{\circ}C$].
T_{Psim}	Simulerad returvattentemperatur på primärsidan [$^{\circ}C$].
$T_{Smätt}$	Uppmätt tappvattentemperatur [$^{\circ}C$].
T_{Ssim}	Simulerad tappvattentemperatur [$^{\circ}C$].

Med förlustfunktionen får man ett bra mått på hur mycket de simulerade temperaturerna avviker från de verkliga. Förlustfunktionen för implementeringarna med en till tre sektioner visas i figur 3.2.

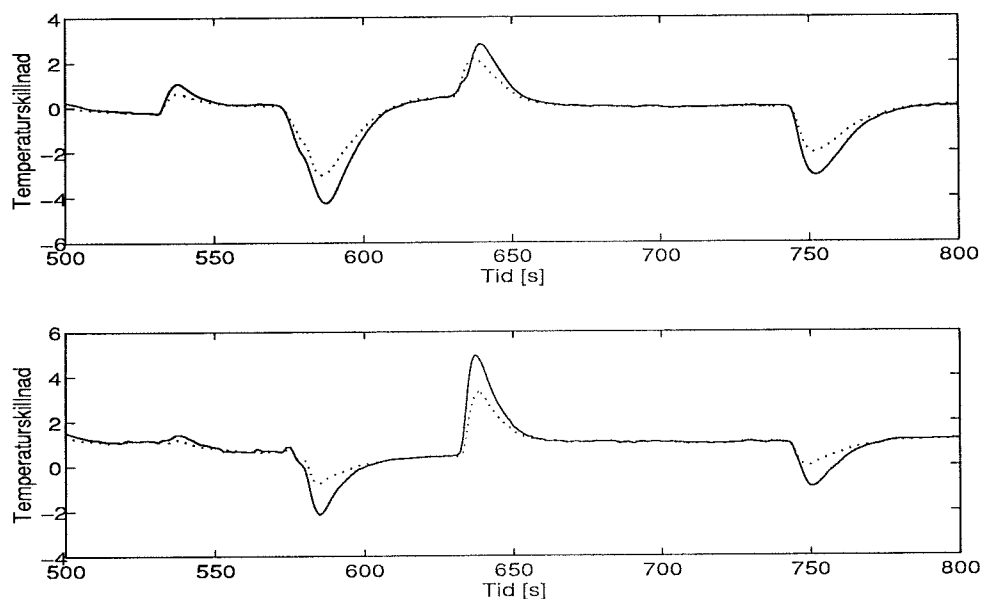


Figur 3.2 Förlustfunktionen för värmeväxlare med en uppdelning i en till tre sektioner.

Kurvan som ligger överst visar förlustfunktionen vid implementering med endast en sektion. Förlustfunktionen minskar sedan succesivt med varje sektion som läggs till vid implementeringen. Man ser en klar förbättring när man går från en till två sektioner. När man lägger till ytterligare sektioner sker endast en marginell förbättring. I figuren ovan kan man inte se någon skillnad mellan kurvorna med två och tre sektioner. I det fortsatta arbetet har en implementering med tre sektioner i värmeväxlarmodellen valts, då simuleringstiden ökar med att antalet sektioner.

I figurerna 3.9 och 3.10 på sidan 21 visas temperaturförändringarna genom värmeväxlaren med tillhörande flödesändringar av tappvattnet för en simulering med tre sektioner.

Ser man bara på hur förlustfunktionen varierar med de olika implementeringarna är det lätt att avgöra om en implementering är bättre än en annan. Det är däremot svårt att bilda sig en uppfattning om hur stor temperaturskillnaden mellan mätta och simulerade värden är då värmeväxlarmodellen delas in i olika antal sektioner. Figur 3.3 visar felet, dvs skillnaden mellan uppmätt och simulerad temperatur i °C.

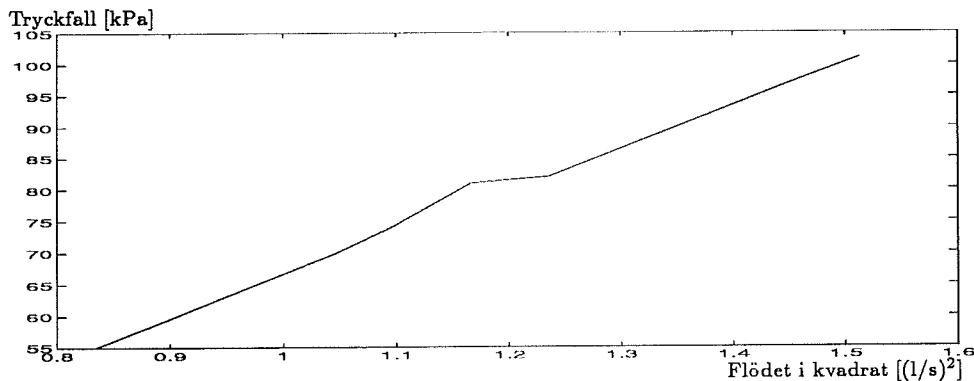


Figur 3.3 Den övre bilden visar temperaturskillnaden mellan uppmätta och simulerade temperaturer på primärsidan medan den undre bilden visar motsvarande kurvor för sekundärsidan. De heldragna linjerna visar felet med en sektion medan de prickade visar felet med tre sektioner.

Förlustfunktionen med tre sektioner är mycket mindre än den för en sektion men när man tittar på temperaturskillnaderna är emellertid inte skillnaden så drastisk.

Verifiering av tryckfallet över en värmeväxlare

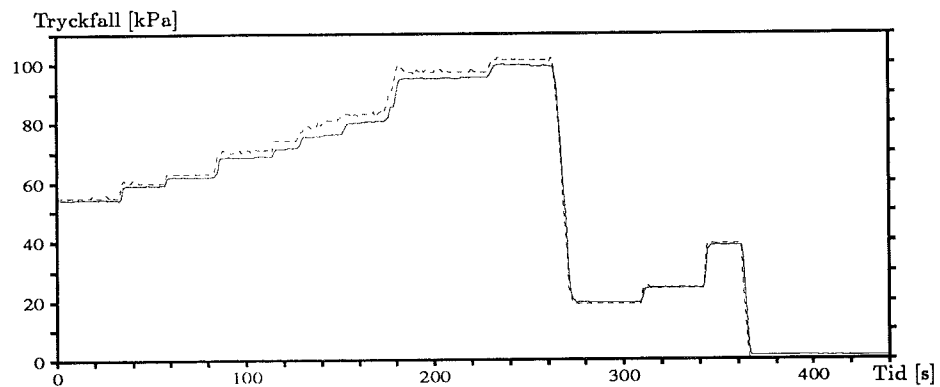
I denna verifiering mättes trycket över och flödet genom värmeväxlaren. För att bestämma konstanten C_v samt för att verifiera det proportionella sambandet i ekvation (2.1) på sidan 6, undersöktes sambandet mellan trycket och flödet, se figur 3.4.



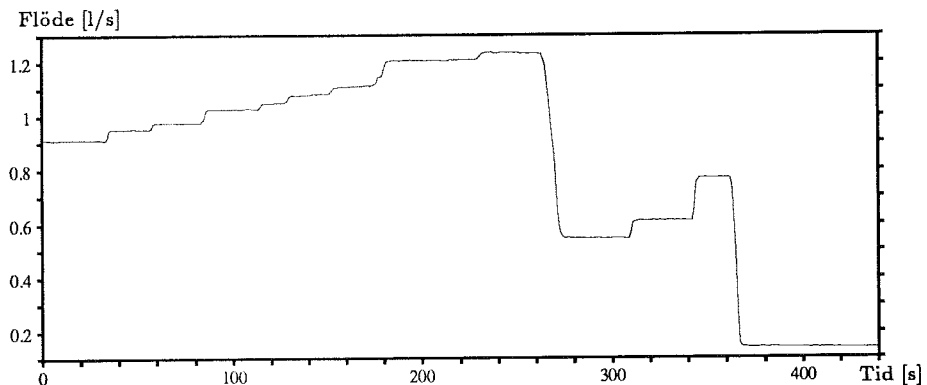
Figur 3.4 Tryckfallet över värmeväxlaren som funktion av flödet i kvadrat på primärsidan.

Antagandet att tryckfallet över värmeväxlaren är proportionellt mot flödet i kvadrat är riktigt förutom i ett område kring 80 kPa enligt, figur 3.4. Vad detta beror på är osäkert. Med ett ρ på 998 kg/m^3 blir $C_v = 1.22 \cdot 10^{-4} [\text{kgm}^3/(\text{s}^2\text{Pa})]^{1/2}$ för värmeväxlaren.

En verifiering av modellen med vårt uträknade C_v gjordes genom att använda det uppmätta flödet som insignal till simuleringsmodellen. Det uppmätta tryckfallet jämfördes med det simulerade, se figur 3.5. I ett område kring 80 kPa är överensstämmelsen mellan simulerat tryckfall och uppmätt något sämre. I figur 3.6 visas hur tappvattenflödet varierar under experimentet.



Figur 3.5 Den heldragna linjen visar det simulerade tryckfallet över värmeväxlarens primärsida med ett C_v på $1.22 \cdot 10^{-4}$, medan den streckade linjen visar det uppmätta tryckfallet.

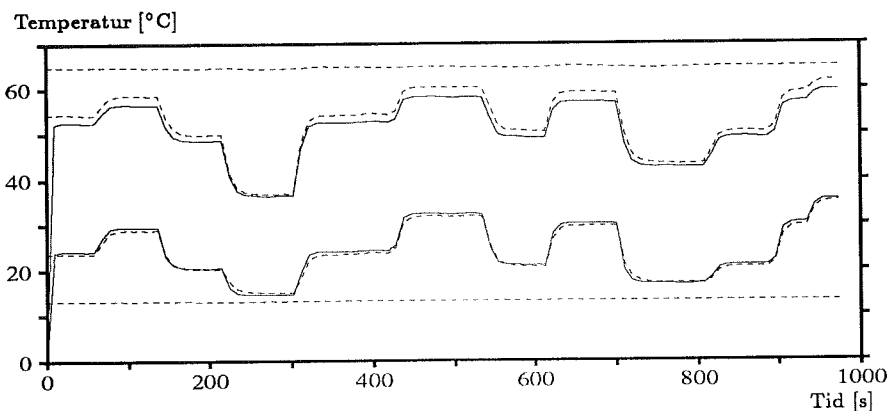


Figur 3.6 Variationen av tappvattenflödet.

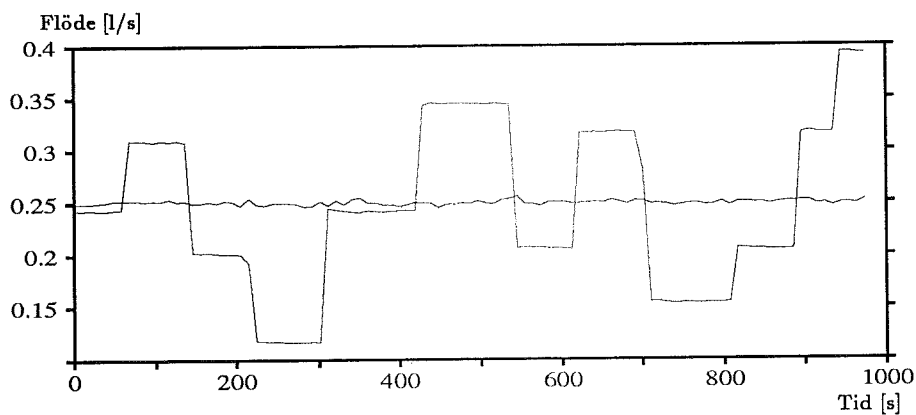
Som figur 3.5 visar, stämmer de simulerade värdena mycket väl överens med de uppmätta utom i ett område kring 80 kPa. Avvikelsen är dock inte stor. Därför har ett proportionellt samband mellan tryckfallet och flödet i kvadrat antagits.

Verifiering av temperaturförändringarna genom värmväxlaren

Här har uppmätt flöde och temperatur använts både på primär- och sekundärsidan som signaler till simuleringsmodellen. För att se hur temperaturen förändras med tiden varierades fjärrvärmevattnets flöde medan tappvattenflödet var konstant, se figurerna 3.7 och 3.8.

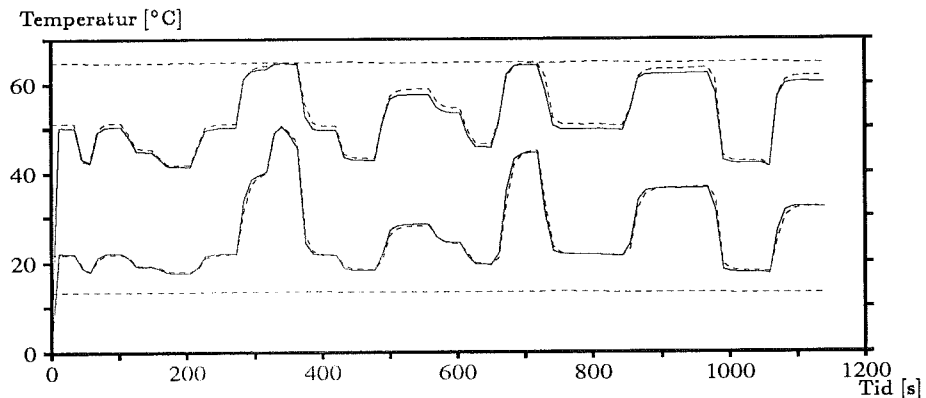


Figur 3.7 Det är konstanta inloppstemperaturer på både primär- och sekundärsidan, $69\text{ }^{\circ}\text{C}$ respektive $12\text{ }^{\circ}\text{C}$. De streckade kurvorna är de uppmätta temperaturförändringarna medan de heldragna visar de simulerade. De övre kurvorna visar temperaturen på tappvattnet och de nedre visar fjärrvärmevattnets returtemperatur.

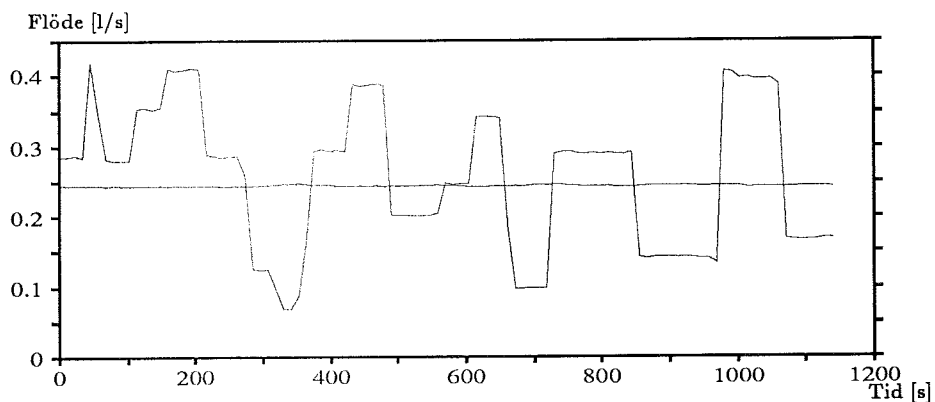


Figur 3.8 Konstant tappvattenflöde på cirka 0.25 l/s och ett varierande flöde på fjärrvärmevattnet.

Simulerade och uppmätta temperaturer ut från värmexlaren stämmer bra överens då primärsidans flöde varierar. För att se om överensstämmelsen blev lika bra då man varierar sekundärsidans flöde undersöktes detta. I denna mätningen varierades tappvattenflödet medan fjärrvärmevattnets flöde hölls konstant, se figurerna 3.9 och 3.10.



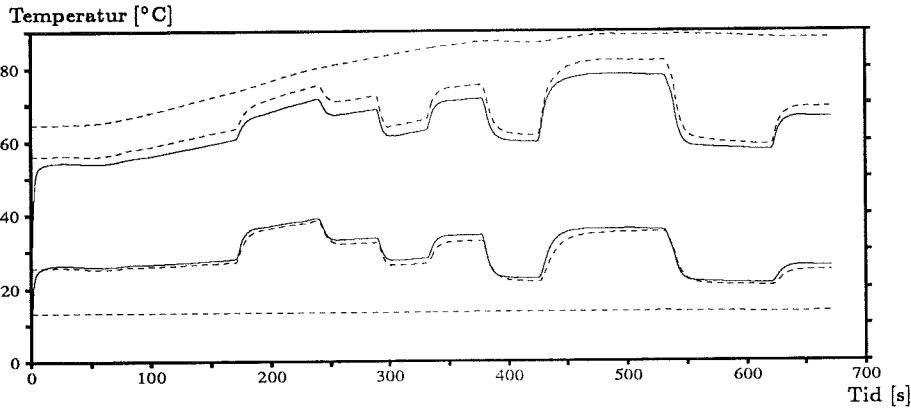
Figur 3.9 Det är konstanta inloppstemperaturer på både primär- och sekundärsidan, $69\text{ }^{\circ}\text{C}$ respektive $13\text{ }^{\circ}\text{C}$. De streckade kurvorna är de uppmätta temperaturförändringarna medan de heldragna visar de simulerade. De övre kurvorna visar tappvattentemperaturen och de nedre visar fjärrvärmevattnets returtemperatur.



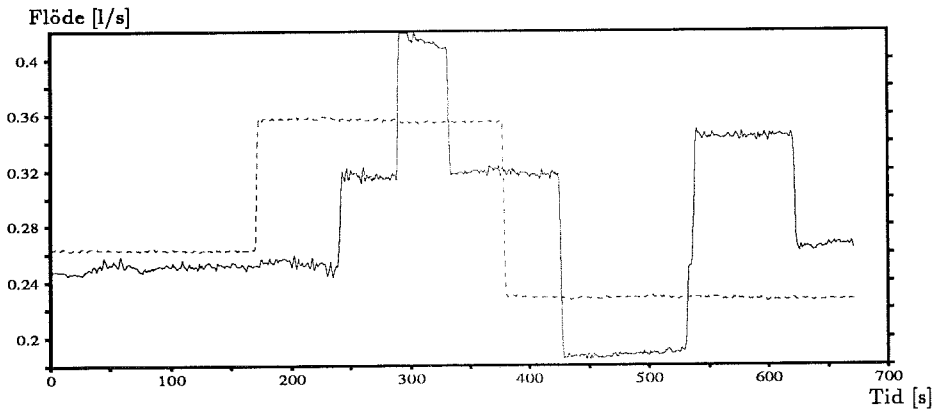
Figur 3.10 Konstant flöde av fjärrvärmevattat på cirka 0.25 l/s och ett varierande tappvattenflöde.

De simulerade temperaturerna stämmer även här mycket väl överens med de verkliga. På sekundärsidan finns stationärt fel. Det kan bero på att temperaturgivaren här inte har en idealisk placering. Precis i början av körningen fås ett insvängningsförlopp, som beror på att alla temperaturer initialt har satts till noll i simuleringsprogrammet.

För att undersöka modellens riktighet vid varierande temperaturer på primärsidans inlopp gjordes ett försök där temperaturen på fjärrvärmevattnet sakta steg från 65 °C till ungefär 90 °C, se figurerna 3.11 och 3.12.



Figur 3.11 Den översta kurvan visar temperaturen på fjärrvärmevattnet. Det övre kurvpåret visar simulerad respektive uppmätt tappvattentemperatur medan det undre kurvpåret visar primärsidans simulerade och uppmätta temperatur på returvattnet. Streckade kurvor visar uppmätta resultat, medan de heldragna visar simulerade värden.



Figur 3.12 Den streckade kurvan är primärflödet medan den heldragna visar sekundärflödet.

I försöket stör vi systemet ytterligare genom att ändra temperaturen på fjärrvärmevattnet samtidigt som primär- och sekundärsidans flöden varieras. Modellen över värmexlaren klarar även detta mycket bra. Temperaturförändringen är ganska långsam så resultatet var kanske inte så överraskande. Önskvärt vore också att undersöka hur modellen uppför sig vid snabbare temperaturförändringar.

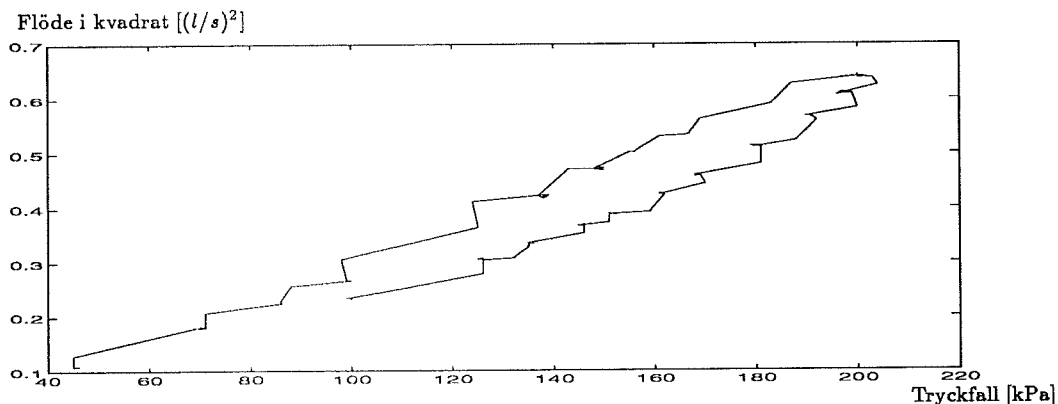
För att undersöka simuleringsmodellens giltighet i ett större område har även de två första försöken utförts med en temperatur på fjärrvärmevattnet in i värmexlaren på ungefär 90 °C. Simuleringarna visade sig stämma nästan lika bra som de för den lägre temperaturen på fjärrvärmevattnet.

3.3 Verifiering av reglerventilen AVTQ20

För att verifiera modellen av reglerventilen AVTQ20 gjordes tre olika mätningar. En för att verifiera ekvation (2.24) på sidan 11. Den andra mätningen gjordes för att kontrollera modellens riktighet vid ett konstant tryckfall över ventilen. Slutligen gjordes en mätning med ett varierande tryckfall över ventilen.

Verifiering av tryckfallet

I den första mätningen mättes tryckfallet över en oreglerad ventil. Ventilen höll en konstant ventilöppning genom att den temperaturgivare som skickar signaler till regulatorn hängdes fritt i luften. Lufttemperaturen var vid mätfallet ungefär $15\text{ }^{\circ}\text{C}$. Flödet på tappvattensidan var i denna mätning avstängt. Ventilen hade en referenstemperatur på $55\text{ }^{\circ}\text{C}$, vilket medförde att reglerventilen hade maximal ventilöppningen under mätningen. Tryckfallet över en ventil är modellerat att vara proportionellt mot flödet i kvadrat, se ekvation 2.24. Genom att plotta flödet i kvadrat som funktion av trycket kan proportionalitetskonstanten räknas ut, se figur 3.13.

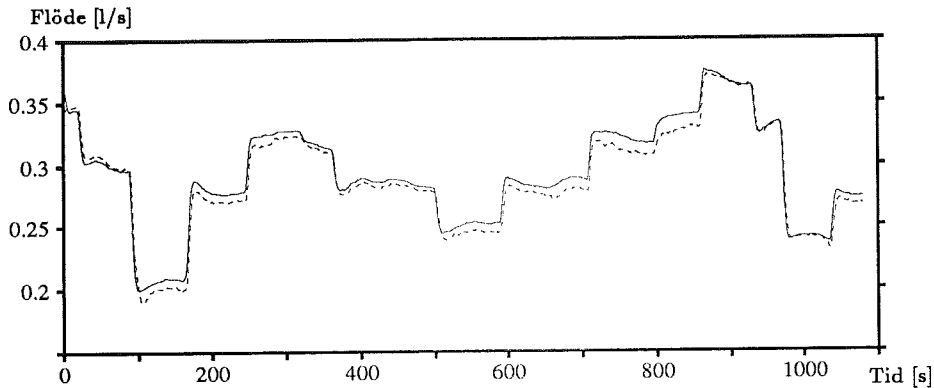


Figur 3.13 Flödet genom ventilen i kvadrat som funktion av tryckfallet över den.

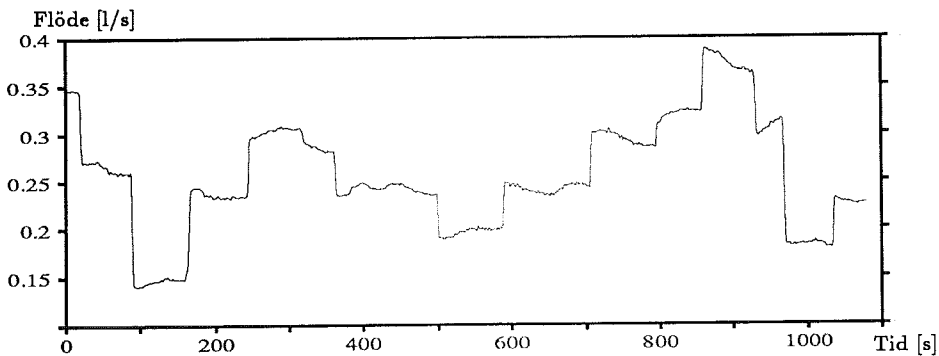
I figuren 3.13 ser man att ventilen inte uppför sig symmetriskt då flödet ökar respektive minskar genom den. Det fenomenet beror troligen på glapp i ventilen.

Verifiering med differenstryckshållare

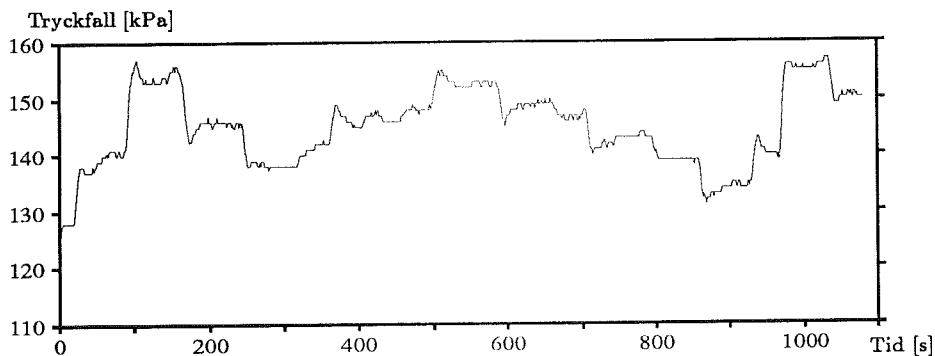
Här utfördes mätningen med reglerventilens temperaturgivare inkopplad på tappvattensidan. På primärsidan var en differenstryckshållare inkopplad. Det medförde att tryckfallet över reglerventilen var omkring 1.5 bar, se figur 3.16. För att verifiera modellen av reglerventilen användes uppmätt tappvattenflöde, tappvattentemperatur och tryckfall över ventilen som insignaler till modellen. Simulerat och uppmätt flöde genom ventilen jämfördes sedan.



Figur 3.14 Flödet genom reglerventilen AVTQ20. Heldragen linje är simulerat flöde och prickat är uppmätt.



Figur 3.15 Uppmätt tappvattenflöde i mätningen med differenstryckshållare inkopplad på primärsidan.

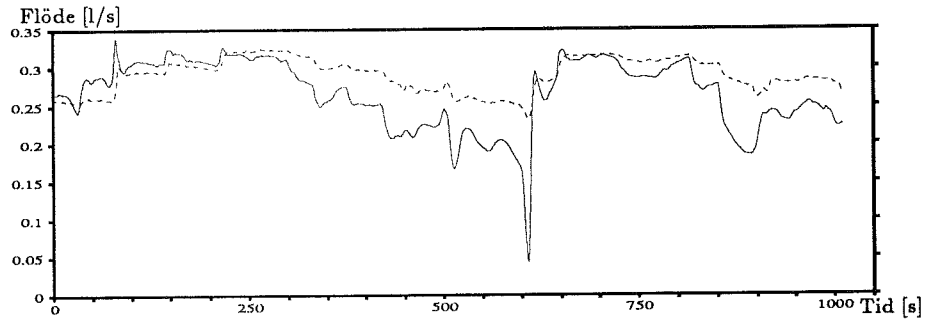


Figur 3.16 Uppmätt tryckfall över ventilen med differenstryckshållare inkopplad.

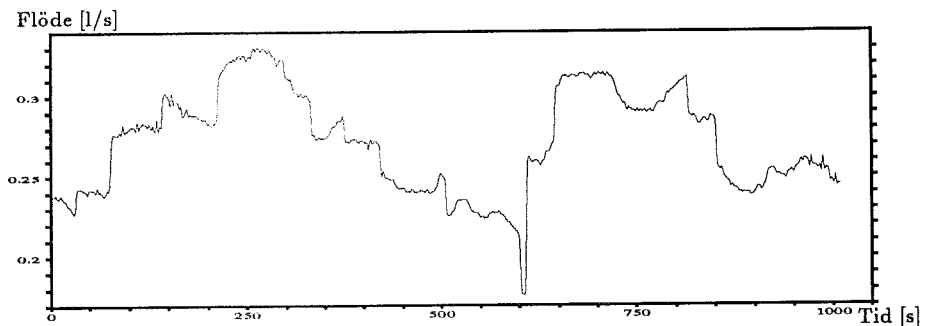
Man ser att det simulerade flödet följer det uppmätta flödet bra. Modellen ser alltså ut att fungera bra när man har ett relativt konstant tryckfall över reglerventilen. Förstärkningen K från tappvattentemperaturen till ventilläget x blev i det här fallet 0.8, se ekvation (2.26) på sidan 11.

Verifiering utan differenstryckshållare

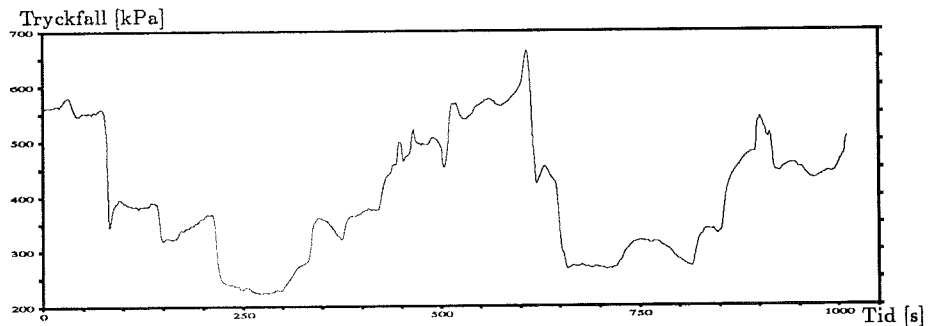
Även i denna verifiering var reglerventilens temperaturgivare inkopplad på tappvattensidan. Skillnaden var att ingen differenstryckshållare var inkopplad på primärsidan. Detta medförde att tryckfallet över ventilen varierade mellan 2 och 6 bar, se figur 3.19. Även här användes uppmätt tappvattenflöde, tappvattentemperatur och tryckfall över ventilen som insignaler till modellen. Simulerade och det uppmätta flöden genom ventilen jämfördes sedan.



Figur 3.17 Flödet genom reglerventilen AVTQ20. Helt dragen linje är simulerat flöde och prickat är uppmätt.

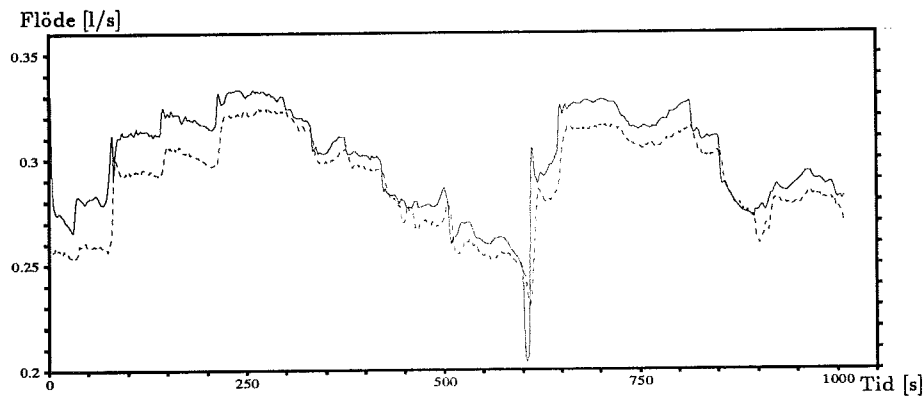


Figur 3.18 Tappvattenflödets variation i mätningen utan differenstryckshållare inkopplad på primärsidan.



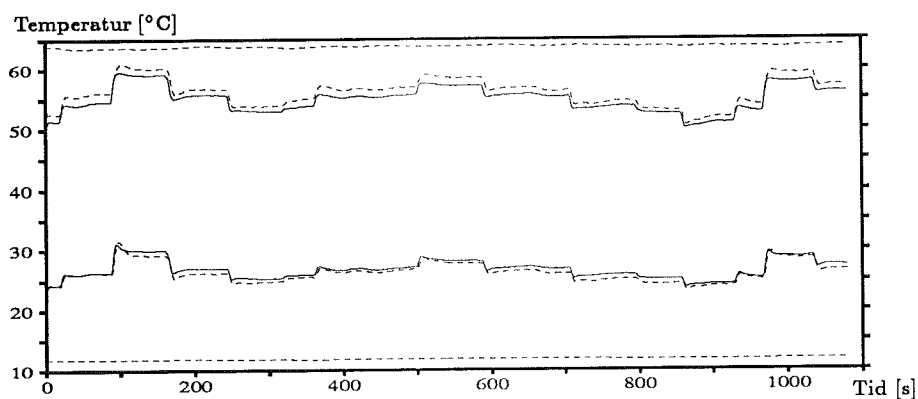
Figur 3.19 Tryckfallet över ventilen.

När tryckfallet över ventilen varierar följer det simulerade flödet inte det uppmätta lika bra vid ökande tryckfall över reglerventilen. Det beror på att trycket påverkar membranen i ventilen, se figur 2.2 på sidan 10. Ett ökat tryck medför att ventilens totala fjäderkonstant blir större. Alltså blir regleringen vid högre tryck långsammare. Sambandet mellan trycket och ventilens fjäderkonstant är ganska komplicerat. Därför är inte detta samband implementerat i vår modell av reglerventilen. Avvikelsen mellan simulerade och uppmätta värden blir dock mindre när man kopplar in modellen av reglerventilen i ett system. Det beror på att man då har en återkoppling av tappvattentemperaturen. Skillnaden mellan de båda fallen är att i det återkopplade fallet är det simulerad tappvattentemperatur man skickar in till modellen av reglerventilen medan man skickar in uppmätt tappvattentemperatur i det andra fallet. Detta medför att i fallet med återkoppling stämmer det simulerade flödet genom ventilen bättre överens med det uppmätta. Figur 3.20 visar uppmätt och simulerat flöde genom ventilen vid simulering med reglerventilen AVTQ20 inkopplad i ett system med en värmeväxlare, se figur 3.21 på sidan 27. Om man jämför figur 3.20 med figur 3.17 ser man en klar skillnad mellan de båda fallen.

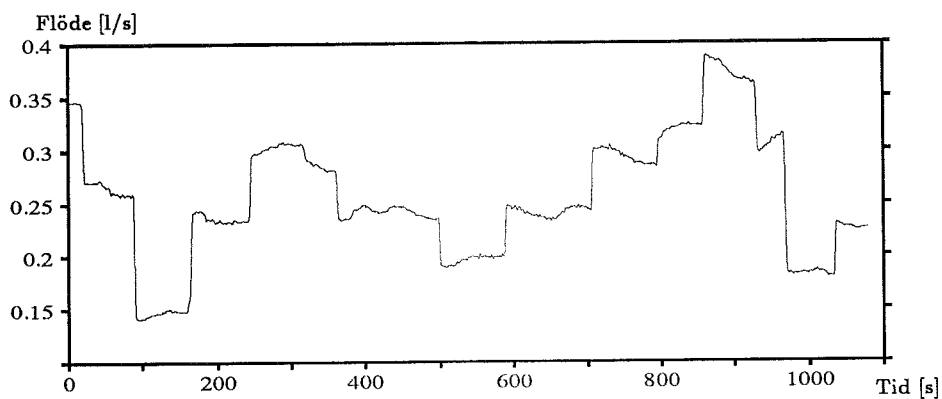


Figur 3.20 Flödet genom reglerventilen AVTQ20. Reglerventilen är här inkopplad i ett system med en värmeväxlare. Helt dragen linje är simulerat flöde och prickat är uppmätt.

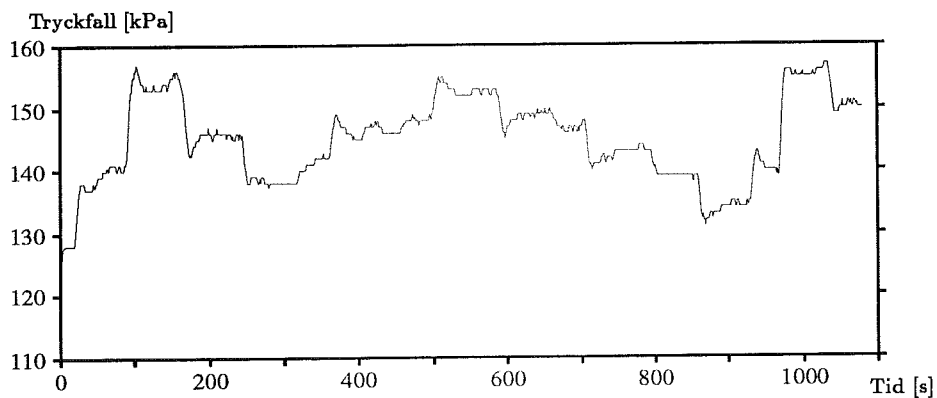
Tappvattenreglering med differenstryckshållare



Figur 3.22 Uppmätta och simulerade utloppstemperaturer.



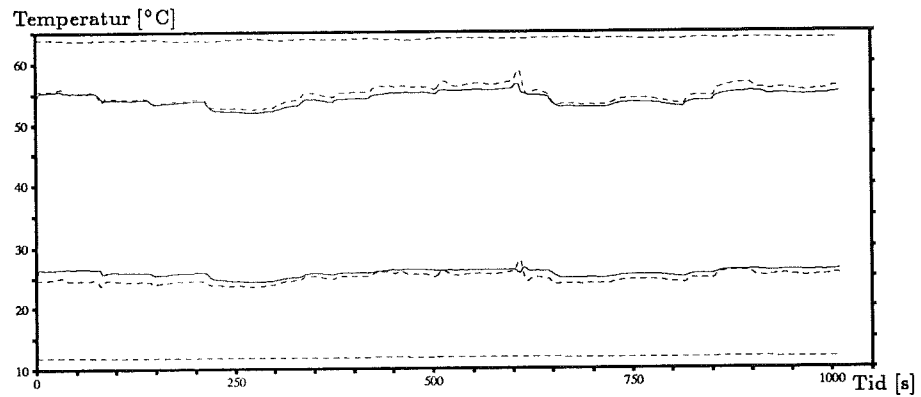
Figur 3.23 Det varierande tappvattenflödet.



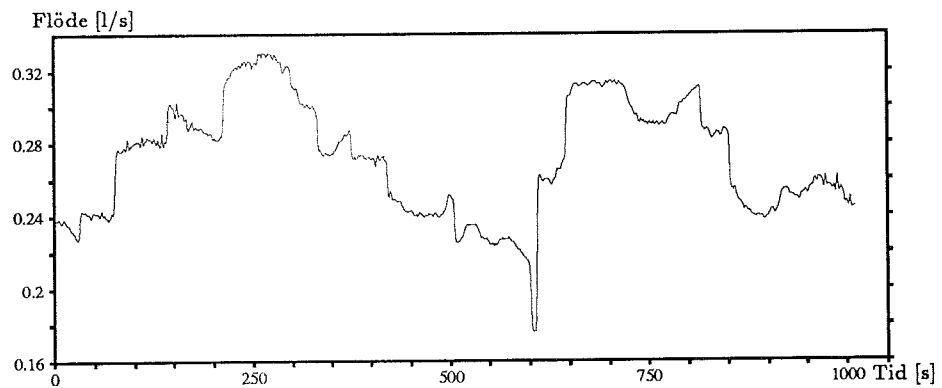
Figur 3.24 Det totala trycket över primärsidan.

Den simulerade tappvattentemperaturen följer den verkliga ganska bra. Modellens dynamiska egenskaper stämmer bra med verkligheten men det finns ett litet stationärt fel, vilket kan bero på felplacerade temperaturgivare.

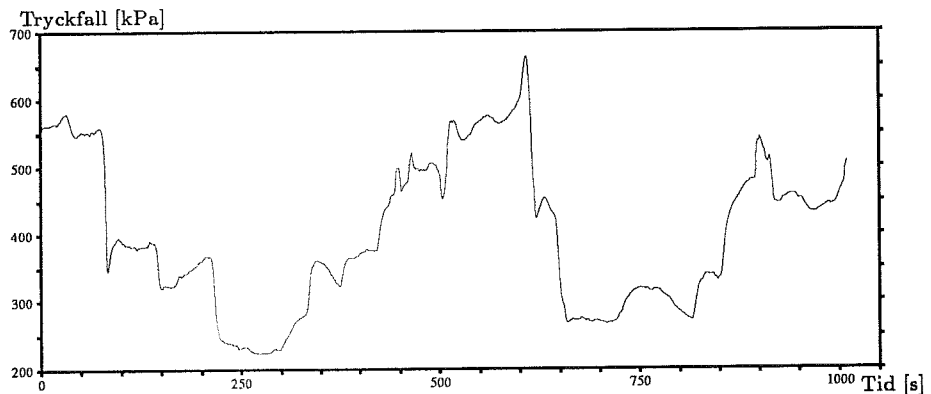
Tappvattenreglering utan differenstryckshållare



Figur 3.25 Uppmätta och simulerade utloppstemperaturer.



Figur 3.26 Det varierande tappvattenflödet.



Figur 3.27 Det totala trycket över primärsidan.

I det här fallet stämmer de simulerade värdena också bra med verkligheten. Eftersom flödesändringarna av tappvatten inte är lika stora som i föregående fall är det svårt att jämföra de två verifieringarna med varandra.

3.5 Sammanfattning

Verifieringarna visar att modellen av värmeväxlaren stämmer bra överens med verkligheten. Den fungerar bra i ett brett område av varierande temperaturer och inflöden. En begränsning i modellen är att den i sin nuvarande form inte klarar av flöden nära noll. Det beror bl a på att i modellen som beskriver värmeväxlaren finns det uttryck där flödet förekommer i nämnaren, med numeriska problem som följd. Dessutom är modellen för värmeövergången i värmeväxlaren inte giltig för flöden nära noll.

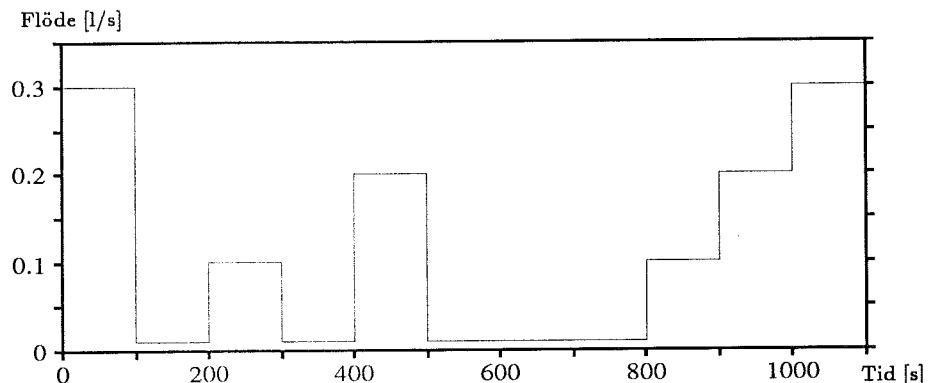
Modellen av reglerventilen ser ut att stämma ganska bra överens med verkligheten då man kopplar in den i ett system. Om den däremot är fristående skiljer sig modellens uppförande relativt mycket från det verkliga då tryckfallet över ventilen varierar. När hela tappvattensidan simuleras blir resultatet avsevärt bättre. Detta beror på att reglerventilen utgör en återkoppling i modulen och återkopplingar brukar ha den egenskapen att de minskar känsligheten för modellfel och olinjäriteter. Alltså är systemet inte lika känsligt för om det ligger ett litet eller högt tryck över primärsidan. Däremot är reglerventilen i sig, då den utgör ett eget system, ganska känslig för vilket tryck den har över sig.

4. Dynamisk analys

Simuleringsbiblioteket utgör ett utmärkt verktyg för att undersöka olika dynamiska förlopp i de termiska komponenterna samt i hela moduler. I kapitlet presenteras ett tappvattenprogram, vilket kommer att användas i analysen. Variationen av väggtemperaturen genom värmeväxlaren vid flödesändringar kommer vidare att undersökas. En undersökning om vad man kan göra för att förbättra temperaturregleringen och reducera returvattnets temperatur på primärsidan kommer också att göras.

4.1 Ett tappvattenprogram

För att kunna genomföra realistiska och reproducerbara simuleringar fastställs ett tappvattenprogram, vilket används som insignal på tappvattenflödet. Tappvattenprogrammet består av ett antal flödesändringar från i stort sett noll upp till 0.3 l/s . Flödesändringarna är rena stegändringar, vilket ungefär motsvarar verkliga flödesändringar av tappvattnet i en abonnentcentral som sitter i en enfamiljsvilla.

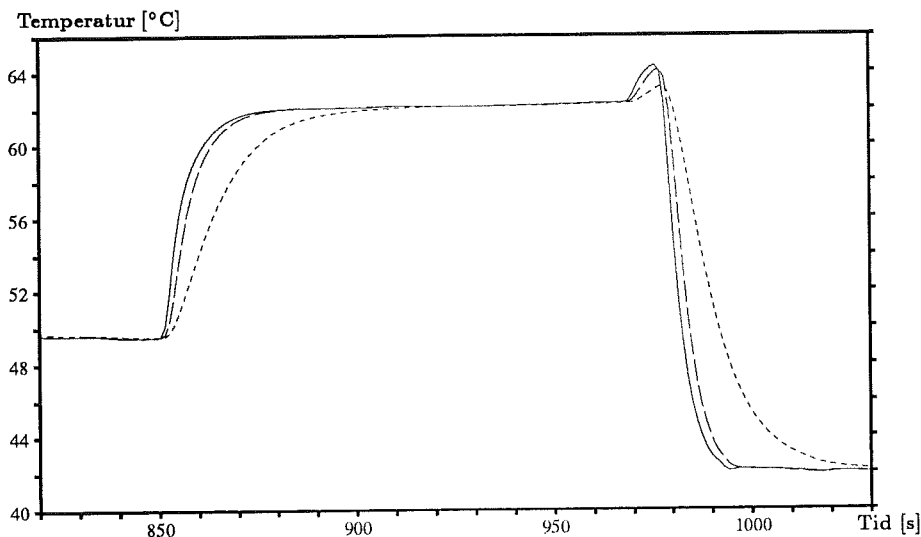


Figur 4.1 Variationer av tappvattenflödet.

I de flesta analysexperiment som utförts har det ovan föreslagna tappvattenprogrammet använts, men med den begränsningen att lägsta flödet är 0.1 l/s . Anledningen är att modellen av värmeväxlaren inte fungerar för flöden nära noll. Idealiskt skulle givetvis variationer ner till noll vara.

4.2 Undersökning av temperaturgivarna

Under mätningarna har två olika typer av temperaturgivare använts. Dels PT 100 med en spets på 6 mm i diameter dels med 3 mm spets. För temperaturgivarna med en spets på 6 mm blev modellkonstanterna $T_k = 9.1$ sekunder och $K = 1.0$, se ekvation (2.35) på sidan 14. T_k och K för temperaturgivarna med en spets på 3 mm blev 1.8 sekunder respektive 1.0. Efter att ha identifierat parametrarna i modellen för respektive temperaturgivare undersöktes hur mycket modelleringen av temperaturgivarna påverkar simuleringsresultatet, se figur 4.2 på nästa sida.



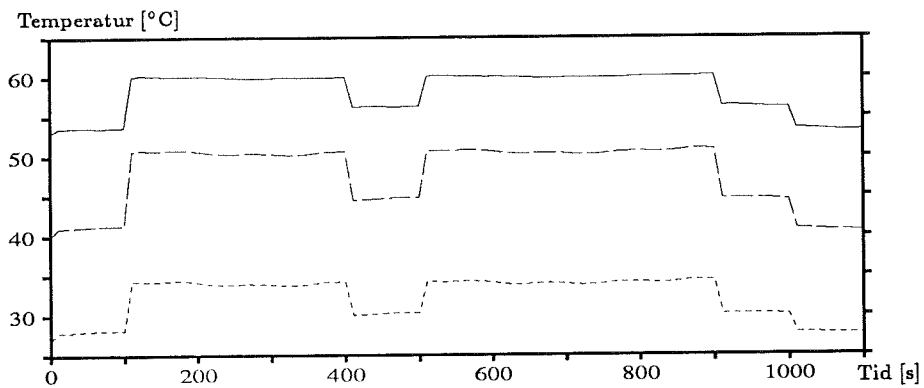
Figur 4.2 Jämförelse av uttemperaturerna i °C från värmeväxlaren vid simulering med respektive utan modellering av temperaturgivarna. Den heldragna linjen visar simulerade värden utan modellering av givarna. Prickad linje är med modell av PT 100; 6mm och streckad med modell av PT 100; 3 mm.

Figur 4.2 visar att med en temperaturgivare PT 100 med 3 mm spets påverkas inte mätresultatet speciellt mycket av givarens dynamik. Om man däremot mäter med en PT 100 med 6 mm spets påverkar temperaturgivarens dynamik mätresultatet avsevärt. Vid större temperaturförändringar och snabbare system skulle skillnaden göra sig ännu mer gällande. För värmeväxlare med en storlek motsvarande CB50 och mindre bör man alltså använda sig av en temperaturgivare med samma snabbhet eller snabbare än PT 100 med 3 mm spets. Detta gäller på primär- och på sekundärsidans utgångar. Vid mätning av de övriga temperaturerna kan man använda en PT 100 med 6 mm spets eftersom temperaturförändringarna där är långsammare.

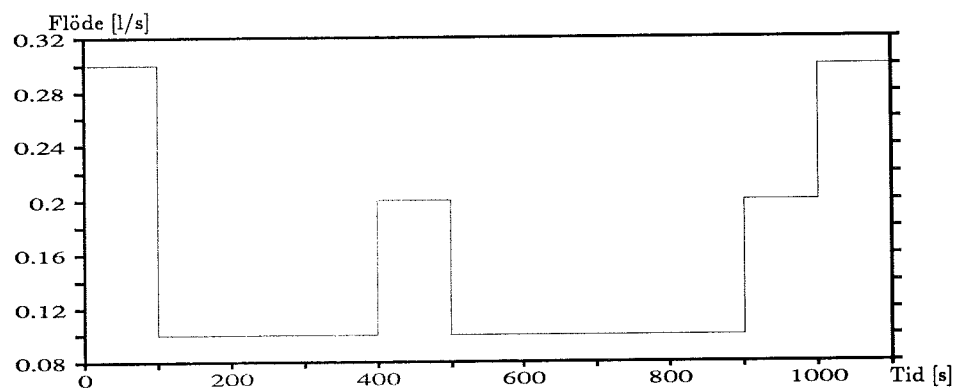
4.3 Väggtemperaturer i värmeväxlaren

Här undersöks väggtemperaturens variationer genom värmeväxlaren för hela modellen av TS30. Två experiment presenteras, ett då trycket över primärsidan varierar sinusformat kring 1.5 bar och ett då det varierar kring 5 bar. Försöken utföres för ett sommarfall, dvs då temperaturen på fjärrvärmewattnet ligger runt 65 °C och kallvattnets temperatur är ungefär 15 °C. Hur pass väl de simulerade väggtemperaturerna stämmer med de verkliga är svårt att säga, eftersom det är svårt att utföra någon realistisk verifiering. De modeller som har använts bör emellertid ge en tillräckligt bra beskrivning av verkligheten, för att belysa de problem som finns och vad man eventuellt kan göra för att rätta till dem.

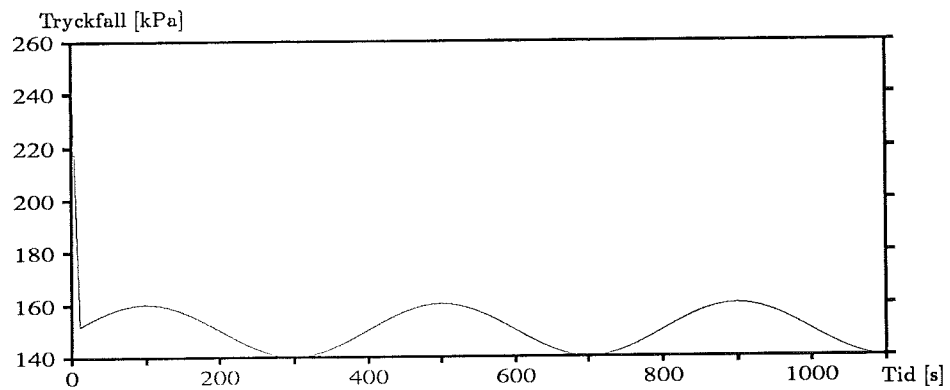
Experiment med 1.5 bar över primärsidan



Figur 4.3 Väggtemperaturer genom sektion ett till tre i värmeväxlaren. Den heldragna kurvan visar väggtemperaturen i sektion ett, den streckade kurvan visar temperaturen i sektion två medan den prickade visar den tredje sektionens väggtemperatur.

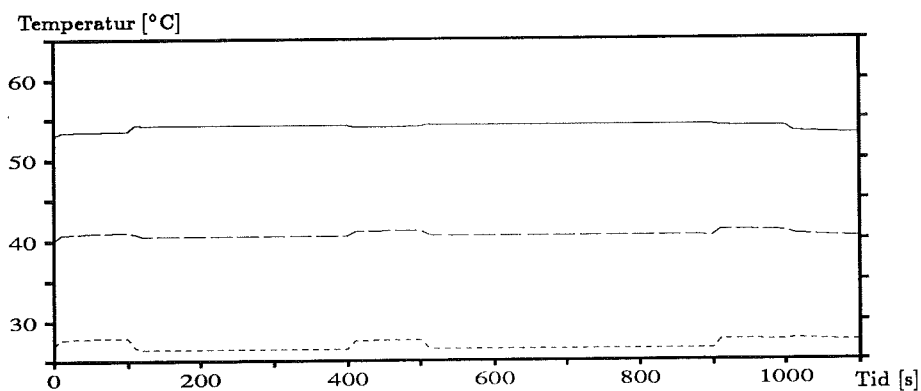


Figur 4.4 Variationer av tappvattenflödet.

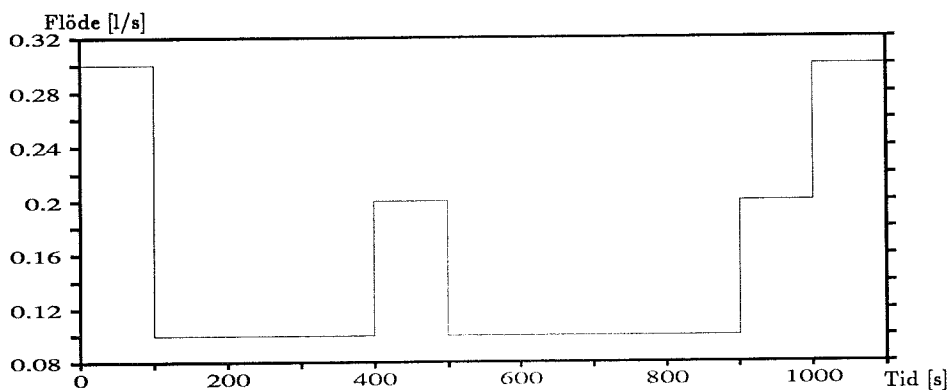


Figur 4.5 Variationer i tryckfallet över primärsidan.

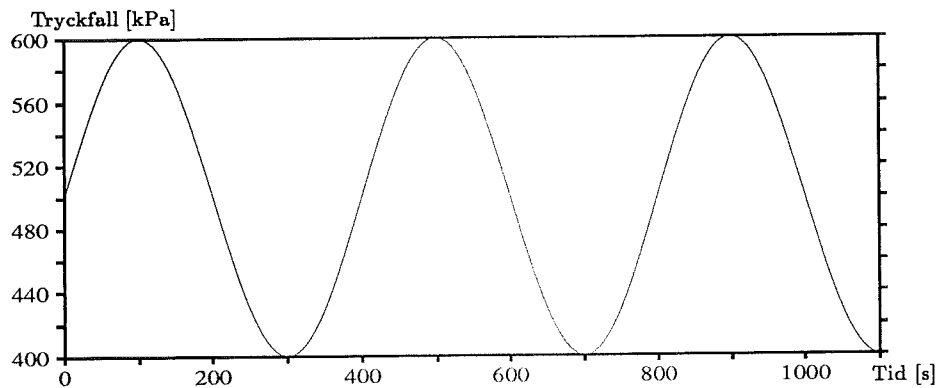
Experiment med 5 bar över primärsidan



Figur 4.6 Väggtemperaturer genom sektion ett till tre i värmeväxlaren. Den heldragna kurvan visar väggtemperaturen i sektion ett, den streckade kurvan visar temperaturen i sektion två medan den prickade visar den tredje sektionens väggtemperatur.



Figur 4.7 Variationer av tappvattenflödet.



Figur 4.8 Variationer i tryckfallet över primärsidan

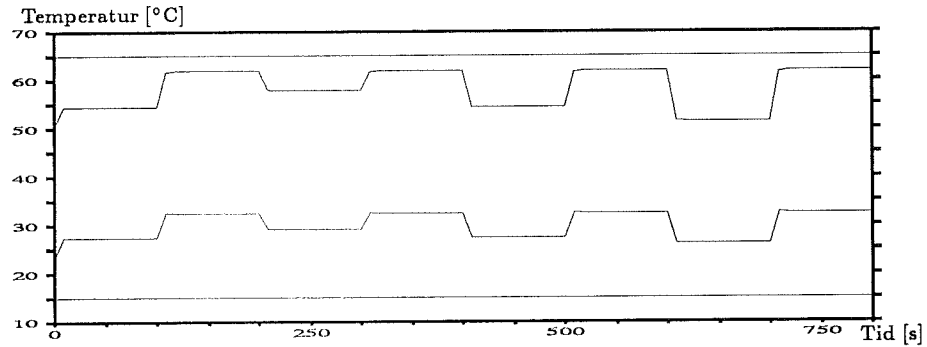
Av de ovan beskrivna experimenten kan man dra följande slutsatser:

- Yttemperaturen är högst i den sektion som ligger närmast inloppet av fjärrvärmevättnet.
- Kraftiga sänkningar av tappvattenflödet resulterar i temperaturtoppar.
- Det är genomgående högre yttemperaturer då det ligger ett lågt tryck över primärsidan.

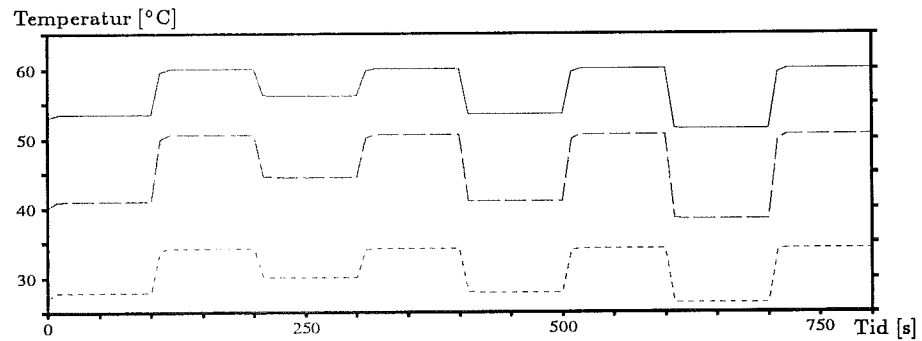
4.4 Jämförelse mellan sommar- och vinterfall

I experimentet jämförs regleringens snabbhet och väggtemperaturer för ett sommar- och ett vinterfall. Sommarfallet ser ut enligt ovan. Vid vinterfallet är temperaturen på fjärrvärmevattnet runt 100 °C och kallvattnets temperatur är ungefär 5 °C.

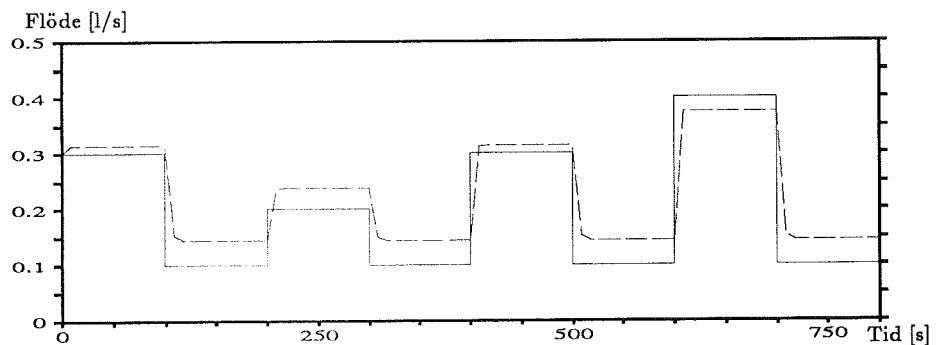
Sommarfallet med 1.5 bars tryck över primärsidan



Figur 4.9 Temperaturregleringen vid sommarfallet. Det är konstanta inloppstemperaturer på både primär- och sekundärsidan, 65 °C respektive 15 °C. Den övre varierande kurvan visar temperaturen på tappvattnet medan den nedre visar fjärrvärmevattnets returtemperatur.

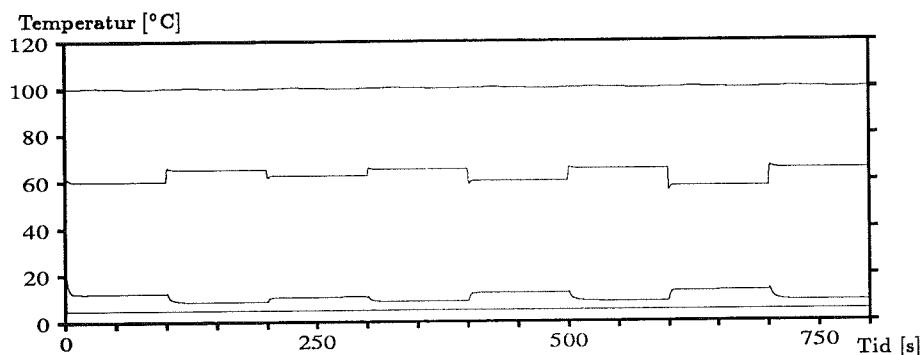


Figur 4.10 Väggtemperaturer genom sektion ett till tre i värmexlaren. Den heldragna kurvan visar väggtemperaturen i sektion ett, den streckade kurvan visar temperaturen i sektion två medan den prickade visar den tredje sektionens väggtemperatur.

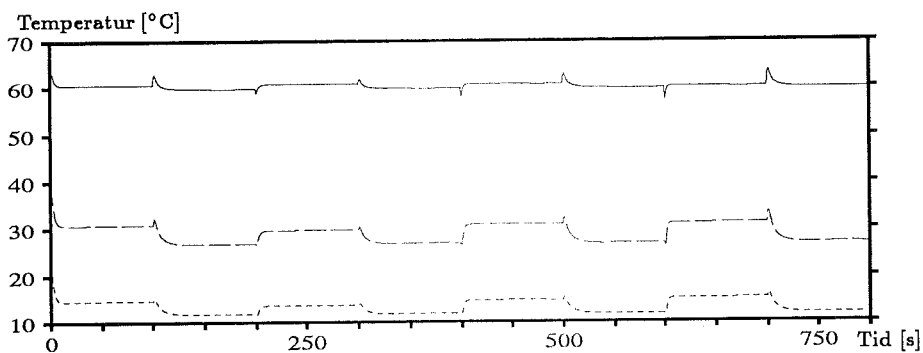


Figur 4.11 Variationerna i tappvattenflödet visas av den heldragna linjen. Den streckade linjen visar fjärrvärmevattnets flöde.

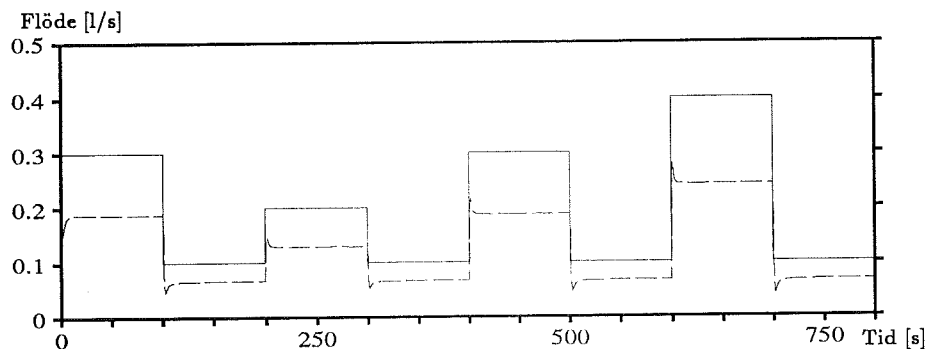
Vinterfallet med 1.5 bars tryck över primärsidan



Figur 4.12 Temperaturregleringen vid vinterfallet. Det är konstanta inloppstemperaturer på både primär- och sekundärsidan, 100 °C respektive 5 °C. Den övre varierande kurvan visar temperaturen på tappvattnet medan den nedre visar fjärrvärmevattnets returtemperatur.



Figur 4.13 Väggtemperaturer genom sektion ett till tre i värmeväxlaren. Den heldragna kurvan visar väggtemperaturen i sektion ett, den streckade kurvan visar temperaturen i sektion två medan den prickade visar den tredje sektionens väggtemperatur.

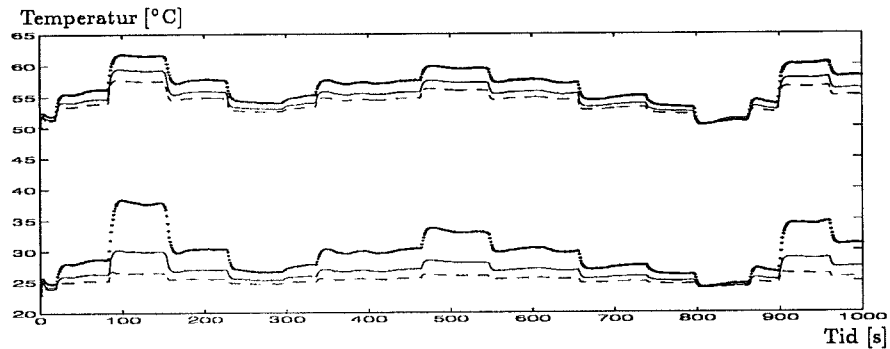


Figur 4.14 Variationerna i tappvattenflödet visas av den heldragna linjen. Den streckade linjen visar fjärrvärmevattnets flöde.

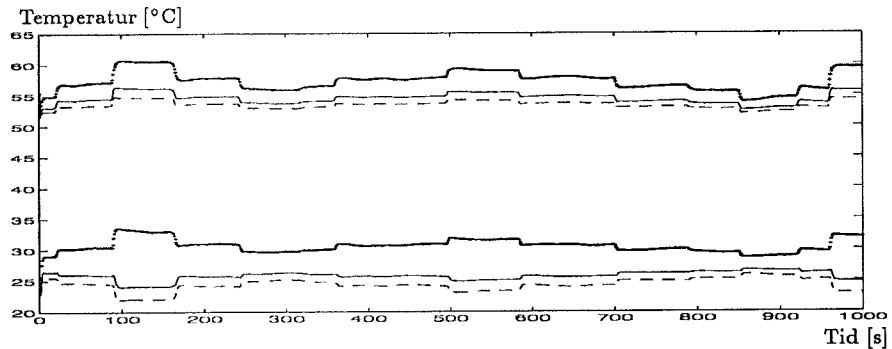
Vid vinterfallet är regleringen snabbare. Det beror på att fjärrvärmevattnets temperatur är högre än för sommarfallet. Vid vinterfallet är det däremot större risk att få överslängar vid regleringen. Överslängarna ser vi tydligast på fjärrvärmevattnets flöde.

4.5 Reglerventilens förstärkning

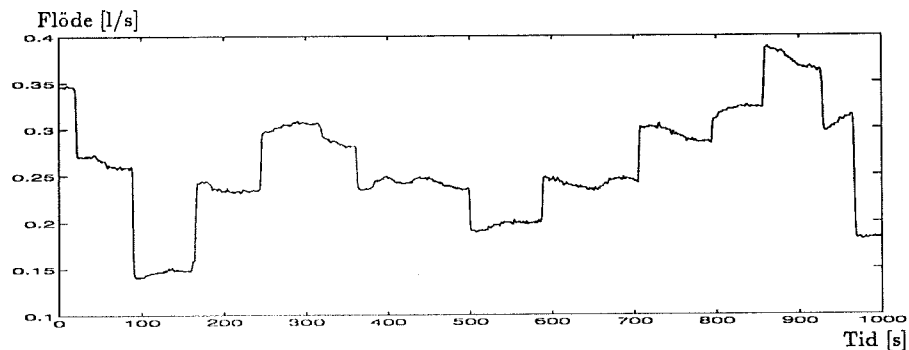
I det här avsnittet undersöks hur förstärkningen från tappvattentemperaturen till ventilläget x , se figur 2.2 på sidan 10, i reglerventilen påverkar systemets uppförande. Simuleringar för tre olika förstärkningar $K = 0.4, 0.8$ och 1.1 i reglerventilen har undersökts. Enligt verifieringar är det verkliga värdet på förstärkningen 0.8 . Simuleringar har genomförts med ett tryckfall över primärsidan på 1 respektive 5 bar. Försöken ska representera fallen med och utan differenstryckshållare på primärsidan. Inloppstemperaturen är 69°C på primärsidan och 15°C på sekundärsidan.



Figur 4.15 Tappvattentemperaturen och returtemperaturen för tre olika förstärkningar. De tjockare heldragna kurvorna representerar $K = 0.4$, de något smalare heldragna $K = 0.8$ och de streckade $K = 1.1$. Tryckfallet över primärsidan är 1 bar.



Figur 4.16 Tappvattentemperaturen och returtemperaturen för tre olika förstärkningar. De tjockare heldragna kurvorna representerar $K = 0.4$, de något smalare heldragna $K = 0.8$ och de streckade $K = 1.1$. Tryckfallet över primärsidan är 5 bar.



Figur 4.17 Variationer av tappvattenflödet för de båda mätningarna.

Om tappvattentemperaturerna i figurerna 4.15 och 4.16 jämförs, ser man att med ett lägre tryckfall på primärsidan blir det högre temperaturtoppar än med ett högre tryckfall. Det beror på att trycket påverkar reglerventilens totala fjäderkonstant. Fjäderkonstanten ökar med trycket, därför blir det mindre svängningar i fallet med det höga tryckfallet. Man ser också att regleringen blir minst svängig med förstärkningen 1.1 i båda fallen. Det är speciellt märkbart då trycket över primärsidan är lågt. Detta visar att det kanske vore önskvärt med en högre förstärkning än 0.8, vilket är fallet för AVTQ20.

4.6 Slutsatser

Följande erfarenheter har vunnits i analysarbetet:

- På snabba system som t ex TS30 är det mycket viktigt att kontrollera mätgivarnas och mätutrustningens snabbhet.
- Dödtiden i TS30 är i stort sett noll. Det beror på framkopplingsdelen i reglerventilen AVTQ20.
- Väggtemperaturen i värmeväxlaren är högst nära inloppet av fjärrvärmevattnet, här ligger den mellan 50° C och 60° C. Väggtemperaturen är lägre då det ligger ett högt tryck över primärsidan. Systemet är speciellt känsligt för kraftiga sänkningar av tappvattenflödet i vinterfallet, med temperaturtoppar som följd.
- Med en högre förstärkning i reglerventilen får vi en bättre reglering. Det är speciellt märkbart vid lågt tryckfall över primärsidan.
- Ett högt tryck över primärsidan medför en snabbare och mer noggrann reglering av tappvattentemperaturen. Eftersom producenter av fjärrvärmevattnet inte garanterar ett högre tryck än 1 bar på primärsidan, är det med en högre förstärkning i ventilen som regleringen kan förbättras.

5. Utprovningensrutin

I det här kapitlet kommer ett dynamiskt provprogram för nykonstruerade abonnentcentraler att presenteras. Utprovningensrutinen är skriven så att Alfa Laval Thermal ska kunna använda den vid en dynamisk utprovning av nykonstruerade fjärrvärmesystem. Den statiska provdel som Alfa Laval Thermal har använt sig av tidigare har inte undersökts närmare, vi föreslår att den behålls i sin nuvarande form.

5.1 Förutsättningar för lyckade provresultat

För att det över huvudtaget ska gå att utreda om en konstruktion uppfyller vissa krav, eller om en jämförelse mellan olika moduler ska vara möjlig måste man kunna genomföra reproducerbara mätningar. Någon form av sådana mätningar går inte att utföra i Alfa-Laval Thermals HTC laboratorium idag. Det första de ansvariga för utprovningen måste införa är alltså möjligheterna till att genomföra mätningarna under nästan exakt samma betingelser vid varje mättillfälle. Under denna punkt faller införandet av ett tappvattenprogram, dvs en serie flödesändringar av tappvattnet. Det är alltså en sekvens av exakt samma flödesändringar man skickar in till systemet vid varje mätning. Ett annat problem är att man idag inte kan genomföra mätningar med ett konstant tryck över primärsidan.

5.2 Införandet av ett tappvattenprogram

Alfa-Laval Thermal borde kunna utnyttja den mät dator som finns på laboratoriet för att skicka ut styrsignaler till någon form av elektriskt ställdon och på så sätt reglera tappvattenflödet. fördelarna med en sådan lösning är många:

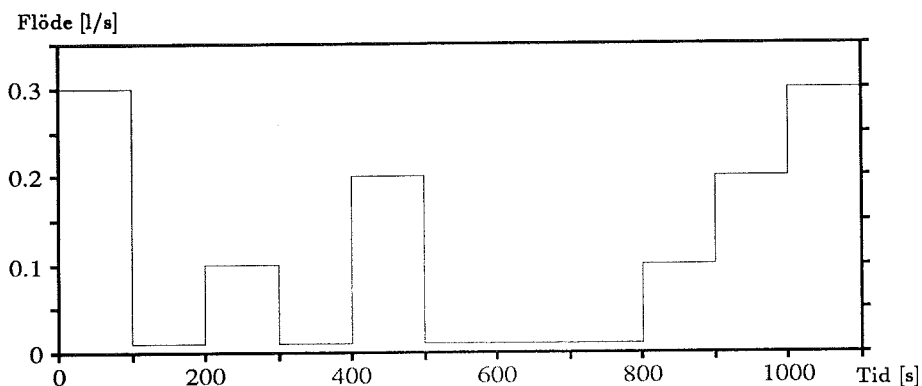
- Befintlig utrustning utnyttjas med undantag av det elektriska ställdonet.
- Möjligheten att få exakta värden på tappvattenflödet.
- Möjligheten att få flödesändringarna vid väl definierade tidpunkter.
- Flexibiliteten i mjukvaran gör det möjligt att välja i stort sett vilka tappvattenflöden som helst och vid vilka tidpunkter flödesändringarna ska inträffa.

Alternativet till en sådan lösning på problemet är att använda en tappvattenkran där det är möjligt att göra väldefinierade stegändringar av flödet. För att få flödesändringarna vid någorlunda väl definierade tidpunkter kommer man vara tvungen att mäta tiden med klocka och manuellt ändra flödet.

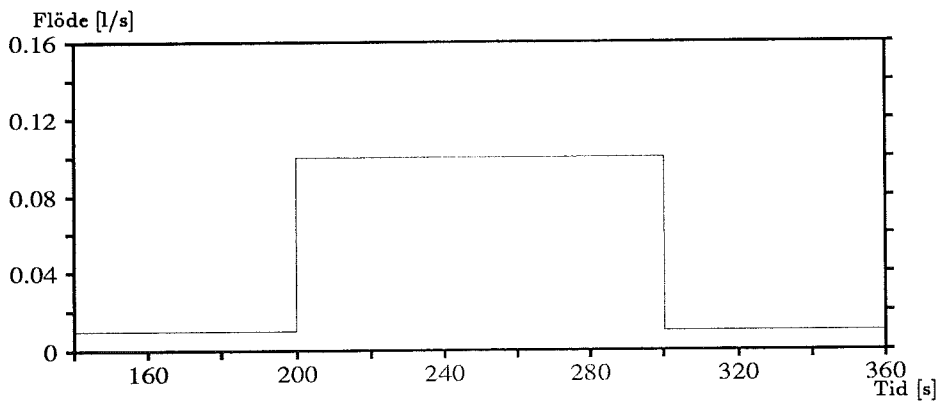
Egentligen är inte den senare lösningen något alternativ. Anledningen är att det borde gå att få ett så pass mycket bättre resultat med mät datorn och ett elektriskt ställdon, att det inte borde vara någon tvekan om vilken lösning som ska väljas.

Förslag på tappvattenprogram

Med den ovan föreslagna lösningen, för att få reproducerbara mätningar, kan man konstruera en mängd olika tappvattenprogram. Det kan ju vara bra att ha ett litet bibliotek av tappvattenprogram att välja mellan, då man vill undersöka eller påvisa olika saker. Vi kommer här att ge förslag på två program för olika tillämpningar.



Figur 5.1 Tappvattenprogram för analys av regleringen.



Figur 5.2 Tappvattenprogram för att bestämma stigtiden vid ökande respektive minskande flöde.

Tappvattenprogrammet i figur 5.1 kan användas till att undersöka regleringen i olika moduler. Figur 5.2 visar ett tappvattenprogram med endast en stegändring av tappvattenflödet. Med hjälp av det kan stigtiderna för olika moduler och komponenter undersökas.

5.3 Konstant tryck över primärsidan

I en utprovning rutin är det önskvärt att prova konstruktionen för ett antal olika tryck över primärsidan. Anledningen är att trycket på fjärrvärmevattnet varierar ganska mycket med var man befinner sig rent geografiskt. Det kan dessutom finnas en viss tryckvariation på varje plats. Ett rimligt utprovningsområde är från en och ända upp till sex bars tryck. Vid varje trycknivå är det rimligt att tänka sig en variation upp till en bar.

För att utföra en mätning i laboratoriet måste man ha någon form av hjälp med att hålla ett någorlunda konstant tryck över primärsidan. Vid de mätningar som gjorts har en differenstryckshållningsventil, AVP20 från Danfoss A/S, använts för att få ett konstant tryck. Problemet är att AVP20 bara klarar av att hålla ett konstant tryck upp till 1.5 bar. Man måste alltså investera i någon ventil som klarar av att hålla ett konstant tryck ända upp till sex bar.

5.4 En dynamisk utprovning rutin

Ett dynamiskt utprovning program för tappvattensidan i en abonnentcentral kommer här att presenteras. Programmet kan givetvis modifieras beroende på hur noggrann analys provbeställaren vill göra på mätobjektet.

Utprovningfallen

Vi föreslår att en noggrann utprovning ska innehålla sex olika utprovningfall. Man provar först ett sommarfall, dvs med en temperatur på fjärrvärmevattnet av 65 °C. På sommarfallet provas konstruktionen med tre olika tryckfall över primärsidan. Efter detta provar man ett vinterfall, dvs med en temperatur på fjärrvärmevattnet av 100 °C. Även här provas konstruktionen med tre olika tryck över primärsidan.

Provfall	Temperatur [°C]	Tryckfall [bar]
1	65	1.5
2	65	3.0
3	65	6.0
4	100	1.5
5	100	3.0
6	100	6.0

Provanalys

I samtliga provfall studeras det dynamiska uppförandet med ett för den aktuella studien lämpligt tappvattenprogram. I den dynamiska analysen bör utprovaren för varje provfall utföra följande undersökningar:

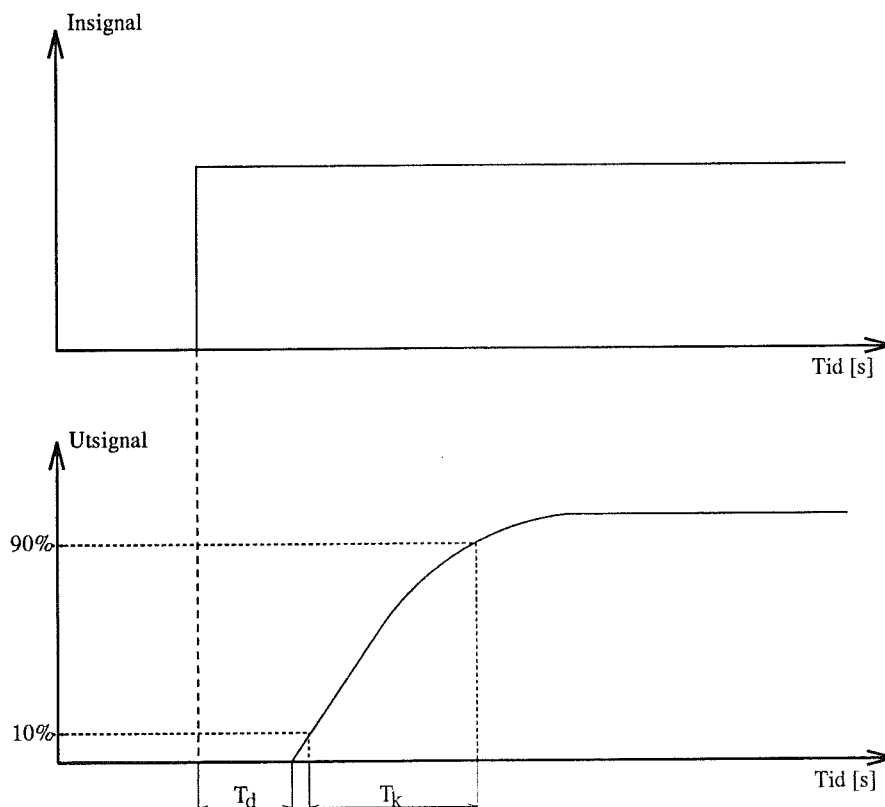
- Hur noggrann temperaturregleringen är kring referenstemperaturen.
- Returvattnets temperatur.
- Snabbheten i regleringen.

Undersökning av regleringens snabbhet

Ett bra sätt att undersöka regleringens snabbhet är att mäta systemets stigtid och dödtid. Det görs lämpligen genom en stegändring av tappvattenflödet, se figur 5.2. Man kan t ex göra tre försök där man gör en stegändring av tappvattenflödet med 0.1, 0.2 och 0.3 l/s. För varje försök mäter man stigtiden både för ökning och strypning av tappvattenflödet. Hur stigtiden mäts beror på vilken definition av stigtiden man vill använda. Användande av följande definition ges som förslag.

Tiden det tar för utsignalen att gå från 10% till 90% av den totala utsignalsförändringen, när man gör en stegändring av insignalen.

I figur 5.3 ges ett exempel på en förändring av utsignalen som följd av en stegändring på en insignal. Stigtiden betecknas med T_k .



Figur 5.3 Schematisk bild över definitionen av begreppen stigtid och dödtid.

På tappvattensidan i en abonnentcentral kan insignalen utgöras av tappvattenflödet och utsignalen kan vara temperaturen på tappvattnet.

Dödtiden T_d påverkar även den, i allra högsta grad resultatet av regleringen. Dödtiden anger den tid det tar för systemet att upptäcka en förändring i insignalerna och till dess att en reglering påbörjas. Som ett direkt mått på hur besvärligt ett objekt är att reglera kan man använda kvoten T_d/T_k . Om kvoten är liten bör regleringen vara ganska enkel, om kvoten däremot är stor kommer det med all sannolikhet att bli betydligt svårare att åstadkomma en bra reglering. Att ha en liten dödtid är speciellt viktigt då vi har mycket snabba system, dvs ofta små system som tex en abonnentcentral avsedd för en villa.

Man strävar hela tiden efter att få så effektiva system som möjligt, dvs man vill bli att stigtiderna ska vara så små som möjligt. För att få ett litet värde på T_d/T_k måste man alltså se till att dödtiden i systemet blir minimal. Som konstruktör av fjärrvärmemoduler kan man påverka dödtiden genom att placera de mätgivare som är kopplade till reglerutrustningen, så att de reagerar så snabbt som möjligt på en förändring av insignalerna. Konstruktören bör även föra en diskussion med leverantörerna av reglerutrustningen, där man gemensamt arbetar för att korta ned dödtiden i systemen. För att konstruktören på ett effektivt sätt ska kunna ställa krav på leverantörerna bör man på något sätt kunna påvisa skillnader i dödtider då man använder olika typer och fabrikat av reglerutrustning. Ett bra sätt att mäta och presentera dödtider för fjärrvärmeställningar är att mäta den tid det tar innan en ändring på tappvattenflödet resulterar i en ändring av fjärrvärmeflödet.

5.5 Presentation av mätresultat

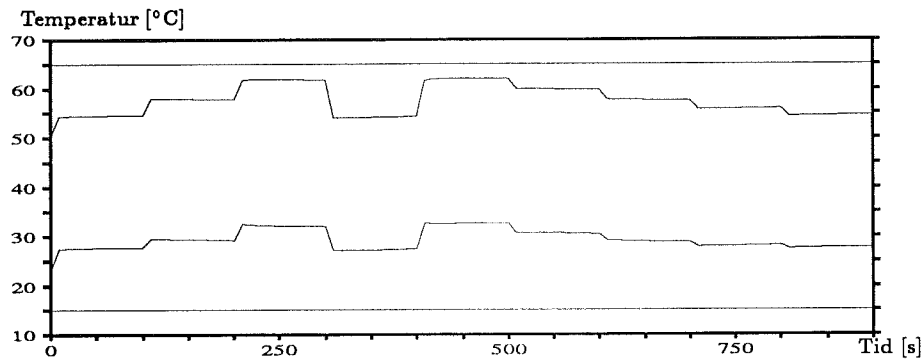
Två förslag kommer att ges på hur en presentation kan se ut av de resultat och slutsatser utprovaren kommit fram till. Ett presentationsförslag kommer att ges för en intern presentation till väl insatta personer, tex produktutvecklare och konstruktörer. Det andra presentationsförslaget riktar sig till personer som arbetar med marknadsföring av de utprovade produkterna.

Intern presentation

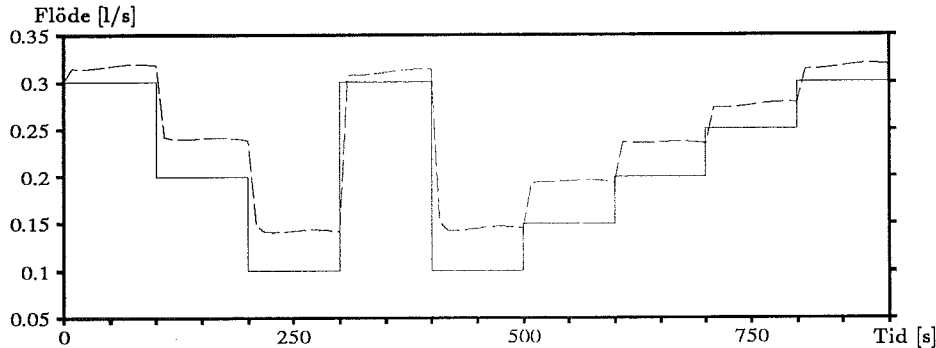
Den interna presentationen syftar till att ge en utförlig information om systemets dynamiska uppförande vid olika driftförhållanden. Informationen vänder sig i första hand till anställda på utvecklingsavdelningar och konstruktörer, vilka i sin tur med hjälp av provresultatet har lättare att formulera önskemål och krav till leverantörer. Presentationen bör innehålla en genomgång av samtliga sex provfall, där man dels visar grafer över temperaturreglering dels presenterar värden på uppmätta stigtider och dödtider.

Vi har simulerat model TS30 för de sex provfallen. För varje simulering har stigtiderna och dödtiden mätts upp. För att visa hur en presentation i princip kan sammanställas, redovisas en graf över temperaturregleringen och en tabell med stigtider och dödtid. Vid en verklig redovisning av resultaten ska det givetvis finnas med grafer över temperaturregleringen för samtliga provfall.

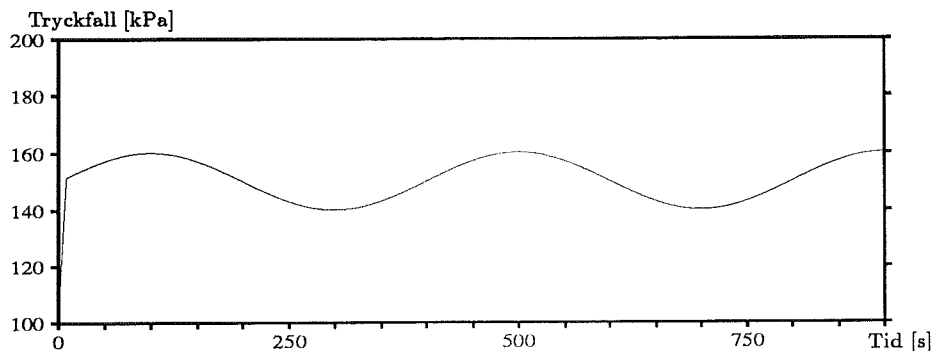
En lämplig uppläggning på presentationen av graferna vid varje provfall visas nedan. Den översta grafen visar temperaturregleringen, under den visas flödesändringarna som ger upphov till variationen i temperaturerna. Den understa grafen visar tryckvariationerna över primärsidan vid försöket. Inloppstemperaturerna och trycket över primärsidan har vi låtit variera något eftersom de i verkligheten inte är helt konstanta. Med grafernas hjälp får man en snabb och översiktlig bild över hur bra modulens prestanda är för det aktuella provfallet.



Figur 5.4 Temperaturregleringen vid provfall 1.



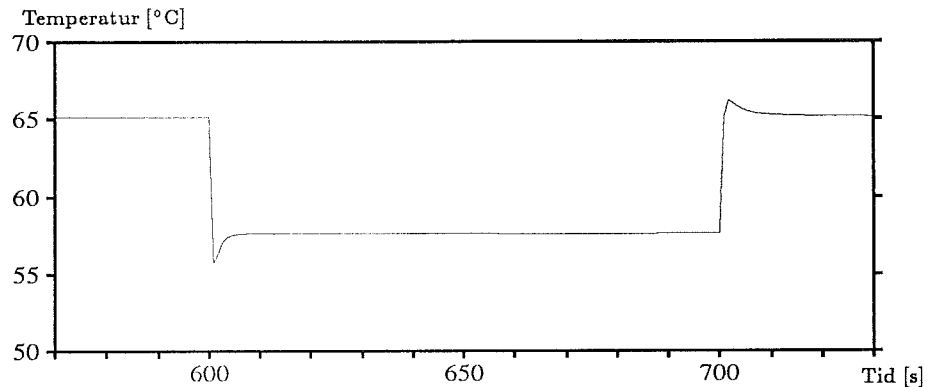
Figur 5.5 Den heldragna kurvan visar ändringarna i tappvattenflödet medan den streckade linjen visar det resulterande flödet på fjärrvärmeverket.



Figur 5.6 Tryckfallet över primärsidan antas variera sinusformat runt 1.5 bar.

För en mer noggrann analys av prestandan i en modul, används i utprovningsprogrammet begreppen stigtid och dödtid. De uppmätta värdena av dem bör man presentera i tabellform. I samtliga utprovningsfall har dödtiden varit så liten att det är svårt att genomföra tillräckligt noggranna mätningar. Vi har då inte funnit det meningsfullt att försöka mäta den och konstaterar bara att den är mycket liten. Storleken på dödtiden beror i vårt fall till stor del på reglerventilen AVTQ 20. Det är främst framkopplingsdelen i ventilen som är anledningen till den korta dödtiden. Man kan alltså konstatera att Danfoss A/S har lyckats väldigt bra med den delen av konstruktionen.

Under simuleringen av de tre sista provfallen, dvs de som berör vinterfallet, visade det sig att regleringen var mycket snabb. En mätning av stigtiderna i de här fallen hade givit ganska osäkra resultat, varför vi nöjer oss med att konstatera att de är mycket små. Däremot upptäcktes ganska kraftiga överslängar vid regleringen av tappvattentemperaturen, se figur 5.7.



Figur 5.7 Exempel på överslängar vid temperaturregleringen av vinterfallet.

Överslängarna är inte bra, dels är det en dålig reglering dels ökar risken för kalkavlagringar på värmväxlarens värmeöverförande ytor. I de fall vi har undersökt är inte överslängarna överdrivet stora. Men man bör vara vaksam på fenomenet. Då överslängarna är mer störande bör det ställas krav på förbättringar av reglerutrustningen.

Ett förslag på hur en tabell med stigtider och dödtider kan byggas upp visas på nästa sida. Stigtiderna betecknas $T_{k,i,d}$, där $i = 1, 2$ eller 3 beroende på om stegändringen av tappvattenflödet är $0.1, 0.2$ eller 0.3 l/s. Beteckningen d har värdena p eller n beroende på om det är en positiv eller negativ ändring av flödet. Dödtiden betecknas T_d . Samtliga stigtider och dödtider har enheten sekunder. De värden som markeras med ett "-" är så små att vi inte har mätt upp dem.

Provfall	Tk _{1,p}	Tk _{1,n}	Tk _{2,p}	Tk _{2,n}	Tk _{3,p}	Tk _{3,n}	Td
1	5.4	6.8	3.9	6.4	3.4	5.3	-
2	4.7	6.4	3.8	5.7	3.0	5.2	-
3	4.4	6.4	3.2	5.4	2.7	4.7	-
4	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-

Vid en analys av resultaten konstateras följande:

- Stigtiderna minskar då trycket över primärsidan ökar, dvs reglering blir snabbare.
- Stigtiderna minskar då stegändringen av tappvattenflödet görs större.

Att stigtiden minskar med ökat tryck beror på att reglerventilens framkoppling då får en större inverkan. Att stigtiden minskar vid ökande stegändringar av tappvattenflödet kan tyckas ganska underligt. Det beror bl a på att värmewäxlarens stigtid minskar med ökat flöde.

Extern presentation

Här vill man presentera en moduls prestanda i ett säljande och marknadsförande syfte. Det är viktigt att presentationen är översiktlig så att kunden snabbt och enkelt kan ta till sig den viktigaste informationen. Vi föreslår en grafisk presentation liknande den på sidan 44. Följande information bör en kund vara speciellt intresserad av:

- Noggrannheten i regleringen, dvs hur bra temperaturen på tappvattnet regleras kring referensvärdet.
- Dödtiden i systemet, dvs den tid det tar innan regleringen av flödet på primärsidan påbörjas efter en ändring av tappvattenflödet.
- Regleringens snabbhet.
- Returvattnets temperatur på primärsidan.

5.6 Sammanfattning

En förutsättning för att genomföra analyser på fjärrvärmemoduler är att det ska gå att genomföra reproducerbara mätningar. I kapitlet ges förslag på nödvändiga förändringar i Alfa Laval Thermals nuvarande utprovningprogram för att detta ska bli möjligt. Vidare ges ett förslag på ett dynamiskt utprovningprogram, där även en diskussion om vad analysen med avseende på noggrannhet och snabbhet i regleringen bör innehålla. Begreppen stigtid och dödtid samt deras påverkan av regleringen berörs också. Vidare innehåller kapitlet ett avsnitt om hur resultaten av utprovningen kan presenteras.

6. Simuleringspaket

6.1 Termiskt komponentbibliotek

Ett antal bibliotek av termiska komponenter har byggts upp i Omola. Komponenterna har en objektorienterad struktur som gör det mycket enkelt att använda dessa för att bygga upp mer komplicerade strukturer. Här kommer biblioteken i korthet att beskrivas. I appendix B finns en komplett dokumentation av programkoden. Biblioteken är följande:

BasicThermo innehåller ett antal modeller av grundkomponenter för termiska tillämpningar.

ThermoGenerators innehåller ett antal modeller som kan liknas vid pumpar för att generera temperatur, tryck och flöde. Man kan konstruera pumpar för generering av ett bestämt tryck och temperatur, flöde och temperatur samt med speciella tappvattenprogram.

Inputs innehåller modeller för att ta in mätdata vid verifiering.

HeatExchangers innehåller en modell av en sektion i en värmeväxlare. Biblioteket innehåller även en värmeväxlare sammansatt av tre enkla sektioner.

BasicMechanics innehåller terminaler för mekaniska tillämpningar.

AVTQ innehåller en modell av en reglerventil AVTQ och tillhörande styrventil.

MeasEquip innehåller modeller för att mäta temperatur, tryck och flöde.

Valves innehåller modeller av ventiler.

Controllers innehåller en modell av en PI-regulator.

6.2 Initialvärdesproblem vid simuleringar i OmSim

I alla simuleringsprogram är uppstarten av simuleringen ett kritiskt moment. Anledningen är att man måste initialisera tillståndsvariablerna med lämpliga värden för att ekvationslösarna ska komma igång. I mindre system behöver detta inte vara ett speciellt svårt problem, men i takt med att systemen blir mer komplexa och de ingående komponenterna blir mer komplicerade så ökar problemen. I de simuleringar vi har utfört på de enskilda komponenterna har det inte varit några problem med att hitta lämpliga initialvärden. Men då vi kopplar ihop flera komponenter har det visat sig att OmSim behöver oerhört exakta värden på tillståndsvariablerna för att komma igång med simuleringen. I de största simuleringarna som vi har kört är vi tvungna att sätta ungefär tjugo variabler initialt. Vid simuleringar av mer komplexa processer är det säkert inte någon ovanlighet att man är tvungen att sätta ett hundratal variabler för att starta simuleringarna. Ett exempel på en sådan process med nära

anknytning till vad vi har arbetat med är simuleringar av ett fjärrvärmenät.

Önskvärt vore att det fanns ett bra verktyg inbyggt i OmSim med vilken man får hjälp att sätta de initialvärden som behövs för att kunna starta en simulering med en rimlig arbetsinsats. I dag finns det tyvärr inget sådant verktyg. För att klara av initialvärdesproblemet har vi skrivit initieringsrutiner som hjälper till att starta upp simuleringen. Nedan följer en beskrivning av hur rutinen i princip fungerar och är uppbyggd.

Initieringsrutin

I vårt arbete har vi haft problem då vi vill specificera ett tryck på vattnet in till systemet, vilket vi i allmänhet gör på primärsidan. På sekundärsidan specificerar vi istället ett flöde på tappvattnet. Fallet på sekundärsidan är mycket lättare eftersom flödet är det samma genom alla komponenter. Det innebär att programmet utifrån givna ekvationer i modellerna kan räkna ut deltrycken över de enskilda komponenterna och summera ihop dem till ett totalt tryck på sekundärsidans ingångsvatten. På primärsidan vet man inte från början vilket flöde som cirkulerar utan bara det totala trycket över sidan. Det innebär att man inte kan räkna ut de olika deltrycken över de ingående komponenterna. Ekvationslösarna i simuleringsprogrammet måste alltså få hjälp med att räkna ut tryckfallet över varje komponent vid uppstarten av simuleringen.

En lösning på problemet är att sätta ett flöde initialt. Med hjälp av det flödet räknar man explicit ut deltrycken över varje komponent i initieringsrutiner som ligger i varje komponents modell. Slutligen summeras deltrycken i en initieringsrutin som ligger i modellen över hela uppkopplingen, här initierar man lämpligen flödet och de variabler som behövs för att räkna ut deltrycken i varje komponent. Nu har man en lösning där ekvationslösarna kan sätta igång att jobba. Problemet är bara det att det totala trycket som man får vid summering av komponenternas deltryck kanske inte stämmer överens med det tryck som man vill skicka in på primärsidan. Det problemet kommer man inte ifrån. Man löser detta genom att först räkna ut skillnaden mellan önskat intryck på pumpen och komponenternas deltryck, se ekvation (6.2). Därefter drar man ifrån den differensen multiplicerat med en avklingningsfaktor från det önskade trycket på pumpen, se ekvation (6.1). Avklingningstermen gör att vi ganska snabbt kommer att svänga in mot det önskade trycket över primärsidan.

$$p = p_0 - \Delta p \cdot e^{-\alpha t} \quad (6.1)$$

$$\Delta p = p_0 - \sum_{k=1}^i p_k \quad (6.2)$$

där

- p Totala trycket över primärsidan [Pa].
- p_0 Önskat tryck över primärsidan [Pa].
- α Avklingningskonstant.
- t Simuleringstiden [s].
- i Antalet komponenter i modulen.
- p_k Deltryck över komponent k [Pa].

Avslutningsvis kan man säga att OmSim är ett utmärkt verktyg för att utföra simuleringar av både enkla men också mer komplexa system. Detta beror bl a på att modelleringspråket Omola är ett objektorienterat språk.

7. Sammanfattning

Arbetet inleds med att modellera viktiga termiska komponenter som värmeväxlare, ventiler och pumpar. Programspråket som använts för att implementera modellerna kallas Omola, det är ett nytt objektorienterat och ekvationsbaserat språk som utvecklats på institutionen för Reglerteknik på Lunds Tekniska Högskola. De framtagna modellerna har verifierats mot mätningar som utförts på en enfamiljs abonnentcentral för fjärrvärme kallad TS30. Rapporten innehåller vidare ett analysavsnitt, där viktiga egenskaper i regleringen av TS30 undersökts. Som avslutning presenteras ett förslag på en dynamisk utprovningrutin för nykonstruerade abonnentcentraler för fjärrvärme.

Modelleringsresultat

Hela modelleringsarbetet bygger på en objektorienterad metod. Där begrepp som bibliotek, klasser och nedärvning av egenskaper är centrala.

- En bra modell av en värmeväxlare har implementerats. Den bygger på väldokumenterad teori inom området.
- Temperaturgivare kan modelleras som första ordningens linjära system, där tidskonstanten lätt bestäms i ett kalibreringsexperiment.
- För att kunna bygga modeller av mekaniska reglerventiler krävs en mycket god kännedom om hur ventilen internt är uppbyggd. De är mycket komplexa ur mekanisk synpunkt med många olinjäriteter. Dock kan man vid simulering av moduler få bra resultat eftersom variationer pga olinjäriteter och omodellerade effekter får mindre inverkan då ventilen sitter i ett återkopplat system. För undersökning av olika designar av ventilen behövs med all sannolikhet en bättre modell av ventilen.
- Arbetet med modelleringen har resulterat i en bra modell av abonnentcentralen TS30, där simulerade och uppmätta resultat stämmer mycket väl överens.

Analysresultat

En undersökning av abonnentcentralen TS30 med avseende på noggrannhet och snabbhet i regleringen har gjorts. Analyser angående temperaturgivarnas snabbhet och värmeväxlarens väggtemperatur har också utförts.

- Temperaturgivare som används i mätningar av snabba system som t ex TS30 bör ha en tidskonstant på två sekunder eller mindre. Stigtiderna i systemet TS30 ligger mellan ungefär fyra och tio sekunder, beroende på hur högt tryck det ligger över primärsidan samt hur stor belastningsökningen är.
- Dödtiden i systemet TS30 är i stort sett noll. Det beror på en framkoppling i konstruktionen av reglerventilen AVTQ20.
- Väggtemperaturen i värmeväxlaren är högst nära inloppet av fjärrvärmevattnet, här ligger den mellan 50 °C och 60 °C. Temperaturen är något lägre då det ligger ett högre tryck över primärsidan. Systemet är speciellt känsligt för kraftiga sänkningar av tappvattenflödet i vinterfallet, med temperaturtoppar som följd.
- Med en högre förstärkning i reglerventilen får vi en bättre reglering. Det är speciellt märkbart vi lågt tryckfall över primärsidan.

Erfarenheter vid utprovning

En dynamisk utprovningrutin för nykonstruerade värmeväxlersystem har tagits fram, den är skriven så att Alfa Laval Thermal ska kunna använda den i sitt HTC laboratorium. I utredningen diskuteras bl a nödvändigheten av reproducerbara mätningar. Provrutinen har arbetats igenom med simuleringar på villamodulen TS30. Följande erfarenheter har vunnits genom arbetet:

- Nödvändigheten av att utföra reproducerbara mätningar.
- Nödvändigheten av att kunna ställa in ett konstant tryck över primärsidan.
- Utprovaren bör koncentrera sig på regleringens snabbhet och noggrannhet. Viktiga begrepp är bl a stigtid och dödtid.
- Resultaten bör presenteras dels i grafiskt i form av kurvor, dels i tabellform där stigtider och dödtid för de olika utprovningfallen redovisas.

Möjliga utvidgningar

Vi är övertygade om att det finns väldigt mycket att vinna på att använda den här formen av simuleringsverktyg i ett brett område av tillämpningar, som t ex vid utvecklingsarbete, konstruktionsarbete, analys och felsökning. Arbetet i programspråket Omola och utvecklingsverktyget OmSim har gått väldigt bra. De objektorienterade principerna som tillämpas passar mycket bra att användas i simuleringstillämpningar.

I det arbete som beskrivits i den här rapporten, klarar inte modellen av abonnentcentralen TS30 att köra simuleringar med flöden nära noll. Önskvärt vore givetvis att modellerna ska klara av även detta, vi har emellertid inte haft tillräckligt med tid för att lösa det problemet. Vidare innehåller arbetet en modell av reglerventilen AVTQ20, vilken skulle kunna förbättras ytterligare genom att bl a implementera även glapp och friktion i modellen.

8. Referenser

- ANDERSSON, M. (1993): "Omsim and Omola — Tutorial and User's Manual." Technical Report ISRN LUTFD2/TFRT--LUTFD2/TFRT-7504-SE--SE, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- ANDERSSON, S., S. FREDERIKSEN, och J. WOLLERSTRAND (1992): "Snabb tappvattenreglering med framkoppling." Technical Report ISRN LUTFD2/TFRT--LUTMDN/TMVK-3151-SE--SE, Department of Heat and Power Engineering, Lund Institute of Technology, Lund.
- HOLMAN, J. P. (1972): *Heat Transfer*. McGraw-Hill Book Company, third edition.
- MATTSSON, S. E., M. ANDERSSON, och K. J. ÅSTRÖM (1993): "Object-oriented modeling and simulation." I LINKENS, red., *CAD for Control Systems*, pp. 31-69. Marcel Dekker, Inc.
- NILSSON, B. (1993): *Object-Oriented Modeling of Chemical Processes*. PhD thesis ISRN LUTFD2/TFRT--1041--SE, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- WOLLERSTRAND, J. (1993): "Parameteridentifiering i en dynamisk modell av genomströmningsberedare för tappvarmvatten." Technical Report LUTMDN/TMVK-3156-SE, Department of Heat and Power Engineering, Lund Institute of Technology, Lund.

A. Enfamiljsmodul TS30

Modulen består av:

- En värmeväxlare modell CB50. Bestående av 42 plattor med radiatordelen kopplad 8/9 och varmvattendelen kopplad 11/12. Kanalvolymen är 0.094 liter per kanal. Den värmeledande ytan är 1.1 m^2 .
- En termostatventil AVTQ-20.
- En ventil VMO-15 med motor AMV123.
- En differenshållningsventil AVP20 från Danfoss A/S.
- En utegivare med kontrollbox ESMT+ECL3310.
- Fyrtråds temperaturgivare PT 100 ESMA från Inor. Temperaturgivare med spetstjocklek 3 mm betecknas TEPS045 och TEPS046, se bild 3.1 på sidan 15, medan de med spetstjocklek 6 mm betecknas TEPS024 och TEPS022.
- Magnetiska flödesgivare med typbeteckning 143110 och 143111 från Proces-Data.
- Differenstrycksmätare med typbeteckning Schlumberger 74ser8dh och serie nummer 33574.247779 från Inor.
- Radiatorpump Prefecta A3-25-4.

B. Programkod

```
LIBRARY BasicThermo;

%%
%% Basic library of general thermodynamical components.
%%

FlowTerminal ISA Base::ZeroSumTerminal WITH
%%
%% Flow through a component.
%%
    unit := "m3/s";
    quantity := "volume.flow.rate";
END;

PressureTerminal ISA Base::SimpleTerminal WITH
%%
%% Pressure over a component.
%%
    unit := "Pa";
    quantity := "pressure";
END;

TemperatureTerminal ISA Base::SimpleTerminal WITH
%%
%% Temperature in a component.
%%
    unit := "K";
    quantity := "thermodynamic.temperature";
END;

ThermoTerminal ISA Base::RecordTerminal WITH
%%
%% A terminal with temperature, flow and pressure used in thermo
%% applications.
%%
    T ISA BasicThermo::TemperatureTerminal;
    %% The temperature in the terminal.
    q ISA BasicThermo::FlowTerminal;
    %% The flow through the terminal.
    P ISA BasicThermo::PressureTerminal;
    %% The pressure in the terminal.
END;

ThermoModel ISA Base::Model WITH
%%
%% The model used in thermo applications
%%
    ra ISA Base::Parameter WITH default := 998.0; END;
```

```

    %% Density.
    q0 ISA Parameter;
    %% The initial value of the flow through a component.
    DPO TYPE Discrete Real;
    %% The initial pressure drop over a component.
END;

```

```

ThermoTwoPole ISA BasicThermo::ThermoModel WITH
%%

```

```

%% Thermo component with two ThermoTerminals.
%%

```

```

terminals:

```

```

    in ISA BasicThermo::ThermoTerminal WITH
        Graphic ISA Base::Layout WITH
            x_pos := 0.0;
            y_pos := 151.0;
            invisible := 1;
    END;

```

```

END;

```

```

    out ISA BasicThermo::ThermoTerminal WITH
        Graphic ISA Base::Layout WITH
            x_pos := 400.0;
            y_pos := 151.0;
            invisible := 1;
    END;

```

```

END;

```

```

types:

```

```

    P TYPE Real;
    %% P is the pressure drop.

```

```

equations:

```

```

    P = in.P - out.P;
    in.q + out.q = 0;

```

```

END;

```

```

ThermoTwoPoleV ISA BasicThermo::ThermoTwoPole WITH
%%

```

```

%% Thermo two pole with vertical terminal layout.
%%

```

```

    out ISA BasicThermo::ThermoTerminal WITH
        Graphic ISA Base::Layout WITH
            x_pos := 201.0;
            y_pos := 300.0;
            invisible := 1;
    END;

```

```

END;

```

```

    in ISA BasicThermo::ThermoTerminal WITH
        Graphic ISA Base::Layout WITH
            x_pos := 201.0;
            y_pos := 0.0;
            invisible := 1;
    END;

```



```

    END;
END;

ThermoFourPole ISA BasicThermo::ThermoModel WITH
%%
%% Thermo component with four ThermoTerminals. InHeat and
%% OutHeat are the ThermoTerminals on the primary side and
%% InCold and OutCold on the secondary.
%%
terminals:
    InHeat ISA BasicThermo::ThermoTerminal WITH
        Graphic ISA Base::Layout WITH
            x_pos := 0.0;
            y_pos := 280.0;
            invisible := 1;
        END;
    END;
    OutHeat ISA BasicThermo::ThermoTerminal WITH
        Graphic ISA Base::Layout WITH
            x_pos := 0.0;
            y_pos := 20.0;
            invisible := 1;
        END;
    END;
    InCold ISA BasicThermo::ThermoTerminal WITH
        Graphic ISA Base::Layout WITH
            x_pos := 400.0;
            y_pos := 20.0;
            invisible := 1;
        END;
    END;
    OutCold ISA BasicThermo::ThermoTerminal WITH
        Graphic ISA Base::Layout WITH
            x_pos := 400.0;
            y_pos := 280.0;
            invisible := 1;
        END;
    END;
END;

types:
    PHeat TYPE Real;
        %% Pressure drop over the primary side.
    PCold TYPE Real;
        %% Pressure drop over the secondary side.

equations:
    PHeat = InHeat.P - OutHeat.P;
    PCold = InCold.P - OutCold.P;
    InHeat.q + OutHeat.q = 0;
    InCold.q + OutCold.q = 0;
END;

RefPoint ISA BasicThermo::ThermoModel WITH

```

```

%%
%% Reference value of the pressure.
%%
terminals:
  T ISA BasicThermo::ThermoTerminal WITH
    Graphic ISA Base::Layout WITH
      x_pos := 200.0;
      y_pos := 300.0;
      invisible := 1;
    END;
  END;

equation:
  T.P = 0.0;

graphics:
  Graphic ISA Base::Layout WITH
    bitmap TYPE String := "ground";
  END;
END;

TimeDelay ISA BasicThermo::ThermoTwoPole WITH
%%
%% Model of a time delay in a pipe.
%%
parameters:
  D ISA Base::Parameter WITH default := 1.0; END;
  %% The pipe diameter.
  L ISA Base::Parameter WITH default := 1.0; END;
  %% The length of the pipe.

types:
  V TYPE Real;
  %% The volume of the pipe.
  Tao TYPE Real;
  %% The time delay.

equations:
  P = 0;
  V := 3.14*sqr(D)*L/4;
  Tao := ra*V/in.q;
  out.T = delay(in.T, Tao);

graphics:
  Graphic ISA Base::Layout WITH
    bitmap TYPE String := "OurBitmaps/timedelay";
  END;
END;

LIBRARY ThermoGenerators;

USES BasicThermo;
%%

```

```

%% Library of generators based on library
%% basicThermo.
%%

TwoPi TYPE REAL := 4.0*acos(0.0);

FlowTempSource ISA ThermoTwoPoleV WITH
%%
%% Device to specify temperature and flow.
%%
parameters:
  Limit ISA Base::Parameter WITH default := 1.0e-5; END;
  %% Minimum flow through a component.
  T0 ISA Base::Parameter WITH default := 1.0; END;
  %% Temperature.
  q0 ISA Base::Parameter WITH default := 1.0; END;
  %% Flow.

equations:
  out.T = T0;
  out.q = if q0 > Limit then
    -q0
  else -Limit;

graphics:
  Graphic ISA Base::Layout WITH
    bitmap TYPE String := "OurBitmaps/FlowTemp";
    invisible := 1;
  END;
END;

FlowTempGenerator ISA FlowTempSource WITH
%%
%% A device which must be connected to a DWaterGenerator
%% or a TPWaterGenerator or its reverses. T0 defines the
%% temperature and q0 defines the flow.
%%
  T0 ISA Base::SimpleInput WITH
    Graphic ISA Base::Layout WITH
      x_pos := 201.0;
      y_pos := 151.0;
      invisible := 1;
    END;
  END;
  q0 ISA Base::SimpleInput WITH
    Graphic ISA Base::Layout WITH
      x_pos := 201.0;
      y_pos := 151.0;
      invisible := 1;
    END;
  END;
END;

```

```

PressureTempSource ISA ThermoTwoPoleV WITH
%%
%% Device to specify temperature and pressure.
%%
parameters:
  TO ISA Base::Parameter WITH default := 60.0; END;
  %% Temperature.
  PO ISA Base::Parameter;
  %% Preferred total pressure.
  alfa ISA Parameter WITH default := 1.0; END;
  %% A constant which specifies how fast DP will decrease.

types:
  DP TYPE Discrete Real;
  %% The differance between preferred total pressure and
  %% the sum of pressure drops over each component.
  t TYPE Real;
  %% Simulation time

equations:
  t := Base::time;
  out.T = T0;
  P = -PO+DP*exp(-alfa*t);
  %% The equation handles the problem with initial values.

graphics:
  Graphic ISA Base::Layout WITH
  bitmap TYPE String := "OurBitmaps/PresTemp";
  invisible := 1;
  END;
END;

PressureTempGenerator ISA PressureTempSource WITH
%%
%% A device which must be connected to a DWaterGenerator
%% or a TPWaterGenerator or its reverses. T0 defines the
%% temperature and PO defines the pressure.
%%
parameters:
  P00 ISA Base::Parameter;
  %% Initial value of pressure.

terminals:
  TO ISA Base::SimpleInput WITH
  Graphic ISA Base::Layout WITH
  x_pos := 201.0;
  y_pos := 151.0;
  invisible := 1;
  END;
  END;
  PO ISA Base::SimpleInput WITH
  Graphic ISA Base::Layout WITH
  x_pos := 201.0;

```

```

        y_pos := 151.0;
        invisible := 1;
    END;
END;
END;

WaterGenerator ISA Model WITH
%%
%% Wave form generator to be connected to FlowTemp
%% or PressureTemp generators. Possible waveforms are sawtooth,
%% sine, square and triangle. For each waveform it is possible
%% to specify amplitude, frequency and bias values.
%%
parameters:
    Func ISA Parameter WITH
        %% Possible waveforms.
        value TYPE (sawtooth, sine, square, triangle);
        default TYPE (sawtooth, sine, square, triangle) := 'sine;
    END;
    Freq ISA Parameter WITH default := 1.0; END;
        %% Frequency in Herz
    Ampl ISA Parameter WITH default := 0.0; END;
        %% Top value amplitude
    DutyCycle ISA Parameter WITH default := 0.5; END;
        %% Fraction of period where a square wave is high or
        %% a triangle wave is increasing.
    Bias ISA Parameter WITH default := 20; END;
        %% Mean value for sine and lowest value for square,
        %% triangle and sawtooth.

types:
    square, triangleDer TYPE Discrete Real;
    triangle, ref TYPE Real;
    switchOffTime TYPE Real := DutyCycle / freq;
    triangle' = triangleDer;
    Init, SwitchOn, SwitchOff ISAN Event;

events:
    OnEvent Init CAUSE SwitchOn;

    OnEvent SwitchOn DO
        schedule(SwitchOff, switchOffTime);
        schedule(SwitchOn, 1/Freq);
        new(square) := Ampl + Bias;
        new(triangle) := 0.0;
        new(triangleDer) := if Func=='triangle then
            Ampl*Freq/DutyCycle else
            Ampl*Freq;

    END;

    OnEvent SwitchOff DO
        new(square) := Bias;

```

```

new(triangleDer) := if Func=='triangle then
                    -Ampl*Freq/(1.0 - DutyCycle) else
                    triangleDer;
END;

equations:
  ref := if (func == 'sine) then ampl*
          sin(freq* ::TwoPi * Base::Time) + Bias
        else if (func == 'square) then square
        else triangle + Bias;

graphics:
  Graphic ISA Base::Layout WITH
    bitmap TYPE String := "OurBitmaps/WaterGen";
  END;
END;

DWaterGenerator ISA Base::Model WITH
%%
%% The double watergenerator consists of two watergenerators.
%% It makes it possible to define the inputs to a
%% FlowTempGenerator or a PressureTempGenerator with one
%% generator.
%%
submodels:
  WaterGenerator1 ISA WaterGenerator WITH
    Graphic ISA Base::Layout WITH
      x_pos := 125.0;
      y_pos := 175.0;
      invisible := 1;
    END;
  END;
  WaterGenerator2 ISA WaterGenerator WITH
    Graphic ISA Base::Layout WITH
      x_pos := 275.0;
      y_pos := 175.0;
      invisible := 1;
    END;
  END;
END;

terminals:
  ref1 ISA Base::SimpleOutput WITH
    Graphic ISA Base::Layout WITH
      x_pos := 400.0;
      y_pos := 275.0;
      invisible := 1;
    END;
  END;
  ref2 ISA Base::SimpleOutput WITH
    Graphic ISA Base::Layout WITH
      x_pos := 400.0;
      y_pos := 25.0;
      invisible := 1;

```

```

        END;
    END;

equations:
    ref1 = WaterGenerator1.ref ;
    ref2 = WaterGenerator2.ref ;

graphics:
    Graphic ISA Base::Layout WITH
        bitmap TYPE String := "OurBitmaps/WaterGen";
    END;
END;

RevDWaterGenerator ISA DWaterGenerator WITH
%%
%% A DWatergenerator with the icon in reverse.
%%
    ref1 ISA Base::SimpleOutput WITH
        Graphic ISA Base::Layout WITH
            x_pos := 0.0;
            y_pos := 275.0;
            invisible := 1;
        END;
    END;
    ref2 ISA Base::SimpleOutput WITH
        Graphic ISA Base::Layout WITH
            x_pos := 0.0;
            y_pos := 25.0;
            invisible := 1;
        END;
    END;
END;

TPDWaterGenerator ISA Base::Model WITH
%%
%% A double water generator with a tap water program,
%% to be connected to a FlowTempGenerator. The temperature
%% is defined by a WaterGenerator as usual. The flow is
%% defined by a tap water program.
%%
submodels:
    WaterGenerator1 ISA WaterGenerator WITH
        Graphic ISA Base::Layout WITH
            x_pos := 125.0;
            y_pos := 175.0;
            invisible := 1;
        END;
    END;

terminals:
    ref1 ISA Base::SimpleOutput WITH
        Graphic ISA Base::Layout WITH

```

```

    x_pos := 400.0;
    y_pos := 275.0;
    invisible := 1;
  END;
END;
ref2 ISA Base::SimpleOutput WITH
  Graphic ISA Base::Layout WITH
    x_pos := 400.0;
    y_pos := 25.0;
    invisible := 1;
  END;
END;

```

types:

```

  Init,e1,e2,e3,e4,e5,e6,e7,e8 ISAN Event;
  %% One event for each change of flow.
  t_e1,t_e2,t_e3,t_e4,t_e5,t_e6,t_e7,t_e8 TYPE Real;
  %% Time when the event will occur.
  q TYPE Discrete Real;
  %% Defines the flow.

```

assignments:

```

  t_e1 := 100;
  t_e2 := 200;
  t_e3 := 300;
  t_e4 := 400;
  t_e5 := 500;
  t_e6 := 600;
  t_e7 := 700;
  t_e8 := 800;

```

initials:

```

  ONEVENT Init D0
    new(q):=0.0003;
    Schedule(e1,t_e1);
    Schedule(e2,t_e2);
    Schedule(e3,t_e3);
    Schedule(e4,t_e4);
    Schedule(e5,t_e5);
    Schedule(e6,t_e6);
    Schedule(e7,t_e7);
    Schedule(e8,t_e8);
  END;

```

events:

```

  ONEVENT e1 D0
    new(q) := 0.0002;
  END;
  ONEVENT e2 D0
    new(q) := 0.0001;
  END;
  ONEVENT e3 D0
    new(q) := 0.0003;

```



```

END;
ONEVENT e4 D0
    new(q) := 0.0001;
END;
ONEVENT e5 D0
    new(q) := 0.00015;
END;
ONEVENT e6 D0
    new(q) := 0.0002;
END;
ONEVENT e7 D0
    new(q) := 0.00025;
END;
ONEVENT e8 D0
    new(q) := 0.0003;
END;

equations:
    ref1 = WaterGenerator1.ref;
    ref2 = q;

graphics:
    Graphic ISA Base::Layout WITH
        bitmap TYPE String := "OurBitmaps/WaterGen";
    END;
END;

RevTPDWaterGenerator ISA TPDWaterGenerator WITH
%%
%% A TPDWatergenerator with the icon in reverse.
%%
    ref1 ISA Base::SimpleOutput WITH
        Graphic ISA Base::Layout WITH
            x_pos := 0.0;
            y_pos := 275.0;
            invisible := 1;
        END;
    END;
    ref2 ISA Base::SimpleOutput WITH
        Graphic ISA Base::Layout WITH
            x_pos := 0.0;
            y_pos := 25.0;
            invisible := 1;
        END;
    END;
END;

Library Inputs;
%%
%% Library containing input equipment.
%%

ThermoInput ISA BasicThermo::ThermoModel WITH

```

```

%%
%% A device to take in one external set of data.
%% For example to look on a serie of measured values.
%%
inputs:
  c ISA ContinuousInput;
  %% The external value.

types:
  x TYPE Real := c;
  %% The internal representation of c.

graphics:
  Graphic ISA Base::Layout WITH
    bitmap TYPE String := "OurBitmaps/Input";
  END;
END;

TwoInputs ISA Base::Model WITH
%%
%% A device to take in two external sets of data.
%% For example to specify the values into a modul.
%%
inputs:
  I1 ISA ContinuousInput;
  %% The first external value.
  I2 ISA ContinuousInput;
  %% The second external value.

terminals:
  ref1 ISA Base::SimpleOutput WITH
    Graphic ISA Base::Layout WITH
      x_pos := 400.0;
      y_pos := 275.0;
      invisible := 1;
    END;
  END;
  ref2 ISA Base::SimpleOutput WITH
    Graphic ISA Base::Layout WITH
      x_pos := 400.0;
      y_pos := 25.0;
      invisible := 1;
    END;
  END;
END;

equations:
  ref1 = I1;
  ref2 = I2;

graphics:
  Graphic ISA Base::Layout WITH
    bitmap TYPE String := "OurBitmaps/Input";
    invisible := 1;

```

```

    END;
END;

RevTwoInputs ISA TwoInputs WITH
%%
%% A TwoInputs with the icon in reverse.
%%
inputs:
    I1 ISA ContinuousInput;
    I2 ISA ContinuousInput;

terminals:
    ref1 ISA Base::SimpleOutput WITH
        Graphic ISA Base::Layout WITH
            x_pos := 0.0;
            y_pos := 275.0;
            invisible := 1;
        END;
    END;
    ref2 ISA Base::SimpleOutput WITH
        Graphic ISA Base::Layout WITH
            x_pos := 0.0;
            y_pos := 25.0;
            invisible := 1;
        END;
    END;
END;

LIBRARY Heatexchangers;

%%
%% The library consists of SimpleHeatExchanger, which is
%% a model of one heat exchanger section. The second model
%% in the library ComplexHeatExchanger, is a heat exchanger
%% consisting of three sections.
%%

SimpleHeatExchanger ISA BasicThermo::ThermoFourPole WITH
%%
%% A model of a section in a heat exchanger.
%%
parameters:
    eps ISA Base::Parameter WITH default := 0.05; END;
    %% A limit when one should use an ln-expression
    %% instead of an approximation..
    Cv ISA Base::Parameter WITH default := 0.0008; END;
    %% Cv is the flow coefficient.
    VHeat ISA Base::Parameter WITH default := 5e-05; END;
    %% The volume of the ducts for each section on
    %% the primary side.
    VCold ISA Base::Parameter WITH default := 5e-05; END;
    %% The volume of the ducts for each section on
    %% the secondary side.

```

```

a ISA Base::Parameter WITH default := 0.4; END;
  %% a is the heat transfer surface in each section.
c ISA Base::Parameter WITH default := 4180; END;
  %% c is the specific heat capacity.
n ISA Parameter WITH default := 0.75; END;
  %% n is a model constant.
T ISA Parameter WITH default := 0.0004; END;
  %% T is the thickness of the plate.
Yf ISA Parameter WITH default := 1.2; END;
  %% Yf is the surface enlargement.
alphakonst ISA Parameter WITH default := 1; END;
  %% Is a model constant.
akonst ISA Parameter WITH default := 0.0; END;
  %% akonst is a model constant.
Tm ISA Parameter WITH default := 57.5; END;
  %% An estimate of the average value of the lowest and
  %% the highest temperature through the heatexchanger.
Rf ISA Parameter WITH default := 0; END;
  %% Rf is a faulting factor.

```

types:

```

lambda TYPE Real;
  %% The Heat conductivity
Rw TYPE Real;
  %% The Wall resistance
alphaH TYPE Real;
  %% surface coefficient of heat transfer on the primary side.
alphaC TYPE Real;
  %% surface coefficient of heat transfer on the secondary side.
k TYPE Real;
  %% ???
QHeat TYPE Real;
  %% Heat power on the primary side.
QCold TYPE Real;
  %% Heat power on the secondary side.
DT1 TYPE Real;
DT2 TYPE Real;
Delta TYPE Real;
DTm TYPE Real;
  %% The LMTD value.
ah TYPE Real;
  %% Model constant.
ac TYPE Real;
  %% Model constant.
bh TYPE Real;
  %% Model constant.
bc TYPE Real;
  %% Model constant.
Tap TYPE Real;
  %% An estimate of the average temperature on
  %% the primary side.
Tas TYPE Real;
  %% An estimate of the average temperature on

```

```

%% the secondary side.
Tw TYPE Real;
%% The temperature in the middle of the plate.
DTempP TYPE Real;
DTempS TYPE Real;
TWallP TYPE Real;
%% The surface temperature of the plate on
%% the primary side.
TWallS TYPE Real;
%% The surface temperature of the plate on
%% the secondary side.

equations:
alphaH := alphakonst*(1 + akonst*((InHeat.T +
    OutHeat.T)/2-Tm))*abs(InHeat.q*ra)^n;
alphaC := alphakonst*(1 + akonst*((InCold.T +
    OutCold.T)/2-Tm))*abs(InCold.q*ra)^n;
lambda := 13.24 + 0.0012*((InHeat.T + OutHeat.T)/2 +
    (InCold.T + OutCold.T)/2)/2;
Rw := T/(lambda*Yf);
k := 1/(1/alphaH + 1/alphaC + Rw + Rf);
PHeat = ra*sqr(InHeat.q)/sqr(Cv);
PCold = ra*sqr(InCold.q)/sqr(Cv);
QHeat = (InHeat.T - OutHeat.T)*InHeat.q*c*ra;
QCold = (OutCold.T - InCold.T)*InCold.q*c*ra;
DT1 = max(InHeat.T - OutCold.T, eps);
DT2 = max(OutHeat.T - InCold.T, eps);
Delta = DT1 - DT2;
DTm = if abs(Delta) < eps*max(DT1, DT2) then
    0.5*(DT1 + DT2)*(1 - sqr(Delta)/(12*DT1*DT2) +
    sqr(Delta)*sqr(Delta)/(24*sqr(DT1*DT2)))
    else Delta/ln(DT1/DT2);
ah := InHeat.q/VHeat;
ac := InCold.q/VCold;
bh := k*a/(InHeat.q*ra*c);
bc := k*a/(InCold.q*ra*c);
OutHeat.T' = ah*(InHeat.T - OutHeat.T - bh*DTm);
OutCold.T' = ac*(-(OutCold.T - InCold.T) + bc*DTm);
%% Wall temperature
Tap := (InHeat.T+OutHeat.T)/2;
Tas := (InCold.T+OutCold.T)/2;
Tw' = (k/(T*ra*c))*((Tap-Tw)-(Tw-Tas));
DTempP := InHeat.T-OutHeat.T;
DTempS := OutCold.T-InCold.T;
TWallP := InHeat.q*c*DTempP*((T)/(2*lambda*Yf*a))+Tw;
TWallS := Tw-InCold.q*c*DTempS*((T)/(2*lambda*Yf*a));

graphics:
Graphic ISA Base::Layout WITH
    bitmap TYPE String := "OurBitmaps/heatexchanger";
END;

initials:

```

```

Init ISA Event;
ONEVENT Init D0
  new(InHeat.q) := q0;
  new(PHeat) := ra*sqr(q0)/sqr(Cv);
  new(DP0) := ra*sqr(q0)/sqr(Cv);
END;
END;

ComplexHeatExchanger ISA BasicThermo::ThermoModel WITH
%%
%% A model of a Heat exchanger with three sections.
%%
parameters:
  VHeat ISA Base::Parameter WITH default := 0.0005; END;
  %% VHeat is the total duct volume on the primary side.
  VCold ISA Base::Parameter WITH default := 0.0005; END;
  %% VCold is the total duct volume on the secondary side.
  a ISA Base::Parameter WITH default := 0.8; END;
  %% a is the total heat transfer surface.
  Cv ISA Base::Parameter WITH default := 0.0008; END;
  %% Cv is flow coefficient.
  c ISA Base::Parameter WITH default := 4180.0; END;
  %% c is specific heat capacity.
  n ISA Parameter WITH default := 0.68; END;
  %% n is a model constant.
  T ISA Parameter WITH default := 0.0004; END;
  %% T is the thickness of the plate.
  Yf ISA Parameter WITH default := 1.2; END;
  %% Yf is the enlargement of the surface.
  alphakonst ISA Parameter WITH default := 18000; END;
  %% alphakonst is model constant.
  Tm ISA Parameter WITH default := 57.5; END;
  %% An estimate of the average value of the lowest and
  %% the highest temperature through the heat exchanger.
  akonst ISA Parameter WITH default := 0.0; END;
  %% akonst is a model constant.

submodels:
  SimpleHeatExchanger1 ISA SimpleHeatExchanger WITH
    Graphic ISA super::Graphic WITH
      x_pos := 100.0;
      y_pos := 150.0;
    END;
  END;
  SimpleHeatExchanger2 ISA SimpleHeatExchanger WITH
    Graphic ISA super::Graphic WITH
      x_pos := 200.0;
      y_pos := 150.0;
    END;
  END;
  SimpleHeatExchanger3 ISA SimpleHeatExchanger WITH
    Graphic ISA super::Graphic WITH
      x_pos := 300.0;

```

```
    y_pos := 150.0;
  END;
END;
```

assignments:

```
SimpleHeatExchanger1.n := n;
SimpleHeatExchanger2.n := n;
SimpleHeatExchanger3.n := n;
SimpleHeatExchanger1.T := T;
SimpleHeatExchanger2.T := T;
SimpleHeatExchanger3.T := T;
SimpleHeatExchanger1.Yf := Yf;
SimpleHeatExchanger2.Yf := Yf;
SimpleHeatExchanger3.Yf := Yf;
SimpleHeatExchanger1.alphakonst := alphakonst;
SimpleHeatExchanger2.alphakonst := alphakonst;
SimpleHeatExchanger3.alphakonst := alphakonst;
SimpleHeatExchanger1.Tm := Tm;
SimpleHeatExchanger2.Tm := Tm;
SimpleHeatExchanger3.Tm := Tm;
SimpleHeatExchanger1.akonst := akonst;
SimpleHeatExchanger2.akonst := akonst;
SimpleHeatExchanger3.akonst := akonst;
SimpleHeatExchanger1.a := a/3;
SimpleHeatExchanger2.a := a/3;
SimpleHeatExchanger3.a := a/3;
SimpleHeatExchanger1.VHeat := VHeat/3;
SimpleHeatExchanger2.VHeat := VHeat/3;
SimpleHeatExchanger3.VHeat := VHeat/3;
SimpleHeatExchanger1.VCold := VCold/3;
SimpleHeatExchanger2.VCold := VCold/3;
SimpleHeatExchanger3.VCold := VCold/3;
SimpleHeatExchanger1.c := c;
SimpleHeatExchanger2.c := c;
SimpleHeatExchanger3.c := c;
SimpleHeatExchanger1.Cv := Cv*sqrt(3);
SimpleHeatExchanger2.Cv := Cv*sqrt(3);
SimpleHeatExchanger3.Cv := Cv*sqrt(3);
SimpleHeatExchanger1.ra := ra;
SimpleHeatExchanger2.ra := ra;
SimpleHeatExchanger3.ra := ra;
SimpleHeatExchanger1.q0 := q0;
SimpleHeatExchanger2.q0 := q0;
SimpleHeatExchanger3.q0 := q0;
```

terminals:

```
InHeat ISA BasicThermo::ThermoTerminal WITH
  Graphic ISA super::Graphic WITH
    x_pos := 0.0;
    y_pos := 270.0;
  END;
END;
OutHeat ISA BasicThermo::ThermoTerminal WITH
```

```

    Graphic ISA super::Graphic WITH
      x_pos := 0.0;
      y_pos := 30.0;
    END;
  END;
  OutCold ISA BasicThermo::ThermoTerminal WITH
    Graphic ISA super::Graphic WITH
      x_pos := 400.0;
      y_pos := 270.0;
    END;
  END;
  InCold ISA BasicThermo::ThermoTerminal WITH
    Graphic ISA super::Graphic WITH
      x_pos := 400.0;
      y_pos := 30.0;
    END;
  END;

connections:
  C1 ISA Base::Connection WITH
    InCold AT SimpleHeatExchanger3.InCold;
    bpoints TYPE Matrix[4, 2] := [400.0, 75.0; 355.0, 75.0;
    355.0, 122.0; 315.0, 122.0];
  END;
  C2 ISA Base::Connection WITH
    SimpleHeatExchanger3.OutCold AT SimpleHeatExchanger2.InCold;
    bpoints TYPE Matrix[6, 2] := [315.0, 176.0; 343.0, 176.0;
    343.0, 200.0; 245.0, 200.0; 245.0, 122.0; 215.0, 122.0];
  END;
  C3 ISA Base::Connection WITH
    SimpleHeatExchanger2.OutCold AT SimpleHeatExchanger1.InCold;
    bpoints TYPE Matrix[6, 2] := [215.0, 176.0; 233.0, 176.0;
    233.0, 200.0; 152.0, 200.0; 152.0, 122.0; 115.0, 122.0];
  END;
  C5 ISA Base::Connection WITH
    InHeat AT SimpleHeatExchanger1.InHeat;
    bpoints TYPE Matrix[4, 2] := [0.0, 250.0; 59.0, 250.0; 59.0,
    176.0; 84.0, 176.0];
  END;
  C6 ISA Base::Connection WITH
    SimpleHeatExchanger1.OutHeat AT SimpleHeatExchanger2.InHeat;
    bpoints TYPE Matrix[5, 2] := [84.0, 122.0; 84.0, 96.0;
    163.0, 96.0; 163.0, 176.0; 184.0, 176.0];
  END;
  C7 ISA Base::Connection WITH
    SimpleHeatExchanger2.OutHeat AT SimpleHeatExchanger3.InHeat;
    bpoints TYPE Matrix[5, 2] := [184.0, 122.0; 184.0, 102.0;
    261.0, 102.0; 261.0, 176.0; 284.0, 176.0];
  END;
  C4 ISA Base::Connection WITH
    OutCold AT SimpleHeatExchanger1.OutCold;
    bpoints TYPE Matrix[3, 2] := [115.0, 176.0; 115.0, 249.0;
    399.0, 249.0];

```



```

END;
C8 ISA Base::Connection WITH
    OutHeat AT SimpleHeatExchanger3.OutHeat;
    bpoints TYPE Matrix[3, 2] := [0.0, 74.0; 284.0, 74.0; 284.0,
    122.0];
END;

graphics:
    Graphic ISA Base::Layout WITH
        bitmap TYPE String := "OurBitmaps/heatexchanger";
    END;

initials:
    Init ISA Event;
    ONEVENT Init DO
        new(DPO) := new(SimpleHeatExchanger1.DPO+
            SimpleHeatExchanger2.DPO+SimpleHeatExchanger3.DPO);
    END;
END;

LIBRARY BasicMechanics;

%%
%% Basic library of general mechanical components.
%%

PositionTerminal ISA Base::SimpleTerminal WITH
%%
%% Position of a mechanical device.
%%
    unit := "m";
    quantity := "length";
END;

ForceTerminal ISA Base::SimpleTerminal WITH
%%
%% Force in a mechanical device.
%%
    unit := "N";
    quantity := "force";
END;

MechanicTerminal ISA Base::RecordTerminal WITH
%%
%% Mechanical terminal with position and force.
%%
    x ISA PositionTerminal;
    F ISA ForceTerminal;
END;

LIBRARY AVTQ;

%%

```

```
%% Library of the controlvalve AVTQ from Danfoss A/S.
%%
```

```
AVTQFlowMeas ISA BasicThermo::ThermoTwoPole WITH
%%
%% Flow measure component used with an AVTQ-valve.
%% Generates a pressure drop depending on the flow
%% through the component. Usually placed on the
%% secondary water side.
%%
```

```
parameters:
```

```
  Pm0 ISA Parameter WITH default := 20000; END;
    %% The initial pressure-drop.
  qm0 ISA Parameter WITH default := 2e-05; END;
    %% The initial flow.
  GainPhase1 ISA Parameter WITH default :=6.6e8; END;
    %% Gain from flow to pressure in phase 1 = low
    %% tap water flow.
  GainPhase23 ISA Parameter WITH default := 3.3e7; END;
    %% Gain from flow to pressure in phase 2 and 3 = high
    %% tap water flow.
```

```
terminal:
```

```
  deltaPm ISA Base::SimpleOutput WITH
    Graphic ISA super::Graphic WITH
      x_pos := 200.0;
      y_pos := 300.0;
      invisible := 1;
    END;
  END;
```

```
equations:
```

```
  out.T = in.T;
  deltaPm = if in.q < qm0 then GainPhase1*in.q else Pm0 +
    GainPhase23*in.q;
  P = deltaPm;
```

```
graphics:
```

```
  Graphic ISA Base::Layout WITH
    bitmap TYPE String := "OurBitmaps/AVTQFlowMeas";
  END;
  END;
```

```
AVTQFlowGenerator ISA BasicThermo::ThermoModel WITH
```

```
%%
%% The part of the AVTQ that generates a force depending
%% on the tap water flow and a reference value.
%%
```

```
parameters:
```

```
  Am ISA Parameter;
    %% Cross area of the membranes.
  F10 ISA Parameter;
    %% Force generated by the flowmeasure with y=0 and x=0.
```

```

F20 ISA Parameter;
    %% Reference force.
C1 ISA Parameter;
    %% Spring constant of flowmeasure spring.
C2 ISA Parameter;
    %% Spring constant of reference spring.
ymax ISA Parameter;
    %% Maximum position of membrane.

types:
    F1 TYPE Real;
        %% Force generated of the tap water flow.
    F2 TYPE Real;
        %% Force generated of the reference spring.
    Mode TYPE DISCRETE Integer;
        %% The mode the component is in.
    y TYPE Real;
        %% Position of the membrane.

terminals:
    deltaPm ISA Base::SimpleTerminal WITH
        Graphic ISA super::Graphic WITH
            x_pos := 0.0;
            y_pos := 150.0;
        END;
    END;
    MecTerm ISA BasicMechanics::MechanicTerminal WITH
        Graphic ISA super::Graphic WITH
            x_pos := 400.0;
            y_pos := 150.0;
        END;
    END;

events:
    %% Changing mode
    ONEVENT (Mode == 1) and (deltaPm > F1/Am) DO
        new(Mode) := 3;
    END;
    ONEVENT (Mode == 3) and (y <= 0) DO
        new(Mode) := 1;
    END;
    ONEVENT (Mode == 3) and (y > ymax) DO
        new(Mode) := 2;
    END;
    ONEVENT (Mode == 2) and (deltaPm < F1/Am) DO
        new(Mode) := 3;
    END;

equations:
    F1 = if Mode == 1 then F10 - C1*MecTerm.x
        else if Mode == 2 then F10 + C1*ymax - C1*MecTerm.x
        else Am*deltaPm;
    y = if Mode == 1 then 0

```

```
    else if Mode == 2 then ymax
    else (F1 - F10)/C1 + MecTerm.x;
F2 = F20 - C2*MecTerm.x;
MecTerm.F = F1 + F2;
```

graphics:

```
Graphic ISA Base::Layout WITH
    bitmap TYPE String := "OurBitmaps/FlowGenerator";
END;
```

END;

AVTQTempGenerator ISA BasicThermo::ThermoModel WITH

%%

%% Generates a force depending on a temperature.

%% Usually measure tap water temperature.

%%

parameters:

```
FO ISA Parameter;
    %% Force generated by temperature element when
    %% Temp is 50 degC and position x=0.
cb ISA Parameter;
    %% Spring constant.
ref ISA Parameter;
    %% Reference temperature. Usually 55 degC.
k ISA Parameter;
    %% Gain from temperature to valve position.
```

type:

```
deltaTw TYPE Real;
    %% Difference between Temp and ref.
```

terminals:

```
Term ISA BasicMechanics::MechanicTerminal WITH
    P ISA BasicThermo::PressureTerminal;
    Graphic ISA super::Graphic WITH
        x_pos := 0.0;
        y_pos := 150.0;
    END;
END;
Temp ISA Base::SimpleTerminal WITH
    Graphic ISA super::Graphic WITH
        x_pos := 200.0;
        y_pos := 0.0;
    END;
END;
```

equations:

```
deltaTw = Temp - ref;
Term.F = FO + k*deltaTw + cb*Term.x;
```

graphics:

```
Graphic ISA Base::Layout WITH
    bitmap TYPE String := "OurBitmaps/TempGenerator";
```

```

    END;
END;

AVTQValve ISA Valves::Valve WITH
%%
%% The AVTQ valve.
%%
parameters:
    kvs0 ISA Parameter;
        %% Parameter affecting the characteristics of the valve.
    kvx ISA Parameter;
        %% Parameter affecting the characteristics of the valve.
    xmax ISA Parameter;
        %% Maximum of the valve position.
    Am ISA Parameter;
        %% Cross area of the membranes.
    Ak ISA Parameter;
        %% Cross area of the cone.
    Amk ISA Parameter;
        %% Difference between Am and Ak.

types:
    Fp TYPE Real;
        %% Force generated by the pressure drop over the valve.

terminals:
    FlowTerm ISA BasicMechanics::MechanicTerminal WITH
        Graphic ISA super::Graphic WITH
            x_pos := 40.0;
            y_pos := 300.0;
        END;
    END;
    TempTerm ISA BasicMechanics::MechanicTerminal WITH
        P ISA BasicThermo::PressureTerminal;
        Graphic ISA super::Graphic WITH
            x_pos := 360.0;
            y_pos := 300.0;
        END;
    END;

equations:
    -Fp - TempTerm.F + FlowTerm.F = 0; %% Force equation
    Amk = Am - Ak;
    TempTerm.P = P;
    Fp = Amk*P/(1 + sqr(kvs0/kvx*FlowTerm.x/xmax));
    FlowTerm.x = TempTerm.x;
    kv = kvs0*FlowTerm.x/xmax/sqrt(1 +
        sqr(kvs0/kvx*FlowTerm.x/xmax));

graphics:
    Graphic ISA Base::Layout WITH
        bitmap TYPE String := "OurBitmaps/ControlValve";
    END;

```

END;

AVTQ ISA BasicThermo::ThermoModel WITH

%%

%% A model of a controlvalve AVTQ.

%%

parameters:

TO ISA Parameter;

%% Initial temperature.

x0 ISA Parameter;

%% Initial valve position.

PO ISA Parameter;

%% Initial pressure drop.

Am ISA Parameter;

%% Cross area of membran.

F10 ISA Parameter;

%% Force of the flowmeasure element with y=0 and x=0.

F20 ISA Parameter;

%% Reference force.

C1 ISA Parameter;

%% Spring constant.

C2 ISA Parameter;

%% Spring constant.

ymax ISA Parameter;

%% Maximum position of membrane.

F0 ISA Parameter;

%% Force generated when TempGenerator.Temp=50

%% and TempGenerator.x=0.

cb ISA Parameter;

%% Spring constant.

ref ISA Parameter;

%% Reference temperature.

kvs0 ISA Parameter;

%% Parameter affecting the characteristics of the valve.

kvx ISA Parameter;

%% Parameter affecting the characteristics of the valve.

xmax ISA Parameter;

%% Maximum valve position.

Ak ISA Parameter;

%% Cross area of the cone.

k ISA Parameter;

%% Gain from temperature to the valve position.

types:

Mode0 TYPE DISCRETE Integer;

%% Initial mode.

terminals:

deltaPm ISA Base::SimpleInput WITH

Graphic ISA super::Graphic WITH

x_pos := 200.0;

y_pos := 0.0;

invisible := 1;

```

    END;
END;
TempMeas ISA Base::SimpleInput WITH
  Graphic ISA super::Graphic WITH
    x_pos := 200.0;
    y_pos := 300.0;
    invisible := 1;
  END;
END;
in ISA BasicThermo::ThermoTerminal WITH
  Graphic ISA super::Graphic WITH
    x_pos := 1.0;
    y_pos := 151.0;
  END;
END;
out ISA BasicThermo::ThermoTerminal WITH
  Graphic ISA super::Graphic WITH
    x_pos := 400.0;
    y_pos := 150.0;
  END;
END;

submodels:
  FlowGenerator ISA AVTQ::AVTQFlowGenerator WITH
    Graphic ISA super::Graphic WITH
      x_pos := 100.0;
      y_pos := 200.0;
    END;
  END;
  Valve ISA AVTQ::AVTQValve WITH
    Graphic ISA super::Graphic WITH
      x_pos := 175.0;
      y_pos := 125.0;
    END;
  END;
  TempGenerator ISA AVTQ::AVTQTempGenerator WITH
    Graphic ISA super::Graphic WITH
      x_pos := 250.0;
      y_pos := 200.0;
    END;
  END;

connections:
  C3 ISA Base::Connection WITH
    TempGenerator.Term AT Valve.TempTerm;
    bpoints TYPE STATIC Matrix[3, 2] := [232.0, 199.0; 189.0,
    199.0; 189.0, 141.0];
  END;
  C5 ISA Base::Connection WITH
    Valve.out AT out;
    bpoints TYPE STATIC Matrix[4, 2] := [191.0, 124.0; 323.0,
    124.0; 323.0, 149.0; 399.0, 149.0];
  END;

```

```

C8 ISA Base::Connection WITH
  FlowGenerator.MecTerm AT Valve.FlowTerm;
  bpoints TYPE STATIC Matrix[3, 2] := [116.0, 199.0; 160.0,
    199.0; 160.0, 141.0];
END;
C11 ISA Base::Connection WITH
  in AT Valve.in;
  bpoints TYPE STATIC Matrix[4, 2] := [0.0, 150.0; 56.0,
    150.0; 56.0, 125.0; 158.0, 125.0];
END;
C4 ISA Base::Connection WITH
  FlowGenerator.deltaPm AT deltaPm;
  bpoints TYPE STATIC Matrix[5, 2] := [83.0, 200.0; 67.0,
    200.0; 67.0, 50.0; 200.0, 50.0; 200.0, -1.0];
END;
C6 ISA Base::Connection WITH
  TempMeas AT TempGenerator.Temp;
  bpoints TYPE STATIC Matrix[7, 2] := [200.0, 300.0; 200.0,
    257.0; 293.0, 257.0; 293.0, 197.0; 293.0, 168.0; 250.0,
    168.0; 250.0, 183.0];
END;

graphics:
  Graphic ISA Base::Layout WITH
    bitmap TYPE String := "OurBitmaps/AVTQ";
  END;
END;

AVTQ20 ISA AVTQ WITH
%%
%% Model of an AVTQ20 with attaching parameters.
%%
assignments:
  Am.default := 0.0045;
  F10.default := 32;
  F20.default := 217.5;
  C1.default := 13000;
  C2.default := 5900;
  ymax.default := 0.005;
  F0.default := 300;
  cb.default := 16400;
  ref.default := 50;
  xmax.default := 0.0016;
  kvs0.default := 4.392e-06;
  kvx.default := 4.0998e-06;
  Ak.default := 0.0044987;
  k.default := 0.8;
  FlowGenerator.Am := Am;
  FlowGenerator.F10 := F10;
  FlowGenerator.F20 := F20;
  FlowGenerator.C1 := C1;
  FlowGenerator.C2 := C2;
  FlowGenerator.ymax := ymax;

```



```

TempGenerator.FO := FO;
TempGenerator.cb := cb;
TempGenerator.ref := ref;
TempGenerator.k := k;
Valve.xmax := xmax;
Valve.kvs0 := kvs0;
Valve.kvx := kvx;
Valve.Am := Am;
Valve.Ak := Ak;

initials:
Init ISA Event;
ONEVENT Init DO
  new(FlowGenerator.Mode) := Mode0;
  new(TempGenerator.Temp) := T0;
  new(TempGenerator.deltaTw) := T0 - ref;
  new(Valve.in.q) := q0;
  new(Valve.FlowTerm.x) := x0;
  new(TempGenerator.Term.x) := new(Valve.FlowTerm.x);
  new(FlowGenerator.MecTerm.x) := new(Valve.FlowTerm.x);
  new(Valve.kv) := kvs0*new(Valve.FlowTerm.x)/xmax/sqrt(1 +
    sqrt(kvs0/kvx*new(Valve.FlowTerm.x)/xmax));
  new(Valve.P) := sqrt(new(Valve.in.q))/sqrt(new(Valve.kv));
  new(Valve.Fp) := new(Valve.P)*(Am - Ak)/(1 +
    sqrt(kvs0/kvx*new(Valve.FlowTerm.x)/xmax));
  new(TempGenerator.Term.F) := FO +
    k*new(TempGenerator.deltaTw) +
    cb*new(TempGenerator.Term.x);
  new(FlowGenerator.F1) := if Mode0==1 then
    F10 - C1*new(Valve.FlowTerm.x)
  else if Mode0 == 2 then
    F10 + C1*ymax - C1*new(FlowGenerator.MecTerm.x)
  else Am*20000;
  new(FlowGenerator.y) := if Mode0==1 then 0
  else if Mode0 == 2 then ymax
  else (new(FlowGenerator.F1) - F10)/C1 +
    new(FlowGenerator.MecTerm.x);
  new(FlowGenerator.F2) := F20 -
    C2*new(FlowGenerator.MecTerm.x);
  new(FlowGenerator.MecTerm.F) := new(FlowGenerator.F1) +
    new(FlowGenerator.F1);
  new(Valve.FlowTerm.F) := new(FlowGenerator.MecTerm.F);
  new(Valve.TempTerm.F) := new(TempGenerator.Term.F);
  new(DPO) := new(Valve.P);
END;
END;

LIBRARY MeasEquip;

%%
%% Library containing equipment for measurements.
%%

```

```

FlowMeas ISA BasicThermo::ThermoTwoPole WITH
%%
%% Sets the terminal Flow to the flow through the component.
%%
terminals:
  Flow ISA Base::SimpleOutput WITH
  Graphic ISA Base::Layout WITH
    x_pos := 200.0;
    y_pos := 300.0;
    invisible := 1;
  END;
END;

equations:
  P = 0;
  out.T = in.T;
  Flow = in.q;

graphics:
  Graphic ISA Base::Layout WITH
    bitmap TYPE String := "OurBitmaps/Flowmess";
  END;
END;

PressureMeas ISA BasicThermo::ThermoTwoPole WITH
%%
%% Sets the terminal Pressure to the pressure into the component.
%%
terminal:
  Pressure ISA Base::SimpleOutput WITH
  Graphic ISA Base::Layout WITH
    x_pos := 200.0;
    y_pos := 300.0;
    invisible := 1;
  END;
END;

equations:
  P = 0;
  out.T = in.T;
  Pressure = in.P;

graphics:
  Graphic ISA Base::Layout WITH
    bitmap TYPE String := "OurBitmaps/Pressuremess";
  END;
END;

TempMeas ISA BasicThermo::ThermoTwoPole WITH
%%
%% Sets the terminal Temp to the temperature via a
%% tranfer function.
%%

```

```

parameters:
  a ISA Parameter WITH default := 1.0; END;
  %% Parameter in the denominator.
  b ISA Parameter WITH default := 1.0; END;
  %% Parameter in the nominator.

terminal:
  Temp ISA Base::SimpleOutput WITH
  Graphic ISA Base::Layout WITH
    x_pos := 200.0;
    y_pos := 300.0;
    invisible := 1;
  END;
END;

equations:
  P = 0;
  out.T = in.T;
  Temp' = -a*Temp + b*in.T;

graphics:
  Graphic ISA Base::Layout WITH
    bitmap TYPE String := "OurBitmaps/Tempmess";
  END;
END;

LIBRARY Valves;

%%
%% Basic library of general valves.
%%

Valve ISA BasicThermo::ThermoTwoPole WITH
%%
%% Valve is a general valve.
%%

types:
  kv TYPE Real;
  %% kv is flow coefficient.

equations:
  P = sqrt(in.q)/sqrt(kv);
  out.T = in.T;

graphics:
  Graphic ISA Base::Layout WITH
    bitmap TYPE String := "OurBitmaps/Valve";
  END;
END;

ControlValve ISA Valve WITH
%%
%% Valve with the flow coefficient as a Terminal.

```

```

%%
terminals:
  kv ISA Base::SimpleTerminal WITH
    Graphic ISA Base::Layout WITH
      x_pos := 300.0;
      y_pos := 300.0;
      invisible := 1;
    END;
  END;

graphics:
  Graphic ISA Base::Layout WITH
    bitmap TYPE String := "OurBitmaps/ContrValve";
  END;
END;

LIBRARY Controllers;

%%
%% Basic library of Controllers.
%%

PIController ISA BasicThermo::ThermoModel WITH
%%
%% A model of a PI Controller.
%%
parameters:
  ref ISA Base::Parameter WITH default := 50.0; END;
  %% Reference value.
  k ISA Parameter WITH default := 2e-05; END;
  %% Gain of the controler.
  Ti ISA Parameter WITH default := 1e+06; END;
  %% Integration time.

types:
  e TYPE Real;
  %% The error.
  i TYPE Real;
  %% The integration part.

terminals:
  u ISA Base::SimpleOutput WITH
    Graphic ISA Base::Layout WITH
      x_pos := 400.0;
      y_pos := 150.0;
      invisible := 1;
    END;
  END;
  meas ISA Base::SimpleInput WITH
    Graphic ISA Base::Layout WITH
      x_pos := 0.0;
      y_pos := 150.0;

```

```
        invisible := 1;
    END;
END;

equations:
    e = ref - meas;
    i' = e/Ti;
    u = k*(e + i);

graphics:
    Graphic ISA Base::Layout WITH
        bitmap TYPE String := "OurBitmaps/Controller";
    END;
END;
```

