CODEN: LUTFD2/(TFRT-5445)/1-90/(1991)

Modelling and Control of a Dairy Filling Machine

Martin Kruciński

Department of Automatic Control
Lund Institute of Technology

November 1991

Department of Automatic Control
Lund Institute of Technology
P.O. Box 118
S-221 00 Lund Sweden

Document name
MASTER THESIS

Date of issue
January 1991

Document Number
CODEN: LUTFD2/(TFRT-5445)/1-90/(1991)

Author(s)
Martin Kruciński

Supervisor
Bert-Ove Bergman, Åke Blomqvist, Tore Hägglund

Sponsoring organisation
Tetra Pak

Title and subtitle
Modelling and Control of a Dairy Filling Machine

Abstract

The goal of this Master Thesis was to model a filling system under investigation at Tetra Pak in Lund, Sweden. The model should be based on physical equations and not be just an ARMAX-model obtained from process identification. The model can handle laminar and turbulent flow of Newtonian liquids and laminar flow of non-Newtonian liquids. The model was implemented in Simnon. Experiments were made to validate chosen parts of the model. The experimental results were processed in Matlab. Two different regulator structures to control the filler were tested, one based on the regulator used today, position reference, and one based on new ideas, a volume reference regulator operating totally in closed loop. Simulation results have shown that the flow errors in the model are around 10 % compared with the real filler. This is quite good considering the raw physical models used to describe fluid and filler dynamics. The regulators both gave satisfactory simulated results.

Key words
Automatic Control, Modelling, Simulation, Fluid Dynamics

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title

ISBN

Language
English

Number of pages
90

Security classification

Recipient's notes

The report may be ordered from the Department of Automatic Control or borrowed through the University Library 2, Box 1010, S-221 03 Lund, Sweden, Telex: 33248 lubbis lund.

Master Thesis

Modelling & Control of a Dairy Filling Machine

Martin Kruciński, D-87

Lund Institute of Technology

November 6, 1991

Contents

1 Abstract

The goal of this Master Thesis was to model a filling system under investigation at Tetra Pak in Lund, Sweden. The model should be based on physical equations and not be just an ARMAX-model obtained from process identification. The model can handle laminar and turbulent flow of Newtonian liquids and laminar flow of non-Newtonian liquids. The model was implemented in Simnon. Experiments were made to validate chosen parts of the model. The experimental results were processed in Matlab. Two different regulator structures to control the filler were tested, one based on the regulator used today, position reference, and one based on new ideas, a volume reference regulator operating totally in closed loop. Simulation results have shown that the flow errors in the model are around 10 % compared with the real filler. This is quite good considering the raw physical models used to describe fluid and filler dynamics. The regulators both gave satisfactory simulated results.

2 Symbol Table

Symbol	Description	Unit
a	acceleration	m^2
A	area	m^2
D	diameter	m
F	force	N
g	gravity acceleration	m/s^2
h	height	m
K	power law parameter	Pl
L	length	m
m	mass	kg
n	power law parameter	
Δp	pressure loss	Pa
p	pressure	Pa
R	radius	m
Re	Reynold's number	
t	time	s
v	velocity	m/s
V	volume	m^3
η	Newtonian fluid viscosity	Pl
$\dot{\gamma}$	shear rate	s^{-1}
λ	loss parameter for Newtonian fluids	
π	constant	
Φ	flow	m^3/s
ρ	density	kg/m^3
σ	shear stress	Pa
ζ	one-time loss factor	

3 Introduction

The goal of this Master Thesis was to develop a simulation model for one of the filling techniques currently investigated at Tetra Pak. If time allowed, different control strategies should be tested too.

The questions to be answered were if we can model a filling mechanism and the flow equations sufficiently good by using quite rude physical models, or if it is necessary to develop advanced physical models to make the simulations correspond to real experiments. We also wanted to see if simulations can aid the development of new filling techniques, machines and regulators.

We chose to implement the model in Simnon, a simulation package. Matlab, a mathematical package, was used for some analysis of simlations and experiments. These packages are very wide spread and are a good platform for developing models. Thus, if the results were satisfactory, Tetra Pak could build up its own simulation environment without too much effort.

This report has been written with the LaTeX document preparation system. Its manual [4] has been of great help.

I would like to thank everybody at Tetra Pak that have helped me with my work. A special thanks to my supervisors Bert-Ove Bergman and Åke Blomqvist at Tetra Pak and Tore Hägglund at the Department of Automatic Control at Lund Institute of Technology for their help and support throughout my work.

Lund, November 6, 1991
Martin Kruciński

4 The Filler

4.1 Introduction

The filler I have modeled is one of the filling techniques that are investigated at Tetra Pak. It fills liquid into prefabricated packages that move under a filling valve. The filler is supposed to be able to handle different products. The packages move in a long row under the filling valve. When it is time for filling a package is lifted up by an AC-motor driven lift and then slowly lifted down simultaneously as the filling valve opens and fills the package. When the package is back at its original position all the packages are moved one step forward (indexing) and the procedure repeats itself. The filling time is 690 ms and the indexing time is 720 ms.

An overview of the filler is shown in figure 1.

4.2 The product pipe

The product comes to the filler through a product pipe that is usually situated quite high. The pressure in this pipe can vary quite a lot because other fillers connected to the same product pipe take different amount of product at different times. Also, the unit producing the product can give pressure surges in the pipe system.

4.3 The product tank

At the top of the filler a big tank is located. It can contain around 50 l of product. This tank separates the product pipe system (with its pressure surges) and the filling system of the filler. It also acts as an reservoir if the product flow is interrupted for a short while.

The product level in the tank is held constant by a regulator that opens and closes the product inlet to the tank. This regulator is not as fast as the filling mechanism so it adjusts the inlet flow to the average outlet flow from the filler. But this means that the product level varies somewhat during operation: decreases when the filling valve is open and increases when the valve is closed. This affects the total driving pressure (p_{filler}) that "pushes" the liquid through the filler.

4.4 The Flowmeter

On its way down from the product tank the product passes the flowmeter. The flow meter works electromagnetically, without any parts being in contact with the product except two electrodes. These are placed on each side of the pipe that goes through the flowmeter. The flowmeter generates a

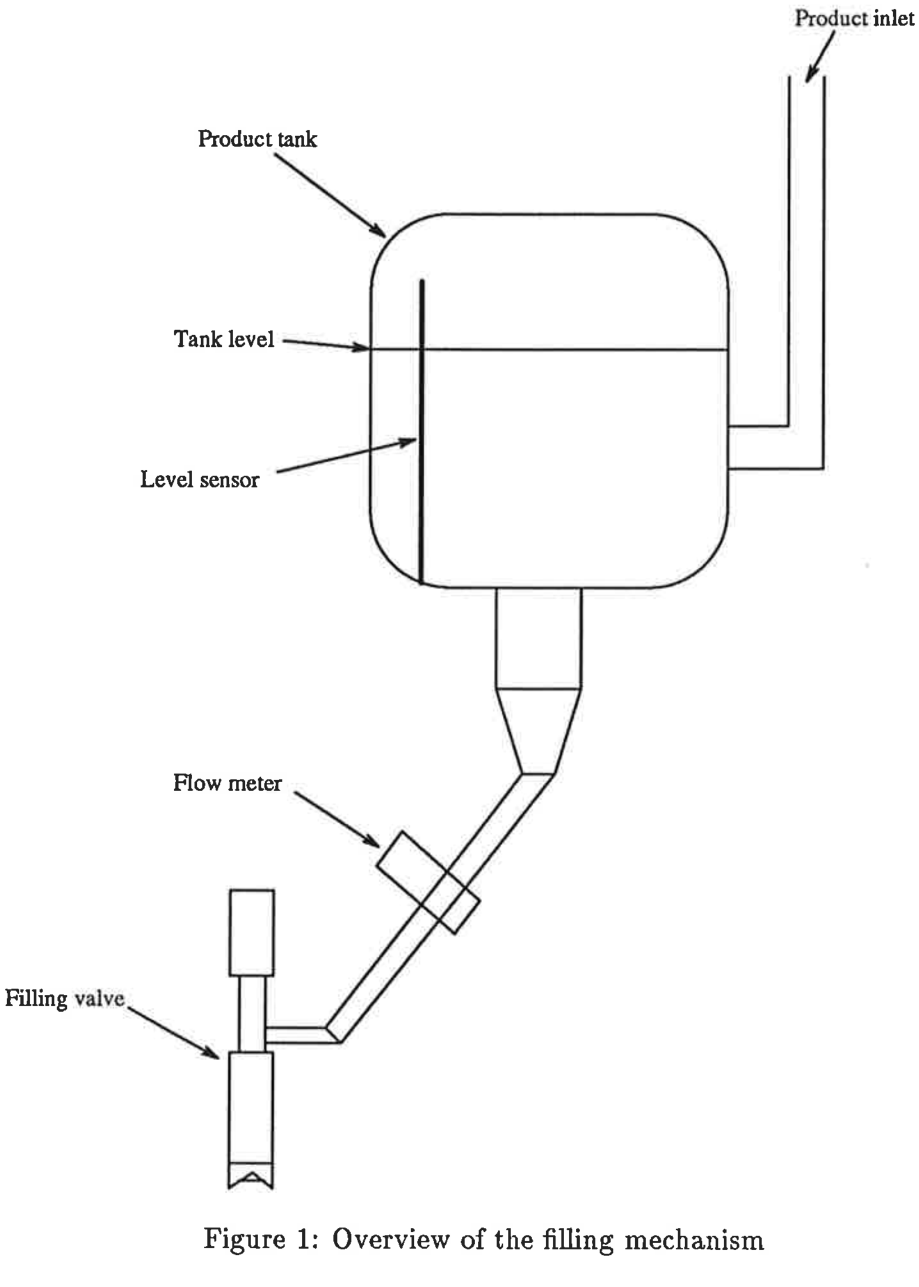

Figure 1: Overview of the filling mechanism

magnetic field inside this pipe. When electrically loaded particles pass the pipe they are acted upon by an electromagnetic force. This gives rise to a small voltage difference between the two electrodes. This signal is further processed by the flow meter and is converted to the actual flow through the meter.

4.5 The Pipes

The pipes in the filler are stainless, circular pipes. The filler has been equipped with quite wide pipes compared to other filling fillers at Tetra Pak. This makes it easier to fill thick products.

4.6 The Filling Valve

The filling valve consists of 4 main parts

1. AC-motor
2. Gear box
3. Product inlet
4. Valve tube

An overview of the filling valve is shown in figure 2.

When driving the motor the valve piston moves up and down. In its upper position the valve is closed, and in its lower position, it is open. The valve tube is a bit smaller than the packages so that is can be inserted into the packages at filling time. At the end of the valve tube there is a nozzle where the liquid leaves the valve tube. The piston position has been specified in mm throghout this report. During production the valve operates between the positions 1 and 22.5 mm. 1 mm is the closed position. The valve starts to open at 5.6 mm and its maximal opening postion is 22.5 mm.

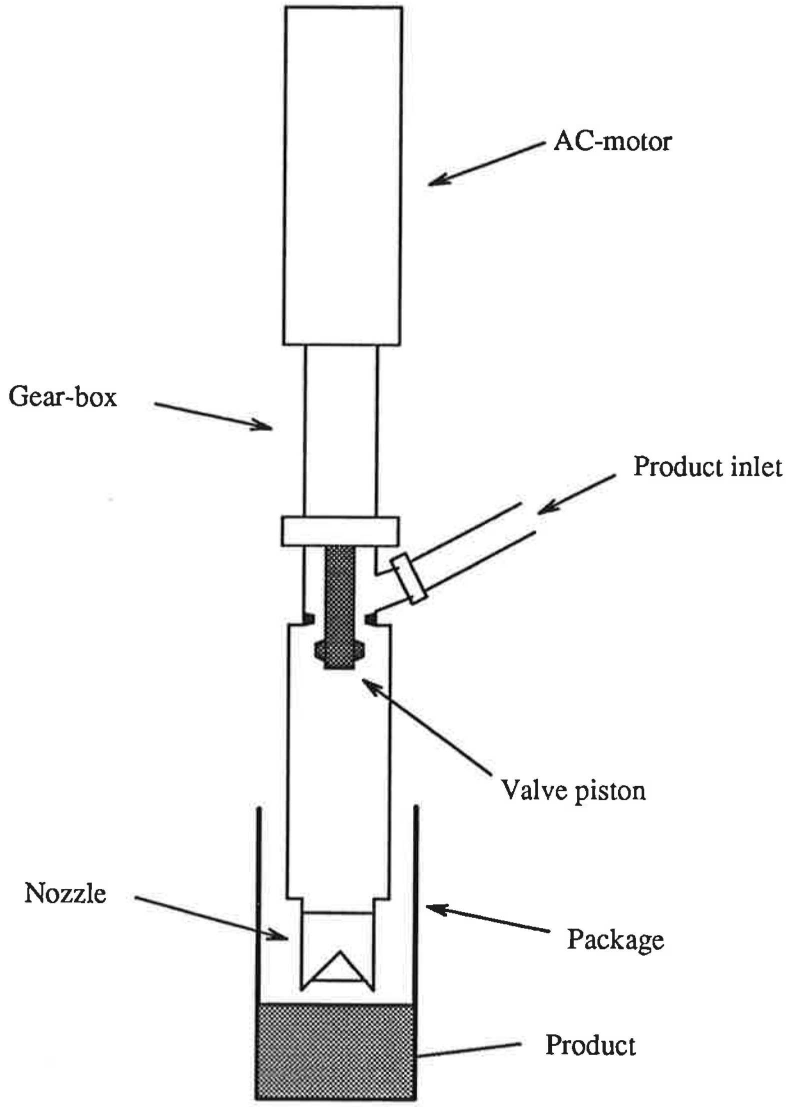

Figure 2: Overview of the filling valve

5 Fluid dynamics

To be able to define a model of the filler we first have to investigate how different fluids behave when set in motion. We want to take a closer look at liquid viscosity, friction losses (due to viscosity, fittings and elbows), other losses due to transformations between static and kinetic energy and the acceleration of the liquid in the filler.

5.1 Viscosity

When dealing with fluids we often need to specify how much resistance a fluid presents to flow. Fluids such as syrup and heavy oil flow much less easily than water. You often do not have such a good knowledge of a liquid's viscosity as of its other physical properties such as density and colour. The relative difference in viscosity between the products the filling machine is supposed to handle can be as much as 30 000!

The viscosity can be illustrated by the following experiment:

Assume we have an experimental setup as depicted in figure 3. Two parallel plates, each of area A are separated by a distance L. The top plate can move freely but the bottom plate is fixed. The region between the plates is filled with a fluid whose viscosity we shall denote by η. In order to move the top plate with speed v a force F is required. We define the viscosity, η, as:

$$\eta = \frac{\sigma}{\dot{\gamma}} \quad \text{(Pl, Poiseulle)}$$

$$\sigma = \frac{F}{A} \quad \text{shear stress(N/m}^2\text{)}$$

$$\dot{\gamma} = \frac{v}{L} \quad \text{shear rate(s}^{-1}\text{)}$$

The shear stress (σ) is a normalized force with respect to the area it is acting on, i.e. a pressure. The shear rate (γ) is a velocity gradient that reflects how fast neighbouring layers in the fluid move relatively eachother.

If we know all the geometrical dimensions and the viscosity in advance the force needed in the above experiment can be calculated as

$$F = \frac{\eta v A}{L} \tag{1}$$

We will later need formulas similar to equation 1 to calculate the friction losses in the machine.

Fluids can behave in a number of ways when set in motion. Some fluids have the same viscosity regardless of the shear rate $\dot{\gamma}$, Newtonian fluids. Others get "thicker" or "thinner", non-Newtonian fluids. The properties of some types of fluids are shown in figure 5.

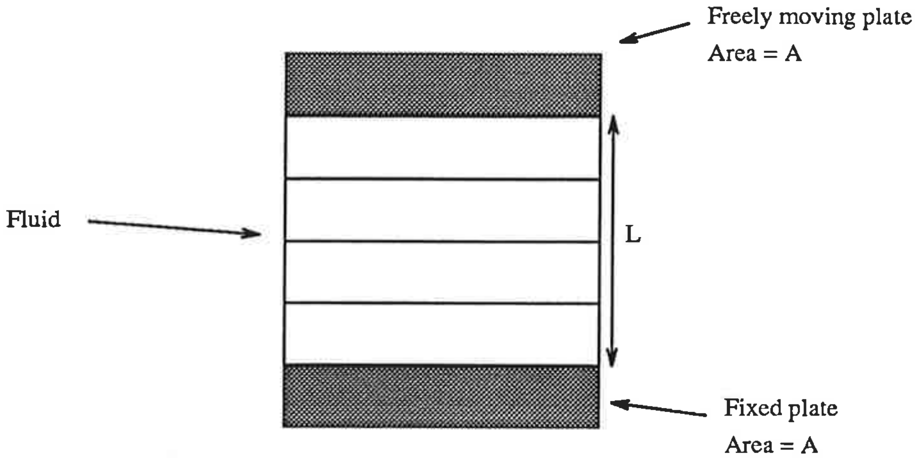

Figure 3: Experimental setup to determine the viscosity of a fluid

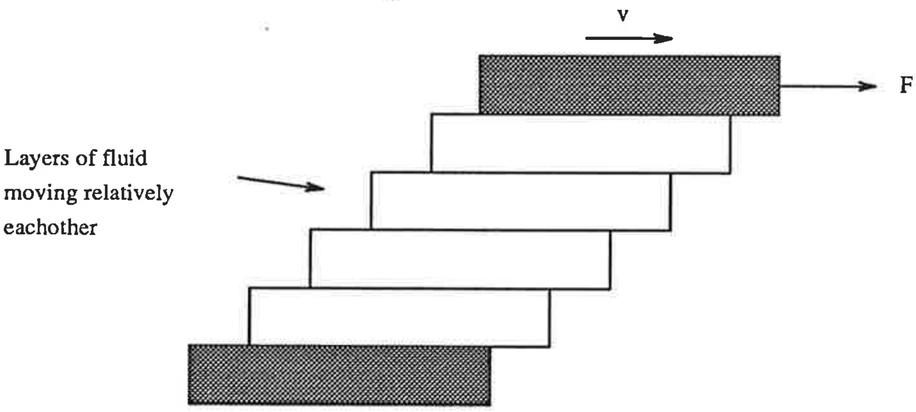

Figure 4: Moving the upper plate

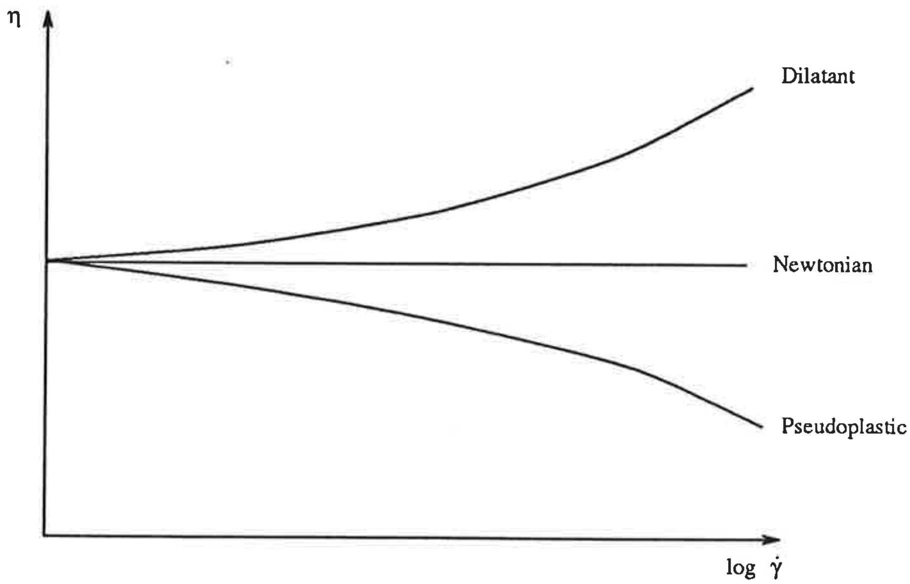

Figure 5: Differences between different fluids

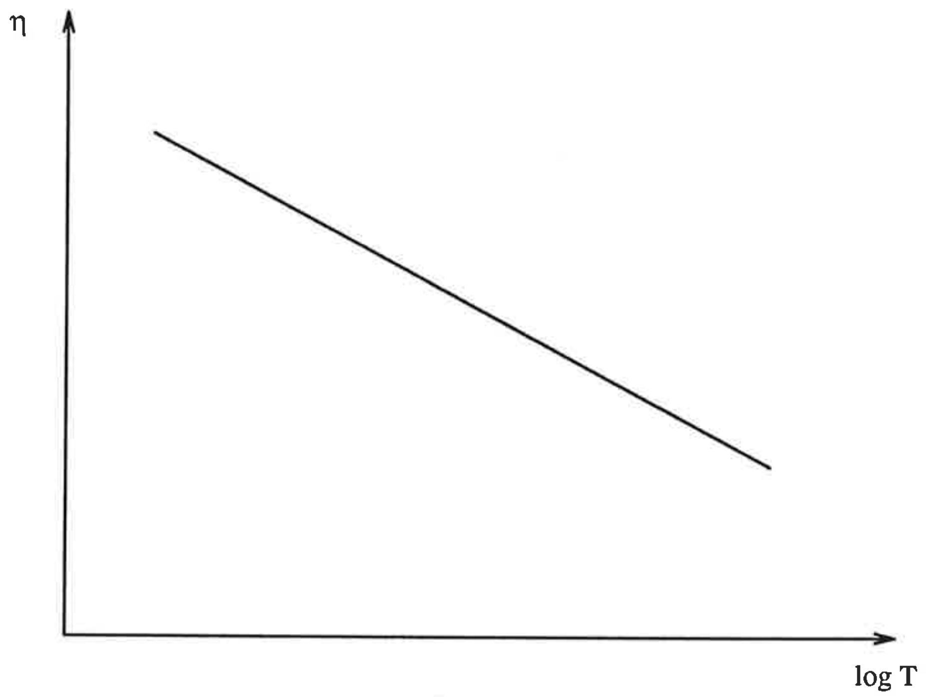

Figure 6: The temperature dependency of the viscosity

The different properties can be handled using different rheological models. The model for Newtonian fluids is

$$\sigma = f(\dot{\gamma}) = \eta\dot{\gamma}$$

Non-Newtonian fluids, dilatant and pseudoplastic, can be described by the Power Law

$$\sigma = f(\dot{\gamma}) = K\dot{\gamma}^n$$

where K is a consistency number (a constant similar to the viscosity for Newtonian fluids, but not the same) and n a dimensionless number that indicates the closeness to Newtonian fluids:

$n = 1$	Newtonian fluid
$n > 1$	Dilatant fluid
$n < 1$	Pseduoplastic fluid

The temperature also affects the viscosity. Liquids float differently when they get warmer. An example of the temperature dependency of the viscosity is shown in figure 6.

5.2 Time dependency

There is a number of fluids in which the shear stress, σ, is a function of both the shear rate, $\dot{\gamma}$, and the time to which it is subjected to a shearing force. These fluids are the so called thixotropic, shear thinning, rheopectic and shear thickening fluids. These aspects have not been included in the model since these effects would require a model where the velocity profile in every pipe is known. One could then calculate which parts of the liquid undergo structural changes and model the effect on the viscosity. Doing these calculations would give us a very complicated model far outside the scope of this model. Also, effects on the viscosity due to chemical and physical processes are not considered.

Examples of different products and their viscosities are shown in table 1 and table 2.

Product (Newtonian)	$\eta * 10^{-3} Pl$
water	1
milk	3
juice	10

Table 1: Example of viscosity of Newtonian fluids

Product (non-Newtonian)	n	K
Raspberry cream	0.35	28

Table 2: Example of viscosity of non-Newtonian fluids

5.3 Pressure losses

When fluids flow through pipes there are energy losses due to the viscosity of the fluid. These losses can be expressed as pressure losses, see [2], [6] and [7]. They are different depending on if the flow is turbulent or laminar. Details of how to calculate the pressure losses for Newtonian fluids are presented in [3] and [6]. Pressure losses for non-Newtonian fluids are handled in [8] and [6].

The following symbols are used in the friction loss calculations:

η – Viscosity (Newtonian fluid)
n K – Viscosity constants (Non-Newtonian fluids)
λ – Loss parameter
D – Pipe diameter
L – Pipe length
v – Average product velocity
Φ – Flow
ρ – Density
Re – Reynolds's number

We have laminar flow when $Re < 2300$, and turbulent flow when $Re > 2300$. Since we have both laminar and turbulent flow in the filler with Newtonian fluids we use different values of λ for these fluids For non-Newtonian fluids we have assumed that we always have laminar flow. (Verified in the simulations).

5.3.1 Newtonian fluids

Friction losses:

$$\Delta p = \frac{\lambda L \rho v^2}{2D}$$

$$\lambda_{Re<2300} = \frac{64}{Re}$$

$$\lambda_{Re>2300} = \frac{0.316}{\sqrt[4]{Re}}$$

Reynold's number:

$$Re = \frac{\rho v D}{\eta}$$

5.3.2 Non-Newtonian fluids

Friction losses:

$$\Delta p = \frac{4KL}{D}\left(\frac{3n+1}{4n}\right)^n \left(\frac{32\Phi}{\pi D^3}\right)^n \qquad Re < 2300$$

Reynold's number:

$$Re = \frac{D^n v^{2-n} \rho}{8^{n-1} K \left(\frac{3n+1}{4n}\right)^n}$$

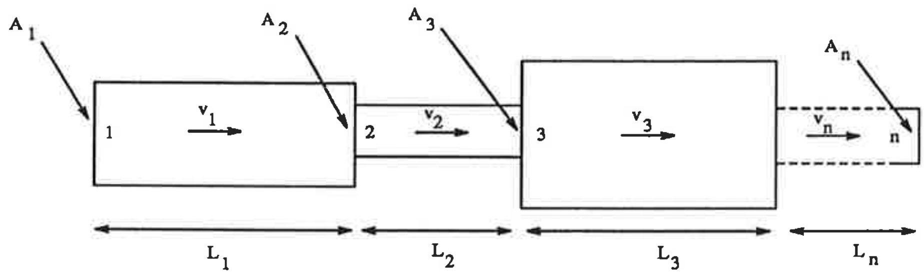

Figure 7: Pipe system used in calculations of liquid acceleration

6 Liquid Acceleration

6.1 Acceleration of the liquid

Assume we have a pipesystem as depicted in figure 7. Also assume that the flow is incompressible, i.e. ρ is constant and that there are no friction losses. These are the symbols used in the calculations

p_i	pressure at point i
h_i	height of point i
L_i	length of pipe-segment i
Δp_i	pressure over pipe-segment i
g	gravity acceleration factor
v_i	liquid velocity in pipe-segment i
F_i	force acting on the liquid in pipe-segment i
a_i	acceleration of the liquid in pipe-segment i
A_i	cross section area of pipe-segment i
Φ_i	liquid flow in pipe-segment i
V_i	volume of the liquid in pipe-segment i
m_i	mass of the liquid in pipe-segment i
ρ	liquid density
h_{tot}	total height between inlet and outlet of the filler
p_{tank}	the pressure in the product tank (relative 1 atm)

Equations 2–12 are used to calculate the acceleration of the liquid in the pipe system. The Bernoulli equation is further explained in [3] and [6].

Bernoulli's equation reads

$$p_i + \rho g h_i + \frac{1}{2}\rho v_i^2 = p_{i+1} + \rho g h_{i+1} + \frac{1}{2}\rho v_{i+1}^2 \tag{2}$$

Rewrite equation 2

$$\Delta p_i = p_i - p_{i+1} = \overbrace{\frac{\rho}{2}(v_{i+1}^2 - v_i^2)}^{\text{dynamic loss}} + \overbrace{\rho g(h_{i+1} - h_i)}^{\text{static loss}} \tag{3}$$

The force acting on the liquid in pipe segment i is

$$F_i = \Delta p_i A_i$$

The acceleration is force divided by mass

$$\frac{dv_i}{dt} = a_i = \frac{F_i}{m_i} = \frac{\Delta p_i A_i}{\rho V_i} = \frac{\Delta p_i A_i}{\rho L_i A_i} = \frac{\Delta p_i}{\rho L_i} \tag{4}$$

The flow and the acceleration of the flow in pipe segment i is

$$\Phi_i = v_i A_i \overset{\frac{d}{dt}}{\Longrightarrow} \frac{d\Phi_i}{dt} = \frac{dv_i}{dt} A_i = a_i A_i = \frac{\Delta p_i A_i}{\rho L_i} \tag{5}$$

The law of continuity gives

$$\Phi_i = \Phi_{i+1} \Longrightarrow \frac{d\Phi_i}{dt} = \frac{d\Phi_{i+1}}{dt} \tag{6}$$

Substituting the acceleration of flow with equation 5 gives

$$\frac{\Delta p_i A_i}{\rho L_i} = \frac{\Delta p_{i+1} A_{i+1}}{\rho L_{i+1}} \Longrightarrow \frac{\Delta p_i A_i}{L_i} = \frac{\Delta p_{i+1} A_{i+1}}{L_{i+1}}$$

Since the above relation holds for all i we can write it as

$$\frac{\Delta p_1 A_1}{L_1} = \frac{\Delta p_2 A_2}{L_2} = \cdots = \frac{\Delta p_N A_N}{L_N}$$

This is the mass transport acceleration and is constant in all pipes due to the law of continuity. The above expression has the SI-unit:

$$\frac{N/m^2\ m^2}{m} = \frac{N}{m} = \frac{kg\ m\ s^{-2}}{m} = \frac{kg}{s^2}$$

We can now write Δp_i as

$$\Delta p_i = \Delta p_1 \frac{A_1}{L_1} \frac{L_i}{A_i}$$

The total pressure over the whole pipe system p_{tot} is

$$p_{tot} = \sum_1^N \Delta p_i = \sum_1^N \Delta p_1 \frac{A_1}{L_1} \frac{L_i}{A_i} = \Delta p_1 \sum_1^N \frac{A_1}{L_1} \frac{L_i}{A_i} \tag{7}$$

Rewriting equation 7 gives us another way to express the pressure over pipe segment 1

$$\Delta p_1 = \frac{p_{tot}}{\sum_1^N \frac{A_1}{L_1}\frac{L_i}{A_i}} \tag{8}$$

The acceleration of the liquid in pipe segment 1 is (see eqution 4)

$$a_1 = \frac{\Delta p_1}{L_1 \rho} \tag{9}$$

Let us rewrite the pressure over pipe-segment 1, Δp_1

$$\Delta p_1 = \frac{p_{tot}}{\sum_1^N \frac{A_1}{L_1}\frac{L_i}{A_i}} = \frac{p_{tot}}{\frac{A_1}{L_1}\sum_1^N \frac{L_i}{A_i}} = \frac{p_{tot}}{\frac{A_1}{L_1}\Psi} \tag{10}$$

where

$$\Psi = \sum_1^N \frac{L_i}{A_i}$$

The acceleration of the liquid can now be expressed with p_{tot}, Ψ and ρ by substituting Δp_1 (equation 10) in equation 5.

$$\frac{d\Phi}{dt} = a_1 A_1 = \frac{\Delta p_1}{L_1 \rho} A_1 = \frac{\frac{p_{tot}}{\frac{A_1}{L_1}\Psi}}{L_1 \rho} A_1 = \frac{A_1}{L_1}\frac{1}{\rho}\frac{p_{tot}}{\frac{A_1}{L_1}\Psi} = \frac{p_{tot}}{\rho \Psi} = \frac{p_{tot}}{\rho \sum_1^N \frac{L_i}{A_i}} \tag{11}$$

This completes the calculations of the acceleration of liquid.

6.2 Acceleration of the liquid in the filler

When calculating the acceleration of the liquid in the filler we start with Bernoulli's extended equation

$$p_i + \rho g h_i + \frac{1}{2}\rho v_i^2 = p_{i+1} + \rho g h_{i+1} + \frac{1}{2}\rho v_{i+1}^2 + \Delta p_{friction,i,i+1} + \Delta p_{one-time,i,i+1} \tag{12}$$

$\Delta p_{friction,i,i+1}$ are the friction losses (viscosity losses) in the pipesegment from point i to $i+1$.
$\Delta p_{one-time,i,i+1}$ are the one-time losses (due to contractions, elbows, sharp edges etc.) between the points i and $i+1$ in the pipesystem.

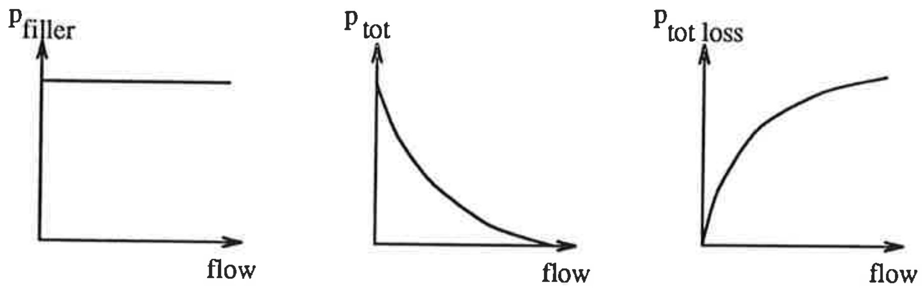

Figure 8: Comparision between different pressures in the machine

The friction and one-time losses appear only as pressure losses in the pipe-segments. This means we can use the results from the case when there are no friction losses etc. and decrease the total pressure with the sum of the losses. The total driving pressure that drives the liquid (p_{tot}) is then the total pressure between inlet and outlet of the filler (p_{filler}) minus the pressure losses in the filler ($p_{tot\ loss}$), i.e. the total pressure p_{filler} depends only on the tank pressure and the tank level. The losses $p_{tot\ loss}$ increase when the flow through the filler increases. This means that the driving pressure decreases as the flow increases, coming to 0 when the flow reaches stationarity conditions. All these pressures as function of the flow through the filler are shown in figure 8.

$$p_{tot} = p_{filler} - p_{tot\ loss}$$

$$p_{filler} = \rho g h_{tot} + p_{tank}$$

The total loss is the sum of all friction, one-time and dynamic losses. The static loss is accounted for in p_{filler} because it is actually one of the driving pressures of the liquid (the height of the filler).

$$p_{tot\ loss} = \sum_{1}^{n-1} \Delta p_{friction,i,i+1} + \sum_{1}^{n-1} \Delta p_{one-time,i,i+1} + \Delta p_{dynamic}$$

The dynamic pressure of the liquid entering the filler is

$$p_{dynamic\_inlet} = \frac{\rho}{2} v_{inlet}^2$$

This pressure is added to the total pressure that drives the liquid through the filler. But the system also loses dynamic pressure, see [9]. The liquid that leaves the filler takes it dynamic energy from the filler pipe-system. This results in a decrease of dynamic pressure that is

$$p_{dynamic\_outlet} = \frac{\rho}{2} v_{outlet}^2$$

The dynamic losses $\Delta p_{dynamic}$ are calculated as the difference in dynamic pressure at the beginning and end of the filler pipe-system.

$$\Delta p_{dynamic} = \frac{\rho}{2}(v_{inlet}^2 - v_{outlet}^2)$$

In the filler, the inlet velocity is the liquid velocity in the tank, which can be approximated with 0. The outlet velocity is the velocity that the liquid leaves the filling valve with. In the model it is v_{10_11}. This gives

$$\Delta p_{dynamic} = -\frac{\rho}{2}v_{10\_11}^2$$

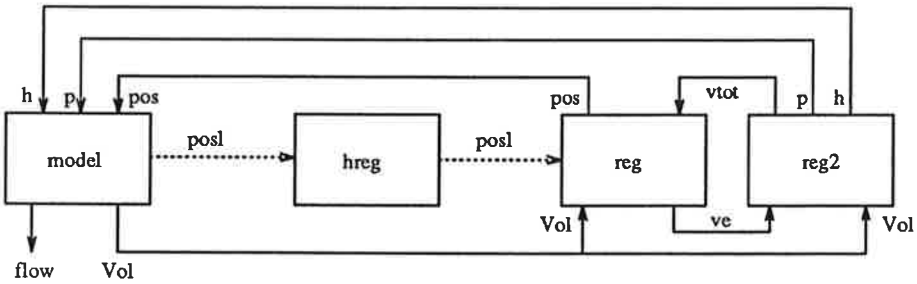

Figure 9: Overview of the Simnon systems for volume reference regulator

7 Simnon model

To simulate the filler we used a total of 4 systems and 2 macros for each regulator type, i.e. volume reference regulator and position reference regulator. An overview of the systems is shown in figure 9 and figure 10. The Simnon listings are found in appendix B.

7.1 What is Simnon?

Simnon is a simulating language developed at the department of Automatic Control at Lund Institute of Technology. It can simulate processes described by both differential (continuous time) and difference (discrete time) equations. Simulation results can be plotted and printed and also saved for later postprocessing by other programs, e.g. Matlab. Simnon facilitate a hiearchical description of systems so separate subsystems can be connected with regulators to simulate the control of a process. In the battles with Simnon we have seeked help in [1] and [5].

7.2 What is Matlab?

Matlab, matrix laboratory, is a numerical mathematics package capable of advanced calculations, filtering and plotting of data. It has been used to do the more complicated calculations involved in the modeling of the filler and for postprocessing and analysis of simulation and experimental data.

7.3 Model

Sampling interval: Time continuous system

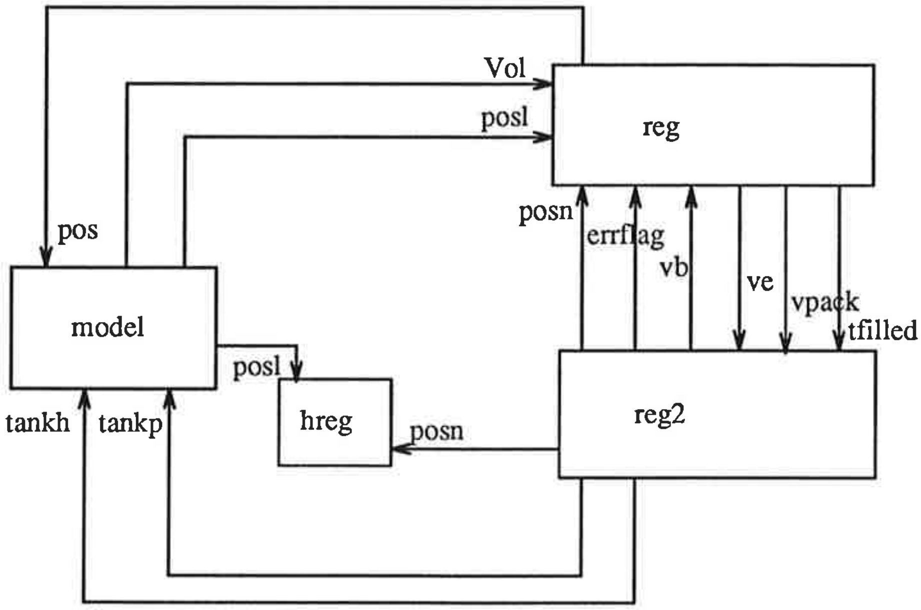

Figure 10: Overview of the Simnon systems for position reference regulator

Input:	tankh	Level in the product tank
	tankp	Pressure in the product tank
	pos	Filling valve piston position set point
Output:	Vol	Volume filled so far
	posl	The filling valve piston position. (This output is internal in the Simnon systems. It is only used by hreg to improve the simulations)

This system describes the physical and dynamic properties of the filler. Using the inputs it calculates the flow through the filler and the volume filled so far, Vol.

Remark: tankh and tankp are the tank level and tank pressure used in the model. pos is a filling valve *setpoint.* The acutal filling valve piston position, posl, is calculated internally.

All the pipes have been modeled as described in chapter 5.3.

All contractions, fittings and elbows have been modeled as one-time losses, i.e. losses localized to one specific point. The pressure losses are calculated as

$$\Delta p = \zeta \frac{\rho v_{highest}^2}{2}$$

where ζ is the loss factor and $v_{highest}$ the highest velocity the liquid travels with, either before or after the obstacle.

All the reference points used in the model can be seen in figure 11.

The filling valve has been simplified a lot in the model. It is modeled as a

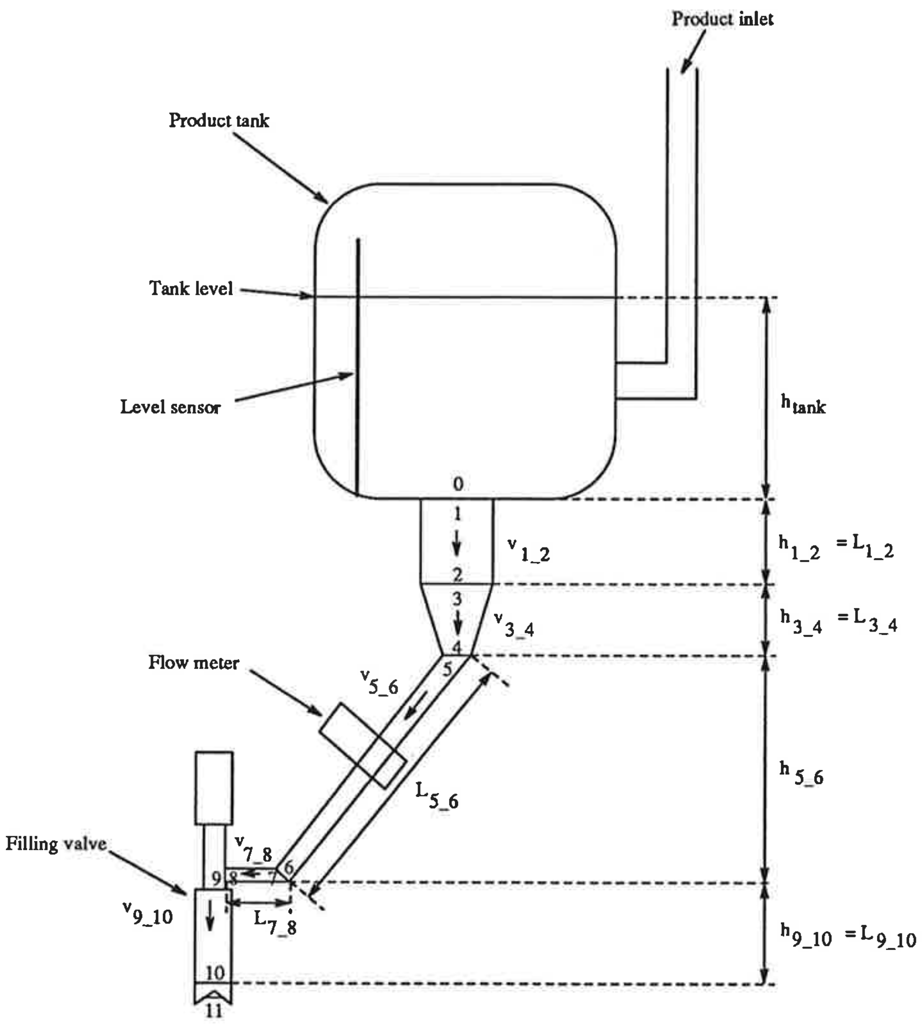

Figure 11: Overview of the filler with all reference points

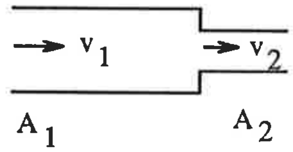

Figure 12: Simplified geometry of the filling valve

contraction with sharp edges, see figure 12. The formula used to calculate the pressure loss over the valve is the normal formula for calculating one-time losses

$$\Delta p = \zeta \frac{\rho v_2^2}{2}$$

$$v_2 = \frac{A_1}{A_2} v_1 = (A_1 v_1) \frac{1}{A_2} = \frac{\Phi}{A_2}$$

In the model we have built in a constant, poseps. This controls how far under the critical position the pressure surge is allowed to build up. A too short pressure surge does not have enough time to stop the liquid flow. The filling valve model in the simnon system does not close completely, but only to a desired level. Otherwise, the valve area would be zero and the simulations would crash. When the valve is open this little bit, a big pressure surge is built up that stops the liquid. By adjusting the constant poseps we can affect how far under the critical position these conditions should be in action. When closing even further the pressure loss is set to the total pressure and the liquid retardation cannot be affected any more (see the listing of system "model4.t" in appendix B). The pressure surge can be studied in figure 14.

We have modeled the filling valve dynamics as a second order system. The valve consists mainly of 3 parts, the AC-motor, the gear-box and the valve piston. The speed of the AC-motor can be modeled roughly as a first order system.

$$G_1(s) = \frac{K_m a}{s + a}$$

The helix screw gear-box can be viewed as an integrator with a specific gain. It integrates the motor speed to a piston position.

$$G_2(s) = \frac{K_h}{s}$$

The piston is a delimiter that limits the piston position to the range 0 to 22.5 mm. The whole thing is today controlled by a drive with an internal regulator. Everything can be viewed in figure 13.

Since it was hard to know the gain of the gear-box, K_h, the gain of the motor, K_m, and the regulator dynamics, we chose to model the filling valve as a second order system with 2 equal time constants, $\tau_1 = \tau_2 = \tau = 40$ ms. These time constants have been extracted from the experimental data in experiment 2, see page 35.

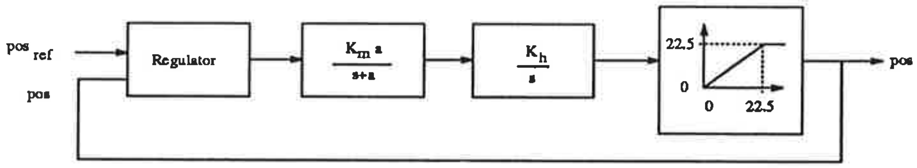

Figure 13: Overview of the filling valve dynamics

$$G(s) = \frac{\tau^2}{(s+\tau)(s+\tau)}$$

The dynamic loss has been modeled as described on page 21. The outlet velocity in the model is v_{10_11}. This velocity depends on the outflow area of the nozzle. Since the nozzle is flexible it has been difficult to measure this area. Different liquids might also give different areas. We have used the velocity inside the filling valve tube instead, v_{9_10}

7.4 Reg

Sampling interval: 10 ms

Volume reference regulator

Input:	Vol	Volume filled so far
	vtot	Volume filled at the start of this filling cycle
	posl	The filling valve piston position
Output:	ve	Volume error, difference between the volume reference and the package volume

Position reference regulator

Input:	Vol	Volume filled so far
	posl	The filling valve piston position
	posn	The filling valve normal opening position
	errflag	Flag that tells us if the pressure regulator is working
	vb	Batch volume level
Output:	ve	Volume error, difference between the volume reference and the package volume
	tfilled	The time the package was filled in
	vpack	The package volume

This system describes the control of the filling valve piston position, pos.

The *volume* reference regulator tries to follow a volume reference during filling. This reference is calculated by multiplying an externally, normalized volume reference signal by the actual package volume.

The *position* reference regulator sets pos to different prespecified values depending on the package volume.

7.5 Reg2

Sampling interval: Once every filling cycle

Volume reference regulator

Input:	ve	Volume error, difference between the volume reference and the package volume
Output:	vtot	The total volume filled into the packages at the start of this filling cycle
	p	The product tank pressure set point
	h	The product tank height set point

Position reference regulator

Input:	ve	Volume error, difference between the volume reference and the package volume
	tfilled	The time the package was filled in
	vpack	The package volume
Output:	errflag	Flag that tells us if the pressure regulator is working
	vb	Batch volume level
	posn	The filling valve normal opening position
	tankp	The product tank pressure set point
	tankh	The product tank height set point

This regulator controls the product tank pressure, p. It samples once at the end of every filling cycle and checks the volume error, ve. If the normal regulator has not managed to keep the filling error below certain limits it adjusts the tank pressure accordingly. The tank level, h, is not controlled in these version of regulator structure.

7.6 Hreg

Sampling interval: Varying

Input:	posl	The filling valve piston position
Output:		None explicit, sampling interval implicit

This is a subsystem needed only in the Simnon simulations, not in a real implementation. It sets down the internal sampling interval for all the systems to a very small value to "help" Simnon through numerically very difficult points in the simulations, see chapter 7.9. This is especially the cases when the filling valve piston position passes its closing point. The model is highly non-linear here and the simulation would crash despite the fact that Simnon has its own safety nets for handling such situations.

7.7 User defined functions

The file *func.t* contains the values of the three user defined functions used in the model. The first describes the filling valve area as a function of piston position. The second describes the filling valve loss factor, ζ_{valve}, as a function of piston position. The third specifies the normalized volume reference as a function of time. In the position reference regulator this file is called "onez.t" and contains only the two first functions.

7.8 Macros

Two macros have been written to aid the simulations. These macros contain Simnon commands for initialization and execution of simulations.

initflow translates the user defined function to Simnon readable form and initializes all the systems.

flow chooses simulation time, which variables should be saved during simulation, simulates the experiment and plots the desired results.

7.9 Simulation problems

One of the greatest problems throghout the simulations has been the point when the filling valve closes on its way down from a fully open position. The closing point is such a strong discontinuity that the simulation crashes or failes to capture what happens at the closing point. To handle this problem we made as follows.

We attached a discrete system, hreg, on top of the other Simnon systems. Now we had the ability to control the simulation step length through this system since Simnon never makes longer steps than prescribed by the discrete system with the shortest step length. This system simply adjusts its step length depending on how close the filling valve piston position is to the critical position, 5.6 mm.

Simultion results are shown in the figures 14, 15 and 16. The totloss is the total pressure loss in the filler. The posl, pos and poscrit are the filling valve piston position, its reference and the critical position respectively. The step length factor is directly proportional to the simulation step length. The flow is the flow through the filler.

One can specify the normal step length, normsper, used when the piston position is far from the critical position, and the critical step length, critsper,

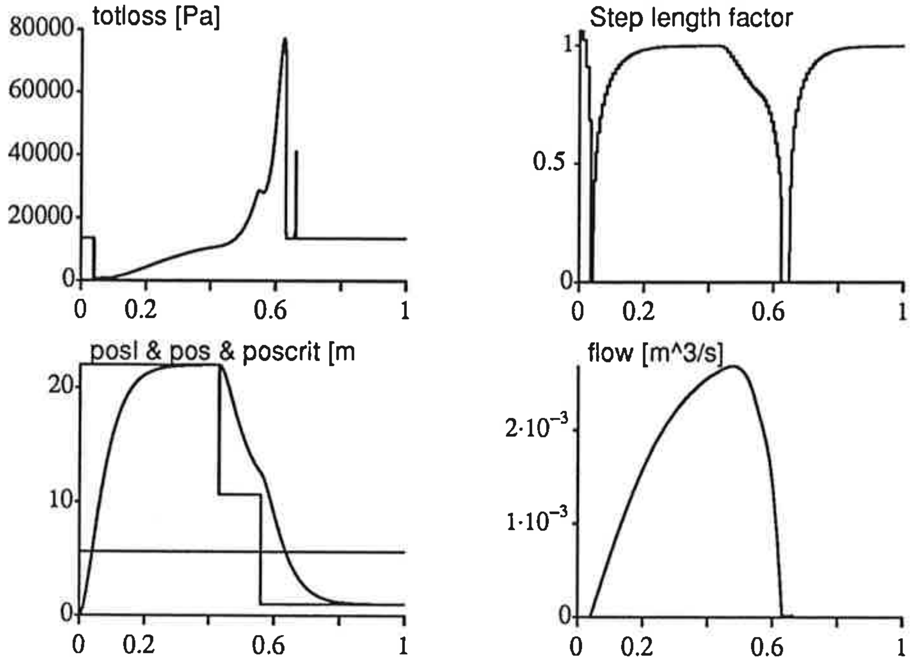

Figure 14: Simulation with cexp=0.25

used when the piston position is passing the critical position. The hysteresis around the critical position is controlled by the constants hu and hd. The transitions between short and long step length is controlled by the constant cexp. In figure 14 cexp=0.25. In figure 15 cexp=1.0. In figure 16 the hysteresis around the critical position has been increased.

The goal of adjusting these constants is to get a simulation step that is as long as possible when the piston position is far from the critical position. This to get as fast simulations as possible. It takes time to simulate with short step lengths. But we also want the step length to be at critsper when passing the critical position. If the simulations fail, try to adjust these constants or poseps in the system "model".

The problem with the critical piston position could also be handled by simulating only with a large step length. This means that the pressure surge required to stop the liquid that is built up at valve closing is missed. But the flow can be reset to zero by resetting the flow state variable when closing the filling valve beyond the critical position. This option has not been investigated since Simnon normally does not allow direct adjustements of the state variables. Neither can this method be used if it is desired to capture the pressure surge in the simulations.

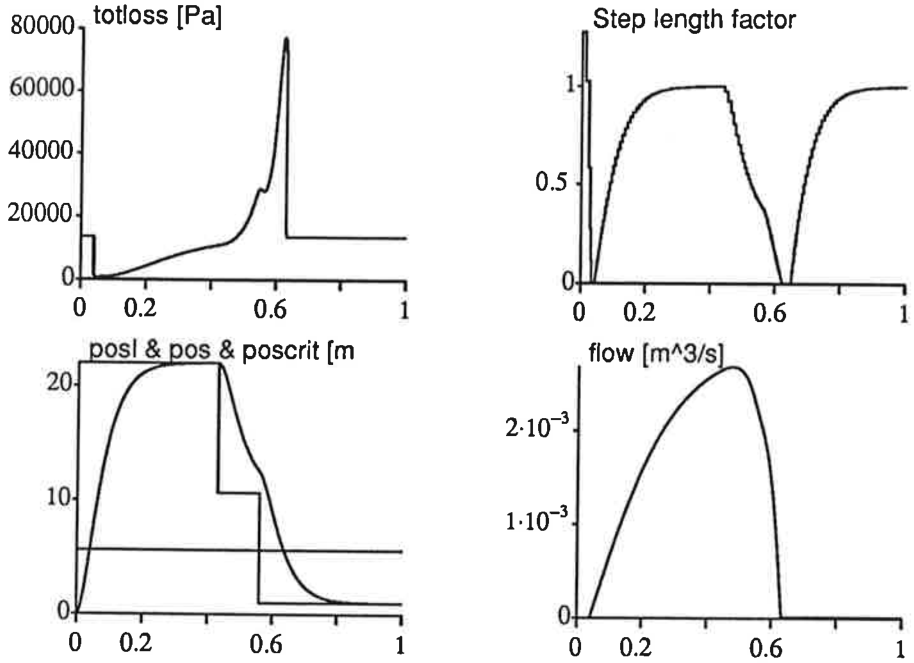

Figure 15: Simulation with cexp=1.0

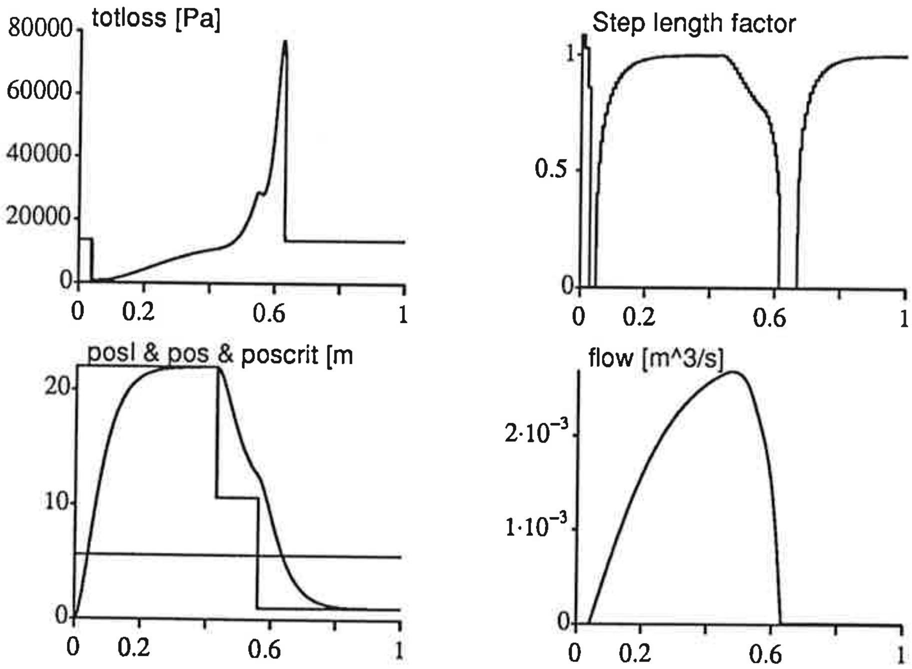

Figure 16: Simulation with large hysteresis around poscrit

8 Model Validation

8.1 Introduction

The aim of the experiments was to gather data that could later be used to achieve a correspondence between the filler and the model. I wanted to do experiments with both thin and thick products, e.g. yoghurt. But to test the filler with a thick product would be considerably more complicated than with a thin product. You have to prepare lots of thick product, at least some hundreds of litres of it, install a special pipe system so that you can get it into the product tank, take care of the outlet (you cannot let it go to the normal water sink) and clean the whole filler afterwards. The time available for the experiments thus gave us only one option, to test with water. This means that the model reliability for thicker products has *not* been investigated.

8.2 How use the experimental results?

The most unsure thing in the model are the pressure losses in the filling valve. The formulas for the frictional losses etc. are more reliable. Another difficult thing was the modelling of the acceleration of the liquid in the filler. The experiments that we have performed have been focused to improve the model acurracy at these two points.

8.3 Experimental procedure

The first part of the experiment was aimed at measuring the stationary flow of water through the filler at different openings of the filling valve.

The second part of the experiment was aimed at measuring the acceleration of the flow when the filling valve opens. We made only tests when the valve opens to its maximal nominal position. To see what effect different opening ramps had on the acceleration of the flow, we made several tests with different ramps, both slow and fast.

A test procedure was programmed to do these steps:

1. Ensure that the tank level is at a prespecified level before each experiment. How this is done is shown in figure 17. First the tank level is decreased (I). Then it is increased until the desired start level is reached (II), and then the experiment starts (III). This procedure assured that we always start every experiment at the same conditions, i.e. at the same tank level every time.

2. Open the filling valve according to a reference. During this step the

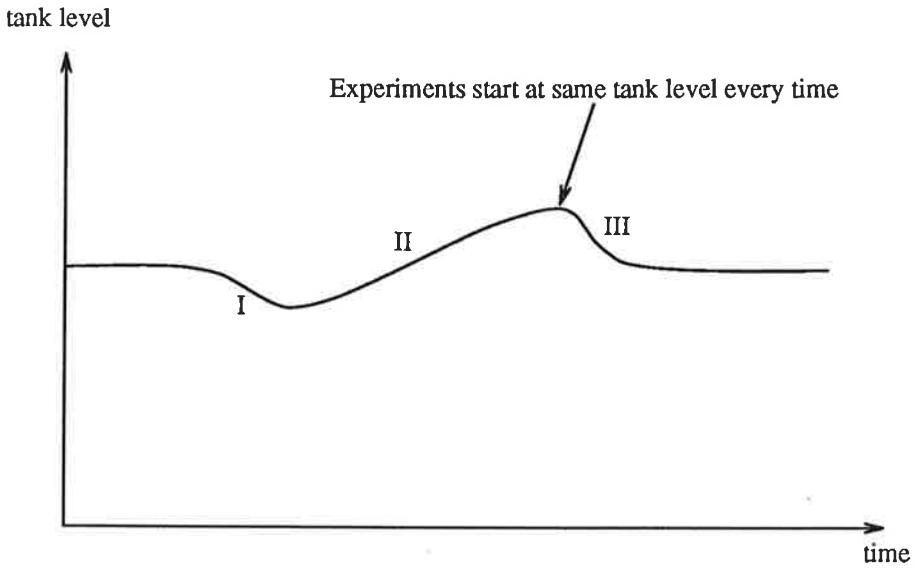

Figure 17: The variations of the tank level during one experiment

product tank level regulator is running with parameters set to minimize the level variations.

These steps were repeated every time flow measurements were made.

All measurements were done using a Hewlett-Packard measurement computer. The flowmeter gives out pulses with a frequency proportional to the flow. The maximal frequency, 10 kHz corresponds to a flow of 4.000 l/s. The computer counts these pulses during a prespecified time, 50 ms in the first experiment and 10 ms in the second experiment. These values are recorded and the counter reset to count pulses during a new period. During the second experiment the measuring period of 10 ms could not be held stable because of hardware trouble. The measurement period varied between 9–11 ms but its mean value was 10 ms. This explains the more "noisy" flow values in this experiment. We have tried to improve the data quality by postprocessing the data.

8.4 Postprocessing of measurement data

All data recorded at the experiments were transferred to the workstations at Lund Institute of Technology. There we have used Matlab to average and filter the data. All data from experiments that were run several times have been averaged to one file. This to get statistically more confident data to work on.

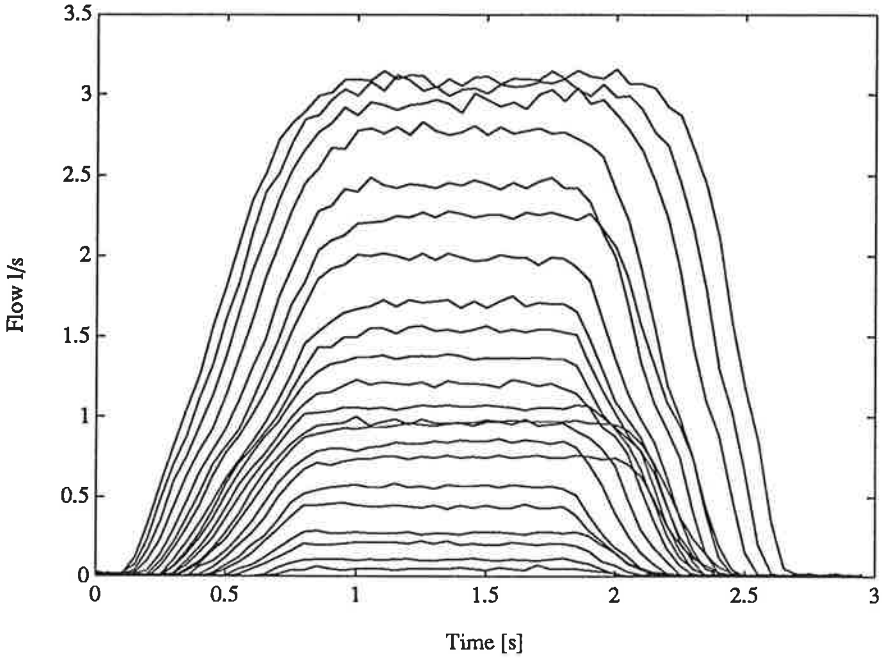

Figure 18: The flows through the filler in experiment 1

8.5 Postprocessing of data from experiment 1

From this data we wanted to extract the stationary flow as a function of the valve piston position. The results from the experiments are shown in figure 18. We wrote a Matlab function "flow", as shown in appendix A. It takes the measurements and extracts the part containing the flow values in stationarity. The mean of these values is converted to flow in litres per second and adjusted according to the height variations in the tank during the experiment. The routine also converts the valve piston position, from the internal units used in the filler control software, to mm. Another program, "Analstat" in appendix A, used the Matlab flow function to calculate the flow for all the different piston positions that were investigated in the experiments. All these flow values were then put together to the static transfer function from the piston position to flow. The result is shown in figure 19.

8.6 Update of the model

We wanted to use the results from experiment 1 to calculate new values for the loss coefficient ζ_{valve} used in the model to describe the valve. This is because the modeling of the valve is the most unsure thing in the model, so it feeled right to modify this part. Here is a description of how the ζ_{valve} should be computed:

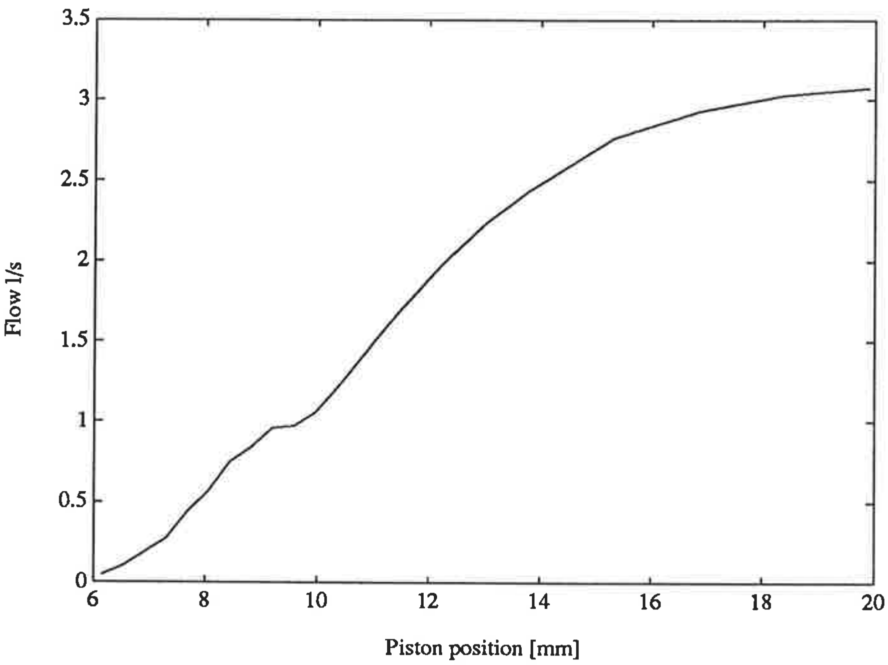

Figure 19: The static transfer function from valve piston position to flow

1. In the Simnon model, calculate the pressure loss in the whole filler, excluding the filling valve. Do this for the same flow values as in the experiments.

2. Calculate the valve velocity, v_{valve}, for every flow value.

3. The equations

$$p_{filler} = p_{tot} + p_{tot\ loss} \ , \ p_{tot} = 0 \Rightarrow$$

$$p_{filler} = p_{tot\ loss} = p_{loss\ excluding\ valve} + \zeta_{valve} \frac{\rho v_{valve}^2}{2}$$

gives us a way to express ζ_{valve} as

$$\zeta_{valve} = \frac{2(p_{filler} - p_{loss\ excluding\ valve})}{\rho v_{valve}^2}$$

$$v_{valve} = \frac{\Phi}{A_{valve}}$$

When comparing simulation and experimental data though, we discovered that the stationary flow in the simulation is *smaller* than the flow in the experiment. This is probably because of overestimation of some losses in the model that are not that big in reality, e.g. some one-time losses. Which one that is we can only guess. The right thing to do would be to measure the pressure losses over each part of the filler, i.e. every bend, fitting, the flow meter etc. We will then know the pressure losses over each part and can make a much better model.

8.7 Postprocessing of data from experiment 2

The measurements in experiment 2 were noisy. The gating signal that controlled when the counter in the measurement computer should count pulses from the flowmeter was not ideal. It was generated in the filling machine PLC/system. The only thing we knew about it was that it had a mean period of 10 ms but that this period varied between 9 to 11 ms.

To study the nature of the disturbances, we let Matlab calculate the covariance function of the stationary flow part of the signal. Before the covariance function was calculated we had to center the flow meter data around 0. The noisy flow and its correspondening covariance function are shown in the left part of figure 20. Now we could clearly see that the covariance function is jumping between 0.8 and -0.8 every sample near the top at 40 (corresponding to $\tau = 0$). So there is a strong opposite correlation between two on eachother following samples. This can be removed by averaging the flow values with an averaging window 2 samples wide, e.g. $flow_i = (flow_i + flow_{i+1})/2 \quad \forall i$. The program that does all this, "Flow", is shown in appendix A.

The filtered flow and its corresponding covariance function is shown in the right part of figure 20. From the covariance function we can see that the remaining noise values are almost uncorrelated with eachother and in the flow we see that they have a small amplitude. The data is now good.

During the experiments we made many experiments with different accelerations of the filling valve. All the results are shown in figure 21. Here we can see at what point we cannot accelerate the liquid and the filling valve any faster. We have averaged the results of the experiments with the fastest accelerations. This gave us a flow curve that we know has the fastest acceleration we can achieve. This is shown in figure 22. The dotted line is a fitted second order model to this flow curve. All the data analysis was done in Matlab. From the model, we could calculate the two time constants of the flow curve. The first, $\tau_1 = 0.160$ s. This is the time constant of the liquid. The other, $\tau_2 = 68$ ms. This is the time constant of the motor and filling valve. We have modeled the motor and filling valve as a second order system. So this constant, which refers to a first order system has to be converted to an equivalent time constant that we can use in our model of the filling valve dynamics. This has been done in figure 23. The equivalent two time constants for a second order system are around 40 ms. We estimated these before the experiments to 20 ms. Here we could see that they are somewhat higher and we adjusted the model accordingly.

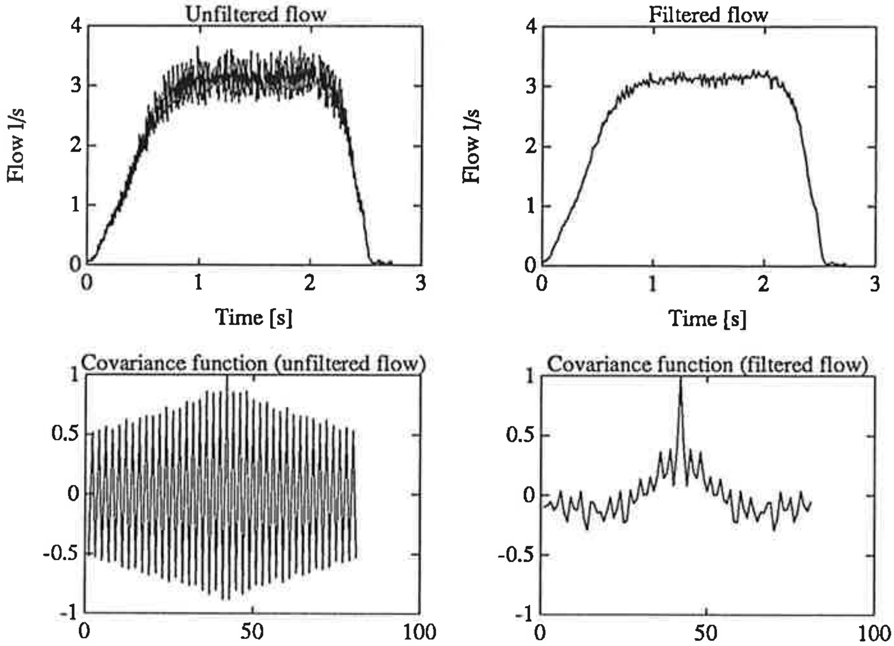

Figure 20: Unfiltered and filtered flow from experiment 2

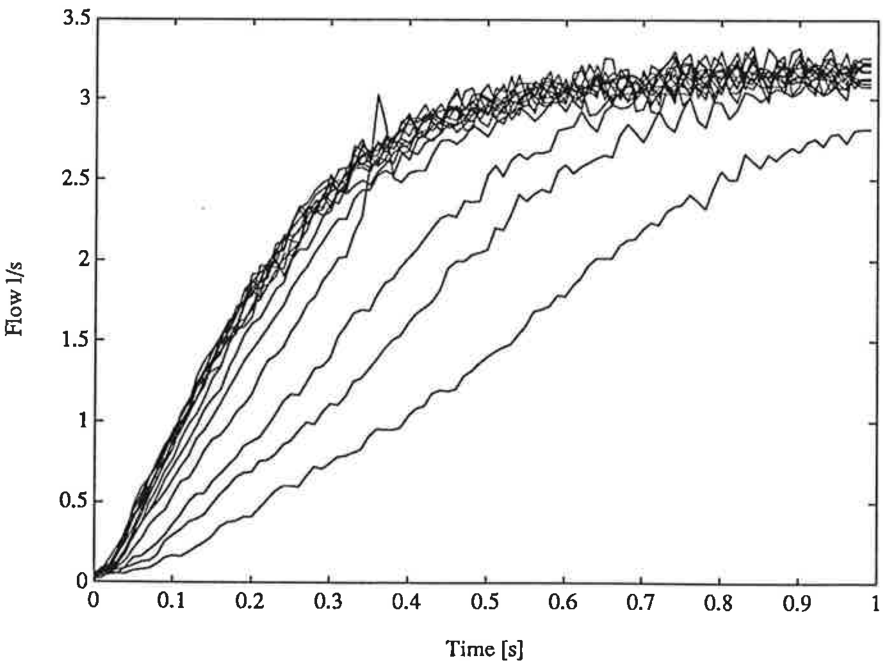

Figure 21: All acceleration profiles from experiment 2

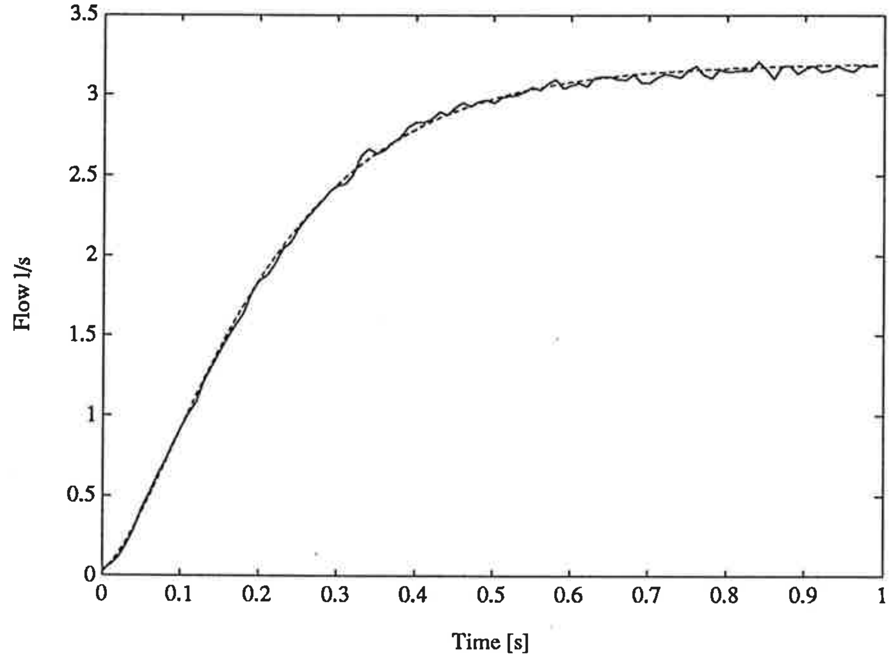

Figure 22: Averaged acceleration profile and fitted model

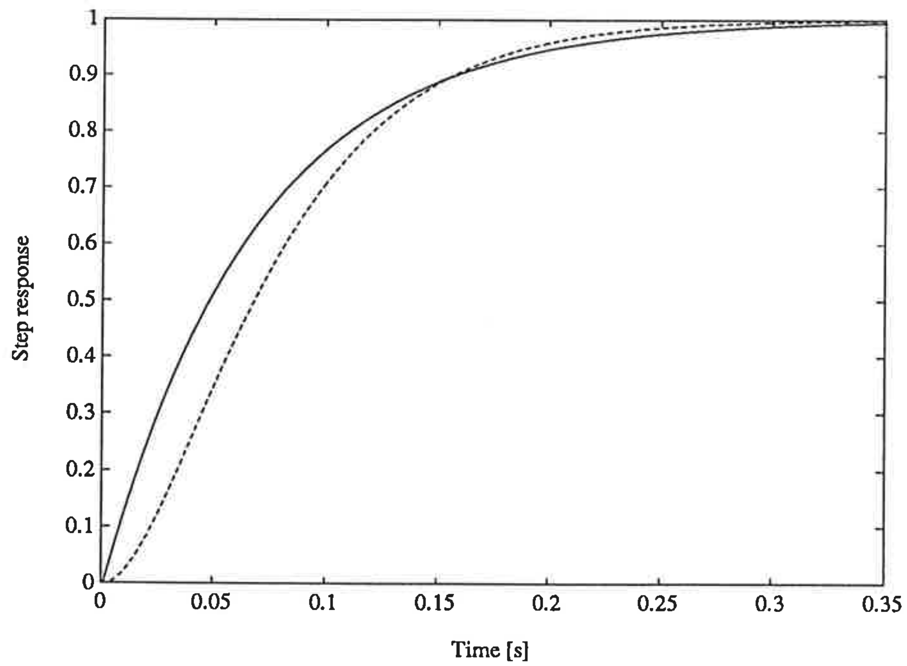

Figure 23: Calculation of the timeconstants of the filling valve

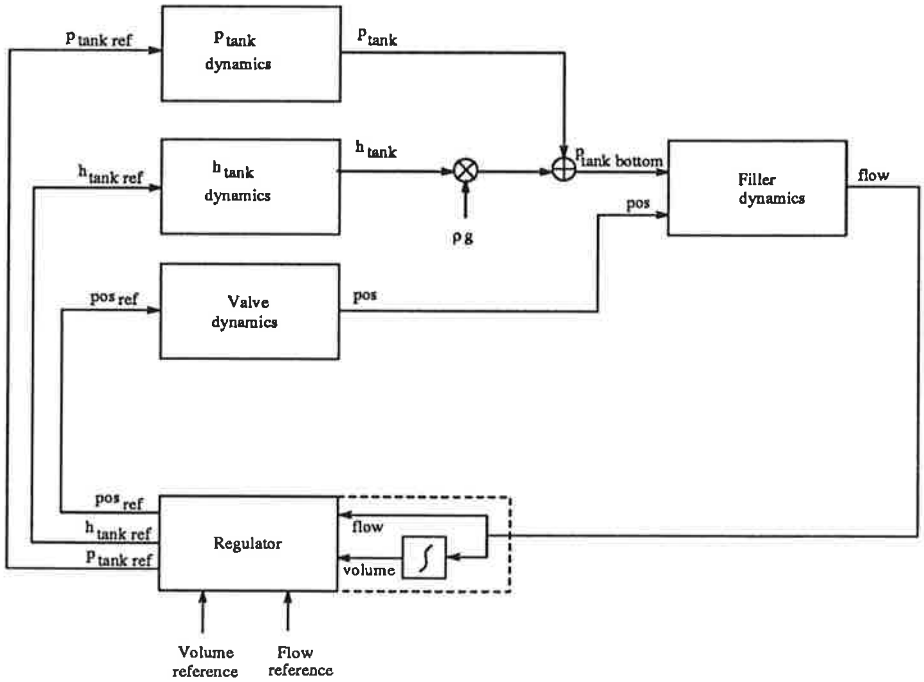

Figure 24: Overview of the regulator and process

9 The control problem

This part describes the control of the filler.

9.1 Aim of the regulator

The aim of the regulator is to fill the desired volume in the packages in a specified time. The filling errors should be as small as possible, preferably a few promille.

An overview of the whole control problem is presented in figure 24. The filler is represented by the transfer functions (shown as boxes) in the upper part of the figure. The regulator can be seen in the lower part.

Starting with the filler, the flow is mainly affected by the filling valve piston position (*pos*) and the pressure at the bottom of the product tank ($p_{tank\ bottom}$). This pressure is affected by the pressure in the product tank (p_{tank}) and by the product level (h_{tank}).

The regulator can measure the flow from the filler (and if wanted, after internal integration, also the volume). The regulator can control the filling valve piston position (pos_{ref}), the tank level ($h_{tank\ ref}$) and the tank pressure ($p_{tank\ ref}$). The changes in these variables do not affect the respective physical variables directly, but through the dynamic systems Valve dynamics, h_{tank} dynamics and p_{tank} dynamics.

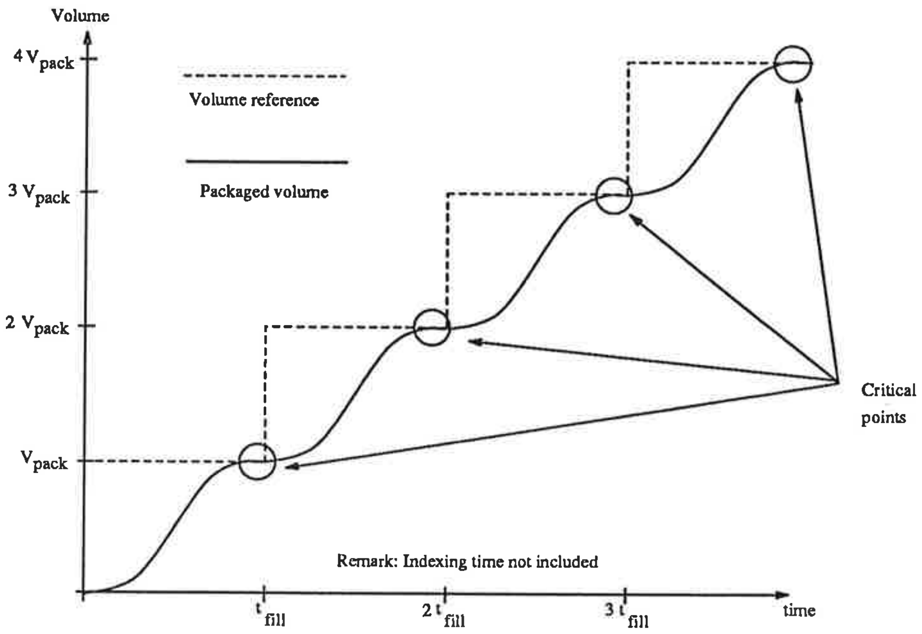

Figure 25: Specification of the desired regulator performance

Specifications of flow

- No flow when indexing packages, i.e. $\Phi(0) = \Phi(t_{fill}) = 0$.
- No negative flows, i.e. $\Phi(t) > 0 \quad \forall t$. We cannot pump liquid from the packages.

Table 3: Constraints on the flow

The specifications for the regulator can be formulated as in figure 25, table 3 and table 4. At the start of a filling cycle ($t = 0$) the regulator receives a volume reference signal that tells it how much liquid it should fill into the packages. At the end of the allowed filling time ($t = t_{fill}$) the filled volume should equal the volume reference. These occasions are marked with small circles. This is the main thing the regulator should achieve. Exactly how the volume in the package varies during the filling cycle is less important. Of course, it (and respectively the flow) has to fulfill certain requirements. These are presented in table 3 and table 4.

The filler should be able to fill packages of 3 different sizes, 1 l, 0.5 l and 0.25 l. A signal that tells us which size that is to be filled is available.

9.2 The two control strategies

We have tried two different control strategies to handle the control problem. The first we have called "posref"-regulator and the second "volref"-regulator.

Practical aspects concerning the flow in the filler

- Soft start of the flow to avoid foaming
- Not too slow closing to avoid the flow range where the flow-meter has big errors
- Not too fast closing to avoid damaging the interior parts of the filling valve
- Smooth increases and decreases of flow to keep the distance between the filling valve nozzle and the liquid level in the package constant

Table 4: Constraints on the flow, cont.

The posref regulator is based on the ideas that the flow meter computer itself uses, augumented with a regulator for the product tank pressure.

The volref regulator is a regulator that operates in closed loop. Instead of following prespecified opening profiles with the filling valve it controls the filling valve piston position.

9.3 Regulator performance

We wanted to measure the performance of the regulators by looking at the filling errors at the end of each filling cycle. But the flow errors in the model are around 10 % so we can not rely on that the filling errors in the simulations would be the same in the real filler. We can only draw conclusions about the main properties of how the regulators behave. Of course, if the model is improved, we can also study the filling errors in the simulation and know that these would be the same in the real filler.

When studying the performance of a real regulator we can use a video camera and transparent packages to registrate the whole filling process. Here we can study phenomenons such as foaming, too fast filling at the start of the filling cycle etc. The filled packages can be weighted after the filling and the real errors measured.

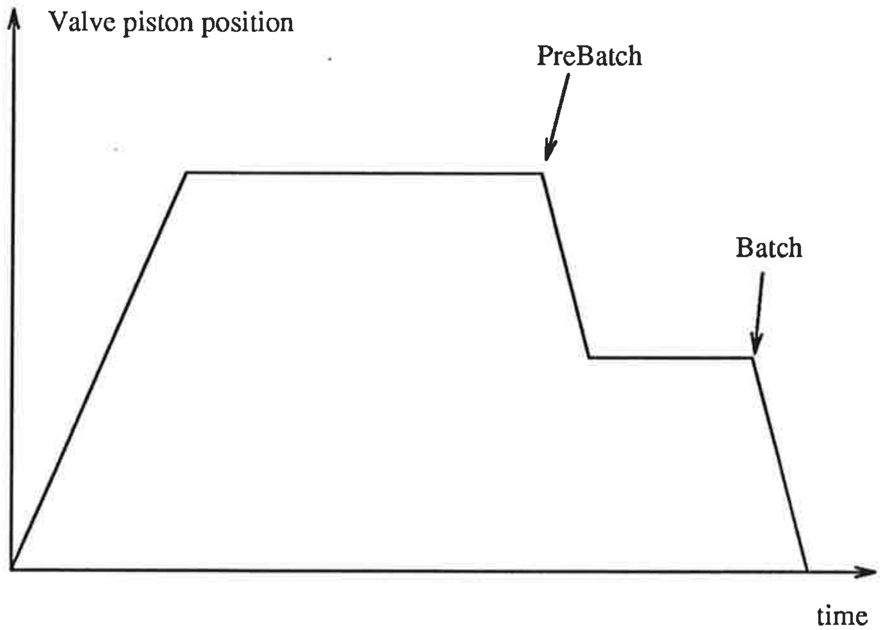

Figure 26: The piston position in the posref-regulator

10 Position Reference Regulator

This regulator is based on the same ideas as the one used today, the one that is internal to the flow meter.

10.1 Function of flow meter

The flow-meter constantly monitors the flow through the filling pipe. When most of the desired volume has been filled, it sends a signal, PreBatch, to the filling valve which then closes approximately half-way. This is shown in figure 26. When only a small portion of the desired volume remains to be filled, a second signal, Batch, is sent. This signal closes the filling-valve totally. The small amount of product that manages to flow into the package before the filling valve closes is just enough to fill up the package to the desired volume.

10.2 Funtion of the flow meter's regulator

We specify the total volume we want to fill into the packages, 1000 ml, the Batch volume, 950 ml, and the PreBatch volume, 800 ml.

During the filling process, the flow-meter computer constantly checks the volume that flows into the package after the Batch signal is given. In this way it knows the total volume filled into every package. If it does not correspond to the prespecified filling volume it changes the Batch volume in the direction that corresponds to a correct filling volume. For example: If the Batch volume is set to 950 ml and the machine fills 985 ml it changes the Batch volume to 965 ml. The volume filled into the packages after the Batch signal is given in this example, is 35 ml. So if the Batch volume is set to 965 ml the total volume filled into the next package will be 1000 ml. These changes are made constantly every filling, the value used to adjust the Batch volume is averaged from the last fillings. How many to average over can be specified. In this way the filling mechanism can handle product variations and is in some way "adaptive".

10.3 Function of the position reference regulator

The position reference regulator consists actually of 3 regulators.

- A modified I-regulator to control the batch volume level. It does not work exactly as the flow meter does. It is an I-regulator with a pre-specified start value (reg)
- A modified I-regulator to control the normal opening position (reg2)
- A modified I-regulator to control the product tank pressure (reg2)

How these regulators are interconnected can be seen in figure 10. When the pressure regulator is running the batch volume regulator is disconnected. Otherwise the 2 regulators would interact in an undesired way. Simulations have shown though, that one can allow both the batch volume regulator and the normal opening position regulator to run simultaneously.

10.4 Advantages

- Knowledge from todays regulator can be used together with the new regulator.
- Does not require fast hardware. It is enough that the hardware can sample at the end of every filling and communicate with the flow-meter.

10.5 Disadvantages

- Difficult to program a regulator that handles all variables that have to be controlled, i.e. batch volume, pressure, valve piston normal opening positon and piston movement. During the simulations we had problems finding the regulator parameters that gave good control.

10.6 Simulation results

The position reference regulator has been tested more than the volume reference regulator. Tetra Pak will probably try to implement this regulator soon and we wanted to make sure that it functions well. We have made simulations with two different liquids, water and raspberry cream. Water is a Newtonian liquid with a low viscosity and raspberry cream is a non-Newtonian liquid with high viscosity. For the raspberry cream we can study the performance of the pressure regulator and for water we can study the opening position regulator. We have simulated with the available package sizes, 1, 0.5 and 0.25 l. These are the variables shown in the simulations.

flow	the flow through the filler
pos	the "sharp edged" signal: the valve piston position reference
posl	the "smooth edged" signal: the valve piston position
poscrit	the critical valve piston position (5.6 mm)
totloss	the total pressure loss in the filler
Batch Volume	the volume level when the regulator sends the Batch singal
Filling time	the time the package was filled in
Filling error	the filling error at the end of the previous filling cycle
	< 0 means overfilled package
	> 0 means underfilled package

Figure 27, 28 and 29:
When filling water the regulator has to decrease the normal opening position from 22.5 mm to a suitable value depending on the package size. It manages this very well. We see it especially good in figure 29 when it has to fill the smallest package in the nominal filling time 690 ms.

Figure 30, 31 and 32:
When filling raspberry cream the regulator has to increase the product tank pressure from 0 Pa to a suitable value depending on the package size. This can be seen especially good in figure 30 where the pressure increase has to be maximal. The pressure increase can be seen as the increase of the mean value of the total pressure loss. It goes up almost to 0.4 bar.

In the simulations, the pressure regulator or the normal position regulator does the main adjustements to achieve correct volume and filling time. Then the batch volume regulator takes over and adjusts the batch volume level to its optimal level.

The simulations have been performed with the following regulator parameters.

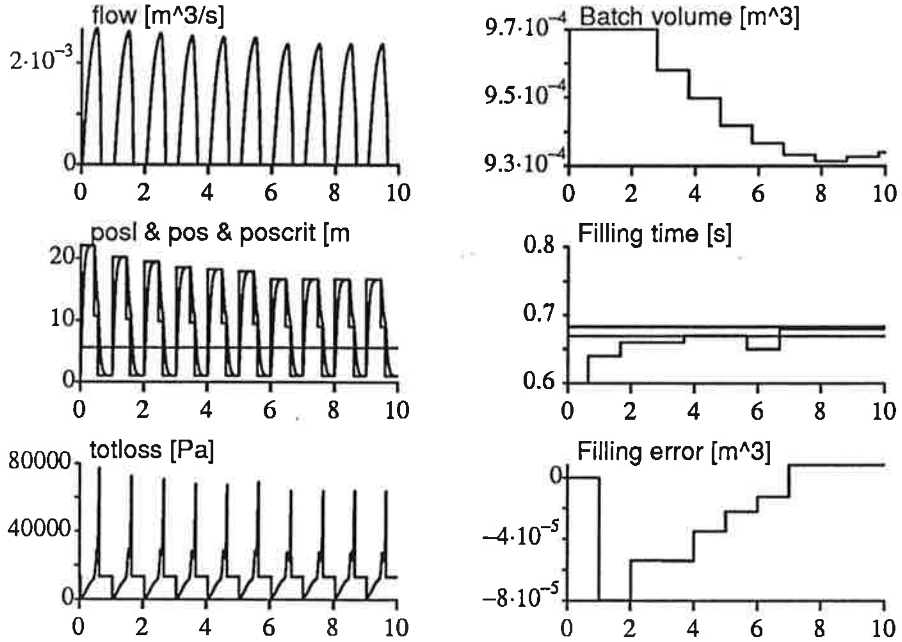

Figure 27: Water, 1.0 l

type	afactor	bfactor	kvb	kn	kp
1.0 l, water	0.99	0.97	0.15	50	2e7
0.5 l, water	0.99	0.97	0.10	40	2e7
0.25 l, water	0.99	0.97	0.05	30	2e7
1.0 l, cream	0.99	0.97	0.15	50	2e7
0.5 l, cream	0.99	0.97	0.10	40	2e7
0.25 l, cream	0.99	0.97	0.05	30	2e7

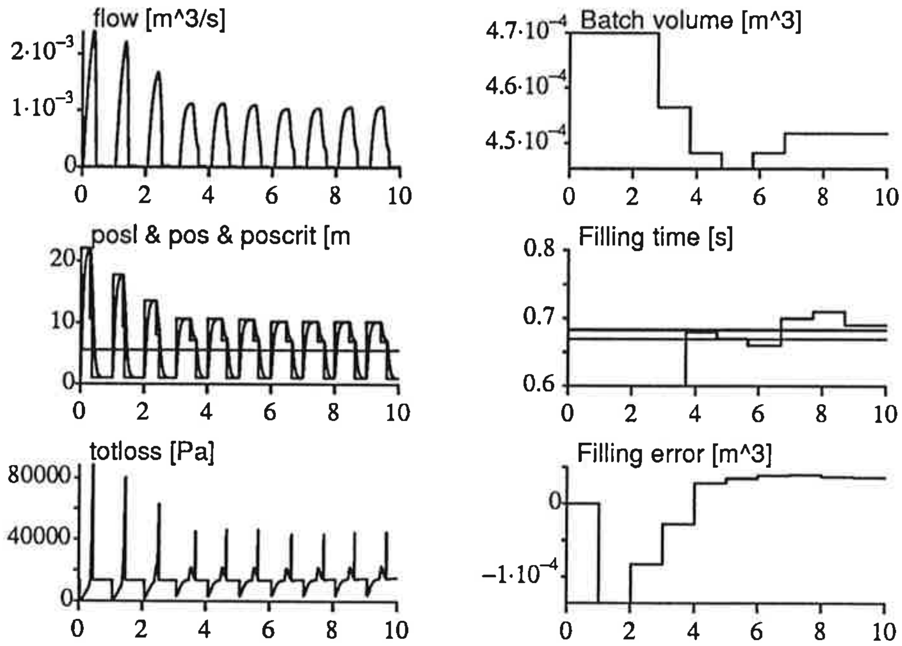

Figure 28: Water, 0.5 l

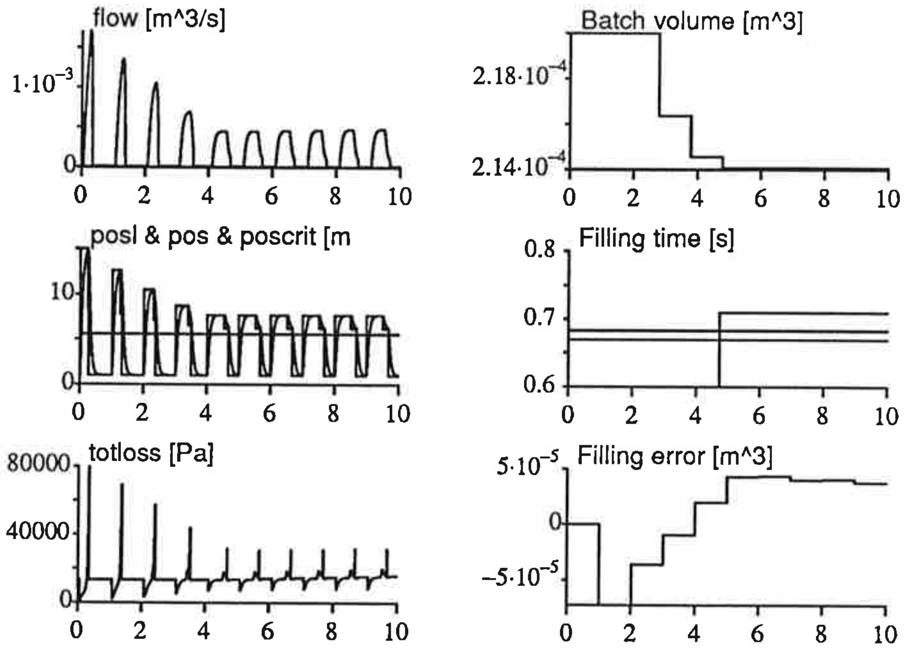

Figure 29: Water, 0.25 l

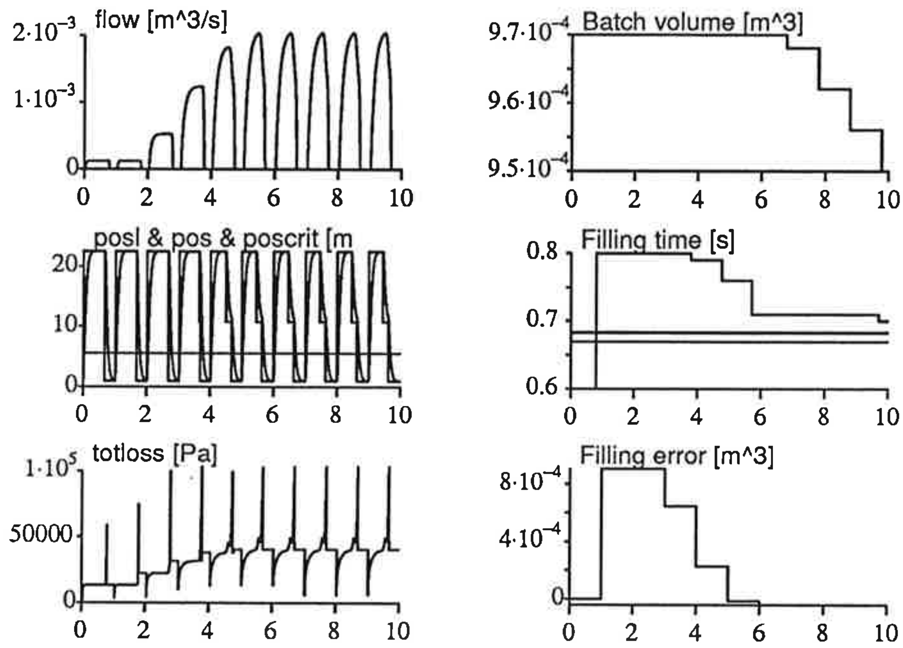

Figure 30: Raspberry cream, 1.0 l

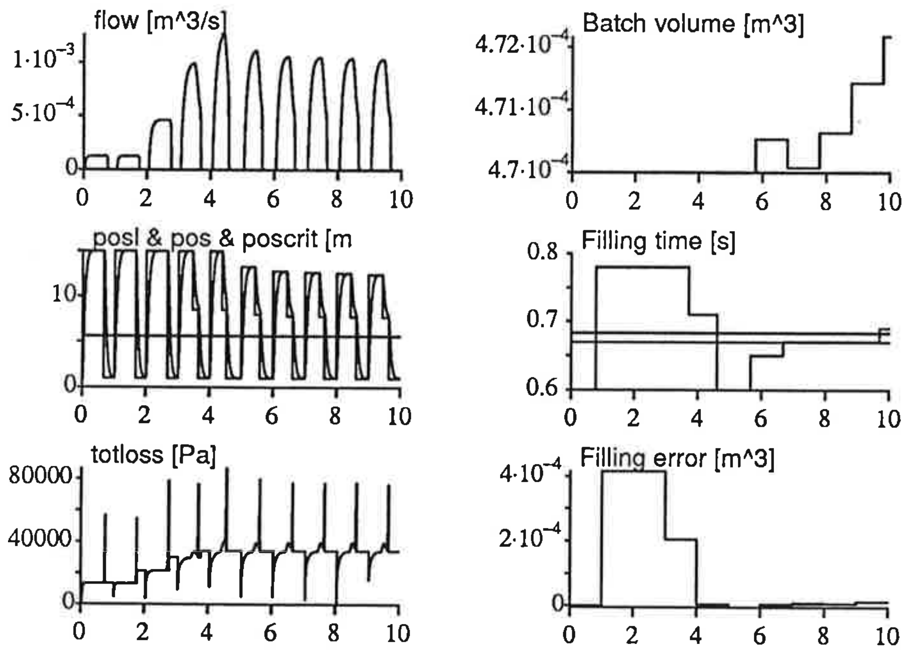

Figure 31: Raspberry cream, 0.5 l

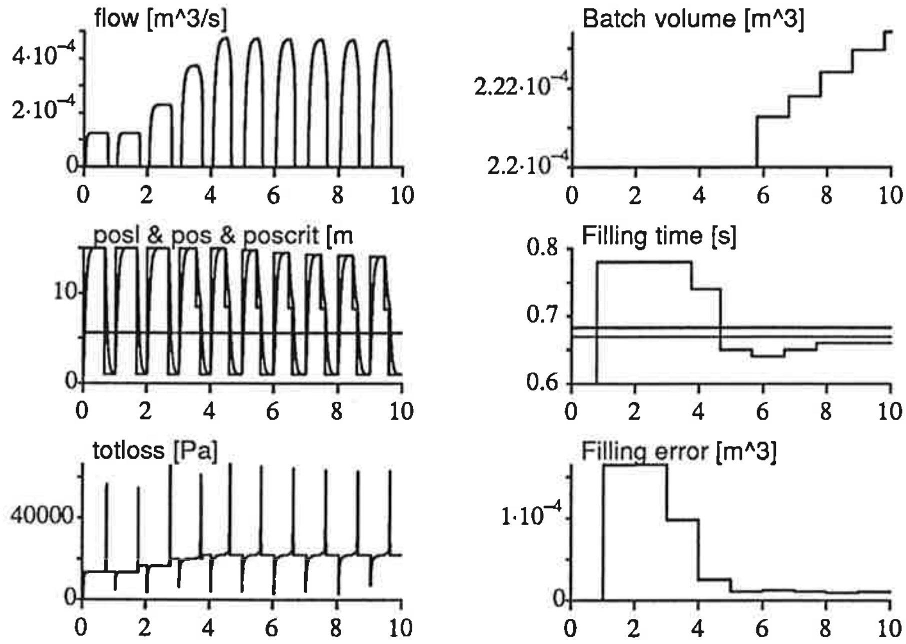

Figure 32: Raspberry cream, 0.25 l

11 Volume reference regulator

11.1 Function of the volume reference regulator

The volume reference regulator is a true closed loop regulator of the filling valve piston position. It tries to follow a prespecified volume reference. It consists actually of 3 regulators.

- A PID-regulator to control the filling valve piston position, sampled every 10 ms (reg)
- A modified I-regulator to control the normal opening position, sampled once per filling cycle (reg2)
- A modified I-regulator to control the product tank pressure, sampled once per filling cycle (reg2)

How these regulators are interconnected can be seen in figure 9.

11.2 Advantages

- The regulator is running in closed-loop. This gives it far better possibilities to handle disturbances.
- The regulator parameters are few and are easy to set.
- The filling valve piston normal opening position is handled automatically by the regulator. This has to be handled as an extra regulator in the posref regulator

11.3 Disadvantages

- Needs hardware that can run such a fast sampled regulator (10 ms).
- A bit difficult to calculate good volume references for the regulator.

11.4 Simulation results

During the simulations we discovered that it was difficult to get good control with only a PI-regulator controlling the piston position. Derivative control action is necessary so the best results have been achieved with a PID-regulator. Anti windup is necessary since the control signal often saturates.

The variables shown in the simulations are

Pack volume	the volume filled into the package
ref volume	the volume reference
flow	the flow through the filler
pos	the "sharp edged" signal: the valve piston position reference
posl	the "smooth edged" signal: the valve piston position
poscrit	the critical valve piston position (5.6 mm)
Filling error	the volume error

The regulator parameters have been kpi=50000, ti=3, kd=0.1, hd=1, hu=1, cexp=0.25 and lin=0.5 for water and the same for raspberry cream except on=1 (switch for pressure regulator), kp=2e7, perr=0, hd=2 and hu=0.5.

Figure 33, 34 and 35:
We have chosen a quite high gain in the regulator. This gives it somewhat jumpy behaviour but we can use the same regulator for all package sizes. To find good regulator parameters is not easy. Perhaps a PI^2D-regulator or a local regulator, as shown in figure 39 is needed. The sudden increase in filling error at the end of each filling cycle has to do with internal updating in the Simnon systems. The final filling error is the value just before this pulse.

Figure 36:
In the volume reference regulator both the pressure regulator and the piston position regulator work simultaneously. In the position reference regulator they were separated in time. This gives rise to a slight problem. At the end of the simulation, the pressure adjustement is too slow, the package volume will never reach 1.0 l. This because of the piston regulator regulating as good as it can and giving small volume errors. The pressure is adjusted with respect to these errors. It would probably work better if these regulators could be somewhat separated in time or if we had an adaptive regulator as described in chapter 12.3.

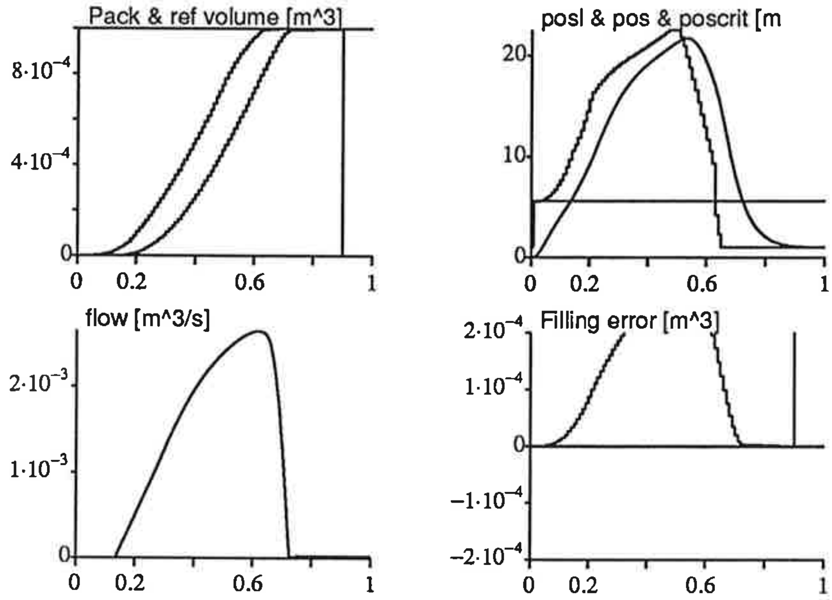

Figure 33: Water, 1.0 l

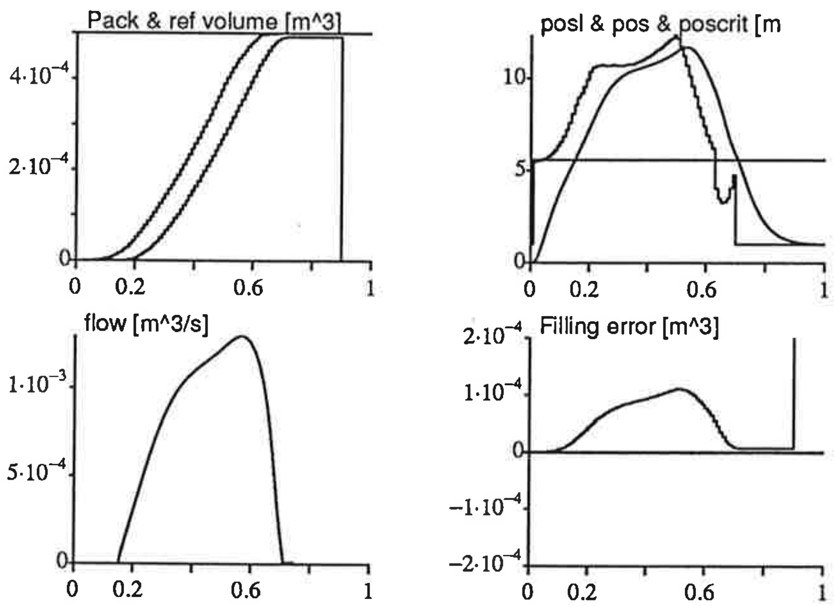

Figure 34: Water, 0.5 l

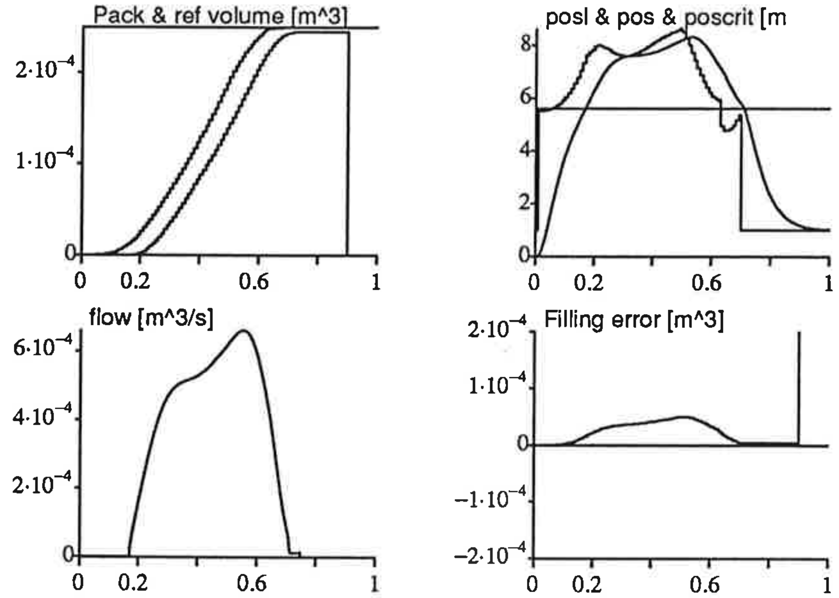

Figure 35: Water, 0.25 l

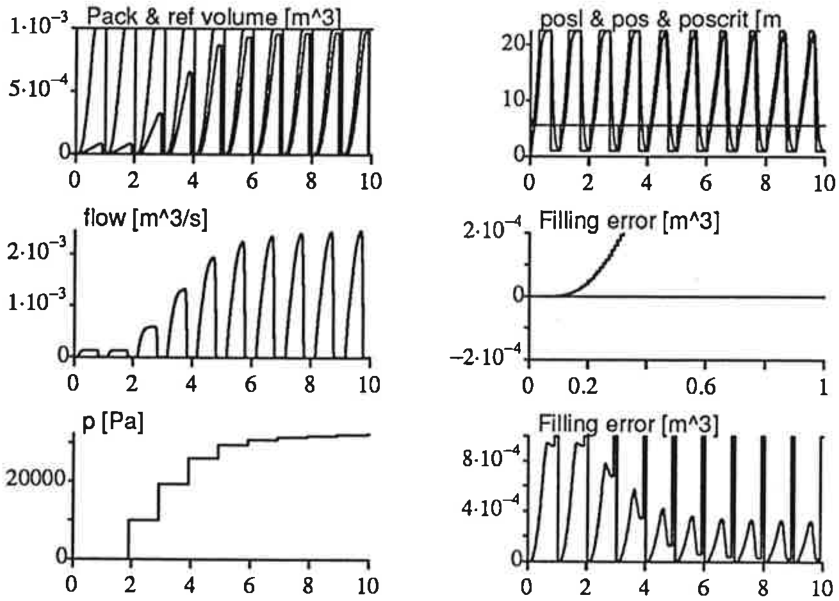

Figure 36: Raspberry cream, 1.0 l

12 Future improvements

If Tetra Pak decides to continue the work presented in this report, here follows some idéas as to what should be done.

12.1 Process Identification

To improve the simulation results (make them closer to reality) one has to make process identification of the different parts of the filler. Many things we have modeled have been intelligent guesses, we did not have access to the filler to make all the experiments needed. The things we think affect the model behaviour the most are the following.

12.1.1 Filling valve dynamics

Measure the real transfer function from pos_{ref} to pos in the filling valve.

12.1.2 Pressure losses in the filling valve

Measure the pressure losses in the filling valve for different liquids and different flows. Use this information in the model of the filling valve pressure losses.

12.1.3 Other liquids

Make experiments and measure the static pos to flow transfer function and acceleration ramps for other liquids than water.

12.2 Cascaded regulators

The static transfer function from the filling valve piston position to flow is not linear, not mentioning the dynamic transfer function! As all designers of analog circuits know one can use internal feedback when using non-linear components (e.g. transistors) to linearize their behaviour and make the behaviour component independent. The filler today is shown in figure 37. The function f can be viewed in figure 19.

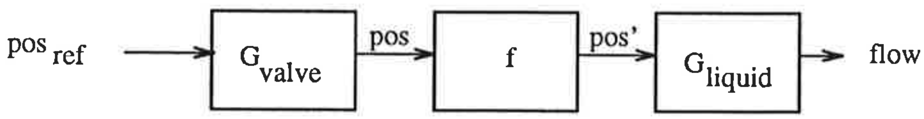

Figure 37: An overview of todays filler

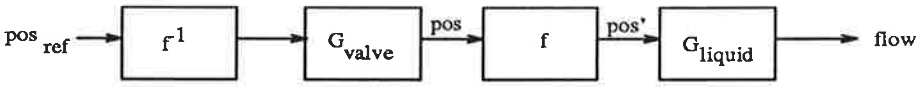

Figure 38: Taking the non-linearity f into account

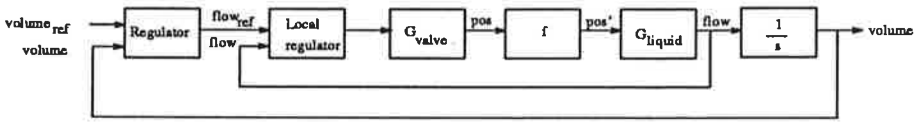

Figure 39: Local feedback to linearize the non-linearity

When trying to control this process with a regulator we can get better results if we take into account the non-linearity in the process. We implement the inverse of the non-linearity, the f^{-1}-function, in the regulator, see figure 38.

Now, $ff^{-1} = 1$, and we have linear behaviour from pos_{ref} to pos' disregarding G_{valve} which was not the case in figure 37. The non-linearity function though, varies with different liquids and filling valves. The f^{-1}-function is implemented in the regulator and if the f-function in the filler changes, we will not have linearity any more. We can do the trick in another way though, by introducing local feedback. This is shown in figure 39

In our case it turns out that we can build the local regulator into the normal regulator. It seems that they are taking different signals as input, but the volume signal is integrated internally in the regulator and the only signal that really enters the regulator is the flow. But it is a different way to view the regulator structure with 2 different regulators. It gives you another perspective on the regulator design.

12.3 Volume Reference Regulator

The volume reference regulator seems to be the most promising of the regulators and has a potential of getting even better. A systematic approach to choosing volume reference signals and regulator parameters is recommended. It can perhaps be extended to an adaptive regulator that can "measure" the apparent viscosity, η_A, or some constant proportional to it, by studying the liquid behaviour. If we apporoximate the liquid dynamics with a first order system we know that the liquid viscosity directly affects two things: The liquid time constant, τ_{liquid}, and the static gain, K_{liquid} in the filler. They relate as low τ_{liquid} and low $K_{liquid} \Rightarrow$ high viscosity and high τ_{liquid} and high $K_{liquid} \Rightarrow$ low viscosity. Of course, the adaptive regulator would have to handle the increase in K_{liquid} as it rises the product tank pressure to be able to fill thick products. If it manages to estimate the apparent viscosity, η_A, it could directly find suitable settings for tank pressure and regulator parameters. In the simulations shown in this report, it takes the regulator 5-15 packages to find the right pressure and opening position and the regulator parameters have to be specified in advance. An adaptive regulator could calculate all these parameters after only one filling!

A Matlab listings

Here follow the programs used in the postprocessing of experimental data.

```
%
% Show
%
% Shows the unfiltered and filtered  dynamic measurement
% together with their respective covariance functions
% of the stationary top
%
load dynamic
scale = 4/100; % scale factor for flow
d1 = d11_1(1:399); % get flows
d2 = d11_2(1:399); % get flows
d3 = d11_3(1:399); % get flows
d = (d1+d2+d3)/3*scale; % average 3 measurements
d = d(1:275); % cut trailing zeroes
df = (d(1:274)+d(2:275))/2; % average
m = d(120:220); % get stationary part
mf = df(120:220); % get stationary part
m = m-mean(m); % throw away bias
mf = mf-mean(mf); % throw away bias

mcov = xcorr(m,'coeff'); % calculate covariance functions
mfcov = xcorr(mf,'coeff'); % for the stationary part

t = (0:0.01:2.74); % get time-axis right
tf = t(1:274);
clg;
subplot(221); % divide graphics screen
plot(t,d); % plot unfilterd flow
title('Unfiltered flow');
ylabel('Flow l/s');
xlabel('Time ÄsÅ');
axis; % freeze axis
plot(tf,df) % plot filtered flow
title('Filtered flow');
ylabel('Flow l/s');
xlabel('Time ÄsÅ');
axis; % unfrezze axis

plot(mcov(60:140)); % plot covariance function
title('Covariance function (unfiltered flow)');
axis;
plot(mfcov(60:140)); % plot covariance function
```

```
title('Covariance function (filtered flow)');
axis;
```

```
function Äpos,yÅ=flow(puls,s,deltah)
% function Äpos,yÅ=flow(puls,s,deltah)
%
% calculates the average flow from a static variable
% flow(puls,deltah) where deltah are the tank heigth
% variations during the experiments (in mm)
%
p = s(21:37); % pick out stationary part
pm = mean(p); % calculate the mean value
pml = pm/500*4; % convert to l/s
toth = 1.200; % total heigth of machine
tankh = 0.253; % nominal tank level
drivh = toth+tankh; % total driving heigth
deltah = deltah/1000; % convert to m
deltah = deltah/2; % the average heigth offset
factor = (drivh+deltah)/drivh;
y = pml*factor; % return adjusted flow value

pos = (puls-4100)/1638 + 5.6; % return piston position
% adjusted for wrong closing point

%
% macro to analyze the static flow data
% calculates the static transfer function pos->flow
%
load static;
Äp2 f2_1Å = flow(27500,s2_1,-22.5);
Äp2 f2_2Å = flow(27500,s2_2,-22.5);
f2 = (f2_1+f2_2)/2;
Äp3 f3Å = flow(25000,s3,-21);
Äp4 f4Å = flow(22500,s4,-19.5);
Äp5 f5Å = flow(20000,s5,-18);
Äp6 f6Å = flow(17500,s6,-13);
Äp7 f7Å = flow(15000,s7,-8);
Äp8 f8_1Å = flow(12500,s8_1,-1);
Äp8 f8_2Å = flow(12500,s8_2,-4);
f8 = (f8_1+f8_2)/2;
Äp9 f9_1Å = flow(10000,s9_1,-0);
Äp9 f9_2Å = flow(10000,s9_2,-1);
Äp9 f9_3Å = flow(10000,s9_3,-3);
Äp9 f9_4Å = flow(10000,s9_4,-8);
f9 = (f9_1+f9_2+f9_3+f9_4)/4;
Äp10 f10Å = flow(7500,s10,-4);
Äp11 f11Å = flow(5000,s11,-0);

Äp14 f14Å = flow(6250,s14,-2);
Äp15 f15Å = flow(8750,s15,-6);
```

```
Äp16 f16Å = flow(11250,s16,-9);
Äp17 f17Å = flow(13750,s17,-20);
Äp18 f18Å = flow(16250,s18,-13);

Äp19 f19_1Å = flow(5625,s19_1,+8);
Äp19 f19_2Å = flow(5625,s19_2,+8);
f19 = (f19_1+f19_2)/2;
Äp20 f20Å = flow(6875,s20,+6);
Äp21 f21Å = flow(8125,s21,+6);
Äp22 f22Å = flow(9375,s22,+0);
Äp23 f23Å = flow(10625,s23,+12);
Äp24 f24Å = flow(11875,s24,+0);
Äp25 f25Å = flow(13125,s25,-4);

%allf = Äf2;f3;f4;f5;f6;f7;f8;f9;f10;f11;f14;..
f15;f16;f17;f18;f19;f20;f21;f22;f23;f24;f25Å;
%allp = Äp2;p3;p4;p5;p6;p7;p8;p9;p10;p11;p14;..
p15;p16;p17;p18;p19;p20;p21;p22;p23;p24;p25Å;
%
% f & p should preferably be in increasing order to get good plot
%
allf =Äf11;f19:f14;f20;f10;f21;f15;f22;f9;f23;..
f16;f24;f8;f25;f17;f7;f18;f6;f5;f4;f3;f2Å;
allp =Äp11;p19:p14;p20;p10;p21;p15;p22;p9;p23;..
p16;p24;p8;p25;p17;p7;p18;p6;p5;p4;p3;p2Å;

plot(allp,allf);
ylabel('Flow l/s');
xlabel('Piston position ÄmmÅ');
```

B Simnon listings

Listings of model4 and reg, reg2, creg, hreg3, func and onez for the position and volume reference regulators.

Remark: In some listings Ä means [and Å means].

B.1 model4.t

```
continuous system model

"
" System for the 4th model of machine
" More detailed model handling all different pipes,
" valve dynamics and dynamic pressure losses
"


state flow Vol possf posf p1 p2

" -3 -3 0 1 1 1
" Order of magnitude
"
der dflow dVol dpossf dposf dp1 dp2

input tankp tankh pos
output outflow posl

time t

INITIAL

"-----------------------------------------------------------------------------
" Calculated constants
"-----------------------------------------------------------------------------

" Filling valve

a = 1/Tm " see Filling valve motor time-constant below
b = 1/Tv " see Filling valve time-constant below

" Equivalent radius for pipe with decreasing radius

R3_4 = SQRT(R3*R3+R3*R4+R4*R4)

" Areas

A1_2 = pi*R1_2*R1_2
A3_4 = pi*R3_4*R3_4
A5_6 = pi*R5_6*R5_6
A7_8 = pi*R7_8*R7_8
```

```
A9_10 = pi*R9_10*R9_10

" Ratios between lengths and areas

K1_2 = L1_2/A1_2
K3_4 = L3_4/A3_4
K5_6 = L5_6/A5_6
K7_8 = L7_8/A7_8
K9_10 = L9_10/A9_10

" Sum of all ratios

sumK = K1_2 + K3_4 + K5_6 + K7_8 + K9_10

" Total height in machine

toth = H1_2 + H3_4 + H5_6 + H7_8 + H9_10

" Total volume and mass of liquid in the machine

vol1_2 = L1_2*A1_2
vol3_4 = L3_4*A3_4
vol5_6 = L5_6*A5_6
vol7_8 = L7_8*A7_8
vol9_10 = L9_10*A9_10

totvol = vol1_2 + vol3_4 + vol5_6 + vol7_8 + vol9_10
totm = totvol*rho

" Half rho

hrho = rho/2

SORT

"--------------------------------------------------------------------------------
" Filling valve dynamics, version 1
"--------------------------------------------------------------------------------

" posl - position used in machine, "filtered"-pos
" pos - position reference from regulator
" The regulator used here was an on/off regulator which gabe
" long simulaiton times. Will probably be better with a P-reg

pose = pos - posl
poss = if pose>0 then 1 else if pose<0 then -1 else 0
dpossf = a*(poss-possf)
dposf = kH*possf
" maybe move limitations here as this:
"dposf = if (posf<poslow) OR (posf>poshigh) then 0 else kH*possf

post =if posf<poslow then poslow else if posf>poshigh then poshigh else posf
pos1 = if dynamic>0.5 then post else pos
```

```
"------------------------------------------------------------------------------
" Filling valve dynamics, version 2 (2nd order system)
"------------------------------------------------------------------------------

" posl - position used in machine, "filtered"-pos
" pos - position reference from regulator

dp1 = -2*b*p1-b*b*p2+pos
dp2 = p1
pos2 = b*b*p2


posd = if ver<1.5 then pos1 else pos2
posl = if dynamic>0.5 then posd else pos
downflag= dp2<0 "*** temporary solution, used to know when the pressure
"*** surge should come


"------------------------------------------------------------------------------
" Calculation of total driving pressure
"------------------------------------------------------------------------------

totp = tankp + rho*g*tankh + rho*g*toth


"------------------------------------------------------------------------------
" Acceleration of liquid in the pipe
"------------------------------------------------------------------------------

drivp = totp-totloss "*** new version
dflow= drivp/rho/sumK


"------------------------------------------------------------------------------
" Calculation of losses
"------------------------------------------------------------------------------

" Velocities in pipe segments

v1_2 = flow/A1_2
v3_4 = flow/A3_4 " ????***???***???***???
v5_6 = flow/A5_6
v7_8 = flow/A7_8
v9_10 = flow/A9_10

" One-time losses

pz1_2 = hrho*z1_2*v1_2*v1_2
pz2_3 = hrho*z2_3*v1_2*v1_2
pz3_4 = hrho*z3_4*v5_6*v5_6
pz4_5 = hrho*z4_5*v5_6*v5_6
pz5_6 = hrho*z5_6*v5_6*v5_6
```

```
pz6_7 = hrho*z6_7*v7_8*v7_8
pz7_8 = hrho*z7_8*v7_8*v7_8
pz8_9 = hrho*z8_9*v7_8*v7_8
pz10_11 = hrho*z10_11*v9_10*v9_10

zloss = pz1_2+pz2_3+pz3_4+pz4_5+pz5_6+pz6_7+pz7_8+pz8_9+pz10_11

" Dynamic losses

dynloss = hrho*v9_10*v9_10

" Frictional losses

" Newtonian fluids

lambda1 = if Re1<Rel then 0 else if Re1<2300 then 64/Re1 else 0.316*Re1^(-0.25)
lambda2 = if Re2<Rel then 0 else if Re2<2300 then 64/Re2 else 0.316*Re2^(-0.25)
lambda3 = if Re3<Rel then 0 else if Re3<2300 then 64/Re3 else 0.316*Re3^(-0.25)
lambda4 = if Re4<Rel then 0 else if Re4<2300 then 64/Re4 else 0.316*Re4^(-0.25)
lambda5 = if Re5<Rel then 0 else if Re5<2300 then 64/Re5 else 0.316*Re5^(-0.25)

new1 = lambda1*L1_2*rho*v1_2*v1_2/4/R1_2
new2 = lambda2*L3_4*rho*v3_4*v3_4/4/R3_4
new3 = lambda3*L5_6*rho*v5_6*v5_6/4/R5_6
new4 = lambda4*L7_8*rho*v7_8*v7_8/4/R7_8
new5 = lambda5*L9_10*rho*v9_10*v9_10/4/R9_10

lossnew = new1+new2+new3+new4+new5

" Ostwaldian fluids

ost11 = L1_2/R1_2
ost21 = L3_4/R3_4
ost31 = L5_6/R5_6
ost41 = L7_8/R7_8
ost51 = L9_10/R9_10

osta2 = 2*k*((3*n+1)/4/n)^n " not machine specific

ost13 = if flow<feps then 0 else (4*flow/pi/(R1_2^3))^n
ost23 = if flow<feps then 0 else (4*flow/pi/(R3_4^3))^n
ost33 = if flow<feps then 0 else (4*flow/pi/(R5_6^3))^n
ost43 = if flow<feps then 0 else (4*flow/pi/(R7_8^3))^n
ost53 = if flow<feps then 0 else (4*flow/pi/(R9_10^3))^n

ost1 = ost11*osta2*ost13
ost2 = ost21*osta2*ost23
ost3 = ost31*osta2*ost33
ost4 = ost41*osta2*ost43
ost5 = ost51*osta2*ost53

lossost = ost1+ost2+ost3+ost4+ost5
```

```
fricloss= if new>0.5 then lossnew else lossost

" Filling valve

valtA = FUNC(1,posl) + valmin
valA = valtA/1e6 " Conversion to m^2
valv = v9_10*A9_10/valA
zvalve = FUNC(2,posl)
valtemp1= zvalve*hrho*valv*valv
"valloss = if posl>poscrit then valtemp1 else totp
fl1 = flow>feps and downflag and posl>poscrit-poseps
fl2 = NOT downflag and posl>poscrit
valloss = if fl1 then valtemp1 else if fl2 then valtemp1 else totp


" Total loss

totloss = fricloss + zloss + valloss + dynloss

"------------------------------------------------------------------------------
" Calculation of flow
"------------------------------------------------------------------------------

outflow = flow

"------------------------------------------------------------------------------
" Calculation of volume
"------------------------------------------------------------------------------

dVol = flow

"------------------------------------------------------------------------------
" Reynolds number
"------------------------------------------------------------------------------

" Newtonian fluids

rhoeta = rho/eta
Re1 = v1_2*2*R1_2*rhoeta
Re2 = v3_4*2*R3_4*rhoeta
Re3 = v5_6*2*R5_6*rhoeta
Re4 = v7_8*2*R7_8*rhoeta
Re5 = v9_10*2*R9_10*rhoeta

" Reynolds number NOT calculated for Ostwald-de Waele fluids
" I assume that there is always laminar flow
" Verify this assumption with simulations on simpler models

"------------------------------------------------------------------------------
" Constants
"------------------------------------------------------------------------------

" Lengths
```

```
L1_2 : 126e-3 " ÄmÅ
L3_4 : 190e-3 " ÄmÅ
L5_6 : 590e-3 " ÄmÅ
L7_8 : 150e-3 " ÄmÅ
L9_10 : 365e-3 " ÄmÅ

" Heights

H1_2 : 126e-3 " ÄmÅ
H3_4 : 190e-3 " ÄmÅ
H5_6 : 440e-3 " ÄmÅ
H7_8 : 000e-3 " ÄmÅ
H9_10 : 365e-3 " ÄmÅ

" Radius

R1_2 : 23.75e-3 " ÄmÅ
R3 : 23.75e-3 " ÄmÅ
R4 : 16e-3 " ÄmÅ
R5_6 : 16e-3 " ÄmÅ
R7_8 : 16e-3 " ÄmÅ
R9_10 : 30e-3 " ÄmÅ

" One-time losses

z1_2 : 0.0
z2_3 : 0.0
z3_4 : 0.04
z4_5 : 0.1
z5_6 : 0
z6_7 : 0.20
z7_8 : 0
z8_9 : 0.5
"z9_10 = zvalve
z10_11 : 1.0

pi : 3.141593
g : 9.80665 " Äm/s^2Å

new : 1 " 1 - Newtonian fluid, 0 - Ostwald-de Waele fluid

n : 0.35 " n and
k : 28 " k values for an Ostwald-de Waele fluid (hallonkram)
eta : 10e-3 " ÄPasÅ viscosity for water
rho : 1000 " Äkg/m^3Å density for the liquids

kH : 300 " Helix-screw gain
Tm : 40e-3 " Filling valve motor time-constant (ver 1)
Tv : 40e-3 " Filling valve time-constant (ver 2)

poscrit : 5.6 " ÄmmÅ critical piston position (exactly when
" valve closes)
```

```
poslow : 0 " ÄmmÅ piston min-pos
poshigh : 22.5 " ÄmmÅ piston max-pos
dynamic : 1 " 1 - filling valve dynamics ON  0 - OFF
ver : 2 " 1 - version 1   2 - version 2

Rel : 1e-4 " Treshold for lambda calculations
feps : 1e-6 " Treshold for non-Newtonian loss calculations
valmin : 2 " Ämm^2Å minimal valve area
poseps : 2 " Treshold for stopping up liquid

end
```

B.2 Position Reference Regulator

B.2.1 reg.t

```
discrete system reg

" Regulator for model4
" Fixed ramps with BATCH and PREBATCH
" Regulates the BATCH-time to achieve 1, 0.5 and 0.25 l in packages

time t
tsamp ts

state  filled  intvtot  oldpos  intve
new nfilled nintvtot noldpos nintve

input posl Vol errflag vb posn
output pos tfilled vpack ve

INITIAL

"------------------------------------------------------------------------------
" Calculate initial values and variables
"------------------------------------------------------------------------------

tmax = 0.98*tfill " max time allowed for filling
vref = if pack<1.5 then vref1 else if pack<2.5 then vref2 else vref4
vpb = if pack<1.5 then vpb1 else if pack<2.5 then vpb2 else vpb4

SORT

"------------------------------------------------------------------------------
" Calculate next sampling instant
"------------------------------------------------------------------------------

ts = t + normsper

"------------------------------------------------------------------------------
" Calculate the filling time
"------------------------------------------------------------------------------

noldpos = posl

"------------------------------------------------------------------------------
" Calculate the (real) time that the package was filled in
"------------------------------------------------------------------------------

modt = MOD(t,ttot)
filledf = oldpos>poscrit AND posl<poscrit
" *** this is the instant when the valve closes
nfilled = if filledf then modt else filled
tfilled = filled
```

```
endflag = modt>ttot-normsper

"------------------------------------------------------------------------
" Calculate packet volume
"------------------------------------------------------------------------

vpack = Vol - vtot " Actual volume - volume in packaged containers
nintve = if endflag then vref-vpack else intve
ve = intve
nintvtot= if endflag then Vol else intvtot
vtot = intvtot

"------------------------------------------------------------------------
" Calculate position reference
"------------------------------------------------------------------------

pos1 = if vpack<vpb then posn else if vpack<vb then pospb else posclose
otflag = modt>tmax
ovflag = vpack>vref
totflag = otflag OR ovflag
pos =  if totflag then posclose else pos1
"*** ensure that max filling time and max volume (vref) is not exceeded

pospb = (posn-poscrit)*pbfactor+poscrit

"------------------------------------------------------------------------
" Constants
"------------------------------------------------------------------------

normsper: 10e-3 " ÄsÅ should be 10 but functions bad
critsper: 0.2e-3 " ÄsÅ

vref1 : 1.000e-3 " Äm^3Å reference volume (1.0 l)
vpb1 : 0.630e-3 " Äm^3Å pre-batch volume (1.0 l
vref2 : 0.500e-3 " Äm^3Å reference volume (0.5 l)
vpb2 : 0.315e-3 " Äm^3Å pre-batch volume (0.5 l
vref4 : 0.250e-3 " Äm^3Å reference volume (0.25 l)
vpb4 : 0.158e-3 " Äm^3Å pre-batch volume (0.25 l

pbfactor: 0.3077 " = (10.8-5.6)/(22.5-5.6)
posclose: 1.0 " ÄmmÅ close position
poscrit : 5.6 " ÄmmÅ Critical piston position

posu1 : 3.5 " ÄmmÅ interval for fast sampling, opening
posu2 : 6 " ÄmmÅ  -"-
posd1 : 5 " ÄmmÅ interval for fast sampling, closing
posd2 : 8.5 " ÄmmÅ  -"-

tfill : 0.690 " ÄsÅ filling time
tindex : 0.310 " ÄsÅ indexing time (shorter for faster simulations)
ttot : 1.000 " ÄsÅ total time
pack : 1 " type of package  1 - 1 l  2 - 0.5 l  4 - 0.25 l
```

```
end
```

B.2.2 reg2.t

```
discrete system reg2

"
" Regulates the pressure in the tank
" Sampled regulator at t = directly after filling
"

time t
tsamp ts

state  intp  inth   intvb  intn
new nintp ninth nintvb  nintn

input tfilled vpack ve
output h p errflag vb posn

INITIAL

ts = 0
vref = if pack<1.5 then vref1 else if pack<2.5 then vref2 else vref4
vbstart = if pack<1.5 then vbstart1 else if pack<2.5 then vbstart2 else vbstart4

maxtime = afactor*tfill
mintime = bfactor*tfill
tmean = (afactor+bfactor)/2*tfill
inth = hstart
intp = pstart
intvb = vbstart
intn = posnst
nop = 0


SORT

"---------------------------------------------------------------------------
" Calculate next sampling time
"---------------------------------------------------------------------------

fflag = if t>0 then 0 else 1
tstep = if NOT fflag then ttot else tfill+0.1
ts = t + tstep

"---------------------------------------------------------------------------
" Calculate new batch volume
"---------------------------------------------------------------------------

tvb = intvb + kvb*ve
nintvb = if NOT errflag then tvb else intvb
vb = nintvb

"---------------------------------------------------------------------------
```

```
" Check if adjustement of the normal opening position is needed
"-------------------------------------------------------------------------------

nflag = if fflag then 0 else tfilled<tmean
te = tfilled-tmean
tnintn = intn + kn/pack*te
nintn = if nflag then tnintn else intn
posn = nintn

"-------------------------------------------------------------------------------
" Check if adjustement of the pressure is needed
"-------------------------------------------------------------------------------

pflag = tfilled>maxtime and ve>perr/pack
tnintp = intp + kp*ve
nintp = if pflag then tnintp else intp
p = nintp

errflag = pflag

"-------------------------------------------------------------------------------
" Calculate new tank height
"-------------------------------------------------------------------------------

ninth = inth
h = inth


"-------------------------------------------------------------------------------
" Constants
"-------------------------------------------------------------------------------

tfill : 0.690 " ÄsÅ filling time
tindex : 0.310 " ÄsÅ indexing time (shorter for faster simulations)
ttot : 1.000 " ÄsÅ total time

afactor : 0.98 " adaption factor, length of max allowed tfilled
" before p-adaption occurs
bfactor : 0.95 " adaption factor, length of min allowed tfilled
" before posn-adaption occurs

vref1 : 1.000e-3 " Äm^3Å reference volume (1.0 l)
vref2 : 0.500e-3 " Äm^3Å reference volume (0.5 l)
vref4 : 0.250e-3 " Äm^3Å reference volume (0.25 l)

vbstart1: 0.970e-3 " Äm^3Å start batch volume (1.0 l)
vbstart2: 0.470e-3 " Äm^3Å start batch volume (0.5 l)
vbstart4: 0.220e-3 " Äm^3Å start batch volume (0.25 l)
posnst : 22 " ÄmmÅ  start posn

pstart : 0 " ÄPaÅ
hstart : 0.253 " ÄmÅ
kvb : 0.4 " gain of batch volume adaption
```

```
kp : 1e7 " gain of pressure adaption
perr : 50e-6 " Äm^3Å max allowed error before p-adaption occurs
nerr : 75e-6 " Äm^3Å max allowed error before posn-adaption occurs
pack : 1 " which package type
kn : 40 " adjustement factor in adaption of posn

end
```

B.2.3 hreg3.t

```
discrete system hreg

"--------------------------------------------------------------------------------
" Handles necessary transitions between fast sampling and slow samplin
" Remark! The pos-regulator always samples every 10 ms. The transitions
" are only to handle numerical problems at valve closing
"
" Tests 911015 - continous modification of critsper with derivative
" action
"
"--------------------------------------------------------------------------------

time t
tsamp ts

state  oldpos
new noldpos

input posl posn

"--------------------------------------------------------------------------------
" Calculate next sampling instant
"--------------------------------------------------------------------------------

downflag= oldpos>posl
noldpos = posl

d = posl-oldpos
s2 = min(1,abs(d/maxd))
s3 = (1-s2)^dexp

s4a = (posl-poscrit-hu)/(posn-poscrit-hu)
"(* to get equal weighting *)
"(* on posl under or over  *)
"(* poscrit   *)
s4b = (poscrit-hd-posl)/(poscrit-hd-posclose)
s4 = if s4a>0 then s4a else if s4b>0 then s4b else 0
"s5 = if downflag then s4^cexp else 1
s5 = s4^cexp
"*** water seems OK with no slow sampling at opening
"*** but not raspberry cream. Even this didn't help:
"*** decreasing critsper, increasing cexp, changing algor to rkf45

"s1 = if ason then normsper*s3*s5+critsper else normsper
s1 = if ason then (normsper-critsper)*s5+critsper else normsper
swposl = posl/2200
swcrit = poscrit/2200

ts = t + s1

"--------------------------------------------------------------------------------
```

```
" Constants
"--------------------------------------------------------------------------------

normsper: 10e-3 " ÄsÅ
critsper: 2e-4 " ÄsÅ

"***911028posn : 22.5 " ÄmmÅ normal pos
poscrit : 5.6 " ÄmmÅ critical pistion position
posclose: 1.0 " ÄmmÅ close position

maxd : 1
dexp : 1
cexp : 0.25
hu : 1 " hysteresis over
hd : 1 " hysteresis under

ason : 1

end
```

B.2.4 flow.t

```
macro flow

"
" Macro to simulate model4 with only a step in pos
"

"import valve3 < valve3 / 2
"syst model4 reg reg2 creg
"error 0.0001

store VolÄmodelÅ tankpÄmodelÅ flowÄmodelÅ totlossÄmodelÅ posÄmodelÅ
store poslÄmodelÅ poscritÄmodelÅ vallossÄmodelÅ -add
store tsÄregÅ vbÄregÅ vpackÄregÅ -add
store dflowÄmodelÅ veÄreg2Å errflagÄreg2Å  -add
store tfilledÄregÅ filledfÄregÅ totflag otflag ovflag modt -add
store posnÄregÅ nflagÄreg2Å maxtimeÄreg2Å mintimeÄreg2Å -add
store s5 -add

split 1 1
axes V 0 1 H 0 1
plot tÄmodelÅ

simu 0 1 / / / valve3

split 2 2

ashow totlossÄmodelÅ
text 'totloss ÄPaÅ'

ashow poslÄmodelÅ posÄmodelÅ poscritÄmodelÅ
text 'posl & pos & poscrit ÄmmÅ'

ashow s5
text 'Step length factor'

"ashow tankpÄmodelÅ
"text 'p ÄPaÅ'

"ashow dflowÄmodelÅ
"text 'dflow Äm^3/s^2Å'

"ashow VolÄmodelÅ
"text 'Vol Äm^3Å'

ashow flowÄmodelÅ
text 'flow Äm^3/sÅ'

"ashow valtAÄmodelÅ
"text 'valve area Ämm^2Å'

"ashow tsÄregÅ
```

```
"text 'sampling time   ts'

"ashow vbÄregÅ
"text 'Batch volume Äm^3Å'

"ashow vpack
"text 'Package volume Äm^3Å'

"axes V 0.6 0.8
"show tfilledÄregÅ maxtime mintime
"text 'Filling time ÄsÅ'

"ashow  ve
"text 'Filling error Äm^3Å'

"ashow posnÄregÅ
"text 'posn ÄmmÅ'

"ashow nflagÄreg2Å
"text 'nflag'

end
```

B.2.5 initflow.t

```
macro initflow

"
" Macro to initialize model2
"

import valve3 < onez / 2
syst model4 hreg3 reg reg2 creg
error 0.0001

end
```

B.2.6 onez.t

```
0 0
6 0
13 350
22 1450
30 1450
30 1450
0 1
30 1
30 1
```

B.3 Volume Reference Regulator

B.3.1 reg.t

```
discrete system reg

"-----------------------------------------------------------------------------
" PI-regulator that regulates pos in the volume reference regulator
"-----------------------------------------------------------------------------

time t
tsamp ts

state i   olde  oposl
new ni nolde noposl

input Vol vtot posl
output pos vpack ve

INITIAL

vpackref= if pack<1.5 then vref1 else if pack<2.5 then vref2 else vref4

SORT

"-----------------------------------------------------------------------------
" Calculate next sampling instant
"-----------------------------------------------------------------------------

ts = t + sper

"-----------------------------------------------------------------------------
" Calculate packaged volume so far
"-----------------------------------------------------------------------------

vpack = Vol - vtot

"-----------------------------------------------------------------------------
" PI-regulator
"-----------------------------------------------------------------------------

modt = MOD(t,ttot)
start = modt<(2*sper)

bt = sper/ti
rtime = rfact*tfill
f = lin + modt/rtime*(1-lin)
linfact = if modt>rtime then 1 else f
"
" Try different volume references
"
Volref = FUNC(3,modt)*vpackref*linfact
```

```
"Volref = vpackref
"Volref = vpackref*linfact
e = Volref-vpack
oflag = abs(e-olde)>olev
olev : 0.5e-3
ve = if oflag then olde else e

nolde = e
d = (e - olde)/sper*kpi
pos1 = kpi*e + i + kd*d
noposl = posl
pos2 = if pos1<posclose then posclose else if pos1<posn then pos1 else posn
otflag = modt>tmax
ovflag = vpack>vpackref
totflag = otflag OR ovflag
pos = if totflag then posclose else pos2
what = pos - pos1
tni = i + kpi*e*sper/ti + bt*(pos - pos1) "*** maybe try posl inst. pos

ni = if start then poss else tni

"------------------------------------------------------------------------------
" Constants
"------------------------------------------------------------------------------

sper : 10e-3 " ÄsÅ sampling period

vref1 : 1.000e-3 " Äm^3Å reference volume (1.0 l)
vref2 : 0.500e-3 " Äm^3Å reference volume (0.5 l)
vref4 : 0.250e-3 " Äm^3Å reference volume (0.25 l)

pack : 1 " package type 1 - 1 l  2 - 0.5 l  4 - 0.25 l
tmax : 0.690 " ÄsÅ max filling time
tfill : 0.690 " ÄsÅ filling time
ttot : 1.000 " ÄsÅ total time
kpi : 3e4 " PID-regulator gain
ti : 1e20 " PID-regulator integration time
kd : 0.2 " PID-regulator derivative gain
posclose: 1.0 " ÄmmÅ close position
poscrit : 5.6 " ÄmmÅ critical position
posn : 22.5 " ÄmmÅ normal position
poss : 5.5 " ÄmmÅ start position
zero : 0.0 " constant 0
lin : 0.0 " linearity constant for volume reference

rfact : 0.9 " Time relatively tfill when vol ref reaches
" its final value
upfact : 0.4 " Specify where to start on vol ref curve

end
```

B.3.2 reg2.t

```
discrete system reg2

"--------------------------------------------------------------------------------
" Regulates p and h in the volume reference regulator
"--------------------------------------------------------------------------------

time t
tsamp ts

state  intp  inth
new nintp ninth

input Vol ve
output p h vtot

"--------------------------------------------------------------------------------

INITIAL

ts = 0
inth = hstart
intp = pstart

"--------------------------------------------------------------------------------

SORT

"--------------------------------------------------------------------------------
" Calculate next sampling time
"--------------------------------------------------------------------------------

tstep = if t>0.1 then ttot else ttot-0.1
ts = t + tstep

"--------------------------------------------------------------------------------
" Adaption of tank pressure
"--------------------------------------------------------------------------------

errflag = ve>perr
tnintp = intp + kp*ve
nintp = if errflag AND on then tnintp else intp
p = intp

"--------------------------------------------------------------------------------
" Calculate new tank heigth
"--------------------------------------------------------------------------------

ninth = inth
h = inth

"--------------------------------------------------------------------------------
```

```
" Calculate packaged volume
"------------------------------------------------------------------------------

vtot = Vol

"------------------------------------------------------------------------------
" Constants
"------------------------------------------------------------------------------

hstart : 0.253 " ÄmÅ starting height
pstart : 0 " ÄPaÅ starting tank pressure
kp : 1e7 " pressure adaption gain
perr : 50e-6 " max allowed filling error before pressure adaption occurs
ttot : 1.000 " ÄsÅ total time
tfill : 0.690 " ÄsÅ filling time
on : 0 " reg2 on/off-switch 0-off 1-on

end
```

B.3.3 hreg3.t

```
discrete system hreg

"------------------------------------------------------------------------------
" Handles necessary transitions between fast sampling and slow samplin
" Remark! The pos-regulator always samples every 10 ms. The transitions
" are only to handle numerical problems at valve closing
"
" Tests 911015 - continous modification of critsper with derivative
" action
"
"------------------------------------------------------------------------------

time t
tsamp ts

state  oldpos
new noldpos

input posl

"------------------------------------------------------------------------------
" Calculate next sampling instant
"------------------------------------------------------------------------------

downflag= oldpos>posl
noldpos = posl

d = posl-oldpos
s2 = min(1,abs(d/maxd))
s3 = (1-s2)^dexp

s4a = (posl-poscrit-hu)/(posn-poscrit-hu)
"(* to get equal weighting *)
"(* on posl under or over  *)
"(* poscrit   *)
s4b = (poscrit-hd-posl)/(poscrit-hd-posclose)
s4 = if s4a>0 then s4a else if s4b>0 then s4b else 0
"s5 = if downflag then s4^cexp else 1
s5 = s4^cexp
"*** water seems OK with no slow sampling at opening
"*** but not raspberry cream. Even this didn't help:
"*** decreasing critsper, increasing cexp, changing algor to rkf45

"s1 = if ason then normsper*s3*s5+critsper else normsper
s1 = if ason then (normsper-critsper)*s5+critsper else normsper
swposl = posl/2200
swcrit = poscrit/2200

ts = t + s1

"------------------------------------------------------------------------------
```

```
" Constants
"-----------------------------------------------------------------------------------------

normsper: 10e-3 " ÄsÅ
critsper: 2e-4 " ÄsÅ

posn : 22.5 " ÄmmÅ normal pos
poscrit : 5.6 " ÄmmÅ critical pistion position
posclose: 1.0 " ÄmmÅ close position

maxd : 1
dexp : 1
cexp : 0.15
hu : 1.5 " hysteresis over
hd : 0.5 " hysteresis under

ason : 1

end
```

B.3.4 flow.t

```
macro flow

"
" Macro to simulate model2 with volume references
"


"import func < func / 2
"syst model3 reg reg2 hreg creg
"error 0.0001

store VolÄmodelÅ tankpÄmodelÅ flowÄmodelÅ totlossÄmodelÅ posÄmodelÅ
store poslÄmodelÅ poscritÄmodelÅ vallossÄmodelÅ -add
store dflowÄmodelÅ tsÄregÅ vpackÄregÅ -add
store veÄregÅ errflagÄreg2Å downflagÄhregÅ -add
store otflag ovflag totflag VolrefÄregÅ vpackrefÄregÅ zero -add
store linfactÄregÅ s1ÄhregÅ what i dÄhregÅ s2 s4 s3 s5 -add
store swposl swcrit -add

split 1 1

"axes V 0 2 H 0 2
"plot tÄmodelÅ

axes V 0 0.010 H 0 2
plot s1

simu 0 2 / / / func

split 3 2

ashow vpack Volref vpackref
text 'Pack & ref volume Äm^3Å'

ashow valloss
text 'valloss ÄPaÅ'

"axes V 0 1 H 0 1.78
"show s3
"text 'Factor, derivative'

"axes V 0 1
"show s5
"text 'Factor, critical pos'

ashow i
text 'i'

ashow s1 swposl swcrit
text 'sper posl poscrit'
```

```
"ashow dÄhregÅ
"text 'd'

"ashow what
"text 'pos - pos1'

"axes V 0 1
"show linfact
"text 'linfact'

ashow flowÄmodelÅ
text 'flow Äm^3/sÅ'

"ashow tankpÄmodelÅ
"text 'p ÄPaÅ'

"ashow poslÄmodelÅ posÄmodelÅ poscritÄmodelÅ
"text 'posl & pos & poscrit ÄmmÅ'

"ashow totlossÄmodelÅ
"text 'totloss ÄPaÅ'

"ashow valtAÄmodelÅ
"text 'valve area Ämm^2Å'

"ashow tsÄregÅ
"text 'sampling time   ts'

"ashow  ve zero
"text 'Filling error Äm^3Å'

end
```

B.3.5 initflow.t

```
macro initflow

"
" Macro to initialize model3 in volume reference regulator
"

import func < func / 2
syst model4 reg reg2 hreg3 creg
error 0.0001

end
```

B.3.6 func.t

```
0 0
5.5 0
13 350
22 1450
30 1450
30 1450
0 1
30 1
30 1
0.0000     0.0000
0.0100     0.0000
0.0200     0.0002
0.0300     0.0005
0.0400     0.0013
0.0500     0.0025
0.0600     0.0043
0.0700     0.0068
0.0800     0.0101
0.0900     0.0144
0.1000     0.0197
0.1100     0.0262
0.1200     0.0340
0.1300     0.0432
0.1400     0.0540
0.1500     0.0664
0.1600     0.0806
0.1700     0.0967
0.1800     0.1148
0.1900     0.1350
0.2000     0.1575
0.2100     0.1811
0.2200     0.2047
0.2300     0.2283
0.2400     0.2520
0.2500     0.2756
0.2600     0.2992
0.2700     0.3228
0.2800     0.3465
0.2900     0.3701
0.3000     0.3937
0.3100     0.4173
0.3200     0.4409
0.3300     0.4646
0.3400     0.4882
0.3500     0.5118
0.3600     0.5354
0.3700     0.5591
0.3800     0.5827
0.3900     0.6063
0.4000     0.6299
0.4100     0.6535
```

0.4200	0.6772
0.4300	0.7008
0.4400	0.7244
0.4500	0.7480
0.4600	0.7717
0.4700	0.7953
0.4800	0.8189
0.4900	0.8425
0.5000	0.8650
0.5100	0.8852
0.5200	0.9033
0.5300	0.9194
0.5400	0.9336
0.5500	0.9460
0.5600	0.9568
0.5700	0.9660
0.5800	0.9738
0.5900	0.9803
0.6000	0.9856
0.6100	0.9899
0.6200	0.9932
0.6300	0.9957
0.6400	0.9975
0.6500	0.9987
0.6600	0.9995
0.6700	0.9998
0.6800	1.0000
0.6900	1.0000
0.6900	1.0000

C Experimental results

C.1 Stationary flow

These are the flow values as a function of the filling valve opening position that were measured during experiments. These values are plotted in figure 19.

Piston position [mm]	Flow [l/s]
6.15	0.049
6.53	0.106
7.29	0.274
7.68	0.439
8.06	0.568
8.44	0.746
8.82	0.838
9.20	0.957
9.58	0.968
9.97	1.055
10.35	1.206
10.73	1.370
11.11	1.538
11.49	1.696
12.25	1.986
13.02	2.237
13.78	2.433
15.31	2.761
16.83	2.928
18.36	3.029
19.89	3.074

References

[1] A Simnon Tutorial
Karl Johan Åström, Department of Automatic Control
Lund Institute of Technology, Sweden
CODEN: LUTFD2/(TFRT-3176)/1-87/(1985)

[2] Formelsamling KL2
Kemisk Apparatteknik (1985), Lund Institute of Technology, Sweden

[3] Fysik Gas, vätska, fast fas
Bodil Jönsson & Rune Kullberg
Studentlitteratur, Lund, Sweden 1986
ISBN 91-44-25551-9

[4] LaTeXUser's Guide & Reference Manual
Leslie Lamport
Addison-Wesley Publishing Company
ISBN 0-201-15790-X

[5] *Simnon*™ User's Guide for MS-DOS Computers
SSPA System, Box 24 001, S-400 22 GÖTEBORG

[6] Grundläggande Teknisk Strömningslära
Bo Appelqvist & Dan Loyd
Ingenjörsförlaget AB, Stockholm, Sweden
ISBN 91-7284-092-7

[7] Strömningslära, kompendium
Lund Institute of Technology, Sweden

[8] Tetra Pak Development Report 2719 *(confidential)*
Bengt Nilsson, Tetra Pak, Lund, Sweden

[9] Tetra Pak Development Report 4827 *(confidential)*
Guido Moruzzi, Tetra Pak, Modena, Italy