Master's Thesis 2013

Candidate:

Sobhan Shafiee

Title:

Automatic updating of hydrological models for runoff/inflow forecasting to hydropower system

Telemark University College

Faculty of Technology
Kjølnes
3914 Porsgrunn
Norway

Telemark University College
Faculty of Technology
M.Sc. Programme

MASTER'S THESIS, COURSE CODE FMH606

Student:	**Sobhan Shafiee**
Thesis title:	**Automatic updating of hydrological models for runoff/inflow forecasting to hydropower system**
Signature:	. . . . .
Number of pages:	67
Keywords:	**Hydrology, Prediction model, Observability, Sensitivity analysis, Model calibration**

. . . . . .

Supervisor:	Bernt Lie	sign.: . . . . .
2nd Supervisor:		sign.: . . . . .
Censor:		sign.: . . . . .
External partner:	Skagerak Kraft dr. Beathe Furenes	sign.: . . . . .
Availability:	Close	

Archive approval (supervisor signature)**:** sign.: . . . . . **Date :** . . . . . .

Abstract:

This report is the result of my master thesis, which was performed to improve the production planning in hydropower system, based on forecasting runoff/inflow of water to a reservoir in a catchment area. For fulfilling thesis tasks, a new reformulated runoff/inflow hydrological model was introduced based on mass balance, physical laws and functional model description that was developed for whole catchment area. The model was implemented in MATLAB programming and was validated for both created and available data. Observability of the system was checked, which showed some parts of the system were not structurally observable. That led to selection of nonlinear least squares method as an alternative estimation method instead of Ensemble Kalman filtering.

The model uses many parameters, which were determined for the catchment system by few measurements. The model has been updated before simulation by changing the most important parameters. Sensitivity analysis and identifialbility analysis were conducted to find the most important identifiable parameters.

Automatic model updating (model calibration) was performed to estimate parameters and states of the system simultaneously. Estimator performance was verified by feeding estimated parameters and states to model and confirming good-fit between model runoff and real one.

This report provides an overview on what different kind of information Skagerak Kraft needs to use in the hydrological model in MATLAB and automating computation of this information including XML as proper data storage format.

During this master thesis and based on what was done in modeling part, one conference paper submitted with "Prediction of daily runoff from hydrological catchment area" title in 54rd SIMS conference in Bergen, Norway.

Table of Contents

List of Figure

List of Table

Preface

This thesis is part of fulfillment for my Master of Science degree in System and Control Engineering at Telemark University College in Porsgrunn, Norway. This master thesis ties in important concepts of modeling and simulation of dynamic systems and control theories at a higher level than I learned before, by including mathematical modeling, model developing, structural observability and simultaneous parameter and state estimation in nonlinear large scale system. I have used MATLAB as programming language and excel file as a database.

I greatly appreciate the help and guidance of my supervisor professor Bernt Lie, and acknowledge the external partner Dr. Beathe Furenes from Skagerak Kraft in Porsgrunn, Norway for providing data and many valuable inputs and helps. I would also like to thank Mohammad Saeed Rostami for reading this report and providing useful feedbacks.

Porsgrunn, Norway, June 2013

Sobhan Shafiee

Nomenclature

$\dot{m}$	mass flow rate	$[kg/\sec]$
$\dot{V}$	volume rate	$[mm^3/\sec]$
ρ	density	$[kg/mm^3]$
A	area	$[mm^2]$
V	volume	$[mm^3]$
V''	volume per unit area	$[mm]$
T	temperature	$[°C]$
T_T	threshold temperature	$[°C]$
$\dot{V}''_p$	precipitation	$[mm/day]$
$\dot{V}''_{p,s}$	precipitation in mainland in the form of snow	$[mm/day]$
$\dot{V}''_{p,r}$	precipitation in mainland in the form of rain	$[mm/day]$
$V''_{s.d}$	dry snow	$[mm]$
$V''_{s.s}$	soggy snow	$[mm]$
$\dot{V}''_{d.2.w}$	melting rate from dry snow form to water snow form	$[mm/day]$
$\dot{V}''_{w.2.d}$	freezing rate from water snow form to dry snow form	$[mm/day]$
$\dot{V}''_{s.2.g}$	runoff rate from snow zone to ground zone	$[mm/day]$
α_L	lake surface fraction	dimensionless
k_m	melting factor	$[mm^3/°C.day]$
$V''_{g.w}$	water content in ground zone	$[mm]$
g_T	ground saturation threshold	$[mm]$
$\dot{V}''_{s.2.g}$	runoff from snow zone to ground zone	$[mm/day]$
$\dot{V}''_{g.2.s}$	runoff rate from ground zone to soil zone	$[mm/day]$
$\dot{V}''_{g.e}$	evapotranspiration rate from ground zone	$[mm/day]$
α	snow surface fraction	dimensionless
α_w	saturation coefficient	dimensionless
β	ground zone shape coefficient	dimensionless
$\dot{V}''_{gepot}$	potential evapotranspiration rate	$[mm/day]$

$V''_{s.w}$	water content in soil zone	$[\,mm\,]$
$\dot{V}''_{s.2.b}$	runoff rate from soil zone to base zone	$[\,mm/day\,]$
$\dot{V}''_{s.2.sr}$	runoff rate from soil zone to surface runoff	$[\,mm/day\,]$
$\dot{V}''_{s.2.fr}$	runoff rate from soil zone to fast runoff	$[\,mm/day\,]$
s_T	soil zone saturation threshold	$[\,mm\,]$
$PERC$	percolation	$[\,mm/day\,]$
α_1	discharge frequency for surface runoff	$[\,day^{-1}\,]$
α_2	discharge frequency for fast runoff	$[\,day^{-1}\,]$
α_3	discharge frequency for base runoff	$[\,day^{-1}\,]$
$V''_{b.w}$	water content in base zone	$[\,mm\,]$
$\dot{V}''_{pl}$	precipitation in lake	$[\,mm/day\,]$
$\dot{V}''_{b.2.br}$	runoff rate from base zone to base runoff	$[\,mm/day\,]$
$\dot{V}''_{l.e}$	rate of evapotranspiration from lake	$[\,mm/day\,]$

1 Introduction

Despite that many attempts have been done during the past decades on developing hydrological models, they are still imprecise because they have been developed based on physical laws such as mass balance and energy balance. These models typically contain parameters the exact values of which are not known.Consequently, the uncertainties of parameter values influence hydrological model performance. (Dong, 2009) As a result optimal estimation of model parameters (calibration) is essential to improve the predictability and accuracy of the hydrological model.

Manual calibration is trial-and-error method is time consuming procedure that often requires needs broad knowledge of hydrology, while automatic calibration uses optimization of an appropriate objective function to seek best-fit parameter possibilities. It is faster and more straightforward in cases with complex nonlinear system and a large number of parameters (Liu, 2010). However, automatic calibration methods do not take any uncertainties such as input data and model error into account (Chen, 2012).

From 1990s, sequential data assimilation methods such as Ensemble Kalman filtering (Moradkhani, 2005) have been applied in hydrology and became more popular for its capability to handle various sources of uncertainties, especially, for nonlinear hydrological model due to computational efficiency, straightforward implementation and its ability to estimate time varying parameters and simultaneous estimation of states (Chen, et al., 2012).

This master thesis is continuous of one of Telemark University College's group project with *"Hydrological modeling for Runoff/inflow forecasting to hydropower system"* title and attempts to implement automatic updating of hydrological model to obtain more accurate future runoff/inflow based on weather forecasting by discussing two methods of model calibration (Ensemble Kalman filtering and nonlinear least squares) to select and test better one, based on model structure and available data .

1.1 Background

Regard to deregulation and linearization of electricity market in the Nordic countries the Nordic electricity exchange was established to run the leading power market and offers three different as is shown in Table 1-1.

Table 1-1: Three different Nordic electricity exchange markets (Nord Pool Spot, 2012)

Market name	Validation time	Adjustability	Market share
Elspot market	1day before delivery	Yes	70%
Elbas market	1hour before delivery	Yes	30%
Balancing market	Close to delivery	Yes	

Skagerak Kraft generates significant part of electricity production from hydro power stations, consequently, interested in a hydrological model for runoff forecasting as a desirable tool that is capable of estimating how much water could be available next days for obtaining the maximum profit of its available water resources based on defined cost function and considering operational constraints. Skagerak Kraft is interested in minimizing waste of its water resources and try to feed the turbine as much as possible. However variation in precipitation and a lag of runoff from catchments cause variation in the amount of water that flows through the turbines. In addition discharge limitations to satisfy the required maximum and minimum flow in rivers for safety reasons such as preventing flood, etc. and environmental concerns such as water need for fishery, etc. should be considered as constraints simultaneously.

1.2 Previous work

Adjusting model parameters to find unique and conceptually realistic parameters that results in an acceptable match between model prediction and observed data is an important but difficult step in model development. As a result, automatic updating of hydrological models for runoff/inflow forecasting to hydropower system is needed as a good water resource management tool for production planning and utilize as much as possible of available water at any time.

Automatic calibration has been developed using different schemes. A few examples are the models by Bergström (1976), Sugawara (1979), Sorooshian and Dracup (1980), Gupta and Sorooshian (1985), Zhang (1988), and Harlin (1991). The common problems in automatic calibration are interdependency between the model parameters; indifference of the objective function to the values of inactive parameters, (Jain, 1993) and ignoring the model and input data uncertainty (Chen, et al., 2012) and (Xingnan & Lindström, 1997).

1.3 Overview of thesis

The reformulated hydrological model for runoff/inflow forecasting which is presented here is based on mass balance and physical laws which I learned during Modeling and simulation of dynamic systems course. I should point out that this model is based on concepts that are similar to HBV-3 model (Bergström, 1973). I have used different symbols and formulations which make it more meaningful and comprehensible. The model is simulated in MATLAB programming language. Model validation is performed using created data and also available information from a real catchment region.

The main purpose of this study is discuss methods of model calibration and check the feasibility of EnKF [1], and test out the nonlinear least squares method as a selected calibration method and updating the model with available data. Sensitivity analysis is also conducted to find out how many parameters and which of them should be selected during the calibration procedure.

[1] Ensemble Kalman Filter

2 Process description

2.1 Current Skagerak Kraft's model and software

Currently, Skagerak is using the LANDPINE model which is implemented inside a framework as license-based software to predict the runoff from catchments and it is called PINE, Windows NT based. The model is hidden inside the framework without access to the source code, which makes it impossible for users to make modifications in the program. It is desirable for Skagerak to use an open source framework where they have some knowledge about details of the model and its parameters. The Skagerak Kraft is interested in looking into the possibility of developing a hydrological model with an open source code which makes the company able to use it for runoff prediction in catchments (Shafiee, et al., 2012).

2.2 Model Hypothesis

The core feature of any hydrological study is the hydrological cycle which is shown in Figure 2-1.Hydrological cycle describes how water in solid, liquid and gas forms move above and below the surface of the Earth continuously (Moreda, 1999). The hydrological model which we want to develop for prediction of runoff for hydroelectric power station is also based on this water cycle (one direction).

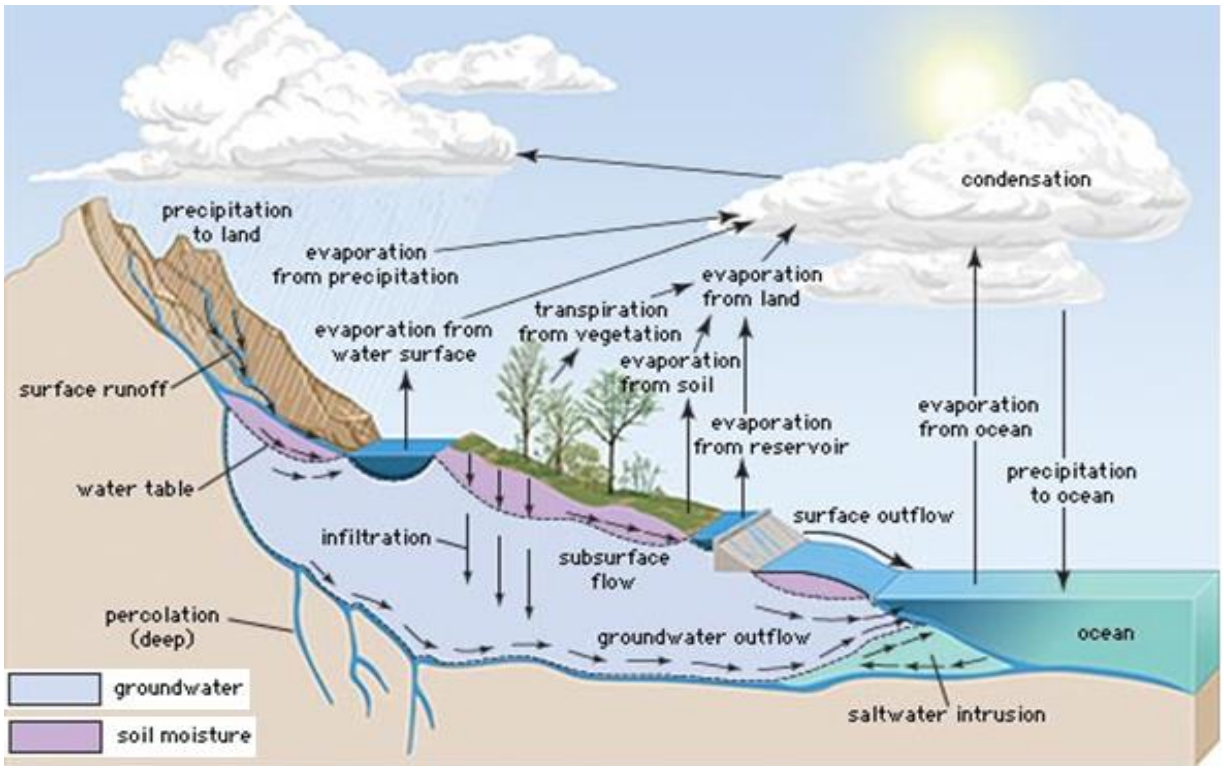


Figure 2-1: Hydrological cycle (Encyclopædia Britannica, 2008)

The term catchment is used to refer to a hydrological unit area where any source of water like precipitation and surface water converge and ends up points that could be a river, lake, reservoir and sea. A catchment could be further be divided into sub catchments depends on its characteristics.

According to process concepts and functional model description, each hydrologic unit (catchment or subcathment) accepts precipitation and temperature as inputs, operates on them internally, and generates total runoff as output as indicated in Figure 2-2, which represents the system operation.

Figure 2-2: Schematic of catchment system operation

As illustrated in Figure 2-3, which describes the hydrological cycle, the condensed vapor water precipitates in snow or rain forms depending on atmospheric temperature. According to environmental temperature and land vegetation, precipitation could be stored as solid form (snow/ice) or liquid form (water) on the upper soil level zone. Accumulated water in upper soil level can increase to the saturation level by precipitation as rain or melting snow, then water can runoff to the soil level zone.

The upper soil level runoff, which reaches to the soil level zone, could disappear due to evapotranspiration or infiltration into porous soil layer. If the water content in soil is equal to the soil saturation value, it streams up to the upper ground zone. Evapotranspiration in soil level is also correlated with soil moisture and upper soil level runoff, which is why maximum evapotranspiration could be reached when the soil moisture is more than soil saturation level.

The infiltration runoff which is coming out from soil level zone constitutes the Surface flow, Fast runoff, Percolation runoff and storage. In the other word, infiltrated water could be percolated to the ground zone, flows down the slope as surface flow, streams as Fast runoff which feeds river or either stored as the water content in the upper ground zone.

Percolated water is coming into ground zone, emerges as spring and constitutes the last compartment of total runoff or store as water content in ground zone. Lakes also play an important non-dynamic role in hydrological models due to water transformation that it has with ground zone as the last layer. It is reasonable to assume that the maximum evapotranspiration should be considered for the lake. (Moreda, 1999)

The advantages of using modeling method outlined above is as following:

- It is constructed based on conceptual foundations
- Semi-distributed model applicable by use of elevation coefficient
- It is understandable for users
- Water content in each layer are known as states of the system

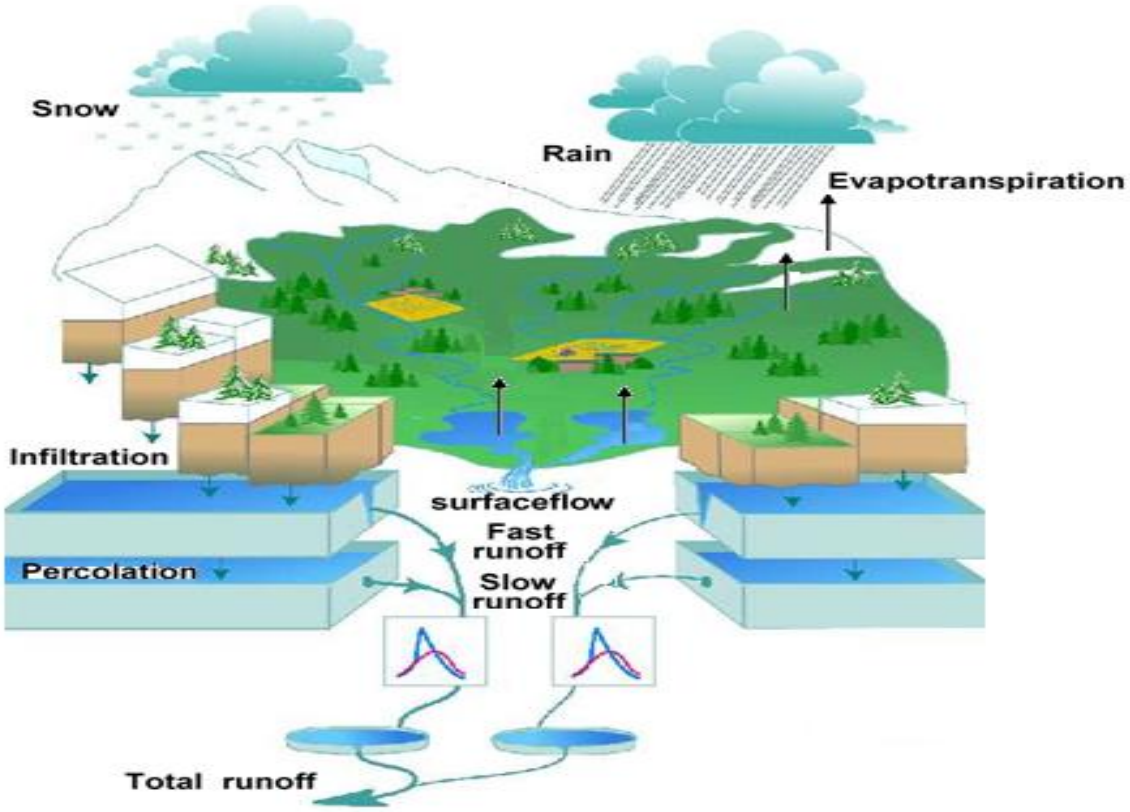


Figure 2-3: Hydrological components of catchment model (Emily Youcha, n.d.)

2.3 Model functional description

According to hydrological components of catchment model and model hypothesis, the schematic of the catchment system operation could be represented as detailed in Figure 2-4. Each module is characterized by inputs and outputs.

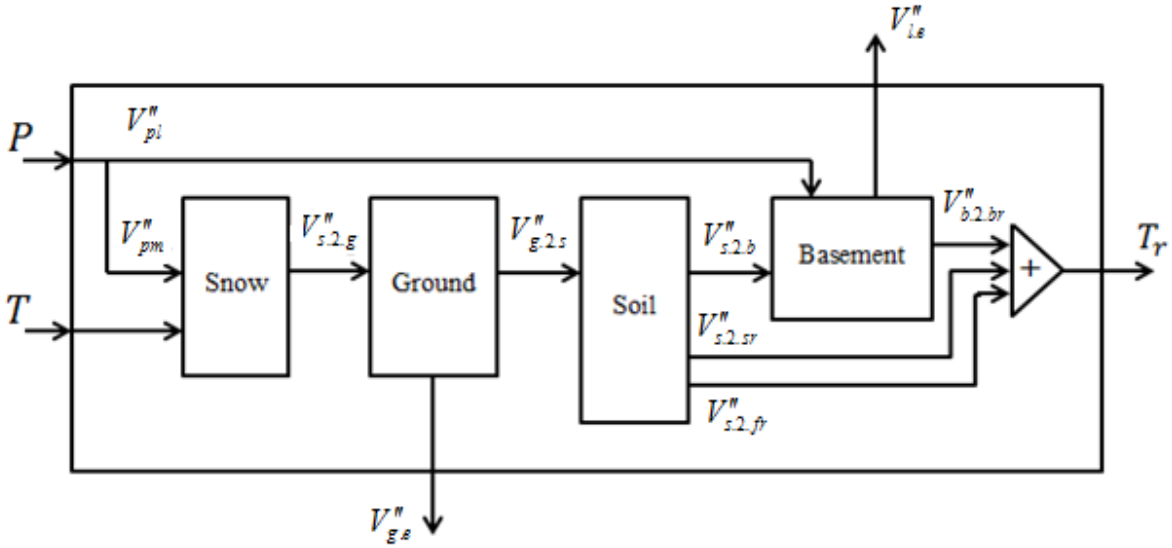


Figure 2-4: Functional description of catchment hydrologic model

3 Model development

3.1 Model zones

As shown in Figure 2-4, the whole model consist of snow zone, ground zone, soil zone and basement zone that are connected to each other in cascade form. Each zone has its own model which connects inputs to outputs by a specific sub-model which is obtained based on mass balance.

$$\frac{dm}{dt} = \dot{m}_{input} - \dot{m}_{output} \tag{3-1}$$

$$m = \int \rho dV \Rightarrow \quad \dot{m} = \rho \dot{V} \tag{3-2}$$

By assuming a constant density, (ρ =constant) we have:

$$\frac{d(\rho V)}{dt} = \rho \dot{V}_{input} - \rho \dot{V}_{output} \tag{3-3}$$

$$\frac{dV}{dt} = \dot{V}_{input} - \dot{V}_{output} \tag{3-4}$$

Since in hydrology and meteorology the amount of precipitation is defined with meter per day, I rewrite the precipitation relationship as follow:

$$\dot{V} = \dot{V}'' \cdot A \tag{3-5}$$

It is also convenient to operate with volume per area

Where,

- V is volume of water with m^3 dimension
- V'' is volume of water per unit area with m dimension
- A is area of water with m^2 dimension

Consequently, mass balance in new form expressed as:

$$\frac{dV''}{dt} = \dot{V}''_{input} - \dot{V}''_{output} \tag{3-6}$$

3.1.1 Snow zone

By considering the ambient temperature (T) in comparison with threshold temperature (T_T), and implementing water balance on snow zone model which is given in Figure 3-1 we have:

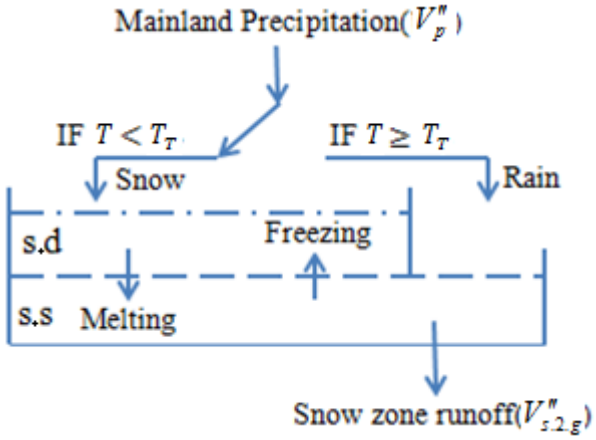


Figure 3-1: Snow zone model (Magne Fjeld, 1980)

$$\frac{dV''_{s.d}}{dt} = \dot{V}''_{p,s} + \dot{V}''_{w.2.d} - \dot{V}''_{d.2.w}$$
$$\frac{dV''_{s.s}}{dt} = \dot{V}''_{p,r} - \dot{V}''_{w.2.d} + \dot{V}''_{d.2.w} - \dot{V}''_{s.2.g} \quad (3\text{-}7)$$

Here:

$$\dot{V}''_{p,s} = \begin{cases} (1-\alpha_L)\dot{V}''_p & if \quad T \leq T_T \\ 0 & if \quad T > T_T \end{cases} \quad (3\text{-}8)$$

$$\dot{V}''_{p,r} = (1-\alpha_L)\dot{V}''_p - \dot{V}''_{p,s} \quad (3\text{-}9)$$

$$\dot{V}''_{d.2.w} = \begin{cases} k_m(T-T_T) & if \quad T > T_T \quad and \quad V''_{s.d} \geq 0 \\ 0 & otherwise \end{cases} \quad (3\text{-}10)$$

$$\dot{V}''_{w.2.d} = \begin{cases} k_m(T_T - T) & if \quad T \leq T_T \quad and \quad V''_{s.s} \geq 0 \\ 0 & otherwise \end{cases} \quad (3\text{-}11)$$

$$\dot{V}''_{s.2.g} = \begin{cases} \dot{V}''_{p,r} & if \quad T > T_T \quad and \quad V''_{s.s} = .V''_{s.d} = 0 \\ (1+\alpha_w)\dot{V}''_{d.2.w} + \dot{V}''_{p,r} & if \quad V''_{s.s} > 0 \quad and \quad V''_{s.s} = (\alpha_w.V''_{s.d}) > 0 \\ 0 & otherwise \end{cases} \quad (3\text{-}12)$$

Where,

- $\dot{V}''_{p}$: precipitation
- $\dot{V}''_{p,s}$: precipitation in mainland in the form of snow
- $\dot{V}''_{p,r}$: precipitation in mainland in the form of rain
- T : ambient temperature
- T_T : threshold temperature
- $\dot{V}''_{d.2.w}$: melting rate form $\overbrace{\text{dry snow form}}^{\text{d}}\text{-}\underbrace{\text{to}}_{2}\text{-}\overbrace{\text{water snow form}}^{\text{w}}$
- $\dot{V}''_{w.2.d}$: freezing rate form $\overbrace{\text{water snow form}}^{\text{w}}\text{-}\underbrace{\text{to}}_{2}\text{-}\overbrace{\text{dry snow form}}^{\text{d}}$
- $\dot{V}''_{s.2.g}$: runoff rate from $\overbrace{\text{snow zone}}^{\text{s}}\text{-}\underbrace{\text{to}}_{2}\text{-}\overbrace{\text{ground zone}}^{\text{g}}$
- $V''_{s.d}$: dry snow (water equivalent of accumulated snow)
- $V''_{s.s}$: soggy snow (free water stored in snow zone)

Water content in the snow zone includes dry accumulated snow ($s.d$) and soggy snow ($s.s$) which are known as two state variables of this layer. When the amount of free water in the snow zone reaches to saturation level ($\alpha_w s.d$), it can't be stored as soggy snow, hence causes $V''_{s.2.g}$. (Magne Fjeld, 1980)

On the other hand, the melting rate of snow between dry snow and soggy snow is dependent on many factors such as snow density and topography, but for simplicity a linear correlation between ambient temperature and melting rate of snow ($k_m(T - T_T)$) could be assumed as a good estimation (Göran Lindström, 1997).

3.1.2 Ground zone

By implementing water balance on ground zone as displayed in Figure 3-2 we have:

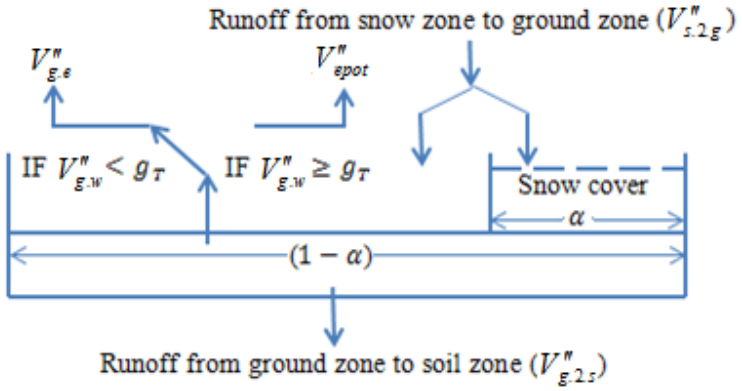


Figure 3-2: Ground zone model

$$\frac{dV''_{g.w}}{dt} = \dot{V}''_{s.2.g} - \dot{V}''_{g.2.s} - (1-\alpha)\dot{V}''_{g.e} \tag{3-13}$$

Here:

$$\dot{V}''_{s.2.g} = \begin{cases} \dot{V}''_{p,r} & \text{if} \quad T > T_T \quad \text{and} \quad V''_{s.s} = .V''_{s.d} = 0 \\ (1+\alpha_w)\dot{V}''_{s.2.w} + \dot{V}''_{p,r} & \text{if} \quad V''_{s.s} > 0 \quad \text{and} \quad V''_{s.s} = (\alpha_w.V''_{s.d}) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3-14}$$

$$\dot{V}''_{g.2.s} = \begin{cases} \left(\frac{V''_{g.w}}{g_T}\right)^{\beta} \times \dot{V}''_{s.2.g} & \text{if} \quad 0 < V''_{g.w} < g_T \\ \dot{V}''_{s.2.g} & \text{if} \quad V''_{g.w} \geq g_T \end{cases} \tag{3-15}$$

$$\dot{V}''_{g.e} = \begin{cases} \left(\frac{\dot{V}''_{g.w}}{g_T}\right) \times \dot{V}''_{epot} & \text{if} \quad V''_{g.w} < g_T \\ \dot{V}''_{gepot} & \text{if} \quad V''_{g.w} \geq g_T \end{cases} \tag{3-16}$$

Where,

- $V''_{g.w}$: water content in ground zone
- g_T : ground saturation threshold
- $\dot{V}''_{s.2.g}$: runoff rate from $\overbrace{\text{snow zone}}^{\text{s}}$ - $\underbrace{\text{to}}_{2}$ - $\overbrace{\text{ground zone}}^{\text{g}}$

– $\dot{V}''_{g.2.s}$: runoff rate from $\overbrace{\text{ground zone}}^{\text{g}}$ - $\underbrace{\text{to}}_{2}$ - $\overbrace{\text{soil zone}}^{\text{s}}$

– $\dot{V}''_{g.e}$: evapotranspiration rate from ground zone

In ground zone the evapotranspiration just happens to that surface of area which is not covered by snow. To account for fractional snow coverage, the average fraction snow covered area, α_w is defined. In the ground zone when the amount of water, gw is equal to its saturation value, g_T the water will flow on to the Soil zone. In this situation potential amount of evapotranspiration, V''_{epot} occurs. On the other hand, if ground water value is less than saturation value, the runoff could be represented by $\left(\frac{V''_{g.w}}{g_T}\right)^{\beta} \times V''_{s.2.g}$. (Magne Fjeld, 1980)

3.1.3 Soil zone

By implementing water balance on soil zone model as displayed in Figure 3-3, we have:

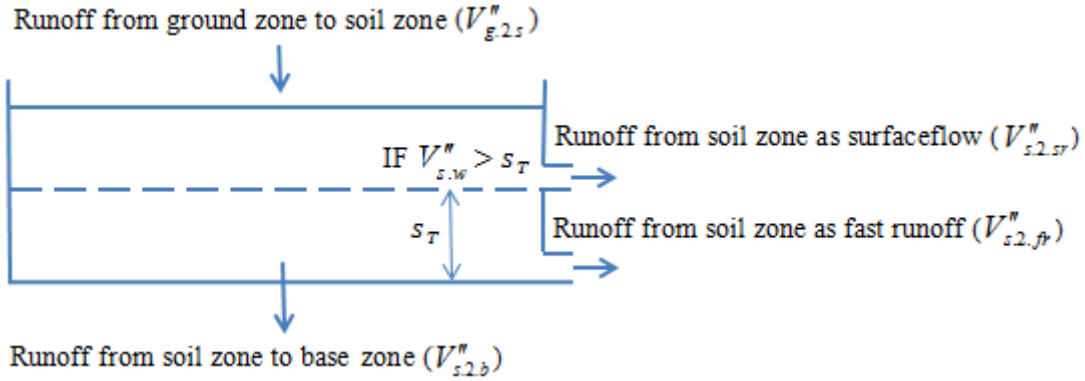


Figure 3-3: Soil zone model (Magne Fjeld, 1980)

$$\frac{dV''_{s.w}}{dt} = \dot{V}''_{g.2.s} - \dot{V}''_{s.2.b} - \dot{V}''_{s.2.sr} - \dot{V}''_{s.2.fr} \tag{3-17}$$

Here:

$$\dot{V}''_{g.2.s} = \begin{cases} \left(\frac{\dot{V}''_{g.w}}{g_T}\right)^{\beta} \times \dot{V}''_{s.2.g} & \text{if} \quad 0 < V''_{g.w} < g_T \\ \dot{V}''_{s.2.g} & \text{if} \quad V''_{g.w} \geq g_T \end{cases} \tag{3-18}$$

$$\dot{V}''_{s.2.b} = (1 - a_L) \cdot PERC \tag{3-19}$$

$$\dot{V}''_{s.2.sr} = \begin{cases} a_1(V''_{s.w} - s_T) & if \quad V''_{s.w} > s_T \\ 0 & if \quad V''_{s.w} \le s_T \end{cases} \tag{3-20}$$

$$\dot{V}''_{s.2.fr} = a_2(V''_{s.w}) \tag{3-21}$$

Where,

- $V''_{s.w}$: water content in soil zone
- $\dot{V}''_{g.2.s}$: runoff rate from $\overbrace{\text{groungd zone}}^{\text{g}}\text{-}\underbrace{\text{to}}_{2}\text{-}\overbrace{\text{soil zone}}^{\text{s}}$
- $\dot{V}''_{s.2.b}$: runoff rate from $\overbrace{\text{soil zone}}^{\text{s}}\text{-}\underbrace{\text{to}}_{2}\text{-}\overbrace{\text{base zone}}^{\text{b}}$
- $\dot{V}''_{s.2.sr}$: runoff rate from $\overbrace{\text{soil zone}}^{\text{s}}\text{-}\underbrace{\text{to}}_{2}\text{-}\overbrace{\text{surface runoff}}^{\text{sr}}$
- $\dot{V}''_{s.2.fr}$: runoff rate from $\overbrace{\text{soil zone}}^{\text{s}}\text{-}\underbrace{\text{to}}_{2}\text{-}\overbrace{\text{fast runoff}}^{\text{s}}$
- s_T: soil zone saturation threshold

If the water content in the soil zone becomes more than its saturation level, s_T then surface runoff flows as one compartment of total runoff, otherwise only fast runoff flows in the soil zone. Parameters a_1 and a_2 represent as the discharge frequencies for surface runoff and fast runoff respectively. Furthermore, based on hydrological concepts, for simplification purpose the rate of $V''_{s.2.b}$ is considered as a constant parameter when $V''_{s.w}$ is larger than zero. $V''_{s.2.b}$ varies depending on soil materials which means the rocky soil should have smaller value as compared to sandy soil.

3.1.4 Basement zone

By implementing water balance on base zone model as displayed in Figure 3-4 we have:

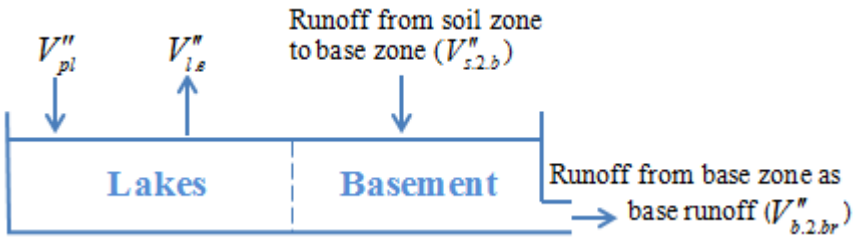


Figure 3-4: Base zone model

$$\frac{dV''_{b.w}}{dt} = \dot{V}''_{s.2.b} + \dot{V}''_{pl} - \dot{V}''_{b.2.br} - \dot{V}''_{l.e} \tag{3-22}$$

Here:

$\dot{V}''_{s.2.b} = (1 - a_L) \cdot PERC$

$\dot{V}''_{pl} = a_L \cdot \dot{V}''_{pm}$

$\dot{V}''_{b.2.br} = a_3(V''_{b.w})$

$\dot{V}''_{l.e} = a_L \cdot \dot{V}''_{epot}$

Where,

- $V''_{b.w}$: water content in base zone
- $\dot{V}''_{pl}$: precipitation in lake
- $\dot{V}''_{s.2.b}$: runoff rate from $\overbrace{\text{soil zone}}^{\text{s}}$ - $\underbrace{\text{to}}_{2}$ - $\overbrace{\text{base zone}}^{\text{b}}$
- $\dot{V}''_{b.2.br}$: runoff rate from $\overbrace{\text{base zone}}^{\text{b}}$ - $\underbrace{\text{to}}_{2}$ - $\overbrace{\text{base runoff}}^{\text{br}}$
- $\dot{V}''_{l.e}$: rate of evapotranspiration from lake

In the lowest layer beside base zone, there are lakes that are assumed to link in a nondynamic connection with basement zone. Therefore the optimal amount of evapotranspiration will happen where we have lakes. Parameter α_L is a coefficient that illustrates the fractional area covered by lakes and a_3 is discharge frequency for base runoff.

3.1.5 Total runoff

The total catchment runoff rate is obtained by summing $\dot{V}_{b2br}$, $\dot{V}_{s2\,fr}$ and $\dot{V}_{s2sr}$ as in what follow:

$$\frac{d\dot{V}_{tot}}{dt} = \begin{cases} \left[a_1(V''_{s.w} - s_T) + a_2(V''_{s.w})\right] \times (1 - \alpha_L)A + Aa_3(V''_{b.w}) & if \quad V''_{s.w} > s_T \\ a_2 V''_{s.w}(1 - \alpha_L)A + Aa_3(V''_{b.w}) & if \quad V''_{s.w} \le s_T \end{cases} \tag{3-23}$$

Where,

- A : total catchment area.

3.2 Implementation of developed model

This section will simulate and implement the developed HBV model for the Eggedal catchment. According to given data the distribution of elevations on the surface of Eggedal catchment, 10 different elevations are defined that each of them has 10% of whole area. Figure 3-5 shows the schematic of catchment based on hypsography and topographic map, the red hatched zones in all levels have same area.

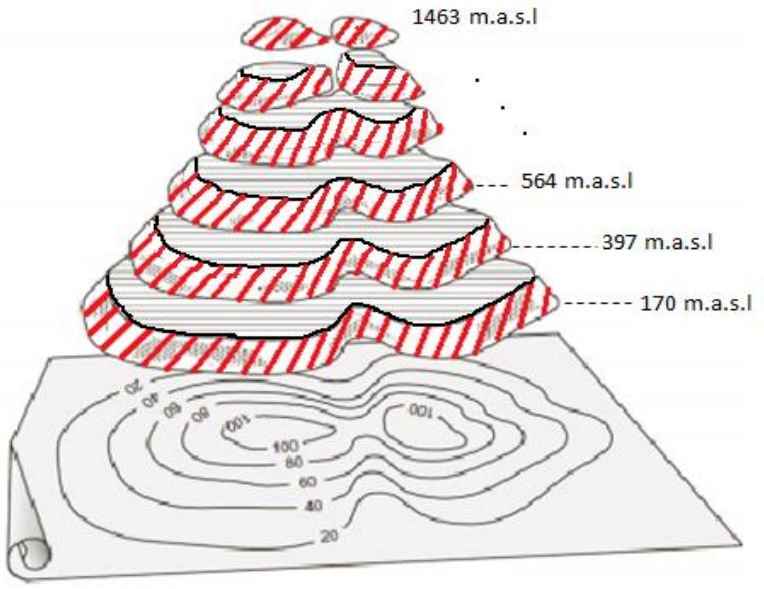


Figure 3-5: Schematic of catchment based on hypsography and topography map

Since temperature will changes depending on altitude change, each altitude range that is shown in Figure 3-5, has its own dry snow and soggy snow values which are known as snow zone states ($V''_{s.d}, V''_{s.s}$). The other states ($V''_{g.w}, V''_{s.w}, V''_{b.w}$) are similar for all altitudes. As Figure 3-6 illustrates, for simplicity it assume that the precipitation in all altitudes is equal, but the difference in temperature causes different dry snow and soggy snow in each altitude.

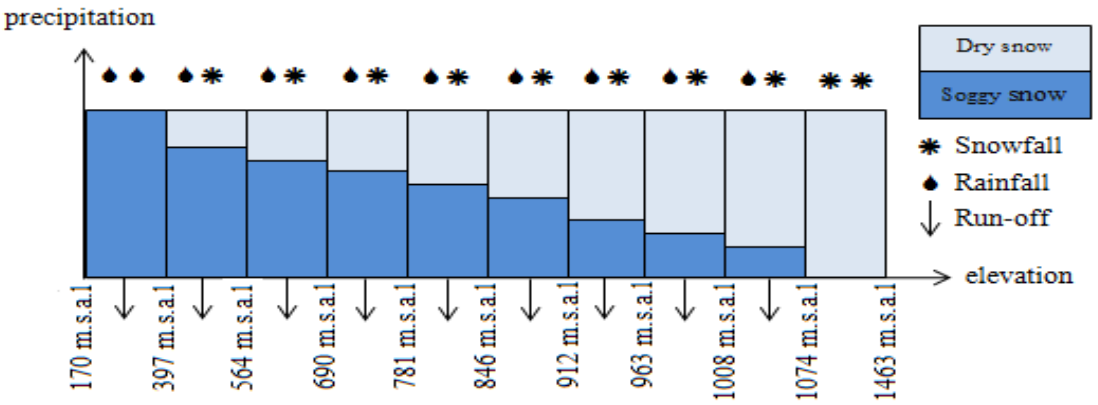


Figure 3-6: Distributed snow zone according to elevation

Consequently, total runoff from the snow zone to the ground zone is obtained by summation of the runoff for all altitudes. With this assumption of dividing catchment into 10 different altitude ranges, we will have 23 states (20 for snow zone and 3 for the other zones).

Table 3-1: Model components (states, parameters and inputs)

State	Description	State	Description	Parameter	Description
$V''_{s.d\_h_1}$	Dry snow in h_1	$V''_{s.d\_h_7}$	Dry snow in h_7	$k_m(\frac{m^3}{°C.day})$	melting factor
$V''_{s.s\_h_1}$	Soggy snow in h_1	$V''_{s.s\_h_7}$	Soggy snow in h_7	$T_T(°C)$	threshold temperature
$V''_{s.d\_h_2}$	Dry snow in h_2	$V''_{s.d\_h_8}$	Dry snow in h_8	α_w	saturation coefficient
$V''_{s.s\_h_2}$	Soggy snow in h_2	$V''_{s.s\_h_8}$	Soggy snow in h_8	α	snow surface fraction
$V''_{s.d\_h_3}$	Dry snow in h_3	$V''_{s.d\_h_9}$	Dry snow in h_9	$g_T(mm)$	ground saturation threshold
$V''_{s.s\_h_3}$	Soggy snow in h_3	$V''_{s.s\_h_9}$	Soggy snow in h_9	β	ground zone shape coefficient
$V''_{s.d\_h_4}$	Dry snow in h_4	$V''_{s.d\_h_{10}}$	Dry snow in h_{10}	$s_T(mm)$	soil saturation threshold
$V''_{s.s\_h_4}$	Soggy snow in h_4	$V''_{s.s\_h_{10}}$	Soggy snow in h_{10}	α_L	lake surface fraction
$V''_{s.d\_h_5}$	Dry snow in h_5	$V''_{g.w}$	Ground zone water	$a_1(day^{-1})$	discharge frequency for surface runoff
$V''_{s.s\_h_5}$	Soggy snow in h_5	$V''_{s.w}$	Soil zone water	$a_2(day^{-1})$	discharge frequency for fast runoff
$V''_{s.d\_h_6}$	Dry snow in h_6	$V''_{b.w}$	Base zone water	$a_3(day^{-1})$	discharge frequency for base runoff
$V''_{s.s\_h_6}$	Soggy snow in h_6			$A(km^2)$	catchment area
				$PERC(\frac{mm}{day})$	percolation from soil zone to base zone

3.2.1 Model verification based on created data

In order to make sure that the model is working in a reasonable manner and also correct implementation in MATLAB, it is necessary to verify the results to check the model matches the specifications and assumption according to model concept. For more simplicity and better analysis, one year set of data was created by separating precipitation and temperature of given data from 01/08/1978 to 31/07/1979 and assuming the average precipitation and temperature in each month. As indicated in Figure 3-7, during September (day 30 to day 60) there is a small runoff due to precipitation rise, but from the first of October to end of March (day 60 to day 240) because of cold weather, precipitation will be stored as dry snow in snow zone, consequently the snow zone runoff is become zero. The maximum amount of runoff was reached during May, while it can't be continued in June because of high amount of

evapotranspiration. Because of high precipitation in July and warm weather, all rain contributes to making runoff that is obvious from Figure 3-7.

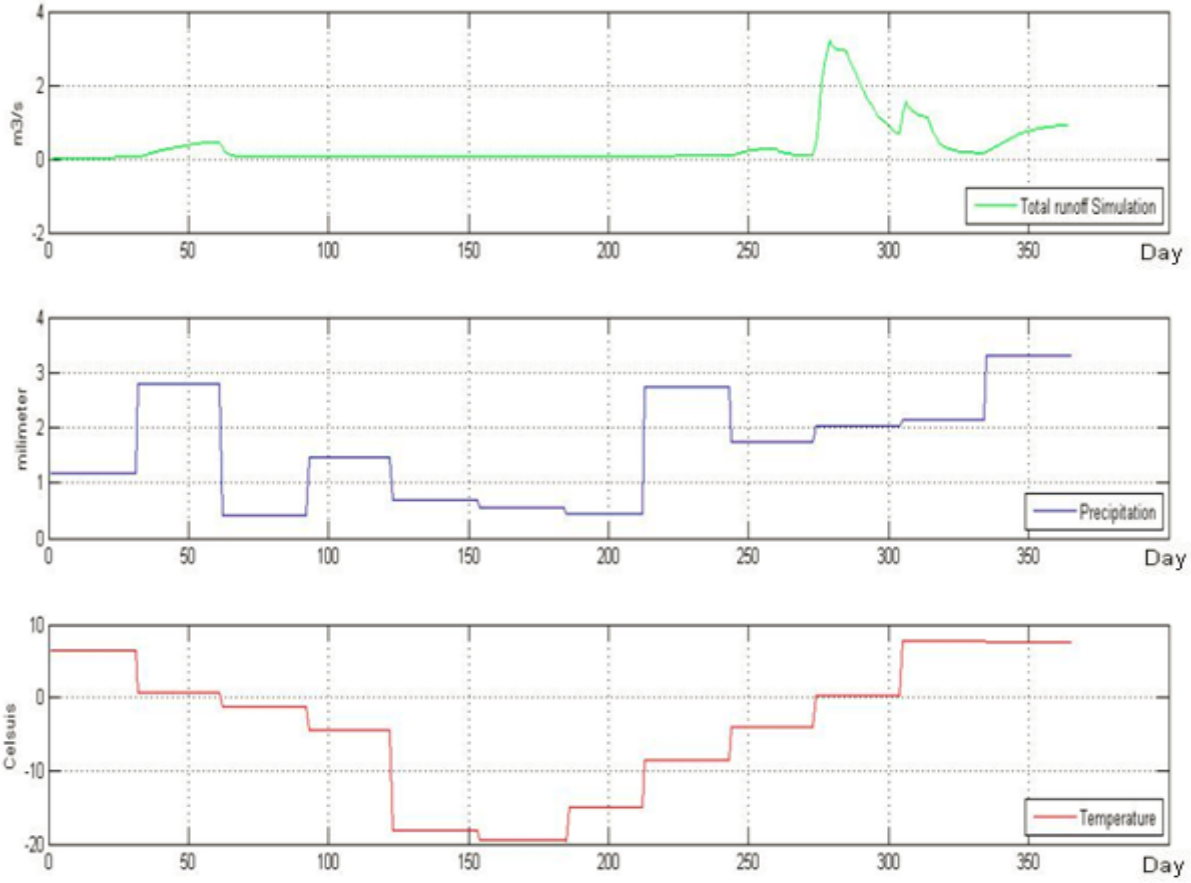


Figure 3-7: Simulated runoff variation with respect to precipitation and temperature change

3.2.2 Model validation based on real catchment information

As a part of model development the accuracy of the model as a representation of the real system should be checked to find out how much the simulated model is imitating the real system.

Table 3-2 contains parameter values that are used in order to get a good approximate result. Some parameters could be found by studying the catchment area (A ,α_L, α), some of the other could be estimated based on hydrological knowledge (average variation in temperature as a function of altitude (k_m , α_w, T_T), and finally, rest of them are those parameters that could be estimated from other catchment studies like g_T , β, a_1, a_2 and a_3 (Magne Fjeld, 1980).

Table 3-2: Parameters value

Parameter symbol	Value	Unit	Description
k_m	5.2	$\frac{m^3}{°C.day}$	melting factor
T_T	0.8	$°C$	threshold temperature
α_w	0.08	-	saturation coefficient
α		-	snow surface fraction
g_T	50	mm	ground saturation threshold
β	2	-	ground zone shape coefficient
s_T	20	mm	soil saturation threshold
α_L	0.032	-	lake surface fraction
a_1	0.547	day^{-1}	discharge frequency for surface runoff
a_2	0.489	day^{-1}	discharge frequency for fast runoff
a_3	0.0462	day^{-1}	discharge frequency for base runoff
A	309.42	km^2	catchment area
PERC	0.25	$\frac{mm}{day}$	percolation from soil zone to base zone

Results of snow zone simulation which is shown in Figure 3-8, indicates that dry snow ($V''_{s.d}$) is accumulating when the temperature falls to below $0.8°C$ through autumn, winter and first month of spring (day 30 to day 240). During the rest of spring, when the temperature value becomes more than $0.8°C$, an increase of soggy snow ($V''_{s.s}$) can be observed. In the summer season, since there is no area covered by snow, it seems reasonable that the dry snow and soggy snow become zero, therefore all precipitation that comes as rain without the presence of snow will continue its path to the ground zone.

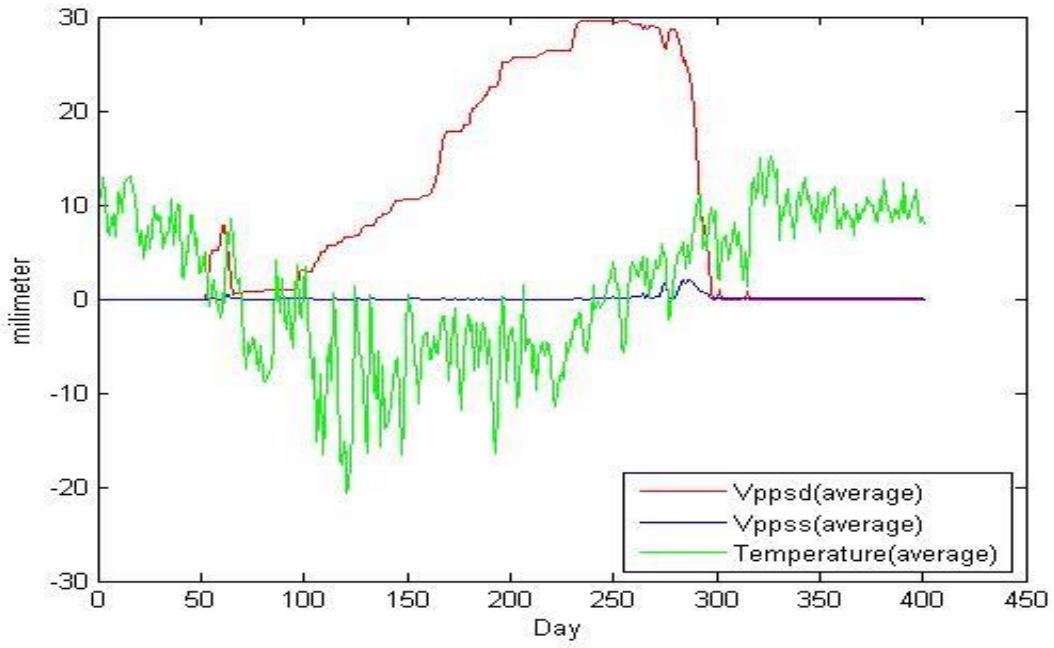


Figure 3-8: Snow zone simulation regards to average temperature

As illustrated in Figure 3-9, during August and September the water content in ground zone $V''_{g.w}$ is affected by evapotranspiration and small amount of snow zone runoff, while from October to end of March it is going to reach the ground zone threshold ($g_T = 50mm$), but it is not happening due to non-zero value of evapotranspiration. Small amount of precipitation during May and June lead to small $V''_{s.2.g}$ and drastic ground water drop. On the other hand, the ground water has been increased since of $V''_{s.2.g}$ rise from day 330 to day 360.

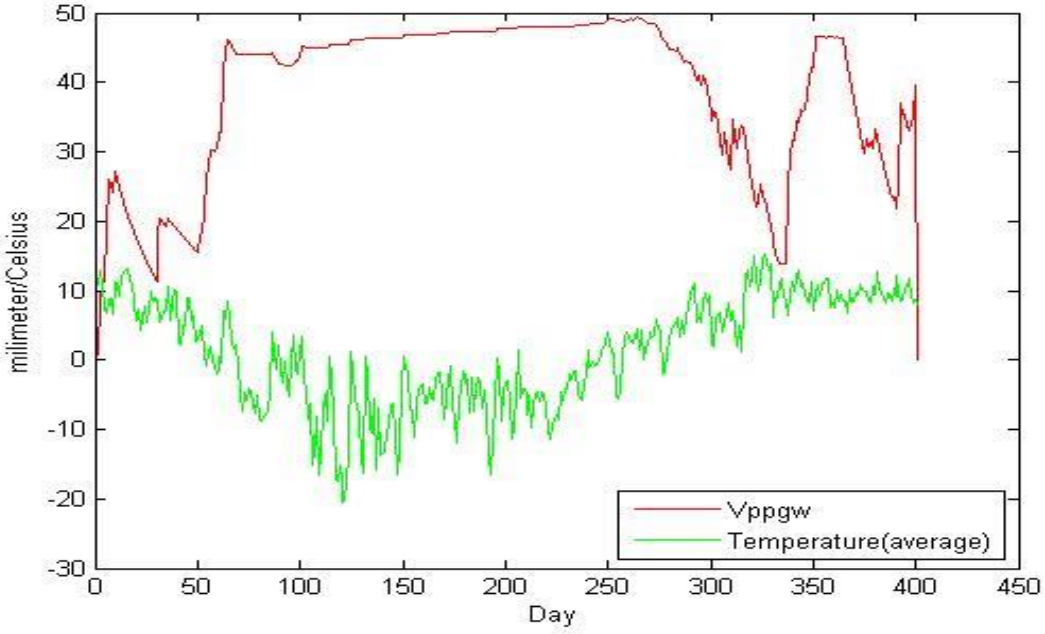


Figure 3-9: Ground zone simulation

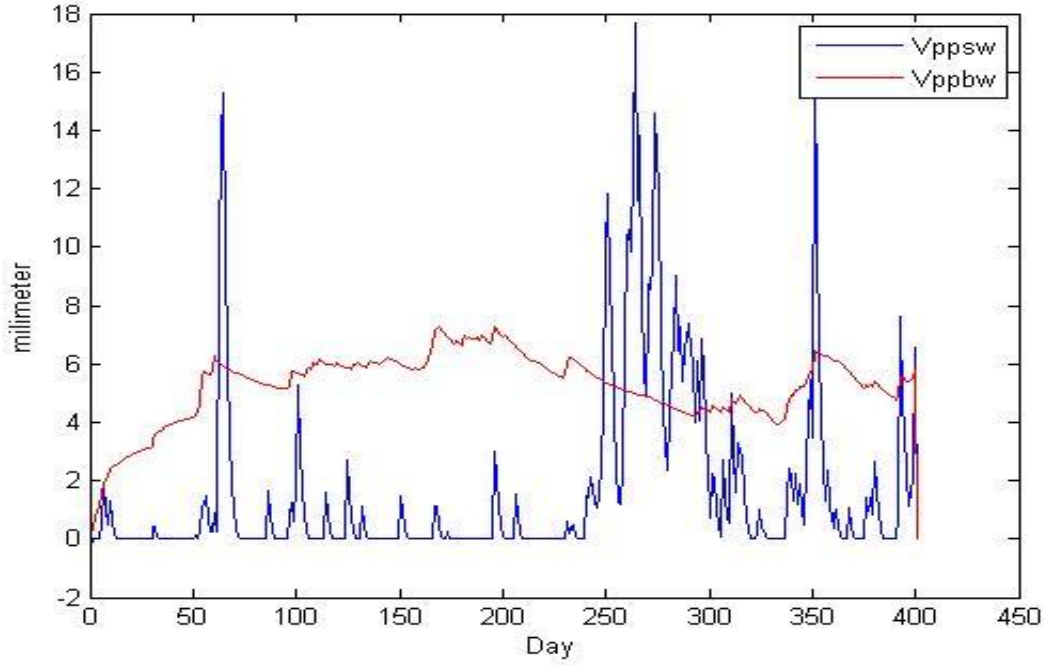


Figure 3-10: Simulation of soil zone and base zone

As indicated in Figure 3-10, and with more detail investigation it could be found that soil water ($V''_{s.w}$) is highly affected by ground zone runoff ($V''_{g.2.s}$), that is caused by the ground water ($V''_{g.w}$) change. In other words, both $V''_{s.w}$ and $V''_{g.w}$ have the same break points and peaks, which seems logical, and corresponds to the model concepts and equations.

In summery, from first of October to end of March and during August and September (day 0 to day 60), because of cold weather and low snow zone runoff respectively, we expect to have a constant value (around zero) for soil water ($V''_{s.w}$). On the other hand during spring season (day 240 to day 300) because of high $V''_{g.2.s}$ and warm weather, more $V''_{s.w}$ expected.

As indicated in Figure 3-10, the water content of base zone is somewhat stable after September, but it has a little drop during the spring season due to lower precipitation with compared to Summer time and more evapotranspiration compared winter and autumn.

It can be concluded from Figure 3-11 that the simulated runoff is similar to the real runoff from the Eggedal catchment.

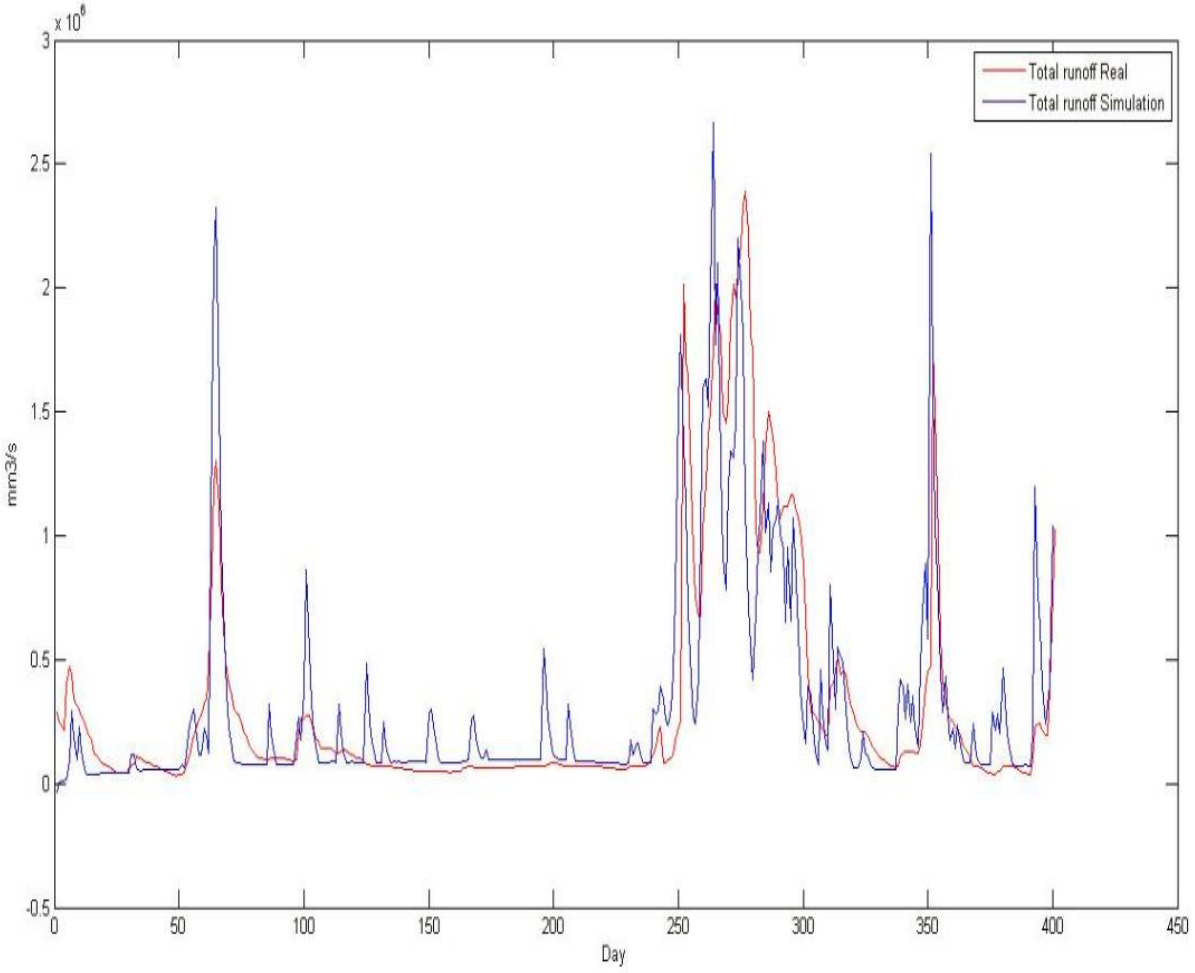


Figure 3-11: Comparing the real runoff with simulated runoff in the catchment area

The simulation was carried out with guessed parameters as listed in Table 3-2. It can be seen that most of the main trends in the real data are predicted by the simulation.

In part of runoff simulation (day 100 to day 240), some undesirable fluctuations were observed. By more careful investigation of temperature as an input of system, it is found that the fast sign change of temperature (from negative to positive and then to positive) causes snow melting and snow zone runoff increment is part of the most probable reason. As a solution to this problem it

is recommended to increase the number of measured temperature and precipitation in order to have a more smooth data. The other difference between simulated runoff and real runoff are due to inaccurate values of some parameters which have been used in the model. In order to achieve a more accurate result, the model parameters need to be calibrated further.

4 Model calibration

As discussed previously, it may be reasonable to conclude that simulated model inaccuracy is caused by parameter values which were obtained from HBV model in other catchment areas. Consequently, parameters tuning is necessary for Eggedal catchment model.

For nonlinear systems, parameters could be estimated by applying different calibration methods like, least squares, maximum likelihood and PEM[2] (where any different kinds of Kalman filtering are included).

4.1 Available measurements

The accuracy of the runoff prediction depends on the accuracy of data obtained by Skagerak. Currently Skagerak gathers measurements on precipitation (measured based on accumulated rain or snow over 24 hours), temperature (given as average daily value), and runoff density (field variable measured in $\frac{m^3}{\sec}$).

4.2 Structural Observability

Necessary condition for implementing any Kalman filtering is observability check to make sure that the system is observable. In our case since the precise linear models cannot describe the catchment (because of lack of knowledge of the plan from nonlinear or time-varying properties) some system properties such as Observability can be investigated without knowing precise system parameters, and can be hold for a large variety of parameter values. Therefore, if system is Structurally Observable then system becomes Observable (Lunze, 1991).

Structural observability is checked for nonlinear system by doing following steps:

1) Model linearization to reach $\overbrace{\begin{cases}\dot{x} = Ax + Bu \\ y = Cx + Du\end{cases}}^{linear}$ from $\overbrace{\begin{cases}\dot{x} = f(x,u) \\ y = g(x,u)\end{cases}}^{nonlinear}$
2) Construct Structure matrices (S_a, S_b, S_c, S_d) based on A, B, C and D.

[2] Prediction error method

$$S_a = \begin{pmatrix} \frac{\partial \dot{x}_1}{\partial x_1} & \cdots & \frac{\partial \dot{x}_1}{\partial x_n} \\ \vdots & \cdots & \vdots \\ \frac{\partial \dot{x}_n}{\partial x_1} & \cdots & \frac{\partial \dot{x}_n}{\partial x_n} \end{pmatrix} \quad S_b = \begin{pmatrix} \frac{\partial \dot{x}_1}{\partial u_1} & \cdots & \frac{\partial \dot{x}_1}{\partial u_m} \\ \vdots & \cdots & \vdots \\ \frac{\partial \dot{x}_n}{\partial u_1} & \cdots & \frac{\partial \dot{x}_n}{\partial u_m} \end{pmatrix} \quad S_c = \begin{pmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \vdots & \cdots & \vdots \\ \frac{\partial y_n}{\partial x_1} & \cdots & \frac{\partial y_n}{\partial x_n} \end{pmatrix} \quad S_d = \begin{pmatrix} \frac{\partial y_1}{\partial u_1} & \cdots & \frac{\partial y_1}{\partial u_m} \\ \vdots & \cdots & \vdots \\ \frac{\partial y_n}{\partial u_1} & \cdots & \frac{\partial y_n}{\partial u_m} \end{pmatrix}$$

Where, n is the number of states and m is the number of outputs.

For each element of structure matrices of nonzero value, a path connect the correlated elements to each other.

3) Check system to satisfy both following conditions:
   - Output-connectable (at least 1 path have to existed between state and output)
   - Rank of $\begin{pmatrix} S_a \\ S_c \end{pmatrix}$ should be equal to n (state number)

Model observability should be checked based on states of the system and runoff as the only measurable output. By considering this fact that model is divided into five cascade zones, the effects of all states are transfered to last zone's output by means of each zone's runoff. Thus, by investigating the structural observability of each layer separately, the whole system observability could be checked. (Shafiee, et al., 2012)

According to aforementioned method, one may conclude that none of the snow zone's states ($V''_{s.d}, V''_{s.s}$) are observable, while the other states of the system ($V''_{g.w}, V''_{s.w}, V''_{b.w}$) are structurally observable. Fortunately, since our system is detectable (unobservable modes are stable), by considering this fact that after the summer season $V''_{s.d}$ and $V''_{s.s}$ are zero, we can use the snow zone model to estimate these state values in an open-loop manner. (Magne Fjeld, 1980)

4.3 Identifiability analysis

In hydrological systems, setting up the experiments to measure all state variables might not possible. In other words, when only some of the state variables are measurable, it is possible that some parameter's effects cannot influence the measured state variables, so such parameters are unidentifiable.

To be identifiable, the parameter subset has to meet two conditions. First, the model output has to be enough sensitive to each parameter's change (sensitivity). Second, changes in the model output that is caused by single parameter change are not compensated by the other parameter changes (collinearity) (Roland Brun, 2002).

Sensitivity analysis could be helpful to find out which parameters are the most important ones and how model runoff can be affected by each parameter change.

In general, the sensitivity function for each parameter could be obtained by calculating the model output twice with the prior parameter set, and after changing the parameter by specific arbitrary increment as following:

$$\frac{\partial Tr}{\partial p_i} \approx \frac{Tr(p_i + \delta p_i) - Tr(p_i)}{\delta p_i} \tag{4-1}$$

Where,

Tr - Total model runoff

p_i - Parameter

δp_i - Arbitrary parameter increment

As listed in Table 3-2, there are 13 parameters in our model, but for conducting the sensitivity analysis just 9 of them were selected (especially those parameters which values are almost not known, but estimated ranges exist from studies of other catchment areas).

In the begging step, the sensitivity indexes were calculated based on equation (4-1) by considering 5% of the nominal value increment for each parameter.

As illustrated in Figure 4-1, in specific periods of time the model output are more sensitive to specific parameters which means that such parameters play a crucial role in comparison to the other parameters. Consequently, it is reasonable to assume that both identifiability and sensitivity are time-variant characteristics. In general, dynamic nature of sensitivity should be accounted to find out the specific periods where parameters have higher identifiability and crucial role in representing model runoff (Nibret A. Abebe, 2010).

Monthly or seasonally implementation of sensitivity analysis might be helpful for finding the most dominant parameters in a specific time period that will be investigated and studied in next section.

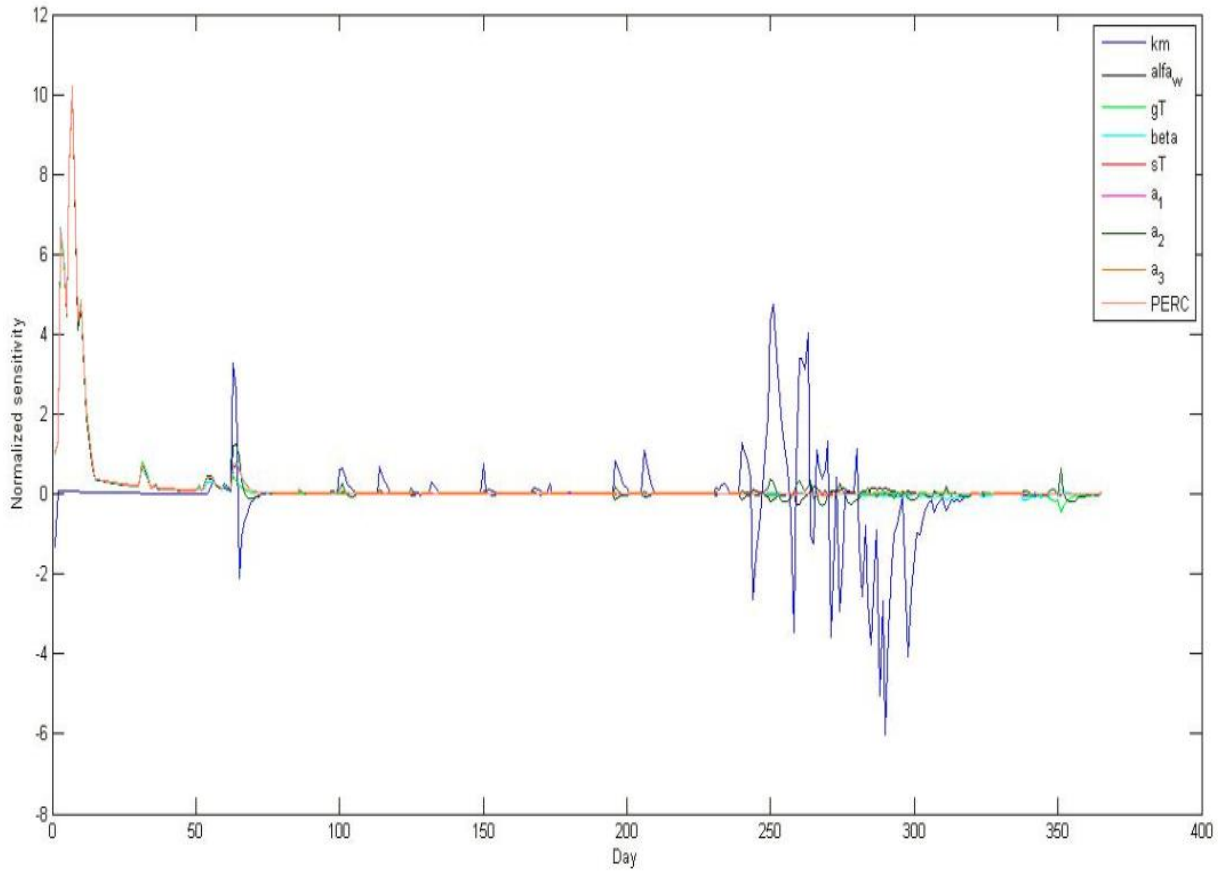


Figure 4-1: Normalized sensitivity index plot for 9 identifiable parameters of August in a year (01/08/1973 - 31/07/1974)

4.3.1 Sensitivity analysis by PCA[3]

This method has been developed to discard nonsignificant parameters by ranking all of them first, and then determining them which one is identifiable or unidentifiable according to their ranks. This method was implemented in following manner:

1. Construct normalized sensitivity matrix:

$$S^{i}_{k,j} = \frac{p_j}{Tr_i} \cdot \frac{\partial Tr_i(t_k, p)}{\partial p_j} \tag{4-2}$$

Where

i - denotes the index of system outputs

j - denotes the index of parameters

k - denotes the index of measurement time points

[3] Principle Component Analysis

2. Run the PCA for each month to find out nonsignificant parameters by ranking all parameters according to their component coefficients.

Table 4-1 shows parameters importance ranking with respect to time-variant characteristic (monthly analysis). Second and third columns indicate the most important parameters (from left to right) for each principal component.

Table 4-1: Parameter importance rankings

Month	**Parameters importance ranking during 01/08/1973 - 31/07/1974 (1st year)**	**Parameters importance ranking during 01/08/1973 - 31/07/1974 (2nd year)**
August	PC1 (89%): $\alpha_\omega, g_T, \beta, s_T, \alpha_1, \alpha_2, \alpha_3, PERC, k_m$	PC1 (89%): $\alpha_\omega, g_T, s_T, \alpha_1, \alpha_3, PERC, \beta, \alpha_2, k_m$
September	PC1 (89%): $s_T, \alpha_1, \alpha_3, \alpha_\omega, PERC, \alpha_2, g_T, \beta, k_m$	PC1 (82%): $PERC, \alpha_\omega, \alpha_3, \beta, \alpha_1, g_T, \alpha_2, s_T, k_m$
October	PC1 (89%): $PERC, s_T, \alpha_1, \alpha_3, \beta, \alpha_\omega, g_T, \alpha_2, k_m$	PC1 (66%): $s_T, \alpha_3, \alpha_1, \beta, g_T, \alpha_\omega, PERC, \alpha_2, k_m$ PC2 (14%): $k_m, \alpha_2, g_T, PERC, \beta, \alpha_1, \alpha_\omega, s_T, \alpha_3$
November	PC1 (78%): $s_T, \alpha_1, \alpha_3, \alpha_\omega, \beta, g_T, \alpha_2, k_m, PERC$	PC1 (50%): $\alpha_\omega, k_m, \beta, g_T, PERC, \alpha_2, s_T, \alpha_1, \alpha_3$ PC2 (32%): $\alpha_1, s_T, \alpha_3, g_T, PERC, \beta, \alpha_2, \alpha_\omega, k_m$
December	PC1 (71%): $\beta, \alpha_1, s_T, \alpha_3, g_T, \alpha_\omega, \alpha_2, k_m, PERC$	PC1 (48%): $g_T, \beta, \alpha_\omega, k_m, PERC, \alpha_2, \alpha_3, \alpha_1, s_T$ PC2 (33%): $s_T, \alpha_1, \alpha_3, \alpha_2, PERC, \beta, \alpha_\omega, k_m\ g_T$
January	PC1 (70%): $g_T, \beta, s_T, \alpha_1, \alpha_3, k_m, PERC, \alpha_2, \alpha_\omega$	PC1 (59%): $g_T, s_T, \alpha_1, \alpha_3, \beta, k_m, \alpha_2, \alpha_\omega, PERC$ PC2 (28%): $\alpha_\omega, PERC, k_m, \beta, \alpha_2, \alpha_3, \alpha_1, s_T, g_T$
February	PC1 (75%): $g_T, \alpha_\omega, s_T, \alpha_1, \alpha_2, \alpha_3, k_m, \beta, PERC$	PC1 (99%): $k_m, \alpha_\omega, g_T, \beta, s_T, \alpha_1, \alpha_2, PERC, \alpha_3$
March	PC1 (85%): $\alpha_1, s_T, \beta, g_T, k_m, \alpha_2, \alpha_3, PERC, \alpha_\omega$	PC1 (89%): $k_m, \alpha_\omega, g_T, \beta, s_T, \alpha_1, \alpha_2, PERC, \alpha_3$
April	PC1 (55%): $\beta, s_T, \alpha_3, \alpha_\omega, \alpha_1, \alpha_2, k_m, PERC, g_T$ PC2 (28%): $PERC, g_T, k_m, \alpha_3, s_T, \alpha_1, \alpha_\omega, \beta, \alpha_2$	PC1 (56%): $\alpha_\omega, g_T, \beta, s_T, PERC, \alpha_1, \alpha_\omega, \alpha_2, k_m, \alpha_3$ PC2 (19%): $\alpha_3, \alpha_1, \alpha_2, \alpha_\omega, s_T, \beta, k_m, PERC, g_T$
May	PC1 (31%): $PERC, \beta, k_m, \alpha_\omega, \alpha_3, \alpha_2, g_T, \alpha_1, s_T$ PC2 (28%): $g_T, s_T, \alpha_1, PERC, \alpha_2, \beta, \alpha_\omega, k_m, \alpha_3$ PC3 (16%): $\alpha_3, \alpha_1, \beta, \alpha_2, \alpha_\omega, s_T, k_m, g_T, PERC$	PC1 (42%): $\alpha_3, \alpha_1, k_m, \alpha_2, \beta, \alpha_\omega, s_T, PERC, g_T$ PC2 (24%): $PERC, s_T, \beta, g_T, \alpha_2, \alpha_1, \alpha_3, \alpha_\omega, k_m$ PC3 (18%): $\alpha_\omega, g_T, \beta, s_T, \alpha_2, PERC, k_m, \alpha_3, \alpha_1$
June	PC1 (45%): $\beta, k_m, g_T, PERC, \alpha_2, \alpha_3, s_T, \alpha_\omega, \alpha_1$ PC2 (32%): $\alpha_1, \alpha_\omega, s_T, \alpha_3, \alpha_2, PERC, g_T, \beta, k_m$	PC1 (68%): $s_T, \alpha_1, \alpha_3, g_T, \beta, \alpha_\omega, PERC, k_m, \alpha_2$ PC2 (23%): $\alpha_2, k_m, PERC, \alpha_\omega, \beta, g_T, \alpha_3, \alpha_1, s_T$
July	PC1 (55%): $s_T, k_m, \alpha_\omega, \alpha_1, \alpha_2, g_T, \alpha_3, PERC, \beta$ PC2 (27%): $\beta, \alpha_3, g_T, PERC, \alpha_1, k_m, \alpha_\omega, s_T, \alpha_2$	PC1 (79%): $k_m, \alpha_\omega, s_T, \alpha_1, \beta, g_T, PERC, \alpha_3, \alpha_2$

Note: The percentage value in front of each principal component explains the corresponding variances of the original variance along each principle component.

4.3.2 Collinearity analysis based on PCA

Interpretation of loading plot indicates the importance and also the correlation between model variables (parameters). For instance in a 2-vector loading plot, those variables which lie far away from the origin and close to PC[4] axis are the significant variables for that PC; while those variables which lie near the origin are the less important ones. It can be also identified if the variables lie on the same side of the origin, they have positive correlation, but if they lie on opposite sides of the origin (more or less along a straight line through the origin), they are negatively correlated. It can be inferred that those variables which are located at 90 degrees to each other through the origin of loading plot are independent. (Esbensen, 2001)

According to explained method, those parameters which are satisfying three following conditions should be selected as desirable identifiable parameters:

1. Not negatively correlated with other parameters
2. Lie far from the origin
3. Located close to the PC axis

Detailed information and related plots on running PCA to find the best parameter set can be found in Appendix 2.

4.3.3 Identifiability analysis based on PCA

By considering both parameters importance ranking (Table 4-1) and parameters collinearity, identifiable parameters could be found as listed in Table 4-2.

[4] Principle component

Table 4-2: Identifiable parameters

Month	**Identifiable parameters (1st year)**	**Identifiable parameters (2nd year)**
August	$\alpha_\omega, g_T, \beta, s_T, \alpha_1, \alpha_2, \alpha_3, PERC$	$\alpha_\omega, g_T, s_T, \alpha_1, \alpha_2, PERC, \beta, \alpha_2$
September	$s_T, \alpha_1, \alpha_3, \alpha_\omega, PERC, \alpha_2, g_T, \beta$	$PERC, \alpha_\omega, \alpha_3, \beta, \alpha_1, g_T, \alpha_2, s_T$
October	$PERC, s_T, \alpha_1, \alpha_3, \beta, \alpha_\omega, g_T, \alpha_2$	$s_T, \alpha_3, \alpha_1, \beta, \alpha_\omega, g_T, k_m$
November	$s_T, \alpha_1, \alpha_3, \alpha_\omega, \beta$	$\beta, PERC, s_T, \alpha_3$
December	α_1, s_T, α_3	g_T, k_m, α_1, s_T
January	$\beta, s_T, \alpha_1, k_m, PERC$	$g_T, s_T, \alpha_1, \alpha_3, PERC$
February	$\alpha_\omega, s_T, \alpha_3$	$k_m, \alpha_\omega, g_T, \beta, s_T, \alpha_1, \alpha_2, PERC$
March	α_1, s_T, α_2	$k_m, \alpha_\omega, g_T, \beta, s_T, \alpha_1, \alpha_2, PERC$
April	$\beta, \alpha_2, g_T, PERC$	$g_T, s_T, \alpha_1, \alpha_3$
May	$k_m, g_T, \alpha_3, s_T, \alpha_1$	$\alpha_3, \alpha_1, k_m, \alpha_\omega, PERC, s_T$
June	$k_m, \alpha_1, \alpha_\omega$	$s_T, \alpha_1, \alpha_3, \alpha_2$
July	$s_T, k_m, \alpha_\omega, \alpha_1, \beta$	$k_m, \alpha_\omega, s_T, \alpha_1, \beta, g_T, \alpha_3$

It is evident that identifiable parameters for the same month in two consecutive years are different. The seasonally identifiability analysis is also lead to the same result. This result is in line with Abebe (2010) study which has confirmed that optimum parameters are not constant in time and range of optimum parameter values varies with time. Consequently, the dynamic identifiability analysis should be conducted based on forecasted temperature and precipitation to find higher identifiable parameters for future days.

4.4 Parameters and states estimation

After finding identifiable parameters for each month such as they were listed in Table 4-2 and considering this assumption that the initial states of the system are equal to zeros on 1st of August, the corrected parameters could be estimated in August at 1973 (1st year) by knowing at least 8 future measurements (temperature, precipitation and real runoff). This number of measurements that is needed to estimate the initial parameters is called estimation interval or estimation horizon which is equal to 8 since there are 8 unknown parameters.

In our case and based on this fact that obtained identifiable parameters in August, September and October are same, the optimum values of parameters could be estimated. Subsequently, by knowing optimal parameters and state estimation horizon which is equal to 23 for August 1973 (since there are 23 unknown states), the initial states of the system could be estimated by implementing optimization method.

In the other words, at each estimation interval an estimator algorithm attempts to decrease the difference between simulated runoff and real runoff by manipulating previous variables (parameters or states) adjustment. The estimated variables are then sent into the model, and the entire calculation is repeated at subsequent estimation intervals. For each iteration the estimation horizon is moving forward in time and the estimator again estimate the variables as are represented in Figure 4-2 for one state as an example.

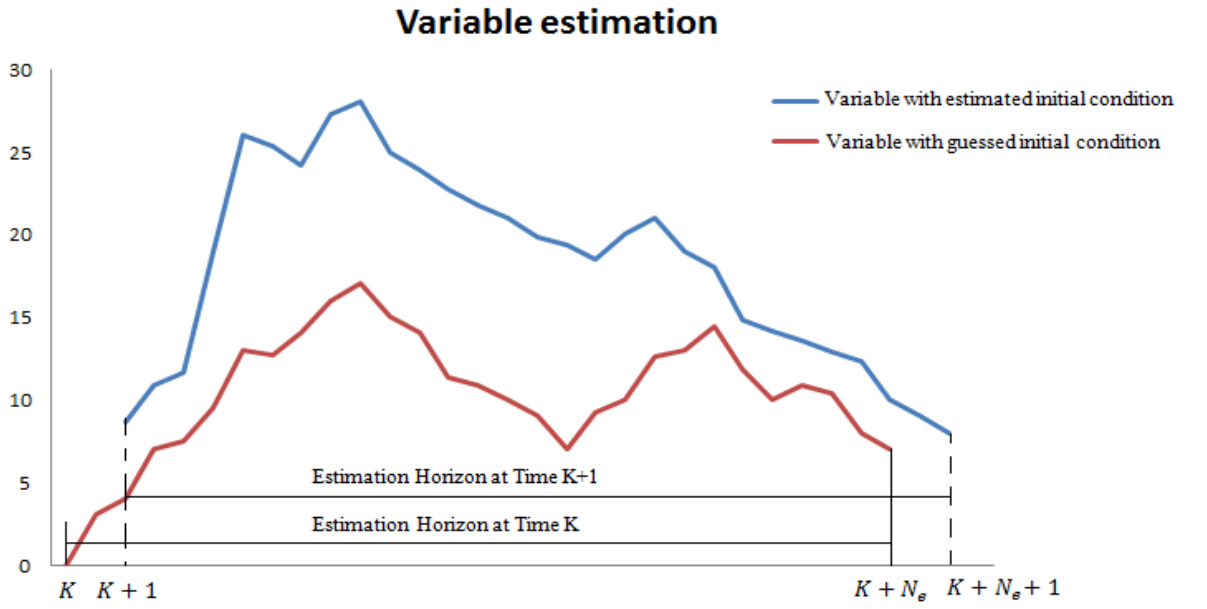


Figure 4-2: Variable estimation with considering estimation horizon (N_e)

The basic structure of the estimator is shown in Figure 4-3.

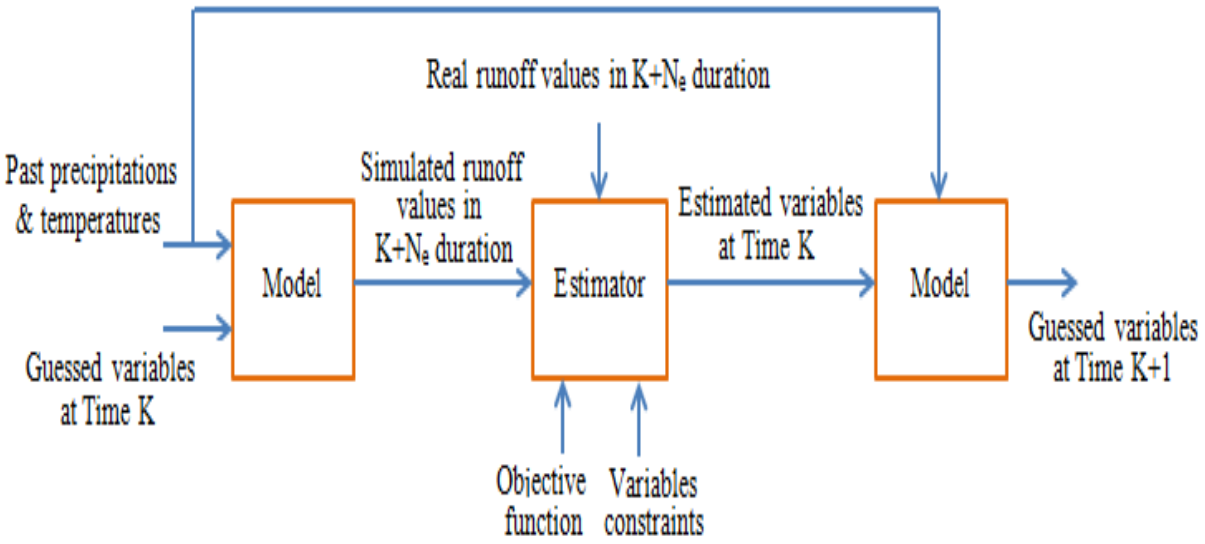


Figure 4-3: Basic structure of estimator

The main point of this estimation problem is to compute a new variable vector, to be fed to the system, and at the same time take variable constraints into consideration (e.g., states have not to be negative).

Among many of the methods used in the optimization Toolbox of MATLAB software I used the constrained multivariables function to find the minimum of variable vector by starting at an initial estimate. It attempts to return a vector of optimal variables with respect to process constraints. Because of having if condition situations in our model the optimization method used the Nelder-Mead algorithm which is a well-defined numerical method when derivatives are not known and it makes problem. Nelder-Mead algorithm uses Lagarias search method instead of using numerical or analytic gradient. (Anon., n.d.)

For verifying the estimator performance, August of 1973 is chosen as a sample duration. By implementing nonlinear least squares optimization method, optimum estimated parameters and states of the system were found. Figure 4-4 shows the comparison of real runoff, simulated runoff with and without estimating states and parameters. It can be seen in Figure 4-4 that the estimator gives acceptable parameters and states that provide a good fit between simulated runoff and real one.

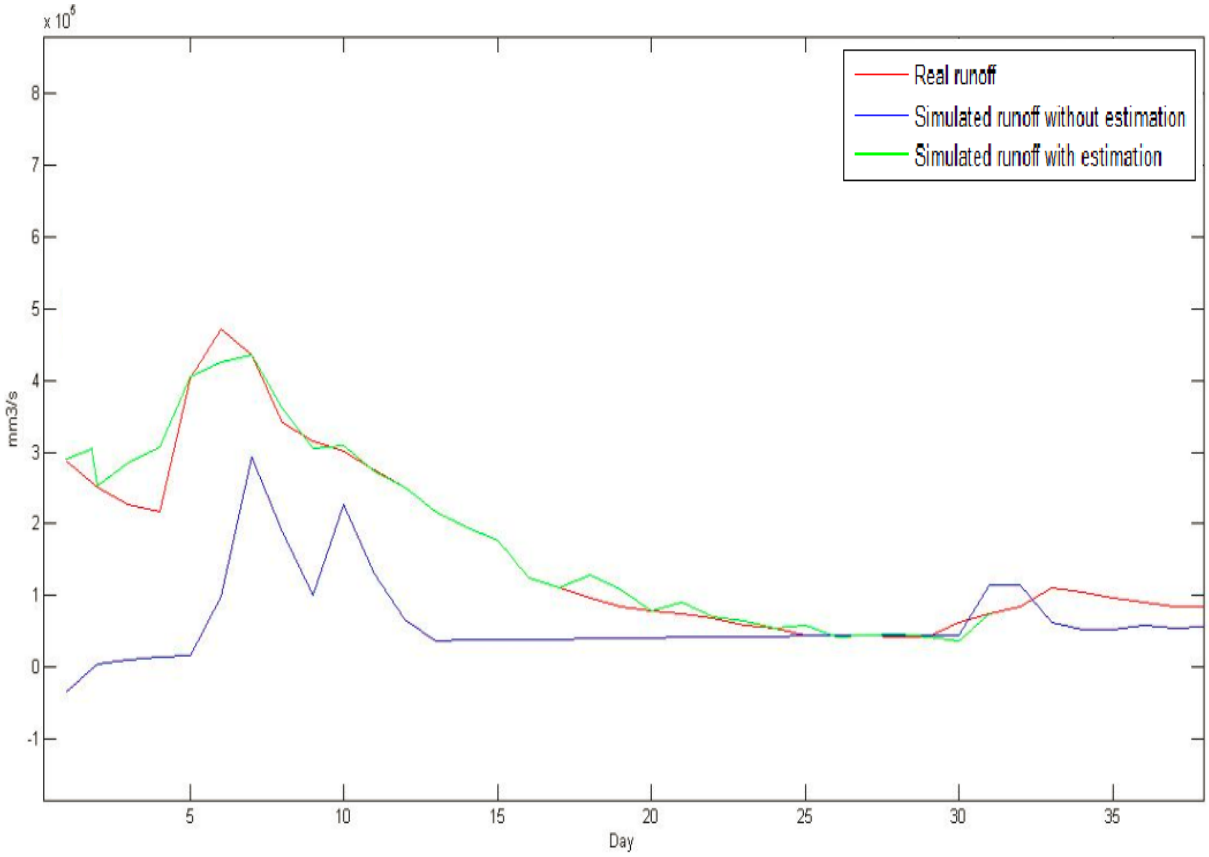


Figure 4-4: Comparison between real runoff and simulated runoff before and after state estimation

Obtained results prove that the automatic model updating and model calibration could be done accurately, based on automatic states estimation and parameters adjusting respectively. Table 4-3 shows calibrated and original parameter comparison in August of 1973.

Table 4-3: Calibrated and original parameters comparison

Parameter symbols	**Unit**	**Original Parameters**	**Calibrated Parameters**
α_w	-	0.08	0.1454
g_T	mm	50	87
β	-	2	3.7
s_T	mm	20	23
a_1	day^{-1}	0.547	0.399
a_2	day^{-1}	0.489	0.240
a_3	day^{-1}	0.0462	0.0004
PERC	$\frac{mm}{day}$	0.25	0.125

Since, hydrological model calibration is done based on automatic parameters and states adjusting according to search scheme and numerical measures of the goodness-of-fit, it requires more availability of observed data and takes more time based on the objective function.

For reducing calculation time in estimation procedure and reaching to the shorter estimation horizon, new measurements could be added.

Snow density measurements could be a possible solution. Snow density measurements will help to define the state of water content in snow, and snow quality. By adding each snow density measurement, two states of the system were calculated directly and consequently estimation horizon decreases two steps.

5 Required Information and automating computation of needed information

In order to get more realistic parameter estimation, it is neseccary to know the reasonable estimation intervals where each parameter could be adjusted in that range. Among 13 parameters which were listed in Table 3-2, some of them such as lake surface fraction (α_L), total area (A) and snow surface fraction (α) could be found by studying catchment area with satellite pictures and maps. Second group of parameters are those that strongly depend on the environmental conditions (average variations in temperature and precipitation) like melting factor (k_m), saturation coefficient (α_w), and threshold temperature (T_T). In our model these kind of parameters should be defined based on physical concepts. The last group of parameters could be found based on geological studies such like ground zone shape coefficient (β), ground saturation threshold (g_T), soil saturation threshold (s_T), percolation rate (*PERC*) and discharge frequencies (a_1, a_2, a_3). Consequently, performing pre study on parameter possibilities could be helpful to find more realistic parameters before implementing any calibration.

The accuracy of the runoff prediction depends on the accuracy of data obtained by Skagerak. Currently Skagerak gathers measurements of precipitation, temperature, and runoff from the catchments by specific measurement devices and send them automatically to the Skagerak office each day.

The third kind of needed information is snow density that was discussed previously. The current practice of Skagerak Kraft is to manually measure snow depth and snow density at certain points in the catchment zone. The snow density measurement is done by inserting a hollow cylinder of known volume and mass into the snow and thereby extracting a core sample. The cylinder is then weighed and the mass of the cylinder subtracted. The density of the core sample is then calculated by

$$\frac{(m_{Total} - m_{cylinder})}{V_{cylinder}} = \rho_{snow} \tag{5-1}$$

The main benefit of automatic measurements and recording equipment is the ability of these methods and equipment to be used in real-time systems and lead to provide more observations for hydrological purposes.

Problems of manual hydrological measurements could be solved by applying automatic computation of needed information. For instance in manual level measuring, gauging station are used that could be replaced by automatic level recorders, where the head and pressure of liquid are measured by piezometers and pressure gauges. The flow velocity of water commonly measured by current meters, tubes, vanes, thermal measurement devices and electronic and

mechanical instruments in manual forms; while in automatic method it is measured under laboratory conditions with kinematography method which is also considering turbulence flow.

Automatic measuring instruments and methods have their own drawbacks. They must be exercised in usage, because automatic equipments are highly variable by comparison with most manual ones. Moreover automatic devices need power to operate sensors, so battery or power lines should be provided.

Among any different data storage format, XML as a standard file format could be a good choice by providing following advantages:

1. Allow hydrologic engineers to leverage many of the existing XML technologies
2. Increasing productivity and accuracy
3. Hydrologic model would be human intelligible, beside machine readable
4. Providing flexibility to incorporate future industry advancements

Figure 5-1, shows the structure of automatic computation of needed information that should be repeated each day.

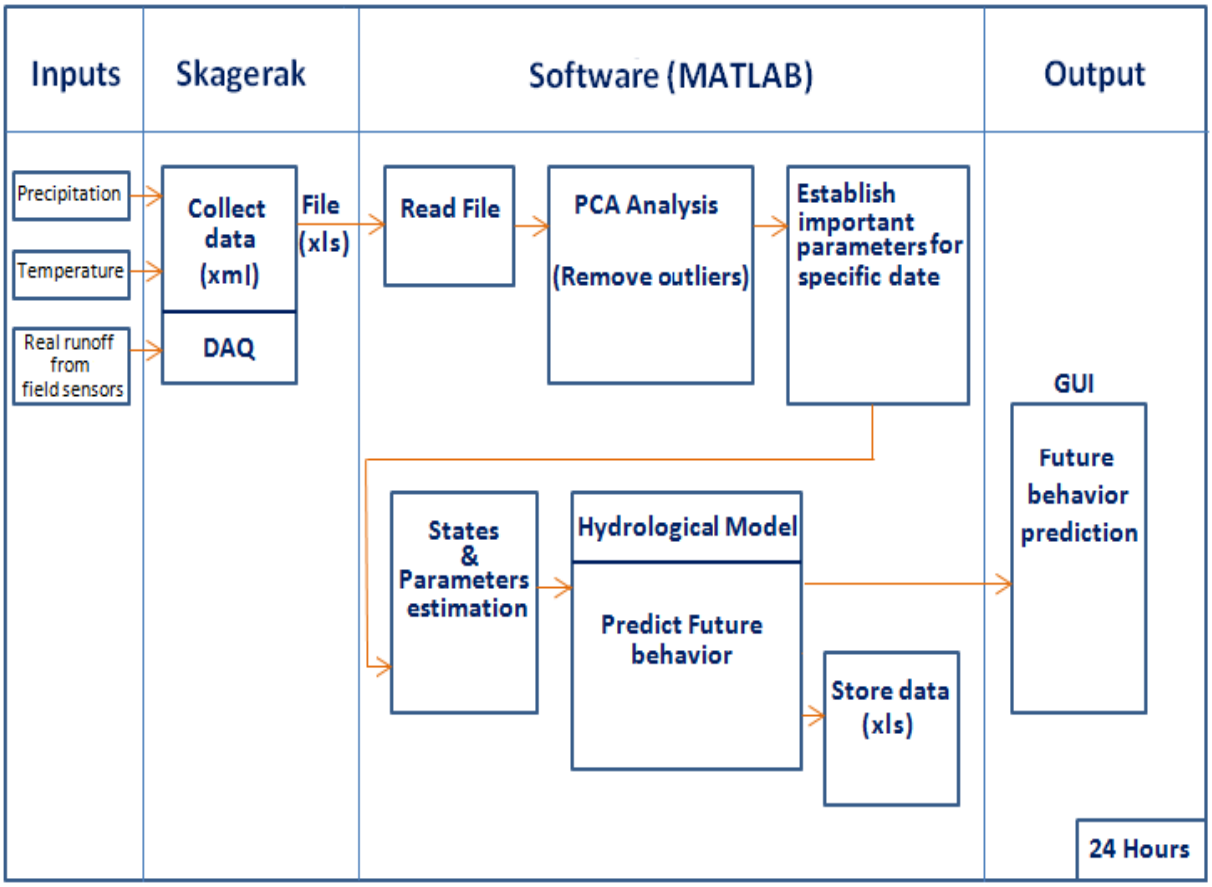


Figure 5-1: Structure of automatic computation of needed information

- **Inputs:** it consists of field measurement sensors to measure real runoff value, precipitation data, and temperature.
- **DAQ:** It consists of Data acquisition system for gathering data input section and prepare suitable data format.
- **Collect data:** Data center to provide a database in proper format (XML).
- **Read File:** Read data in xls format
- **PCA Analysis:** Run principle component analysis to remove outliers.
- **Establish important parameters:** Finding most important identifiable parameters in specific date.
- **States & Parameters estimation:** Nonlinear least squares to find states and parameters.
- **Predict future behavior:** Feed parameters and states to hydrological model.
- **Store data:** Logging data in xls format.
- **Output:** Displaying future behavior in proper method (GUI).

6 Conclusion

In order to improve the production planning in a hydropower system, based on forecast runoff/inflow water to a reservoir, a new runoff /inflow hydrological model was introduced in this master thesis. This model was obtained by reformulating the HBV-3 model based on mass balance and physical laws by using model functional description for one altitude. The model was developed for all altitudes by considering hypsography concept. More accurate results could be reached by increasing number of elevations to get smoother function between snow surface fraction and elevation. More computations and data availability in estimation procedure is the main drawback of aforementioned suggestion.

MATLAB programming language was used to solve state equations and model implementation. Model verification was done based on created data, on Fjeld & Aam parameters. Both model verification and model validation improved the accuracy of prediction, that was discussed in detail through sections 3.2.1 and 3.2.2.

Model inaccuracy has been compensated by conducting Ensemble Kalman Filtering and Non-linear least squares as two different methods of model calibration. Observability of the system as a necessary condition of implementing of any Kalman filtering was discussed and it was found that states of the snow zone were not structurally observable. This result led to select nonlinear least squares as an alternative estimation method.

To find the most important parameters for estimation procedure, sensitivity analysis and identifiability analysis was done based on principal component analysis. Identifiability analysis indicated that identifiable parameters were not constant in time and the range of optimum parameter varies with time. For simplicity and getting more reasonable result, parameters were assumed to be constant in a month. The parameter's estimation horizon was defined according to the number of unknown identifiable parameters in each month. On the other side, the state's estimation horizon was also defined based on number of unkown states of the system that was constant for all the time.

Parameters and state estimation were carried out in August of 1973 (as a sample duration). Afterwards, estimated parameters and states fed to the model for testing procedure. Comparison between simulated runoff with and without estimated variables (parameters and states) verified the estimator performance.

During this master thesis and based on what was done in modeling part, one conference paper submitted with "Prediction of daily runoff from hydrological catchment area" title in 54rd SIMS conference in Bergen, Norway.

Future work:

- To decrease the estimation horizon and computation time, measurements need to be added.
- Combination of Augmented Kalman filtering (for observable part) and nonlinear least squares (for non observable parts) could be studied as an alternative method for states and parameters estimation.
- Model predictive control can be implemented using the model introduced in this report.

7 References

Anon., 2009. *ClimaWater.* [Online]
Available at: Climawater Hydrological Modelling-Literature Review1.pdf
[Accessed 2012].

Anon., 2012. *Nord Pool Spot.* [Online]
Available at: http://www.nordpoolspot.com/Global/Download%20Center/Annual-report/Nord-Pool-Spot_Europe's-leading-power-markets_August-2012.pdf

Anon., 2012. *Nordic Energy Regulators.* [Online]
Available at: https://www.nordicenergyregulators.org/The-Development-on-the-Nordic-Electricity-Market/

Anon., 2012. *Skagerak Kraft AS.* [Online]
Available at:
http://www.skagerakenergi.no/eway/default.aspx?pid=300&trg=MainRight_9194&MainArea_8872=9194:0:&MainRight_9194=9216:0:10,3419

Anon., n.d. *Optimizing Nonlinear Functions.* [Online]
Available at: http://www.mathworks.se/help/matlab/math/optimizing-nonlinear-functions.html#bsgpq6p-10
[Accessed 28 May 2013].

Bergström, S., 1973. Development of a conceptual deterministic rainfall–runoff model. *Nordic Hydrol*, Volume 4, p. 147–170.

Chen, J., 2012. Assimilating multi-site measurements for semi-distributed hydrological model updating. *Elsevier*, Volume 282, p. 122–129.

Chen, J., Zhang, W., Gao, J. & Cao, K., 2012. Assimilating multi-site measurements for semi-distributed hydrological model updating. *Elsevier*, Volume 282, pp. 122-129.

Dong, X., 2009. Automatic calibration of a lumped Xinanjiang hydrological model by genetic algorithm. *IEEE*, pp. 211 - 217.

Emily Youcha, D. K., n.d. *Introduction to runoff modeling on the North Slope of Alaska using the Swedish HBV model.* Fairbanks, Alaska: University of Alaska Fairbanks, Water & Environmental Research Center.

Encyclopædia Britannica, I., 2008. *hydrologic cycle: processes.* [Online]
Available at: http://www.britannica.com/EBchecked/media/112177/In-the-hydrologic-cycle-water-is-transferred-between-the-land
[Accessed 5 March 2013].

Esbensen, K. H., 2001. *Multivariate data analysis - in practice.* 5th ed. Oslo: Camo.

Fjeld, M. & Aam, S., 1980. An Implementation of Estimation Techniques to a Hydrological Model for Prediction of Runoff to a Hydroelectric Power Station. *IEEE Transactions on Automatic Control,* AC-25(NO.2).

Göran Lindström, B. J. M. P. M. G. S. B., 1997. Development and test of the distributed HBV-96 hydrological model. *Journal of Hydrology,* Volume 201, pp. 272-288.

Gosain, A. K., Mani, A. & Dwivedi, C., 2009. *Hydrological Modelling-Literature Review,* s.l.: A Indo-Norwegian Institutional Cooperation Program.

Hongyu Miao, X. X. A. P. H. W., 2011. On identifiability of nonlinear ODE Models and Applications in Viral Dynamics. *NIH.*

Jacquez JA, P. T., 1990. Parameter estimation: local identifiability of parameters. *PubMed.*

Jain, S. K., 1993. Calibration of conceptual models for rainfall-runoff simulation. *Hydrological Science Journal,* 38(5), pp. 431- 441.

Liu, J., 2010. Automatic calibration of hydrological model by shuffled complex evolution metropolis algorithm. *IEEE,* Volume 3, pp. 256 - 259 .

Lunze, J., 1991. Structural Controllability and Stractural Observability. In: *Feedback Control Of Large-Scale System.* London: Prentice-Hall international series in systems and control engineering , pp. 47-52.

Magne Fjeld, S. A., 1980. An Immlementation of Estimation Techniques to a Hydrological Model For Prediction of Runoff to a Hydroelectric Power Station. *IEEE Transaction on Automayic Control,* AC.-25(2), pp. 151-163.

Moradkhani, H., 2005. Uncertainty assessment of hydrologic model states and parameters: Sequential data assimilation using the particle filter. *Water Resources Research,* 41(5).

Moreda, F., 1999. *Conceptual rainfall - runoff models for different time steps with special consideration for semi-arid and arid catchments,* Brussels Faculty of Applied Sciences, Vrije Universiteit Brussel: Laboratory of Hydrology.

Nibret A. Abebe, F. L. O. N. R. P., 2010. Sensitivity and uncertainty analysis of the conceptual HBV rainfall–runoff model: Implications for parameter estimation. *Journal of Hydrology,* Volume 389, p. 301–310.

Nord Pool Spot, 2012. [Online]
Available at: http://www.nordpoolspot.com/Global/Download%20Center/Annual-report/Nord-Pool-Spot_Europe's-leading-power-markets_August-2012.pdf

Norwegian Water Resources and Energy Directorate, n.d. *NVE Atlas.* [Online]
Available at: http://atlas.nve.no/ge/Viewer.aspx?Site=NVEAtlas
[Accessed 2012].

Rakovec, O. et al., 2012. State updating of a distributed hydrological model with Ensemble Kalman Filtering: effects of updating frequency and observation network density on forecast accuracy. *Hydrology and Earth System Sciences Discussions,* pp. 3961-3999.

Raughunath, H. M., 2006. *Hydrology Principles, Analysis, Design.* Delhi: New Age International Ltd.

Rinde, T., 1999. *LANDPINE - A Hydrological to Simulate the influence of Land-Use Changes on Runoff,* Trondheim: SINTEF.

Riverso, C., 2011. *Calibration of rainfall-runoff models,* BOLOGNA: UNIVERSITÀ DI BOLOGNA.

Rochester, R. E., 2010. *Uncertainty in hydrological modelling: A case study in the tern catchment,* Shropshire, UK: UCL.

Rochester, R. E. L., 2010. *Uncertainty in Hydrological Modelling: A Case study in the Tern catchment, Shropshire, UK,* London: Department of Geography, University College London.

Roland Brun, M. K. H. S. W. G., 2002. Practical identifiability of ASM2d parameters—systematic selection and tuning of parameter subsets. *Water Research,* Volume 36, p. 4113–4127.

Seibert, J., 1999. *Conceptual runoff models-fiction or representation of reality? ,* Uppsala: Faculty of Science and Technology, Uppsala University.

Shafiee, S. et al., 2012. *Hydrological modeling for runoff/inflow forecasting to hydropower system,* Porsgrunn: s.n.

Shultz, M. J., 2007. *Comparison of distributed versus lumped hydrologic simulation models using stationary and moving storm events applied to small synthetic rectangular basins and an actual watershed basin,* Arlington, Texas: The University of Texas.

Skagerak Kraft AS, 2012. *Catchment Data Eggedal,* Porsgrunn: s.n.

Skagerak Kraft AS, 2013. *Fakta om Skagerak Energi.* [Online]
Available at:
http://www.skagerakenergi.no/eway/default.aspx?pid=300&trg=MainRight_9194&MainArea_8872=9194:0:&MainRight_9194=9216:0:10,3419

Xingnan, Z. & Lindström, G., 1997. Development of an automatic calibration scheme for the HBV Hydrological model. *Hydrological processes,* Volume 11, pp. 1671 - 1682.

8 Appendices

Appendix 1: Thesis task description

FMH606 Master's Thesis

Title: Automatic updating of hydrological models for runoff/inflow forecasting to hydropower system

TUC supervisor: Bernt Lie

External partner: Skagerak Kraft, dr. Beathe Furenes

Task description:

Based on a project work on a model for forecasting of runoff/inflow of water from catchments to hydropower reservoirs, the following is to be studied.

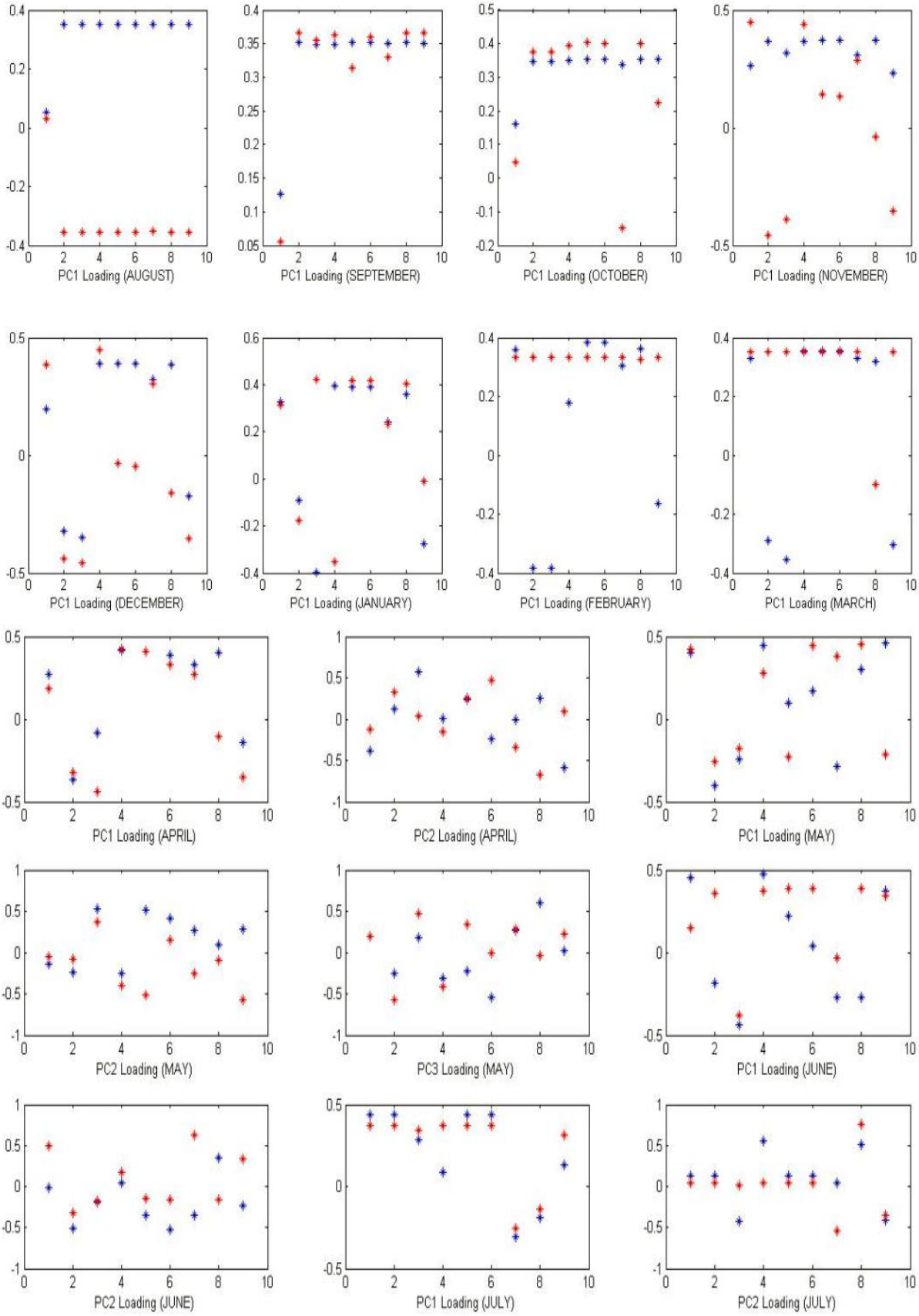
1. Give an overview of model requirements: what will the model be used for, and what must it give information about. (Based on 3rd semester project.)
2. Describe the HBV model in detail. Implement the model in a suitable programming language (MATLAB, Python, Modelica, or similar). Validate the model. (Based on 3rd semester project.)
3. Discuss available information (measurements), whether the model is observable, and whether the model is identifiable. Validate the model using available information from a real catchment region. (Based on 3rd semester project.)
4. Discuss methods of model calibration (model fitting, parameter estimation), and test out a selected method for the model/available data.
5. Discuss methods for updating the model with available measurements (state estimation), and consider flow predictions using weather forecast.
6. Give an overview of the kind of information Skagerak Kraft needs in a hydrological model, and how to compute this information.
7. Discuss automating the computation of the needed information (MATLAB, Python?), including storage formats for the data (gs2-format, or other).
8. Document the work in a report.

Task background:

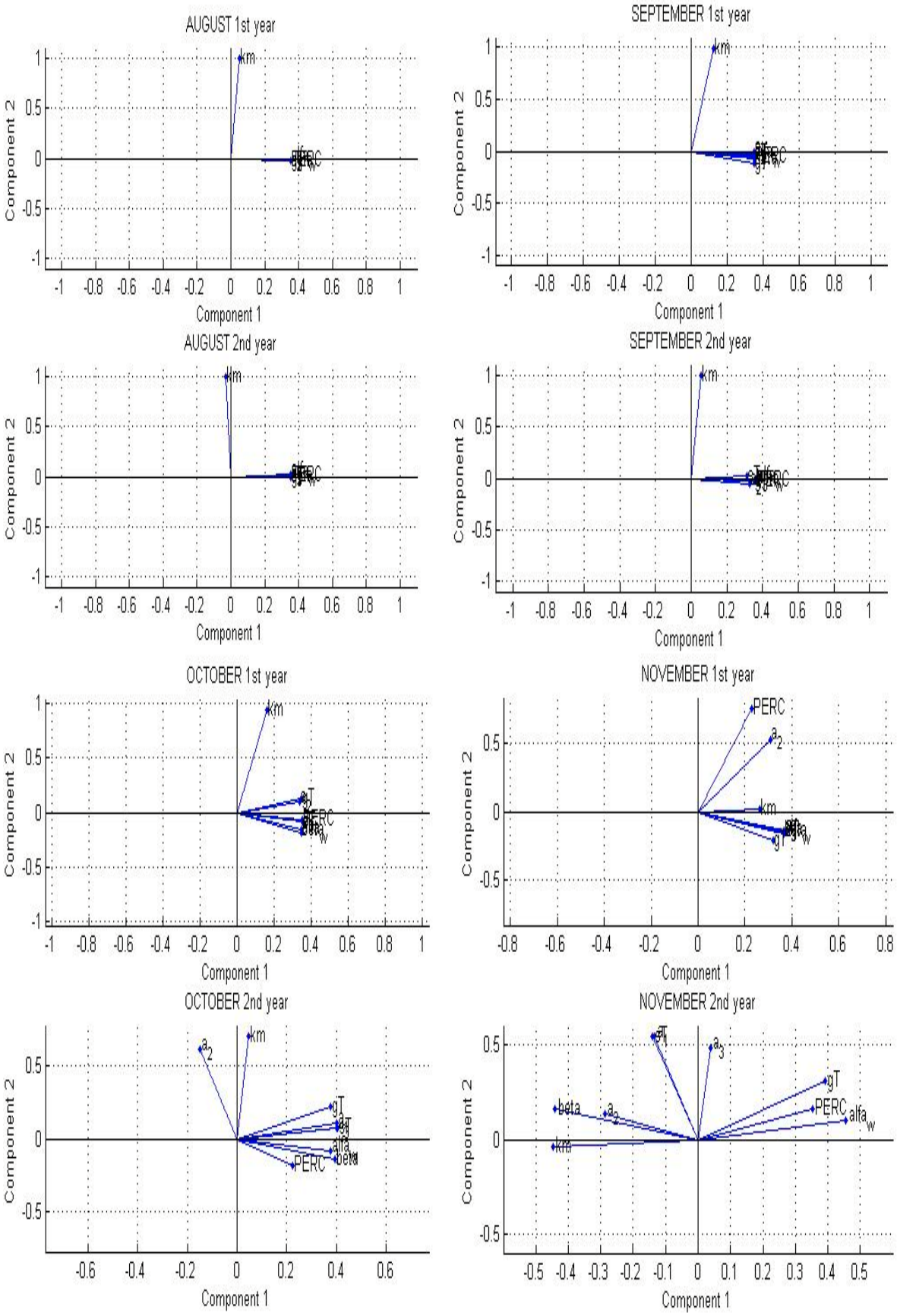
In order to improve the production planning in a hydropower system, forecast/prediction of runoff/inflow of water to a reservoir in a catchment is important. The HBV model (HBV = Hydrologiska Byråns Vattenbalansavdelning, developed in Sweden in the 1970s), is a lumped precipitation-runoff model, and is extensively used for making runoff/inflow forecasts to hydropower systems on most continents. The model has many tuning parameters, and the catchment system is characterized by few measurements. To get realistic forecasts, it is important to update the model before simulation – today the models are usually updated manually by manipulating the inputs until the model match the measured output. Well-known estimation techniques have been tested in the past, e.g. the Kalman filter, but are not commonly used due to problems with observability and identifiability. Alternative estimation methods have led to more abstract formulations of the HBV model (i.e. less based on physical reasoning). With modern formulations of estimation techniques (e.g. the EnKF = Ensemble Kalman Filter) combined with more available measurements (e.g. the use of GIS = Geographical Information Systems, etc), it is of interest to re-investigate some standard/modern estimation techniques in combination with a physically based model (HBV model) to see how the model can be used for on-line estimation/forecast of runoff/inflow to water reservoirs.

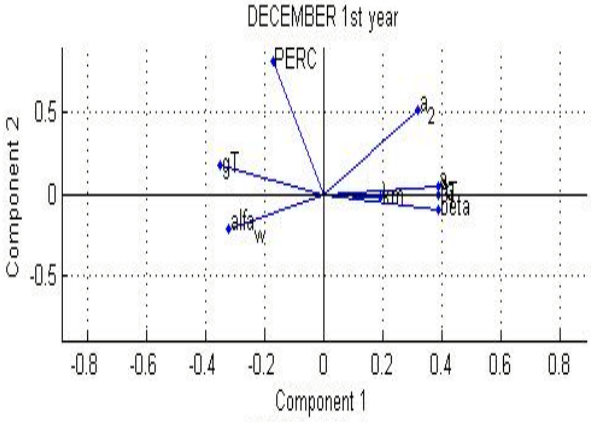
Appendix 2: Principle component coefficients

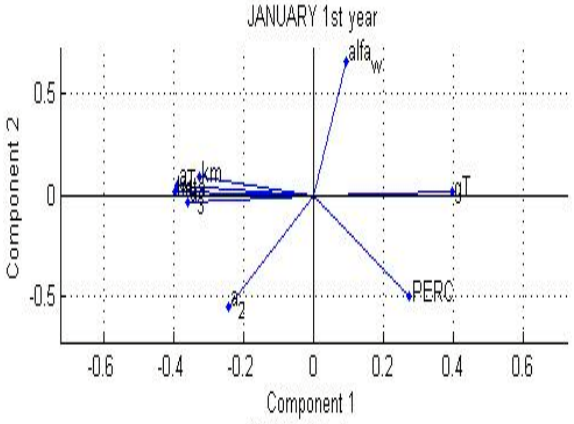
1st year (blue asterisk) and 2nd year (red asterisk)

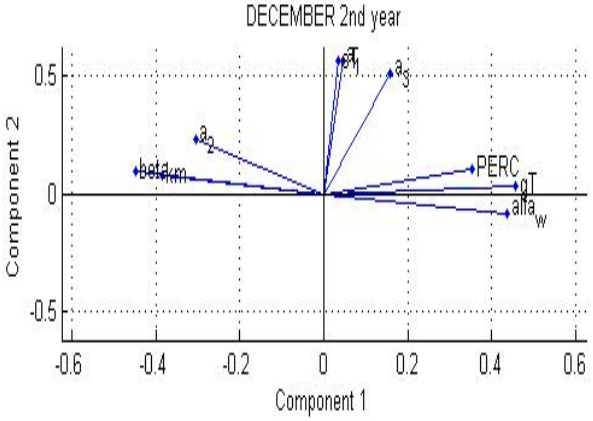


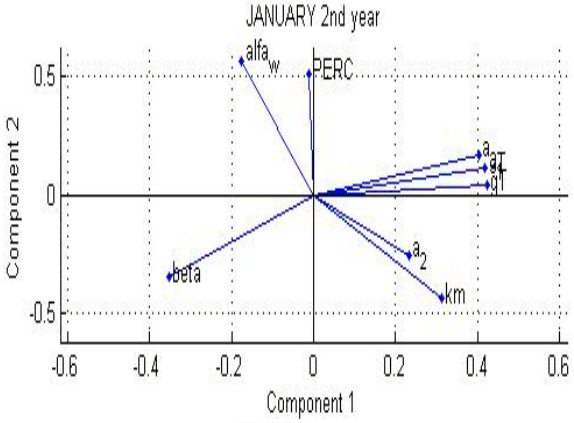
Appendix 3: Principle component loading plots

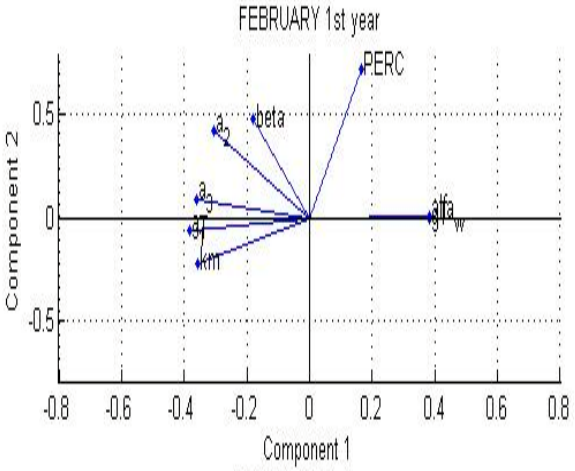


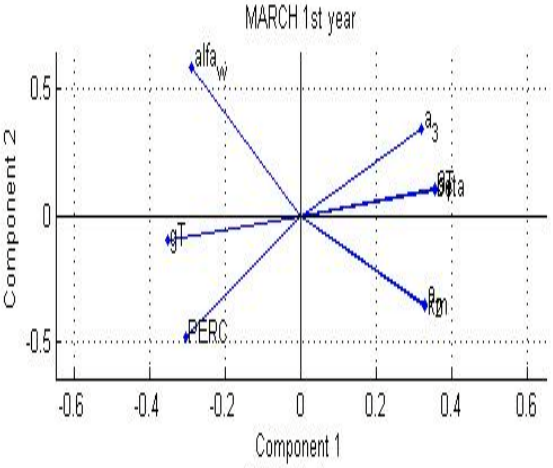

DECEMBER 1st year
PERC
a_2
gT
km
beta
$alfa_w$
Component 2
Component 1

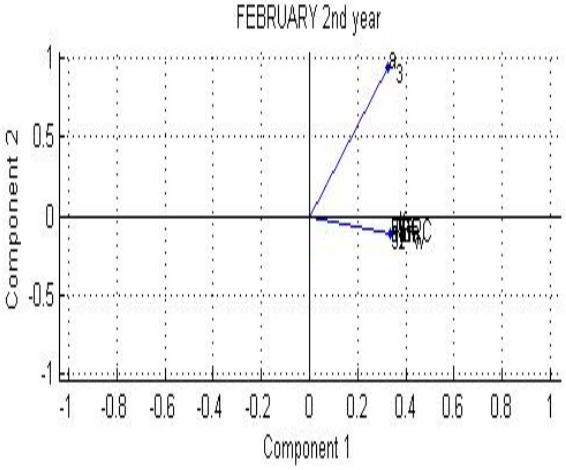

JANUARY 1st year
$alfa_w$
gT
a_2
PERC
Component 2
Component 1

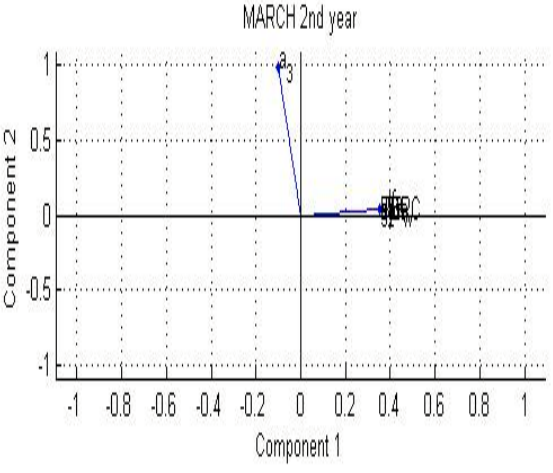

DECEMBER 2nd year
a_3
a_2
beta
km
PERC
gT
$alfa_w$
Component 2
Component 1

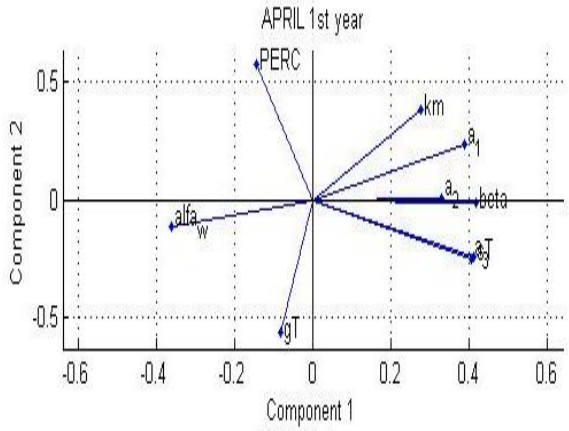

JANUARY 2nd year
$alfa_w$
PERC
a_2
beta
km
Component 2
Component 1

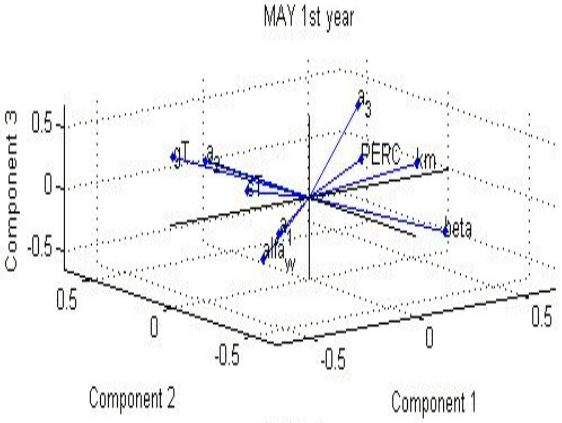

FEBRUARY 1st year
PERC
beta
a_2
a_3
gT
km
$alfa_w$
Component 2
Component 1

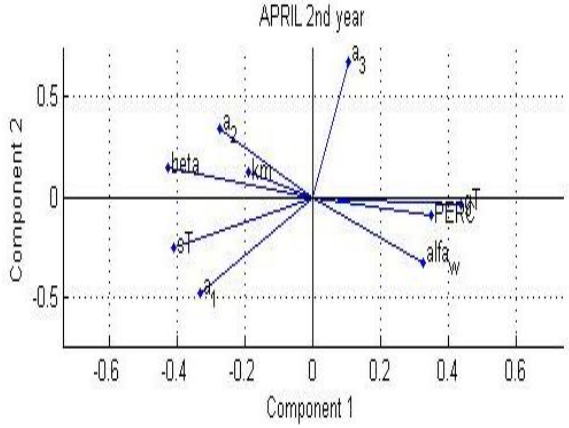

MARCH 1st year
$alfa_w$
a_3
gT
PERC
Component 2
Component 1

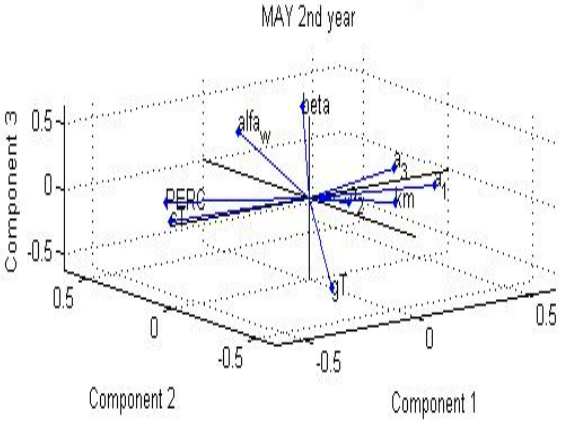

FEBRUARY 2nd year
a_3
Component 2
Component 1

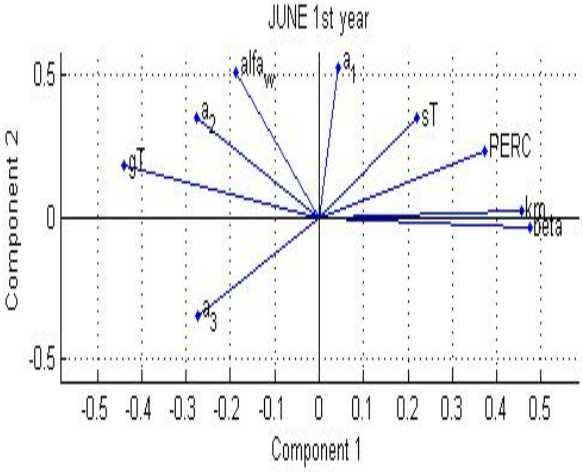

MARCH 2nd year
a_3
Component 2
Component 1

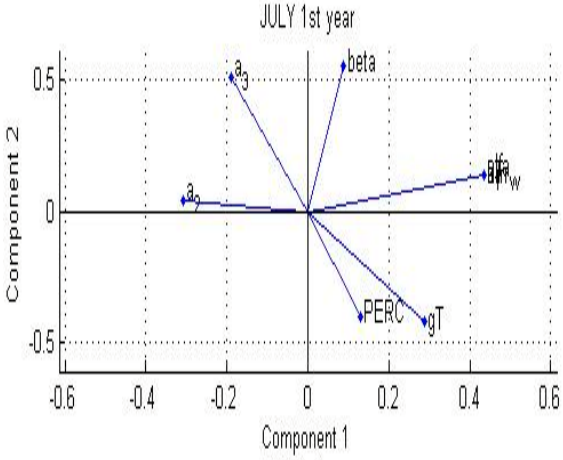

APRIL 1st year
PERC
km
a_1
a_2
beta
$alfa_w$
a_3
sT
gT
Component 2
Component 1

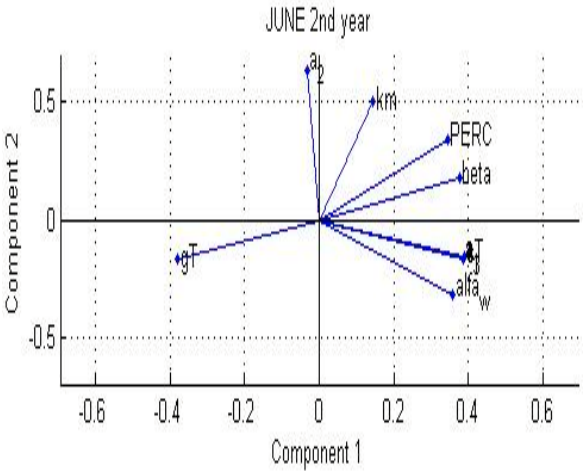

MAY 1st year
a_3
sT
a_2
PERC
km
a_1
beta
$alfa_w$
gT
Component 3
Component 2
Component 1

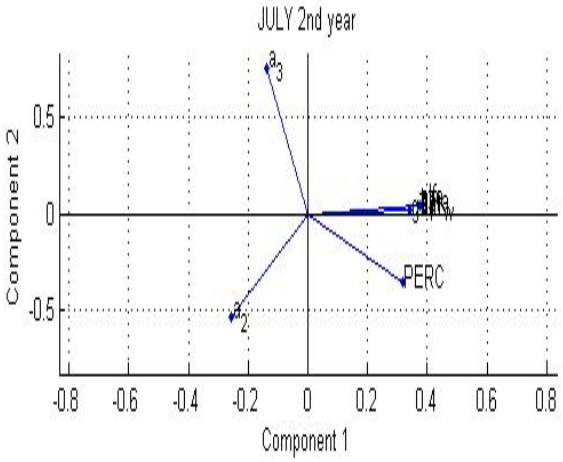

APRIL 2nd year
a_3
a_2
beta
km
PERC
sT
$alfa_w$
a_1
gT
Component 2
Component 1


MAY 2nd year
beta
$alfa_w$
a_3
a_1
PERC
km
gT
Component 3
Component 2
Component 1


JUNE 1st year
$alfa_w$
a_1
a_2
sT
PERC
gT
km
beta
a_3
Component 2
Component 1


JULY 1st year
a_3
beta
a_2
$alfa_w$
PERC
gT
Component 2
Component 1


JUNE 2nd year
a_2
km
PERC
beta
gT
sT
$alfa_w$
Component 2
Component 1


JULY 2nd year
a_3
PERC
a_2
Component 2
Component 1

Appendix 3: MATLAB Code

```
clc
clear all
filename='catchment_data_from_Eggedal_area1.xls';
sheet=3;
xlRange='A582:A982';% Date
xlRange_T_mean='Q582:Q982';%Average temperature
Average_temperature(:,1)=xlsread(filename,sheet,xlRange_T_mean);
xlRange_P='B582:B982';%Precipitation
u(:,1)=xlsread(filename,sheet,xlRange_P);
xlRange_T_h1='F582:F982';%Temperature in 283.5 m.a.s.l
u(:,2)=xlsread(filename,sheet,xlRange_T_h1);
xlRange_T_h2='G582:G982';%Temperature in 480.5 m.a.s.l
u(:,3)=xlsread(filename,sheet,xlRange_T_h2);
xlRange_T_h3='H582:H982';%Temperature in 627 m.a.s.l
u(:,4)=xlsread(filename,sheet,xlRange_T_h3);
xlRange_T_h4='I582:I982';%Temperature in 735.5 m.a.s.l
u(:,5)=xlsread(filename,sheet,xlRange_T_h4);
xlRange_T_h5='J582:J982';%Temperature in 813.5 m.a.s.l
u(:,6)=xlsread(filename,sheet,xlRange_T_h5);
xlRange_T_h6='K582:K982';%Temperature in 879 m.a.s.l
u(:,7)=xlsread(filename,sheet,xlRange_T_h6);
xlRange_T_h7='L582:L982';%Temperature in 937.5 m.a.s.l
u(:,8)=xlsread(filename,sheet,xlRange_T_h7);
xlRange_T_h8='M582:M982';%Temperature in 985.5 m.a.s.l
u(:,9)=xlsread(filename,sheet,xlRange_T_h8);
xlRange_T_h9='N582:N982';%Temperature in 1041 m.a.s.l
u(:,10)=xlsread(filename,sheet,xlRange_T_h9);
xlRange_T_h10='O582:O982';%Temperature in 1268.5 m.a.s.l
u(:,11)=xlsread(filename,sheet,xlRange_T_h10);
xlRange_Runoff='D582:D982';%Runoff
output(:,1)=xlsread(filename,sheet,xlRange_Runoff);
q=size(u,1);
Vdppepot=[0.1 0.2 0.7 1 2.3 3.5 3.5 2.3 1 0.7 0.2 0.1];
[date txt]=xlsread(filename,sheet,xlRange,'Date');
pa=read_date(txt,Vdppepot,q);
u(:,12)=pa;
N=size(u,1);
tspan=0:0.01:1;
x_ini=zeros(1,23);
p=[5.2 0.8 0.08 0.001 50 2 20 0.032 0.547 0.489 0.0462 309420 0.25];
sT=p(7); %soil zone's saturation threshold
alfa_L=p(8); %fractional area covered by lakes
a_1=p(9); % discharge frequency for surface runoff
a_2=p(10); % discharge frequency for fast runoff
a_3=p(11); % discharge frequency for base runoff
A=p(12); % catchment area
X = zeros(N,length(x_ini));
for j=1:1:N-1
    uj=u(j,:)';
    sm_r=@(t,x)sub_model_runoff(t,x,uj,p);
    [t,x]=ode45(sm_r,tspan,x_ini,[]);
    for i=1:size(x_ini,1)
```

```
    x(end,i) = max(0,x(end,i));
    end

    %%%%%%%%%%%%%%%%%%%%TOTAL RUNOFF%%%%%%%%%%%%%%%%%%%%%%
if  x(end,22)>sT
    Tr(j)=(a_1*( x(end,22)-sT)+a_2* x(end,22))*(1-
alfa_L)*A+A*(a_3)*x(end,23);
else
    Tr(j)=a_2* x(end,22)*(1-alfa_L)*A+A*(a_3*x(end,23));
end
    x_ini = x(end,:)';
    X(j,:) = x_ini';
end

%%%%%%%%%%% SNOW ZONE SIMULATION FOR AVERAGE ALTITUDE %%%%%%%%%%%%%
figure (1)
c=1;
Dry_Snow_Sim_average = zeros (size(u),1);
Wet_Snow_Sim_average = zeros (size(u),1);
for i=1:2:20
Dry_Snow_Sim_average =(X(:,i)+Dry_Snow_Sim_average)/c;
Wet_Snow_Sim_average =(X(:,i+1)+Wet_Snow_Sim_average)/c;
c=c+1;
end
plot(Dry_Snow_Sim_average,'r')
hold on
plot(Wet_Snow_Sim_average,'b')
hold on
plot(Average_temperature,'g')
legend('Vppsd(average)','Vppss(average)','Temperature(average)',4);
xlabel('Day')
ylabel('milimeter')

%%%%%%%%%%%%%%%% SNOW ZONE STATE SIMULATON IN ALL ALTITUDES %%%%%%%%%%%%%%%%
figure (2)
c=1;
for i=1:2:20
Dry_Snow_Sim =X(:,i);
Wet_Snow_Sim =X(:,i+1);
subplot(5,2,c);
plot(Dry_Snow_Sim,'r')
hold on
plot(Wet_Snow_Sim,'b')
hold on
plot(u(:,c+1),'g')
hold on
legend(['Vppsd in h',num2str(c)],['Vppss in h',num2str(c)],['Temperature in
h',num2str(c)]);
xlabel('Day')
ylabel('milimeter/Celsius')
c=c+1;
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%% GROUND ZONE SIMULATION %%%%%%%%%%%%%%%%%%%%%%%%%%
figure (3)
plot(X(:,21),'r')
hold on
plot(Average_temperature,'g')
legend('Vppgw','Temperature(average)',4);
xlabel('Day')
ylabel('milimeter/Celsius')

%%%%%%%%%%%%%%%%%%% SOIL ZONE AND BASE ZONE SIMULATION %%%%%%%%%%%%%%%%%%
figure (4)
plot(X(:,22),'B')
hold on
plot(X(:,23),'r')
legend('Vppsw','Vppbw',1);
xlabel('Day')
ylabel('milimeter')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure (5)
xlRange3='D582:D982';
Total_runoff_Real(:,1)=86400*xlsread(filename,sheet,xlRange3);
Total_runoff_Sim=Tr;
plot(Total_runoff_Real,'r')
hold on
plot(Total_runoff_Sim,'b')
legend('Total runoff Real','Total runoff Simulation')
xlabel('Day')
ylabel('mm3/s')

%%%%%%%%%%%%%%%%%%%%%%%%%%%% SENSITIVITY ANALYSIS %%%%%%%%%%%%%%%%%%%%%%%%%%
u_new=u(1:366,:);
N_new=size(u_new,1);
x_ini_new=X(1,:);
p=[5.2 0.8 0.08 0.001 50 2 20 0.032 0.547 0.489 0.0462 309420 0.25];
for k=1:13
    p_new = p;
    delta=p(k)/20;
    p_new(k)=p(k)+delta;
    X_new = zeros(N_new,length(x_ini_new));
    for v=1:1:N_new-1
        uj_new=u_new(v,:)';
        sm_r_new=@(t,x_new)sub_model_runoff(t,x_new,uj_new,p_new);
        [t,x_new]=ode45(sm_r_new,tspan,x_ini_new,[]);
        for f=1:size(x_ini_new,1)
            x_new(end,f) = max(0,x_new(end,f));
        end

        %%%%%%%%%%%%%%%%%%%% NEW TOTAL RUNOFF%%%%%%%%%%%%%%%%%%%%%%%
        if  x_new(end,22)>p_new(7)
            Tr_new(v)=(p_new(9)*( x_new(end,22)-
p_new(7))+p_new(10)*x_new(end,22))*(1-
p_new(8))*p_new(12)+p_new(12)*p_new(11)*x_new(end,23);
        else
```

```
            Tr_new(v)=p_new(10)*x_new(end,22)*(1-
p_new(8))*p_new(12)+p_new(12)*(p_new(11)*x_new(end,23));
        end
        x_ini_new = x_new(end,:)';
        X_new(k,:) = x_ini_new';
        dTrdp(v,k)=(Tr_new(v)-Tr(v))/delta;
        %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    end
end

dTrdp_new=[dTrdp(:,1) dTrdp(:,3) dTrdp(:,5) dTrdp(:,6) dTrdp(:,7) dTrdp(:,9)
dTrdp(:,10) dTrdp(:,11) dTrdp(:,13)];
dTrdp=dTrdp_new;
color=[0 0 1;0 0 0;0 1 0;0 1 1;1 0 0;1 0 1;0 0.3 0;0.9412 0.4706 0;1 0.502
0.502];
stdr = std(dTrdp);
sr = dTrdp./repmat(stdr,size(dTrdp,1),1);
figure(6)
for i=1:9
    plot(sr(:,i), 'color', color(i,:))
    hold on
end
legend('km','alfa_w','gT','beta','sT','a_1','a_2','a_3','PERC')
xlabel('Day')
ylabel('Normalized sensitivity')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PCA METHOD %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% AUGUST 1973
dTrdp_AUG=dTrdp(1:31,:);
stdr_AUG = std(dTrdp_AUG);
sr_AUG = dTrdp_AUG./repmat(stdr_AUG,size(dTrdp_AUG,1),1);
[coefs_AUG,scores_AUG,variances_AUG,t2_AUG] = princomp(sr_AUG);
percent_explained_AUG = 100*variances_AUG/sum(variances_AUG);
figure (7)
subplot(4,3,1);
pareto(percent_explained_AUG)
xlabel('Principal Component for August')
ylabel('Variance Explained (%)')

% SEPTEMBER 1973
dTrdp_SEP=dTrdp(32:61,:);
stdr_SEP = std(dTrdp_SEP);
sr_SEP = dTrdp_SEP./repmat(stdr_SEP,size(dTrdp_SEP,1),1);
[coefs_SEP,scores_SEP,variances_SEP,t2_SEP] = princomp(sr_SEP);
percent_explained_SEP = 100*variances_SEP/sum(variances_SEP);
subplot(4,3,2);
pareto(percent_explained_SEP)
xlabel('Principal Component for September')
ylabel('Variance Explained (%)')
% OCTOBER 1973
dTrdp_OCT=dTrdp(62:92,:);
stdr_OCT = std(dTrdp_OCT);
sr_OCT = dTrdp_OCT./repmat(stdr_OCT,size(dTrdp_OCT,1),1);
```

```
[coefs_OCT,scores_OCT,variances_OCT,t2_OCT] = princomp(sr_OCT);
percent_explained_OCT = 100*variances_OCT/sum(variances_OCT);
subplot(4,3,3);
pareto(percent_explained_OCT)
xlabel('Principal Component for October')
ylabel('Variance Explained (%)')

% NOVEMBER 1973
dTrdp_NOV=dTrdp(93:122,:);
stdr_NOV = std(dTrdp_NOV);
sr_NOV = dTrdp_NOV./repmat(stdr_NOV,size(dTrdp_NOV,1),1);
[coefs_NOV,scores_NOV,variances_NOV,t2_NOV] = princomp(sr_NOV);
percent_explained_NOV = 100*variances_NOV/sum(variances_NOV);
subplot(4,3,4);
pareto(percent_explained_NOV)
xlabel('Principal Component for November')
ylabel('Variance Explained (%)')

% DECEMBER 1973
dTrdp_DEC=dTrdp(123:153,:);
stdr_DEC = std(dTrdp_DEC);
sr_DEC = dTrdp_DEC./repmat(stdr_DEC,size(dTrdp_DEC,1),1);
[coefs_DEC,scores_DEC,variances_DEC,t2_DEC] = princomp(sr_DEC);
percent_explained_DEC = 100*variances_DEC/sum(variances_DEC);
subplot(4,3,5);
pareto(percent_explained_DEC)
xlabel('Principal Component for December')
ylabel('Variance Explained (%)')

% JANUARY 1974
dTrdp_JAN=dTrdp(154:184,:);
stdr_JAN = std(dTrdp_JAN);
sr_JAN = dTrdp_JAN./repmat(stdr_JAN,size(dTrdp_JAN,1),1);
[coefs_JAN,scores_JAN,variances_JAN,t2_JAN] = princomp(sr_JAN);
percent_explained_JAN = 100*variances_JAN/sum(variances_JAN);
subplot(4,3,6);
pareto(percent_explained_JAN)
xlabel('Principal Component for January')
ylabel('Variance Explained (%)')

% FEBRUARY 1974
dTrdp_FEB=dTrdp(185:212,:);
stdr_FEB = std(dTrdp_FEB);
sr_FEB = dTrdp_FEB./repmat(stdr_FEB,size(dTrdp_FEB,1),1);
[coefs_FEB,scores_FEB,variances_FEB,t2_FEB] = princomp(sr_FEB);
percent_explained_FEB = 100*variances_FEB/sum(variances_FEB);
subplot(4,3,7);
pareto(percent_explained_FEB)
xlabel('Principal Component for February')
ylabel('Variance Explained (%)')
% MARCH 1974
dTrdp_MAR=dTrdp(213:243,:);
stdr_MAR = std(dTrdp_MAR);
sr_MAR = dTrdp_MAR./repmat(stdr_MAR,size(dTrdp_MAR,1),1);
```

```
[coefs_MAR,scores_MAR,variances_MAR,t2_MAR] = princomp(sr_MAR);
percent_explained_MAR = 100*variances_MAR/sum(variances_MAR);
subplot(4,3,8);
pareto(percent_explained_MAR)
xlabel('Principal Component for March')
ylabel('Variance Explained (%)')

% APRIL 1974
dTrdp_APR=dTrdp(244:273,:);
stdr_APR = std(dTrdp_APR);
sr_APR = dTrdp_APR./repmat(stdr_APR,size(dTrdp_APR,1),1);
[coefs_APR,scores_APR,variances_APR,t2_APR] = princomp(sr_APR);
percent_explained_APR = 100*variances_APR/sum(variances_APR);
subplot(4,3,9);
pareto(percent_explained_APR)
xlabel('Principal Component for April')
ylabel('Variance Explained (%)')

% MAY 1974
dTrdp_MAY=dTrdp(274:304,:);
stdr_MAY = std(dTrdp_MAY);
sr_MAY = dTrdp_MAY./repmat(stdr_MAY,size(dTrdp_MAY,1),1);
[coefs_MAY,scores_MAY,variances_MAY,t2_MAY] = princomp(sr_MAY);
percent_explained_MAY = 100*variances_MAY/sum(variances_MAY);
subplot(4,3,10);
pareto(percent_explained_MAY)
xlabel('Principal Component for May')
ylabel('Variance Explained (%)')

% JUNE 1974
dTrdp_JUN=dTrdp(305:334,:);
stdr_JUN = std(dTrdp_JUN);
sr_JUN = dTrdp_JUN./repmat(stdr_JUN,size(dTrdp_JUN,1),1);
[coefs_JUN,scores_JUN,variances_JUN,t2_JUN] = princomp(sr_JUN);
percent_explained_JUN = 100*variances_JUN/sum(variances_JUN);
subplot(4,3,11);
pareto(percent_explained_JUN)
xlabel('Principal Component for June')
ylabel('Variance Explained (%)')

% JULY 1974
dTrdp_JUL=dTrdp(335:365,:);
stdr_JUL = std(dTrdp_JUL);
sr_JUL = dTrdp_JUL./repmat(stdr_JUL,size(dTrdp_JUL,1),1);
[coefs_JUL,scores_JUL,variances_JUL,t2_JUL] = princomp(sr_JUL);
percent_explained_JUL = 100*variances_JUL/sum(variances_JUL);
subplot(4,3,12);
pareto(percent_explained_JUL)
xlabel('Principal Component for July')
ylabel('Variance Explained (%)')
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%% PARAMETER RANKING %%%%%%%%%%%%%%%%%%%%%%%%%%%%
LABELS1 = {'PC1 Loading (AUGUST)'; 'PC1 Loading (SEPTEMBER)'; 'PC1 Loading
(OCTOBER)';'PC1 Loading (NOVEMBER)';'PC1 Loading (DECEMBER)'; 'PC1 Loading
(JANUARY)';'PC1 Loading (FEBRUARY)';'PC1 Loading (MARCH)'};
COEFS1 = [coefs_AUG(:,1) coefs_SEP(:,1) coefs_OCT(:,1) coefs_NOV(:,1)
coefs_DEC(:,1) coefs_JAN(:,1) coefs_FEB(:,1) coefs_MAR(:,1)];
figure (8)
for i=1:8
  subplot(2,4,i)
  plot(COEFS1(:,i), 'b*')
xlabel(LABELS1(i))
end
LABELS2 = {'PC1 Loading (APRIL)'; 'PC2 Loading (APRIL)'; 'PC1 Loading
(MAY)';'PC2 Loading (MAY)';'PC3 Loading (MAY)';'PC1 Loading (JUNE)'; 'PC2
Loading (JUNE)';'PC1 Loading (JULY)';'PC2 Loading (JULY)'};
COEFS2 = [coefs_APR(:,1) coefs_APR(:,2) coefs_MAY(:,1) coefs_MAY(:,2)
coefs_MAY(:,3) coefs_JUN(:,1) coefs_JUN(:,2) coefs_JUL(:,1) coefs_JUL(:,2)];
figure (9)
for i=1:9
  subplot(3,3,i)
  plot(COEFS2(:,i), 'b*')
xlabel(LABELS2(i))
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%% COLINEARITY %%%%%%%%%%%%%%%%%%%%%%%%%%%%
vbls = {'km','alfa_w','gT','beta','sT','a_1','a_2','a_3','PERC'};
TITLES = {'AUGUST'; 'SEPTEMBER';
'OCTOBER';'NOVEMBER';'DECEMBER';'JANUARY';'FEBRUARY';'MARCH'};
COEFS = [coefs_AUG(:,1:2) coefs_SEP(:,1:2) coefs_OCT(:,1:2) coefs_NOV(:,1:2)
coefs_DEC(:,1:2) coefs_JAN(:,1:2) coefs_FEB(:,1:2) coefs_MAR(:,1:2)];
j=1;
for i=1:8
    figure (9+i)
    biplot(COEFS(:,j:j+1),'varlabels',vbls);
    title(TITLES(i))
    j=j+2;
end
figure (18)
subplot(2,2,1)
biplot(coefs_APR(:,1:2),'varlabels',vbls);
title('APRIL')
subplot(2,2,2)
biplot(coefs_MAY(:,1:3),'varlabels',vbls);
title('MAY')
subplot(2,2,3)
biplot(coefs_JUN(:,1:2),'varlabels',vbls);
title('JUNE')
subplot(2,2,4)
biplot(coefs_JUL(:,1:2),'varlabels',vbls);
title('JULY')


%%%%%%%%%%%%%% DUAL PARAMETER & STATE ESTIMATION ON AUGUST %%%%%%%%%%%%%%%%%
% By knowing higher idendifiable parameters in each month, and considering
```

```
% zero values as initial states of model on 1st of August(starting point),
% the Augmented nonlinear Least squares methodand can be implemented to
estimate
% both states and parameters simultaniously such as following:
%%%%%%%%%%%%%%%%%%%%%%% PARAMETER ESTIMATION ON AUGUST %%%%%%%%%%%%%%%%%%%%%%%
input_aug_for_param=u(1:8,:);
output_aug_for_param=output(1:8,:);
X_est(1,:) = zeros(1,23);
states_ini=[0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0];
S_aug_est(:,1)=states_ini;
P_aug=[0.08;50;2;20;0.547;0.489;0.0462;0.25];
P_aug_LB=[0.08*0.5;50*0.01;2*0.5;20*0.01;0.547*0.01;0.489*0.01;0.0462*0.01;0
.25*0.5];
P_aug_UB=P_aug*2;
options = optimset('Display','iter','MaxFunEvals',1000);
[P_aug_est(:,1) P_fval] = fminsearchbnd(@(P)
LSCostFunction_for_param(P,states_ini,input_aug_for_param,output_aug_for_par
am),P_aug,P_aug_LB,P_aug_UB,options);

%%%%%%%%%%%%%%%%%%%%%%%% STATES ESTIMATION ON AUGUST %%%%%%%%%%%%%%%%%%%%%%%
x_ini=states_ini';
p=[5.2 0.8 P_aug_est(1,1) 0 P_aug_est(2,1) P_aug_est(3,1) P_aug_est(4,1)
0.032 P_aug_est(5,1) P_aug_est(6,1) P_aug_est(7,1) 309420 P_aug_est(8,1)];
uj=u(1,:)';
sm_r=@(t,x)sub_model_runoff(t,x,uj,p);
[t,x]=ode45(sm_r,tspan,x_ini,[]);
for i=1:size(x_ini,2)
    x(end,i) = max(0,x(end,i));
end
x_ini = x(end,:);
states_ini=x_ini';
param=P_aug_est(:,1);
input_aug_for_state=u(1:23,:);
output_aug_for_state=output(1:23,:);
S_aug_LB=[0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0];
S_aug_UB=[10;0.8;10;0.8;10;0.8;10;0.8;10;0.8;10;0.8;10;0.8;10;0.8;10;0.8;10;
0.8;100;100;100];
options = optimset('Display','iter','MaxFunEvals',500);
[S_aug_est(:,1) fval] = fminsearchbnd(@(S)
LSCostFunction_for_states(S,param,input_aug_for_state,output_aug_for_state),
states_ini,S_aug_LB,S_aug_UB,options);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%% STATES OF 2nd OF AUGUST %%%%%%%%%%%%%%%%%%%%%%%
for k=1:1:31
x_ini=S_aug_est(:,k)';
p=[5.2 0.8 P_aug_est(1,1) 0 P_aug_est(2,1) P_aug_est(3,1) P_aug_est(4,1)
0.032 P_aug_est(5,1) P_aug_est(6,1) P_aug_est(7,1) 309420 P_aug_est(8,1)];
uj=u(k+1,:)';
sm_r=@(t,x)sub_model_runoff(t,x,uj,p);
[t,x]=ode45(sm_r,tspan,x_ini,[]);
for i=1:23
    x(end,i) = max(0,x(end,i));
end
x_ini = x(end,:);
```

```
states_ini=x_ini';
param=P_aug_est(:,1);
input_aug_for_state=u(k+1:k+23,:);
output_aug_for_state=output(k+1:k+23,:);
S_aug_LB=[0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0];
S_aug_UB=[10;0.8;10;0.8;10;0.8;10;0.8;10;0.8;10;0.8;10;0.8;10;0.8;10;
0.8;100;100;100];
options = optimset('Display','iter','MaxFunEvals',1000);
[S_aug_est(:,k+1) fval] = fminsearchbnd(@(S)
LSCostFunction_for_states(S,param,input_aug_for_state,output_aug_for_state),
states_ini,S_aug_LB,S_aug_UB,options);
End

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
p=[5.2 0.8 P_aug_est(1) 0 P_aug_est(2) P_aug_est(3) P_aug_est(4) 0.032
P_aug_est(5) P_aug_est(6) P_aug_est(7) 309420 P_aug_est(8)];
sT=p(7); %soil zone's saturation threshold
alfa_L=p(8); %fractional area covered by lakes
a_1=p(9); % discharge frequency for surface runoff
a_2=p(10); % discharge frequency for fast runoff
a_3=p(11); % discharge frequency for base runoff
A=p(12); % catchment area

%%%%%%%%%%%%%%%% TOTAL RUNOFF WITH ESTIMATED PARAMETERS AND STATES%%%%%%%%%%
for j=1:1:31
if  S_aug_est(22,j)>sT
    Tr_est(j)=(a_1*( S_aug_est(22,j)-sT)+a_2* S_aug_est(22,j))*(1-
0.032)*309420+309420*a_3*S_aug_est(23,j);
else
    Tr_est(j)=a_2* S_aug_est(22,j)*(1-
0.032)*309420+309420*a_3*S_aug_est(23,j);
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure
xlRange3='D582:D982';
Total_runoff_Real(:,1)=86400*xlsread(filename,sheet,xlRange3);
Total_runoff_Sim=Tr;
Total_runoff_est=Tr_est(1:31);
plot(Total_runoff_Real,'r')
hold on
plot(Total_runoff_Sim,'b')
hold on
plot(Total_runoff_est,'g')
legend('Real runoff','Simulated runoff without estimation','Simulated runoff
with estimation')
xlabel('Day')
ylabel('mm3/s')


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% FMINSEARCHBND FUNCTION %%%%%%%%%%%%%%%%%%%%%%%
```

```
function [x,fval,exitflag,output] =
fminsearchbnd(fun,x0,LB,UB,options,varargin)
% created by John D'Emico
% 11 Aug 2005 and updated on 6th of Feb 2012
% Reference:
% http://www.mathworks.com/matlabcentral/fileexchange/8277-fminsearchbnd-
%fminsearchcon
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  LB - lower bound vector or array, must be the same size as x0
%
%       If no lower bounds exist for one of the variables, then
%       supply -inf for that variable.
%
%       If no lower bounds at all, then LB may be left empty.
%
%       Variables may be fixed in value by setting the corresponding
%       lower and upper bounds to exactly the same value.
%
%  UB - upper bound vector or array, must be the same size as x0
%
%       If no upper bounds exist for one of the variables, then
%       supply +inf for that variable.
%
%       If no upper bounds at all, then UB may be left empty.
%
%       Variables may be fixed in value by setting the corresponding
%       lower and upper bounds to exactly the same value.
%

% size checks
xsize = size(x0);
x0 = x0(:);
n=length(x0);

if (nargin<3) || isempty(LB)
  LB = repmat(-inf,n,1);
else
  LB = LB(:);
end
if (nargin<4) || isempty(UB)
  UB = repmat(inf,n,1);
else
  UB = UB(:);
end

if (n~=length(LB)) || (n~=length(UB))
  error 'x0 is incompatible in size with either LB or UB.'
end

% set default options if necessary
if (nargin<5) || isempty(options)
```

```
  options = optimset('fminsearch');
end

% stuff into a struct to pass around
params.args = varargin;
params.LB = LB;
params.UB = UB;
params.fun = fun;
params.n = n;
% note that the number of parameters may actually vary if
% a user has chosen to fix one or more parameters
params.xsize = xsize;
params.OutputFcn = [];

% 0 --> unconstrained variable
% 1 --> lower bound only
% 2 --> upper bound only
% 3 --> dual finite bounds
% 4 --> fixed variable
params.BoundClass = zeros(n,1);
for i=1:n
  k = isfinite(LB(i)) + 2*isfinite(UB(i));
  params.BoundClass(i) = k;
  if (k==3) && (LB(i)==UB(i))
    params.BoundClass(i) = 4;
  end
end

% transform starting values into their unconstrained
% surrogates. Check for infeasible starting guesses.
x0u = x0;
k=1;
for i = 1:n
  switch params.BoundClass(i)
    case 1
      % lower bound only
      if x0(i)<=LB(i)
        % infeasible starting value. Use bound.
        x0u(k) = 0;
      else
        x0u(k) = sqrt(x0(i) - LB(i));
      end

      % increment k
      k=k+1;
    case 2
      % upper bound only
      if x0(i)>=UB(i)
        % infeasible starting value. use bound.
        x0u(k) = 0;
      else
        x0u(k) = sqrt(UB(i) - x0(i));
      end
```

```
      % increment k
      k=k+1;
    case 3
      % lower and upper bounds
      if x0(i)<=LB(i)
        % infeasible starting value
        x0u(k) = -pi/2;
      elseif x0(i)>=UB(i)
        % infeasible starting value
        x0u(k) = pi/2;
      else
        x0u(k) = 2*(x0(i) - LB(i))/(UB(i)-LB(i)) - 1;
        % shift by 2*pi to avoid problems at zero in fminsearch
        % otherwise, the initial simplex is vanishingly small
        x0u(k) = 2*pi+asin(max(-1,min(1,x0u(k))));
      end

      % increment k
      k=k+1;
    case 0
      % unconstrained variable. x0u(i) is set.
      x0u(k) = x0(i);

      % increment k
      k=k+1;
    case 4
      % fixed variable. drop it before fminsearch sees it.
      % k is not incremented for this variable.
  end

end
% if any of the unknowns were fixed, then we need to shorten
% x0u now.
if k<=n
  x0u(k:n) = [];
end

% were all the variables fixed?
if isempty(x0u)
  % All variables were fixed. quit immediately, setting the
  % appropriate parameters, then return.

  % undo the variable transformations into the original space
  x = xtransform(x0u,params);

  % final reshape
  x = reshape(x,xsize);

  % stuff fval with the final value
  fval = feval(params.fun,x,params.args{:});

  % fminsearchbnd was not called
```

```
  exitflag = 0;

  output.iterations = 0;
  output.funcCount = 1;
  output.algorithm = 'fminsearch';
  output.message = 'All variables were held fixed by the applied bounds';

  % return with no call at all to fminsearch
  return
end

% Check for an outputfcn. If there is any, then substitute my
% own wrapper function.
if ~isempty(options.OutputFcn)
  params.OutputFcn = options.OutputFcn;
  options.OutputFcn = @outfun_wrapper;
end

% now we can call fminsearch, but with our own
% intra-objective function.
[xu,fval,exitflag,output] = fminsearch(@intrafun,x0u,options,params);

% undo the variable transformations into the original space
x = xtransform(xu,params);

% final reshape to make sure the result has the proper shape
x = reshape(x,xsize);

% Use a nested function as the OutputFcn wrapper
  function stop = outfun_wrapper(x,varargin);
    % we need to transform x first
    xtrans = xtransform(x,params);

    % then call the user supplied OutputFcn
    stop = params.OutputFcn(xtrans,varargin{1:(end-1)});

  end

end % mainline end

% ======================================
% ========= begin subfunctions =========
% ======================================
function fval = intrafun(x,params)
% transform variables, then call original function

% transform
xtrans = xtransform(x,params);

% and call fun
fval = feval(params.fun,reshape(xtrans,params.xsize),params.args{:});
```

```
end % sub function intrafun end

% ====================================
function xtrans = xtransform(x,params)
% converts unconstrained variables into their original domains

xtrans = zeros(params.xsize);
% k allows some variables to be fixed, thus dropped from the
% optimization.
k=1;
for i = 1:params.n
  switch params.BoundClass(i)
    case 1
      % lower bound only
      xtrans(i) = params.LB(i) + x(k).^2;

      k=k+1;
    case 2
      % upper bound only
      xtrans(i) = params.UB(i) - x(k).^2;

      k=k+1;
    case 3
      % lower and upper bounds
      xtrans(i) = (sin(x(k))+1)/2;
      xtrans(i) = xtrans(i)*(params.UB(i) - params.LB(i)) + params.LB(i);
      % just in case of any floating point problems
      xtrans(i) = max(params.LB(i),min(params.UB(i),xtrans(i)));

      k=k+1;
    case 4
      % fixed variable, bounds are equal, set it at either bound
      xtrans(i) = params.LB(i);
    case 0
      % unconstrained variable.
      xtrans(i) = x(k);

      k=k+1;
  end
end

end % sub function xtransform end
```