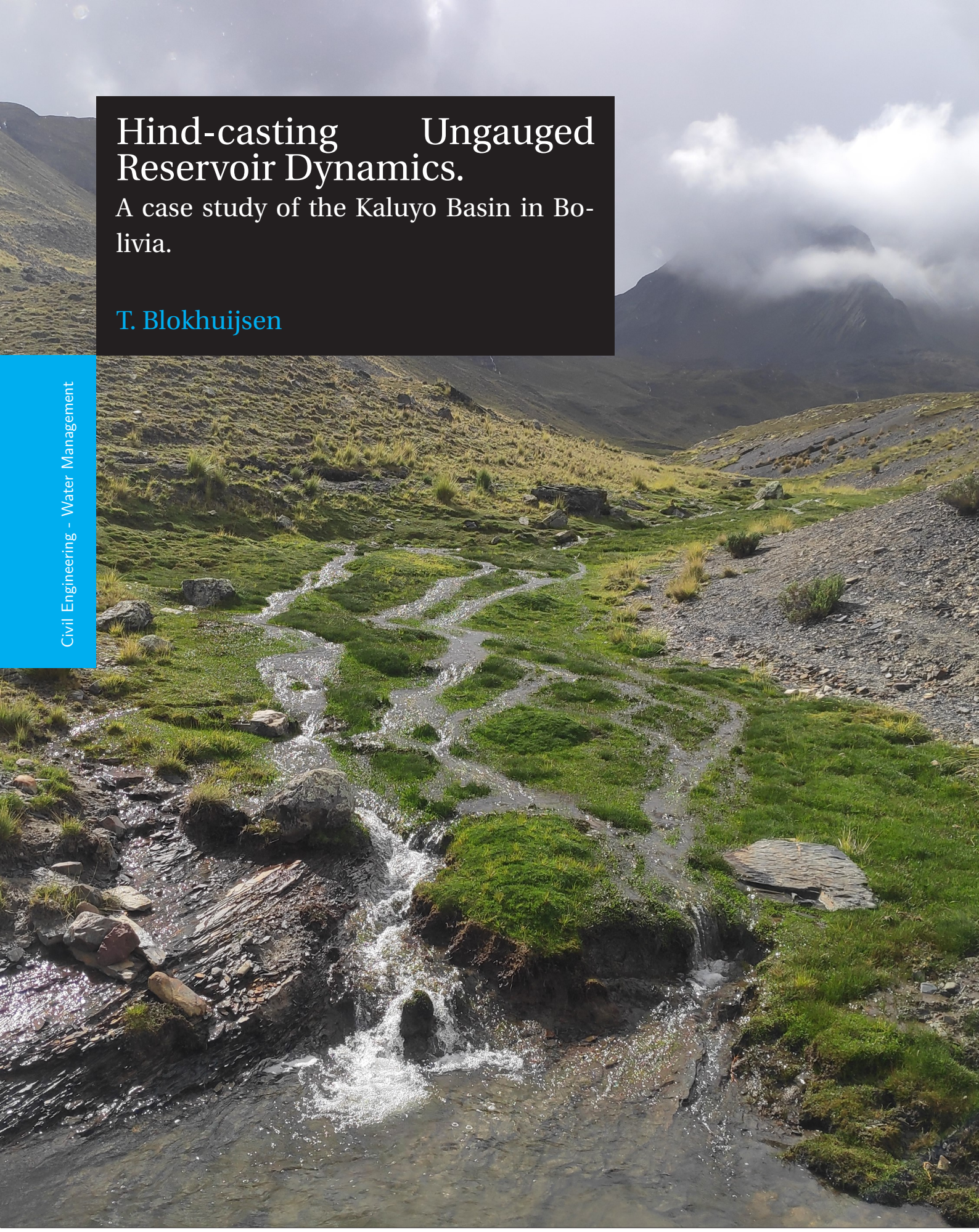
Hind-casting Ungauged Reservoir Dynamics.

A case study of the Kaluyo Basin in Bolivia.

T. Blokhuijsen

Civil Engineering - Water Management



TU Delft Delft University of Technology

Challenge the future

HIND-CASTING UNGAUGED RESERVOIR DYNAMICS.

A CASE STUDY OF THE KALUYO BASIN IN BOLIVIA.

by

T. Blokhuijsen

in partial fulfillment of the requirements for the degree of

Master of Science
in Civil Engineering

at the Delft University of Technology,
to be defended publicly on 11 December, 2020 at 10:00 .

Supervisor:	Assoc. Prof. PhD. MSc. M. Hrachowitz	
Thesis committee:	Dr. ir. G. Schoups,	TU Delft
	Dr. MSc. M.A. Schleiss,	TU Delft
	MSc. H.G. Nomden,	Royal HaskoningDHV

An electronic version of this thesis is available at http://repository.tudelft.nl/.

Royal HaskoningDHV
Enhancing Society Together

TU Delft Delft University of Technology

PREFACE

"It is far more difficult to observe correctly than most men imagine; to behold is not necessarily to observe, and the power of comparing and combining is only to be obtained by education."

Alexander von Humboldt

With this report comes an end to ten years of academic education. A period not only representing growth, discovery and change on an intellectual level, but also on a personal and emotional one. These years led me to countless interactions with people, situations, cultures and landscapes all over the world; which have had a very powerful effect on me. Some more vivid in memory than others, but each one part of the whole.

Although maybe not the most *catchy* of quotes, the epigraph by Alexander von Humboldt embodies much of what my time at the TU Delft signifies for me. Von Humboldt described new mathematics and instruments like the thermometer as *our new organs* that set us in closer contact with the world. Today our ability to behold is immeasurably greater and more extensive than the novel thermometers during Von Humboldts' era. The thousands of satellites orbiting the earth are just one example. I first met Von Humboldt during a lecture *Hydrological Modelling* by prof. Savenije. Where others saw clouds, trees and animals Von Humboldt saw an ecosystem with its intricate interactions. Where his contemporaries saw just the society, Von Humboldt saw an exploitative system for humans and environment alike. He truly *observed*, and truly understood.

Especially in the age of A.I. it's important to understand how things work or at least try to. Not just to apply the latest model without knowing why. This is the most valuable lesson I take away from my education in general and specifically the lectures on Hydrological Modelling of Markus Hrachowitz and Huub Savenije. If someone is so passionate about a modeling philosophy that during the lecture several crayons are destroyed within a minute (while shouting-out: *"I'm getting too emotional now!"* (Markus)) you know there is something special to it. I found the lectures very inspiring and fun.

At the water management section a wide range of subjects is taught. From hard sciences like mathematical optimisation and chemistry to socio-hydrology, ethics and anthropology. This holistic view of what water management is, is why I enjoyed it so much and what makes the master so valuable as basis for my future. Therefore, I first and foremost want to compliment and thank the entire Water Management staff. The greatest asset is the atmosphere created by you all, an incredible learning environment. I'm really fortunate to have received the most formative schooling of my life, from you.

Without the help and opportunities offered to me this thesis would not have been. I want to thank Royal HaskoningDHV for the opportunity to finish my education by working on a real-life, challenging and relevant problem. Seeing my work being applied gave an extra dimension to my research.
Special thanks to the chair of my committee: Markus Hrachowitz. I am very grateful for your and guidance, and in difficult moments encouragement and kindness. Furthermore, I want to thank the other committee-members Harm Nomden, Gerrit Schoups and Marc Schleiss. Harm for your help on a daily basis. I enjoyed working with you and especially our time in Bolivia. It is very unfortunate that due to Covid-19 our cooperation was mostly virtual thereafter. Gerrit and Marc for their support, perspective and advice.

Lastly I want to express my gratitude to my family and friends; for the endless friendship, love, laughs, comfort and inspiration you offer me. Everyday I enjoy our exchange of thoughts.

Through all our interactions; I am hopefully able to observe.

Thies Blokhuijsen
Leiden, December 2020

ABSTRACT

The construction of drinking-water reservoirs in previously free-flowing river basins introduces challenges in operational water management, especially in data scarce environments. Often resulting in effectively ungauged basins. In order to be able to estimate the expected increase in water supply, the hydrology of the ungauged sub-basins must be deduced from the whole basin. This can be done by estimation based on relative area to the whole. However such a method neglects the potential differences in boundary conditions and ecosystems. This research presents an improved hydrological modeling method. Through the development of a process-based, flexible model (FLEX-Topo) with four classes the hydrology of a polar-desert, high-mountainous basin in the Cordillera Real (Bolivia) is investigated. To increase realism additional datasources are used in model development and calibration. Internal model states related to vegetation- and snow-presence are evaluated in the frequency- respectively time domain. The calibrated model is applied to the sub-domains to identify important hydrological differences. To further support operational decision making the correlation between the ENSO-Index and the local climate is investigated, which is not found to be significant. The research shows that the developed FLEX-Topo model is suited to function in a data scarce- and low data quality environment. The model is able to reproduce the natural flow regime to a high degree. Appropriate model realism can be assumed and sensible sub-basin hydrology investigation is achieved. Subsequent reservoir dynamics hind-casting is performed to increase knowledge on the to be expected flows and water levels.

NOMENCLATURE

List of Abbreviations

FDC	Flow Duration Curve.
ANN	Artificial Neural Networks
DEM	Digital Elevation Model
DRR	Drought Risk Reduction
ENSO	El Niño Southern-Oscillation
EPSAS	Empresa Pública Social de Agua y Saneamiento
HAND	Height Above Nearest Drain
HRU	Hydrological Response Unit.
ISM	Integrated System Model, to Kaluyo system including its hydrology and infrastructure.
LPWEEW	La Paz Water Efficiency and Early Warning project
NASA	National Aeronautics and Space Administration
NDSI	Normalized Difference Snow INdex
NDVI	Normalized Difference Vegetation Index
NIR	Measurement of the near-infrared reflectance region.
NSE	Nash-Sutcliffe Efficiency
OF	Objective Function
RCF	Relative Contribution Factor
RC	Runoff Coefficient. The ratio between measured precipitation and discharge, calculated by dividing the discharge by precipitation.
Red	Measurement of the visible red reflectance region.
RHDHV	Royal HaskoningDHV
RMSE	Root mean squared error.
RVE	Relative Volume Error
SRTM-30	Shuttle Radar Topography Mission at 30 meters resolution
SST	Sea-Surface Temperatures
UNISDR	United Nations Office for Disaster Risk Reduction
w.e	Water Equivalent
WSS	Water Supply System
OM	Organic Matter

WTP	Water Treatment Plant

List of Symbols

Δ_T	$T_{max} - T_{min}$ (°C)
γ_T	Temperature Lapse Rate ($°CL^{-1}$)
Ab	Ablation, mass reduction of a glacier (L^3T^{-1})
E	Evaporation (L^3T^{-1})
E_a	Actual Evaporation (L^3T^{-1})
E_a	Minimum daily temperature lapse rate ($°Cm^{-1}$)
E_i	Interception Evaporation (L^3T^{-1})
E_i	Maximum daily temperature lapse rate ($°Cm^{-1}$)
E_P	Potential Evaporation (L^3T^{-1})
E_S	Sublimation, transition from solid to the gaseous phase of an substance (L^3T^{-1})
E_T	Mean daily temperature lapse rate ($°Cm^{-1}$)
E_T	Transpiration Evaporation (L^3T^{-1})
$E_{P,Hg}$	Potential Evaporation, calculated with Hargreaves equation (L^3T^{-1})
G	Solar irradiance (Wm^{-2})
H	Solar Irradiation (Jm^{-2})
h	River stage (L)
M	Melt, transition from liquid to the gaseous phase of an substance(L^3T^{-1})
P	Precipitation (L^3T^{-1})
Q	River discharge (L^3T^{-1})
R_A	Mean extra-terrestrial radiation (PL^{-2})
T	Temperature (°C)
ED	Euclidean Distance of parameter-set.
WI	Weight of objective function i for calculation of Euclidean Distance.

List of Figures

List of Tables

CONTENTS

1

INTRODUCTION

1.1. MOTIVATION

Droughts were abundant in 2016. All over the world major- and minor cities were experiencing or expecting severe water shortages. The city of La Paz in Bolivia was no exception. In November 2016 two of the main drinking water reservoirs ran dry. With no water in storage the company responsible for the water supply system, Empresa Pública Social de Agua y Saneamiento (hereafter: EPSAS), had to close the tap for more than 300.000 people. Water had to be transported by trucks to large parts of the city. When former president Morales declared a national emergency on 21st of November, more than half of Bolivia's municipalities had already done so.[3] Millions of people were affected in all of Bolivia. The emergency lasted for two months until in mid January 2017 rains arrived.

The drinking water for the cities of La Paz, El Alto and surrounding areas is supplied by the the La Paz Water Supply System. Around two million people depend on the system for their drinking water and thus for their livelihoods. Intermittent or absent drinking water supply has far-reaching consequences. A Drought Risk Reduction (hereafter: DRR) mission initiated by the Dutch government following the 2016 drought concluded that the water shortages had negatively impacted the country in terms on the societal-, environmental and economical level. Changing land-use, increasing demographic pressure and climate change are considered to be the main causes [4] [5] [6]. Furthermore, it related the 11-months dry spell to the warm phase of El Niño Southern-Oscillation (hereafter: ENSO) , also known as an El Niño year.

In direct response to the crisis Royal HaskoningDHV (hereafter: RHDHV), EPSAS and The Development Related Infrastructure Investment Vehicle of the Netherlands Enterprise Agency initiated the La Paz Water Efficiency and Early Warning project (hereafter: LPWEEW). In order to ensure a permanent supply and good quality water to consumers. The three main focuses of the project are Water Resource Monitoring and Control, increasing Water Supply Efficiency and improving the Pampahasi Water Treatment Plant. [7]

In essence the LPWEEW project is about Drought Risk Reduction. Risk is defined as the probability of occurrence multiplied by the impact. Risk can be reduced either by reducing the chances of an undesirable event happening, or by reducing the impact or consequences of such an event when it happens.According to the United Nations Office for Disaster Risk Reduction (hereafter: UNISDR) the basis for disaster risk reduction is knowledge: which signals, shocks and drivers compose a risk to the stability of the system.[8] If known, measures can be taken far before a critical state is reached.

The La Paz WSS is made up of four water supply systems, each with its own water sources, a treatment plant and distribution system [7]:

1. Tilata system
2. El Alto System
3. Achachicala System
4. Pampahasi System

The Achachicala system receives the majority of its water the Kaluyo basin. The Kaluyo supply is supplemented by water from the Milluni basin. After the 2016 Drought three drinking water reservoirs have been constructed in Kaluyo, resulting in a new situation. In this research the new Achachicala system is investigated.

1.2. PROBLEM STATEMENT

As stated the bases for risk reduction in knowledge of the signals, shocks and drivers that compose risk to the stability of the system. However, with the construction of the reservoirs, combined with the lack of measurements at the dam locations (i.e. where the dams obstruct upstream flows from entering the free-flowing river), resulted in three new ungauged basins. Currently important knowledge about the hydrology of the system is lacking such as:

1. Expected steam-flows, both in the wet- as well as in the dry season.
2. Contribution of each sub-catchment to the total stream-flow.
3. Dominant flow generating processes.
4. Glacier contribution to the flow.

A problem with respect to increasing the knowledge of the hydrology in the basin is the data; in terms of availability and quality and mostly relating to the discharge and precipitation data.

Next to the hydrology, the systems' dynamics as a reaction to the hydrology is a gap in knowledge. Whether all or any of the reservoirs is likely to fill to its maximum capacity during the wet season, and under what circumstances. Or under which conditions the reservoirs are likely to completely drain during the dry season? If known critical situations can be more easily identified. Unknowns:

1. Expected reservoir contribution.
2. Expected efficiency increase.

The main of the drivers of the system is known: the local climate. But the climate shows high variability, both inter- and intra-annual. Within the hydrological year there is a distinct wet- and dry season. Between years the precipitation can varies often between minus and plus 50 percent of the mean accumulated precipitation of around 600 mm per year. From an operational point of view it is very valuable to have some grasp on the expected variability of climatic variables (e.g. precipitation or temperature) over the coming months. The ENSO cycle is often considered to be such an indicator. Where warm ENSO phases are associated with local warm and dry years, while cold-phases (La Niña) are considered to be cold and wet. Whether the ENSO-cycle is a reliable predictor for local conditions, on which operational decisions can be based is not known.

1.3. RESEARCH OBJECTIVES

The aim of this study is to provide insight into - and increase knowledge of emergent behaviour of new Achachicala system as part of the La Paz Water Supply System in order to reduce the risk of drought. The emergent behaviour of the total system is the result of the interaction of its sub-systems, infrastructure to the climatic conditions.

(1) to gain insight into the hydrological functioning of both the Kaluyo basin as whole, and its sub-basins individually. This will be done through the development a rainfall-runoff model that can represent both the total basin as well as the sub-basins individually.

(2) to increase knowledge through hind-casting about the behaviour of the newly formed system, with a focus on the reservoir dynamics and increase of water availability.

(3) The hypothesis is that the variability of the ENSO-Index can be used to base Water Supply System decisions on. This can only be the case if there is a significant relationship. This study therefore aims to investigate the ENSO-Index and the local climate relationship and determine whether its significant.

1.4. RESEARCH QUESTIONS

The research objective can be met by answering the following research questions related to the three components: the hydrological functioning of the basin, the usefulness of the ENSO-Index for operational decisions and the integrated systems response to the modelled hydrological conditions.

COMPONENT 1

The first component refers to the hydrology of the basin. Through the development of a rainfall runoff model the following research questions will be answered:

1. What are the dominant hydrological processes in the basin, and where are flows created?
2. Can auxiliary data measurements be used in model development and calibration to increase the reflection of reality, and decrease the dependence on discharge data in calibration?

COMPONENT 2

Can a statistically significant linear correlation relationship between the local climate variability and fluctuations in the ENSO-Index be found?

COMPONENT 3

The last component relates to the new Kaluyo system, its response to the hydrology and expected reservoir behaviour. Based on hind-casting of historical precipitation data, through the rainfall-runoff model in combination with a to be developed system model, what can be said about the new system.

1. What is the relative contribution of the sub-basins to the total flow?
2. What is the impact of the reservoirs, in terms of water utilization and dependence on water from the Milluni basin?

1.5. DOCUMENT STRUCTURE

Including this introductory chapter, the research is divided into the following chapters:

2. Materials
3. Area & System Description
4. Methods
5. Results
6. Discussion
7. Conclusions and Recommendations

REMARKS ON NAMING

Please note that *Kaluyo* and *Choqueyapu* refers to the same basin: the basin under study. In principal *Kaluyo* is used, sometimes figures or graphs use *Choqueyapu*. Furthermore, *La Paz* is used as pars pro toto for the greater municipality of *La Paz - El Alto*. Also Riparian, Hillslope(s), Moraine or Glacier with a capital letter is used to refer to the HRUs.

2

MATERIALS

2.1. DATA

2.1.1. DATA SOURCES

This research combines both publicly available- as well as private data sources.

Table 2.1: List of weather measurement stations.

Name	ID	Source	Temporal Resoultion	Period	Observations (Missing)
Alto Achachicala	AAC	SENHAMHI	day	1991 - 2017	9686 (53)
Chuquiaguillo	CHU	SENHAMHI	day	1980 - 2017	13747 (10)
San Calixto	SCA	SENHAMHI	day	1980 - 2017	12901 (856)
El Alto Aeropuerto	EAA	SENHAMHI	day	1980 - 2017	13685 (11)
El Tejar	ELT	SENHAMHI	day	1983 - 2017	12595 (67)
Villa Copacabana	VCO	SENHAMHI	day	1980 - 2017	13745 (13)
Vino Tinto	VTI	SENHAMHI	day	1975 - 2012	13531 (107)

Table 2.2: List of Satellite products used.

Product				Observational Information				
Variable	Owner	Satellite	Productcode	Resolution		Interval	Period	
				°	*m*	*d*	*Begin*	*End*
NDSI	NASA	MODIS	MOD13Q1 V006		0.5	1	1 Sep 2000	31 Aug 2017
NDVI	NASA	MODIS	MOD10A1 V006		0.25	16	18 Feb 2000	29 Aug 2017
DEM	NASA	STRM	STRM V04		30	-	-	-
Precipitation	NASA	TRMM		0.25		0.125	1 Sep 1998	31 Aug 2017

2.1.2. VARIABLE DEFINITION

In this section definitions of the variables used in this study are given. When considered necessary some theoretical background is provided and if the data has been manipulated this is discussed.

RIVER DISCHARGE AND STAGE

In the context of water management discharge refers to the volume of water flowing through the river at a certain point in space and time. In this report stream-flow, river-flow are used are also used. Discharge in the Kaluyo is measured at one location, several hundred meters before the Achachicala WTP intake. Q is used as

symbol to denote stream-flow. Stage (h)is the water level with respect to the river bottom.

$$Q = \text{Discharge } (L^3 T^{-1})$$
$$h = \text{Stage } (L)$$

Figure 2.1: Two problems related to discharge and its data. The photo on the left shows an illegal intake, located before the measurement location. The photo on the right shows the stage scale, on which the scale has faded. Both images are made- and provided by Harm G. Nomden, photos are taken in 2017.

There are several data-quality issues with respect to the discharge. In Figure 2.1 two are shown explicitly, the third implicitly. The exact volume of water is hard to determine. Both in terms of the precision as well as the accuracy of the measurements. Accuracy relates to how close the measured value is to the true value, precision is how precise the method of measurement, i.e. how big the spread (or error) is of two measurements of the same true value. When observing the right image of Figure 2.1, one can see that the scale bars have faded. So it is hard to precisely tell what the stage of the river is. There are far better and more accurate ways to determine the stage.
The measured stage is subsequently translated to a discharge through a *Q-h rating curve*. In which stage is related to discharge. Here the river profile plays an important factor, a profile that is constantly changing in time. Therefore, discharge-stage relationships are in fact always off.
Another issue is shown on the left photo of Figure 2.1. It shows an illegal (or at least ungauged) intake. Where the river has been diverted for the personal use of water. The main problem with this is that the measured discharge downstream, is not the same as the total run-off of the catchment. Therefore, introducing a discrepancy between true- and measured discharge.

PRECIPITATION

Precipitation is *"water that falls from the clouds towards the ground, especially as rain or snow"* [9]. Precipitation is a flux with P as symbol.

$$P = \text{Precipitation } (L^3 T^{-1})$$

Precipitation in highly variable; both in time as well as in space. This is evenmore the case in mountainous areas with complex and pronounced topography. Therefore, ideally one would use spatially distributed remote-sensed precipitation products or multiple ground-measurement stations as input for the hydrological model. Several satelite products were tried, but none was considered useful. Either the spatial resolution was too large (TRMM) or the values of the product were not reasonable (PERSIANN-CSS). The provided data of the latter was more than twice what was to be expected. Although higher than of ground-stations nearby, the TRMM values seemed reasonable. However the gridcells extended over the northern bounds of the catchment. This bound is also the moisture divide between the arid Altiplano and the humid Amazonia. Measured precipitation at this side is often twice as high as within the catchment, therefore decided to use precipitation

from ground measurements.

These measurements are from SENHAMHI. There are some quality issues concerning this data however. Hunziker [10] has investigated the data quality of measurement network of SENHAMHI and concluded that Bolivian precipitation records suffer of three types of quality issues:

1. Heavy precipitation truncation;
2. Small precipitation gaps;
3. Weekly precipitation cycles (approx. 15%).

The first two categories are influencing the total measured precipitation, the third to a lesser extent. The first error category is in essence a deficiency of rain events with an intensities above a certain threshold. In this case 35% of the SENHAMHI stations' daily records showed a temporary lack of measurements above the 20 mm threshold. This error can often be attributed to the measurement equipment. The second error are gaps in small precipitation amounts. Due to low precision instruments these rain events get accumulated over the following days (approximately 60 to 65% of the stations). Resulting in increased evaporation and lower records. The last error category - weekly precipitation cycles - is the result of observations systematically not being taken on a certain day of the week. The data was inspected with respect to the heavy precipitation truncation and small precipitation gaps and seemed to be present.

Precipitation is often assumed to increase with altitude. In this study a 2% increase is taken, based on an internal RHDHV analysis.

TEMPERATURE

[] There are daily records available of the mean (T_{mean}), maximum (T_{max}) and minimum (T_{min}) temperature, measured at station Alto Achachicala.

$$T = \text{Temperature}(°C)$$

The used temperature lapse rates are based on literature. Temperature measurements from other weather stations around Kaluyo not available on the same temporal resolution. The only potentially useful station - El Alto-Aeroporto - resulted in very high values, up to twice the ones found in literature. Possibly explained by the different circumstances: the airport has a lot of paved surface, low shading, no inclination and natural elements that generally lower temperatures (water bodies, vegetation) are absent. The Airport is also located within the city. The high values might could very well be a result of the aforementioned factors and the presence of a urban heat island.

The used lapse rates come from a detailed study on the temperature lapse rates in Southern Ecuadorian Andes. Besides located at a higher latitude (2.9 °S, compared to 16 °S), the study region is quite similar to the Choqueyapu basin. It is situated on the eastern slope of the Andes, with an altitude range from 2600 to 4200 meters. Furthermore, the study distinguishes the rates for the mean, maximum and minimum temperature separately. Lastly, the lapse rate is derived from temperatures of nine stations all within the same basin [11], under similar conditions. Other studies find somewhat lower lapse rate values, but have less similarities with Kaluyo.

$$\begin{aligned}\gamma_{T,mean} &= 6.88 \cdot 10^{-3} °C m^{-1} \\ \gamma_{T,max} &= 8.80 \cdot 10^{-3} °C m^{-1} \\ \gamma_{T,min} &= 5.48 \cdot 10^{-3} °C m^{-1}\end{aligned}$$

$$\begin{aligned}\gamma_{T,mean} &= \text{Mean daily Temperature lapse rate} °C m^{-1} \\ \gamma_{T,max} &= \text{Maximum daily Temperature lapse rate} °C m^{-1} \\ \gamma_{T,min} &= \text{Minimum daily Temperature lapse rate} °C m^{-1}\end{aligned}$$

EVAPORATION

Evaporation is phase change of a element or compound from liquid to gas. In in report evaporation refers this transition the specific molecule H_2O commonly known by its liquid phase name: water. Evaporation is a flux, similar to precipitation.

In terms of the hydrological cycle, evaporation of water can occur in different manners. The three most important ones are evaporation through vegetation (also known as *transpiration*), interception evaporation and open-water evaporation. The theoretical maximum evaporation is the potential evaporation or E_P. The potential evaporation is calculated based on available energy to evaporate and can only be achieved if there is no shortage of water. Generally speaking (if the E_P is calculated correctly) the actual evaporation (E_A) is lower.

$$\begin{aligned} E &= \text{Evaporation } (L^3T^{-1}) \\ E_P &= \text{Potential Evaporation } (L^3T^{-1}) \\ E_T &= \text{Transpiration Evaporation } (L^3T^{-1}) \\ E_i &= \text{Interception Evaporation } (L^3T^{-1}) \\ E_a &= \text{Actual Evaporation } (L^3T^{-1}) \end{aligned}$$

Evaporation is an extremely important part of the hydrological cycle, but very difficult to capture, measure and calculate accurately. Most measurements or calculations of theoretical evaporation rates have serious shortcomings and error margins. The evaporation data used in this research is derived from measured Solar Irradiance, wind-speed, temperature and the moisture content of the air. At weather station Alto Achachicala SENAMHI measures these variables and subsequently calculates the daily potential evaporation with the Hargreaves equation [12]

$$E_{P,Hg} = 0.0023 * R_A * \Delta_T^{0.5} * (T_{mean} + 17.8)$$

Where:

$$\begin{aligned} E_{P,Hg} &= \text{Hargreaves Potential Evaporation } (L^3T^{-1}) \\ R_A &= \text{Mean extra-terrestrial radiation } (PL^{-2}) \\ \Delta_T &= T_{max} - T_{min} \ (°C) \\ T_{mean} &= \text{Mean daily Temperature}(°C) \end{aligned}$$

DIGITAL ELEVATION MODEL

[] A Digital Elevation Model (hereafter: DEM) is a digital representation of the elevation of the terrain. Several products are available. In this study a DEM model of the NASA is used: The Shuttle Radar Topography Mission at a spatial resolution of 1 arc-second, which is approximately 30 meters. (hereafter: SRTM-30). [13]

NORMALIZED DIFFERENCE VEGETATION INDEX

[] The Normalized Difference Vegetation Index (hereafter: NDVI) is used to distinguish the vegetated from the non-vegetated parts of the basin. The theoretical basis is in essence that different types of surface reflect light in different ways. Analyzing the reflectance of a surface gives therefore information about its buildup. Vegetation has evolved to reflect radiation at the Near Infrared spectral region, while absorbing light in the visible Red spectral region . The ratio between the two provides a measurement of indication on the greenness of the area. The NDVI ranges between -1 and 1 and is given by:

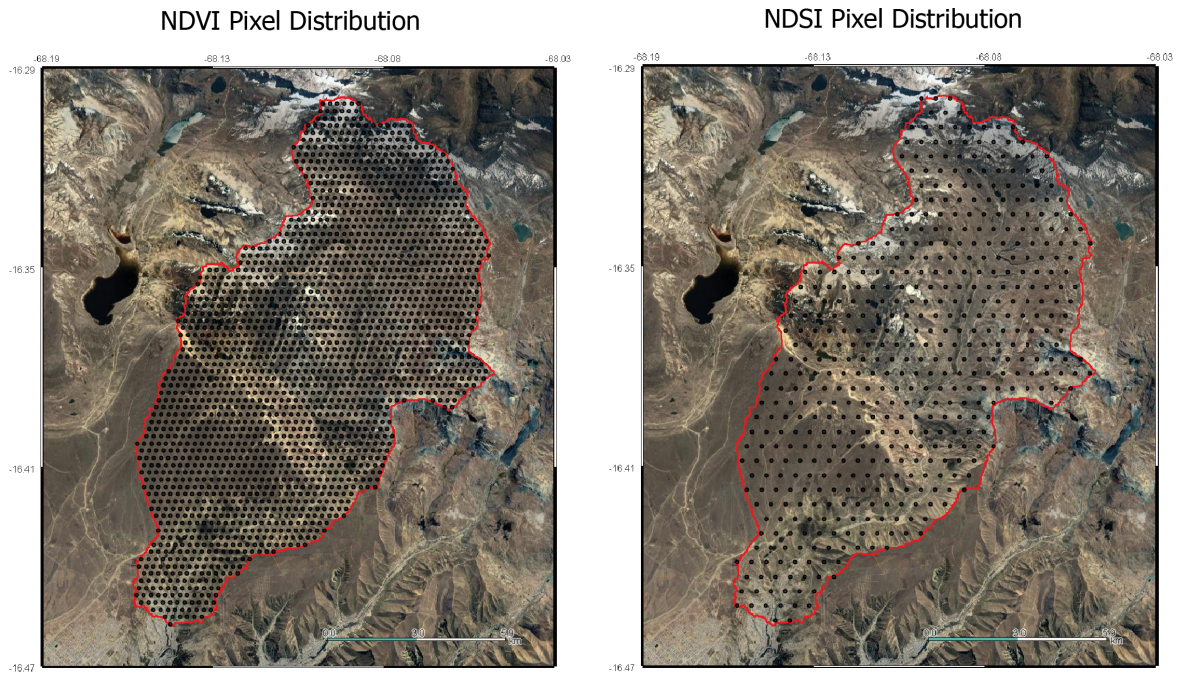


Figure 2.2: Distribution of MODIS NDVI and NDSI observational pixels over the Kaluyo Basin. There are N=2017 NDVI- and N=507 NDSI observational pixels fully or parly within the basin.Data retrieved and maps made with Google Earth Engine.

$$NDVI = \frac{NIR - Red}{NIR + Red}$$

Where:

$NDVI$ = Normalized Difference Vegetation Index (-)
NIR = Near-infrared reflectance (-)
Red = Visible red reflectance (-)

Although exact values are hard to demarcate, the ranges in which certain surfaces reflect are well defined. Low values (below approximately 0.20-0.25) of NDVI indicate the absence of vegetation, while very high values (above approximately 0.80) indicate presence of snow and ice (as perfectly white surfaces reflect everything, and approach 1). NDVI values in the region of 0.25 - 0.70 suggests the presence of vegetation. Within this range, the higher the value, the *greener* the surface. Therefore, the more densely vegetated and well-evaporating the surface.

Introduce MODIS Several freely accessible satellite products are available. Based on the observational period, spatial resolution and retrievable MODIS. [REF] is chosen.

SOLAR IRRADIANCE & IRRADIATION

Solar Irradiance is the rate at which radiant energy is incident per unit area of the surface. This rate depends on the location on the earth, inclination of the surface, aspect and incidence angle, atmospheric properties, and reflectively (i.e. the albedo constant) of the surface.

G = Solar irradiance (Wm^{-2})
H = Solar Irradiation (Jm^{-2})

H is G integrated over a period of time. Here the SI unit for time - a day - will be used throughout the thesis.

How much energy is received by a surface has influence on the near-surface temperature, available energy for evaporating processes and also the rate of snow ablation.

GLACIER ABLATION, SUBLIMATION & MELT

Sublimation and melt refer to two manners of phase change of a substance in the solid phase. The former is the direct transition from the solid- to the gaseous phase, without passing through the liquid phase. The latter is the transition from the liquid phase.

Ablation often refers to the removal of snow and ice by melting, sublimation or any other processes, from a surface covered by snow or ice. In this report ablation is defined as only referring to melt and sublimation. In this report ablation is change of mass of a glacier as a result of sublimation and melt.

For a number of reasons glacial sublimation is hard to quantify. First of all, the exact amount of precipitation is unknown. Therefore, calculating a simple mass balance immediately is highly uncertain. Although sublimation is not an evaporation process, it is in this report represented as E_s, similar to the evaporation symbols. This is done because it is also an flux that results in water leaving the basin as vapor.

$$A = E_S + M$$

Where:

$$Ab = \text{Ablation}(L^3 T^{-1})$$
$$E_S = \text{Sublimation } (L^3 T^{-1})$$
$$M = \text{Snow and Ice melt } (L^3 T^{-1})$$

Melt and sublimation are expressed in rates of water equivalent (w.e.) , meaning the depth of water that would remain after a complete melting of the ice or snow. Hereafter the w.e. is omitted and volumes of E_S and M are given in mm, but note it refers to mm water equivalent.

For glacierized areas in research shows that the ablation ratio $\frac{melt}{sublimation}$ decreases from $70 mm w.e. d^{-1}$ (w.e. is water equivalent) at the low-albedo glacier terminus (around 3000 m), where almost 100 % of total ablation corresponds to melt, to 5 mm w.e. per day at wind-exposed, strong-radiated sites above 5500 m, where surface sublimation represents more than 75% of total ablation [14].
Estimates of surface sublimation rates are between 1 and 6 mm w.e. per day and represent up to 81% of total ablation at glacierized sites above 5000 m a.s.l [15]. [16] estimated the yearly sublimation of Chacaltaya around 166 $mm y^{-1}$ in 1998. This estimation is based on a water balance, in which the precipitation value is important determinant of the sublimation estimate.

NDSI & SNOW COVER

Shortly explain how NDSI is determined.

The Normalized Difference Snow Index (hereafter: NDSI) is a measure of the MODIS satellite and relates to the presence of snow in a pixel. Similar to the NDVI measurement, the NDSI is based on the reflectance distribution. Snow is highly reflective in the visible spectrum and a very low reflectance of the the Infra-red spectrum.[17]

$$NDSI = \frac{band_4 - band_6}{band_4 + band_6}(-)$$

A NDSI value above 0.0 indicates the presence of snow, the higher the value the higher the snow-cover of the pixel.

ENSO

Whereas weather forecast generally are only reliable on a very short time scale, seasonal forecast can be accurate on much larger timescales. The reason is that seasonal climate variability is driven by far slower reacting systems, compared to those of weather. Atmospheric conditions are an important determinant of weather.

Those conditions are very sensitive to the initial state. The signal of the initial conditions quickly loses its influence. In addition, errors in the description of the initial conditions propagate quickly through the system [18]. Discovery of the importance of initial conditions on error propagation - by Edward Lorenz in the 1960's - founded a completely new branch of mathematics: chaos theory [19]. Opposed to the chaos of the atmosphere, is the slow responding and well predictable behaviour of large scale oceanic systems. These systems react and change much slower. These systems retain information from their initial state much longer and are less sensitive to errors in the initial conditions. Because of the enormous amounts of energy stored and transported by them, their influence on the global climate variability is enormous. Therefore anomalies (i.e. deviation from the long-term mean) in surface temperature of Pacific and Atlantic Oceans, especially around the equator can form a robust basis for predictions about climate variability for the coming months, averaged over a large spatial scale. But forecasting on seasonal time-scales has a low predictability, especially on small spatial scales (in the order of hundreds of kilometres) [18]. Since the study aim of is to use seasonal forecast on a relatively small spatial scale caution is required.

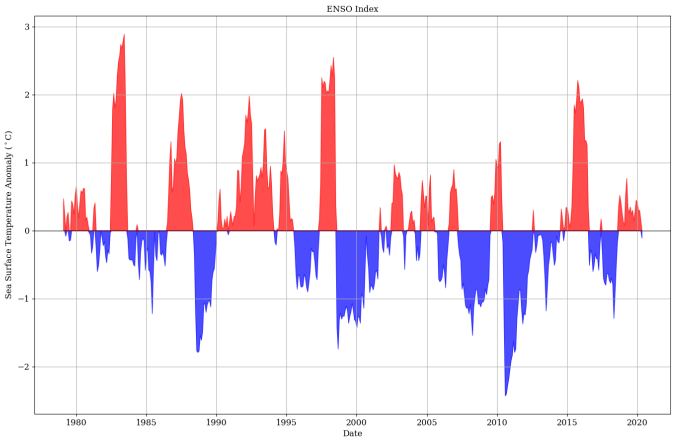


Figure 2.3: The occurrences of warm and cool phases of the ENSO cycle over the last 20 yeas.

To what extent macro-scale systems influence local climate in the Andes and on Altiplano has been topic of extensive research. Most research focuses on the impact of the ENSO cycle. The consensus is that the occurrence of the ENSO warm Phase (El Niño) is often accompanied by dry years (i.e. lower precipitation amounts during the wet season), whereas the ENSO cold phase (La Niña) usually corresponds with wet conditions. However, dry La Niña years and wet El Niño years are not uncommon. [20][21]. There is also consensus that the ENSO-precipitation signal is increasing in strength going from the East to West on the Altiplano [22] . Lastly, [20] found that the most extreme precipitation events seem to be independent on the Southern-Oscillation. This let [23] conclude that the relationship between the sea-surface anomalies in the tropical Pacific (i.e. ENSO) and precipitation anomalies in the central Andes is not straightforward or simple.

On the other hand, although this might be true for the Altiplano and Central Andes region as a whole, it is not necessarily the case for the study region. Precipitation events in the region are generally of convective nature, with a high spacial variability [22]. The predominant and consistent wind directions during El Niño or El Niña events, combined with the topography of the La Paz source region might as well lead to more consistent- and therefore predictable behaviour on local scale. Precipitation in the study region originates mostly in the Amazon forest, therefore it is often hypothesised that in addition to the Sea-Surface Temperature anomalies in the Pacific ocean, Sea-Surface temperature anomalies in the Atlantic ocean might be correlated to precipitation anomalies in the Andes as well.

However, research to this relationship is divided. Where Vuille [20] concluded the relationship was not statistically significant, more recent studies show that there is likely a distinguishable signal. For example [24] showed that Atlantic Sea-Surface Temperatures (SST) anomalies influence the *type and character* of the El Niño/La Niña event. An interesting finding, since dry La Niña and wet El Niño are observed. [25] and [26] both emphasised the influence of the Atlantic Ocean on the South American Monsoon system, which is in turn related to the Bolivian High system (i.e. convective precipitation over the Andes).

For reasons of brevity this study will focus on the relationship between ENSO and the local climate only, and forego a more detailed study to the relationship between ENSO, the Atlantic SST and the local climate.

2.2. PROGRAMS AND TOOLS

2.3 lists the programs that have been used for this research.

Table 2.3: Overview of Open Source computer programs and software that have been used.

Open Source Software	Version(s)	Reference
QGIS	2.18.28	
HDF5 Viewer	3.1.0	
PyCharm Community Edition 2019	3.3	
Google Earth Engine	n/a	
Google Earth Pro	7.1.8.3036 (32-bit)	
Python	3.7.6	
Overleaf	n/a	

Tables 2.4 and 2.5 show the Python Packages and QGIS Plugins.

Table 2.4: List of open source Python Packages used.

Python Packages	Version	Reference
earthengine-api	0.1.217	
earthpy	0.8.0	
fiona	1.8.11	
gdal	3.0.2	
geedataextract	0.0.3	
geojson	2.5.0	
geopandas	0.6.1	
geos	3.8.0	
geotiff	1.5.1	
google-api-core	1.14.3	
h5py	2.9.0	
hdf5	1.10.4	
imageio	2.6.1	
matplotlib	3.1.2	
netcdf4	1.5.3	
numba	0.46.0	
numexpr	2.7.1	
numpy	1.18.1	
pandas	1.0.3	
plotly	4.6.0	
proj	6.2.1	
pyparsing	2.4.6	
pyproj	2.4.1	
pysocks	1.7.1	
pytables	3.6.1	
rasterio	1.1.0	
scikit-learn	0.21.3	
scipy	1.3.1	
seaborn	0.10.0	
shapely	1.6.4	
sqlite	3.31.1	
statsmodels	0.11.1	

Table 2.5: QGIS PlugIns.

QGIS Plugins	Version(s)	Reference
Grass	7.6.0	
SAGA GIS	2.3.2	
TauDEM	5.3.7	

3

Area & System Description

3.1. La Paz

La Paz is one of the highest cities in the world and situated on a vast plateau with an average elevation around 3750 m a.s.l: the Altiplano. Which stretches from southern Peru, through Bolivia all the way down to northern Chile and Argentina. Since 1970 the population of La Paz - El Alto has more than tripled: from around 700 thousand inhabitants, to more than 2 million.

La Paz lies at the Eastern edge of the Altiplano, on the foothills of the Cordillera Real mountain range. This part of the Andes - with its peaks reaching well over 6000 m a.s.l. - acts as the natural moisture divide between the wet Amazon rain-forest, and the (semi-)arid Altiplano. Although the South-Early trade winds generally transport moist air towards the Altiplano: from the Amazon, over the Andes, and towards the equator, the majority of the moisture has already precipitated on the windward side of the Andes upon reaching the Altiplano. When atmospheric circulation changes direction, moving from the Pacific towards the Amazon rain forest, moisture transport is also hindered by a mountain ridge: the Cordillera Occidental. The Cordillera Occidental forms the Western Altiplano border on the latitudes of La Paz.

The pronounced topography of the Altiplano and its surrounding massifs, combined with the high latitude (La Paz is located around 16 °South), and high altitudes result in extreme climate differences on the plateau. Although near the Cordillera Occidental it is are cool and relatively humid, most parts are (semi-)arid, and even one of the the driest place on earth (the Atacama Desert) can be found on the Altiplano.

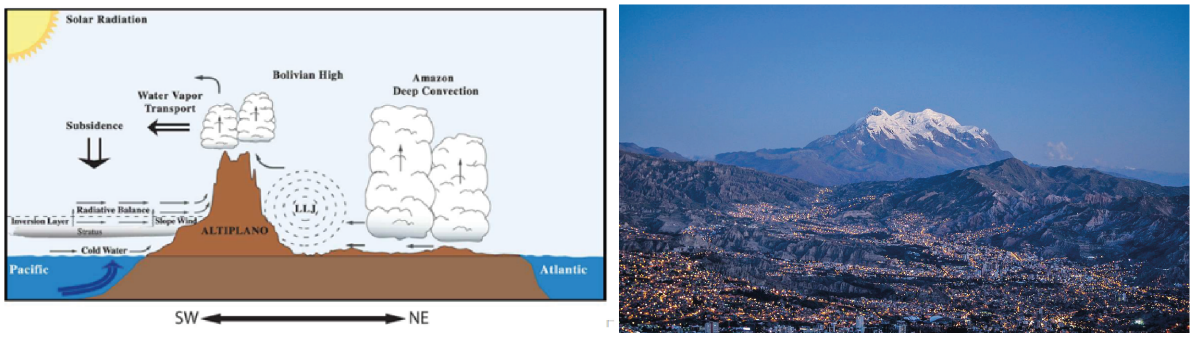


Figure 3.1: Left: Schematized figure of the location of the Altiplano and macro-systems w.r.t. moisture transport over the continent. Figure from [25]. On the right photo of La Paz, with mountain Illumani on the background (Credits photo taken by user EEJCC, distributed under the *Creative Commons* copyright.

The climate of La Paz is also uncommon with a *Polar desert/tundra* climate according to the Köppen-Geiger classification. To fall within this classification the climate must be characterized by low temperatures (mean monthly temperature below 10 °C), semi-arid conditions and high solar irradiation. There is a distinct wet- and dry season.[27] typically, the wet-season is during the austral summer (November to March), while the winter is usually dry (April to October). [28][23]

3.2. THE WATER SUPPLY SYSTEM OF LA PAZ

The city of El Alto/La Paz is dependent on the adjacent Andes massive for it's water supply. Six river catchments supply different parts of the city. To supply water throughout the year the system is dependent on several drinking water reservoirs, because the dry period extents for more than half of the year, During these dry months a significant part of the run-off is contributed to glacier melt water. Estimates from studies allocate between 20 and 30 percent of the dry run-off to glacier.[29][22][30] [4] The WSS is quite complex, with more than 20 reservoirs, some placed in series, some parallel together totalling 70.5 hm^3. Through six drinking water treatment plants the water is distributed over the city. To increase resilience there is a possibility to transfer flows between basins c.q. treatment plants.

Figure 3.2: Left: Water intake at WTP Achachicala of Milluni and Kaluyo (Credits: photo taken and provided by H.G. Nomden). Right: photo of the part of the RHDHV project team (photo taken by author).

The Water Treatment Plant (WTP) Achachicala is supplied by Kaluyo together with Milluni, Figure 3.2 shows the two supply flows. Kaluyo is the primary source. The Milluni supply serves two purposes; to meet the demand if the Kaluyo supply is insufficient, and secondly to reduce treatment costs. The latter is due to heavy metal pollution in Milluni. Mining activities in the past have led to heavy metal and metalloids pollution, such as copper, arsenic and cadmium. [5] [31]. Generally this is undesirable, but whenever appropriately mixed it makes additives in the treatment process less required. WTP Achachicala is currently responsible for 43% of the total water supply for La Paz, see table 3.1. The table shows also the expected increase in demand, from 0.47m^3s^{-1} to 0.58m^3s^{-1} in 2036. This study focuses on the Achachicala system in general, and the Kaluyo basin specifically.

Table 3.1: Demand and capacity of the total system, and the role of WTP Achachicala in it.

System	Capacity	Demand			
	Current (m3/s)	2016 (m3/s)	2018 (m3/s)	2022 (m3/s)	2036 (m3/s)
La Paz	2.5	1.14	1.32	1.41	1.82
El Alto	1.9	1.30	1.43	1.80	2.85
Combined	4.4	2.44	2.75	3.21	4.67
Achachicala (part of La Paz System)	1.0	0.49	0.47	0.49	0.58

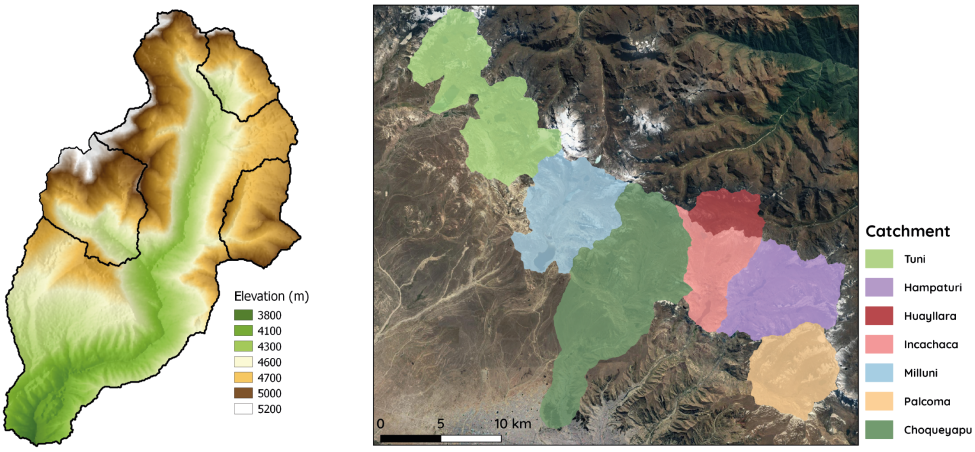


Figure 3.3: Digital Elevation Model of the Kaluyo Basin on the left.The different source regions for the WSS of La Paz shown on the right.

3.3. THE KALUYO BASIN

This section describes in more detail the physical circumstances of the basin that is central in the research. First some information on the man-made infrastructure is provided. Hereafter the geography and climate is discussed.

3.3.1. INFRASTRUCTURE

The Kaluyo river used to run-off freely, but that situation has changed. The construction of the new drinking water reservoirs has led to the formation of new sub-basins. Especially the Pampalarama, Chacaltaya and Al Paquita basins, these basins drain directly into the reservoirs. In addition there is a diversion of the water from Toma Jumalincu towards the basin Pampalarama, see Figure 3.5.

Note that a part of the basin falls within the boundaries of the WTP Achachicala intake drainage basin, but after the discharge measurement station. This sub-basin must be accounted for in the final hinds-cast exercise, but has to be excluded in the development of the hydrological model; the run-off of this part is not gauged.

3.3.2. TOPOGRAPHY, LAND-USE AND SOILS

The Kaluyo basin is located between 16.30° - 16.45° South, and 68.08° - 68.18° East.The total drainage area is 108.12 km^2. The average elevation is 4611 m a.s.l. and ranges from 3809 to 5356 meter a.s.l.. On average the basin is facing west (184 °) and has a slope of 15.8 °.

Table 3.2: The physical sub-basins of Kaluyo, their contribution area and if present resersvoir information. Note that basin Jumalincu drains partly towards reservoir Pampalarama.

Sub-basins			**Reservoir**	
Name	**Area**		**Capacity**	**Surface Area**
	km^2	%	hm^3	km^2
Freely Flowing	72.4	67.0	-	-
Chacaltaya	14.1	13.0	2.7	1.4
Al Paquita	8.5	7.9	1.34	0.16
Pampalarama	6.6	6.1	2.9	0.32
Jumalincu	6.5	6.0	-	-
	108.1	100	6.94	1.88

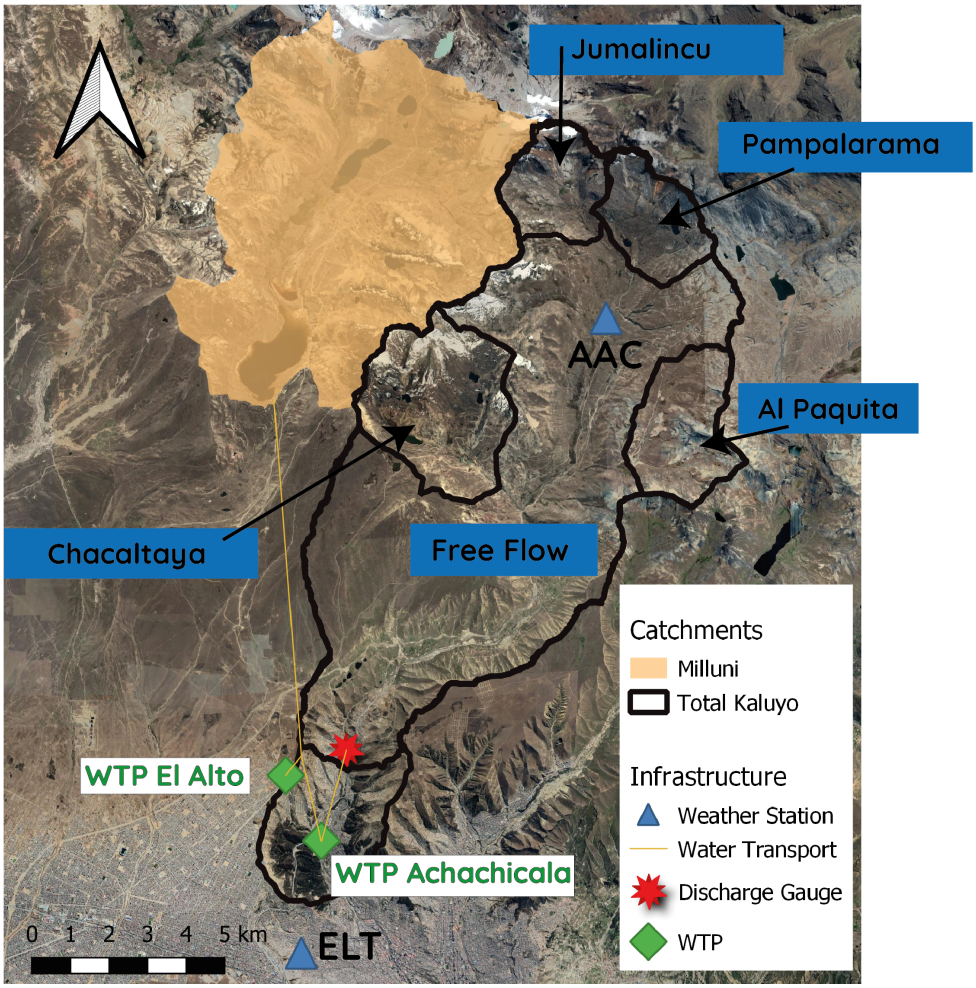


Figure 3.4: Overview of Kaluyo basin, with its most important features and sub-basins for the research.

Figure 3.5: Left: Intake structure to divert water from Toma Jumalincu to Pamalarama reservoir. Right: author (left) with H.Kats and supervisor H.G. Nomden, Kaluyo February 2020.

SOILS

The valley of Kaluyo is small and is fully covered by the *bofedals*, a type of wetland that can be found all over Bolivia. The soil type is peat and the surface is covered with Puna grasses. [32] On these wetlands Alpacas

and Llamas can often be found, as it's typically used for grazing. During a field-visit the author observed that in multiple parts the top-layer of the peat is excavated by the local people.

When moving up-slope the riparian wetlands change to increasingly steep hill slopes. Here also Puna grasses can be found. The lower and more gradual slopes are also used for grazing, but otherwise are not used for economic activities that could be seen. Grasses can grow on altitudes up to 4500 to 5000 meter.[32] At higher altitudes the soils are barren: bedrock and moraine deposits.

Moraine landscapes are generally characterized by high permeable soils. Generally speaking increasing vegetation cover leads to increasing macro-porosity and thus increasing values of the saturated hydraulic conductivity (K_{sat}) and associated infiltration rates. However at moraine landscapes due the absence- or low concentration of organic matter (hereafter: OM) , and the coarse material infiltration rates are very high. OM is hydrophobic and hinders water infiltration. A comparative study to the evolution infiltration capacity of moraine landscapes over time show that recently exposed landscapes, i.e. recent retraction of glaciers, yields soils with very high infiltration capacities: for 30 year old moraine K_{Sat} was on average 4320 $mm\ hr^{-1}$, compared to 540 $mm\ hr^{-1}$ for 30.000 year old moraine [33]. Other studies found similar values [34]. These high infiltration rates do not only apply to the bare-rock moraine landscapes, but also for grass-covered ones. [33] [35] The implication of the extreme infiltration rates is that infiltration excess and saturation excess overland flow are unlikely to occur. Satellite photos show that small ponds are present however. This can be explained by the presence of a bedrock layer at, or just below, the surface.

3.3.3. CLIMATE AND HYDROLOGY

Where the climate on the Altiplano is semi-arid on average, the climate in basin is classified as *"Polar desert/tundra (ET)"* according to the Köppen-Geiger classification system [27]. To fall within this classification the average of the mean daily temperatures in warmest month, must be between 0 - 10 ° C. Observations confirm that this is true as can be seen in Figure 3.7. In this section the different components of the climate and hydrology are discussed.

PRECIPITATION

Between 1998-2017 the average yearly precipitation was 600 mm, measured at Alto Achachicala. The minimum yearly precipitation was 397 mm and the maximum 955 (2016/2017). The hydrological year starts in September and ends twelve months later on 31 of August. Although interpretations on the hydrological year differ, this period is used most often and will be used in this study as well. During the year the precipitation variability is high, and usually precipitation falls concentrated in the months from November to March, sometimes extending into April.

Compared to other, nearby precipitation stations Alto Achachicala has the lowest minimum (397 mm) median (522 mm) and mean (565 mm) yearly precipitation. It shows a high variability however with a the second highest standard deviation (134 mm) and yearly maximum (955 mm). December, January and February are generally the wettest months.

Of the 9739 observational days at Alto Achachicala, 2649 (or 27%) have registered precipitation above zero, with the smallest registered value of 0.1 mm and the heaviest rain event of 30.8 mm.

Table 3.3: Precipitation Statistics for measurements at Alto Achachicala (1998-2017)

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Mean	122	93	80	38	10	8	6	12	25	40	47	87
Maximum	225	200	250	122	76	47	21	47	85	79	101	158
Minimum	39	36	11	10	0	0	0	0	0	9	6	23

.

On a daily basis, the most extreme precipitation events that are measured 30.8 mm. The probability density function of daily precipitation depths is shown below.

TEMPERATURE

The maximum daily temperatures are relatively constant throughout the year, while the minimum temperature vary a lot more. This could be explained by the fact that the solar radiance Kaluyo is fairly consistent

throughout the year, as a result of it being located in the tropical zone. But during the wet season clouds are more present, preventing the emitted energy from the surface to dissipate into the higher parts of the troposphere or into the stratosphere.

Long term trends in temperature are not that clear. Figure 4.3 shows that trends in yearly- or seasonal temperature are not obviously going up or down.

EVAPORATION

In the austral[1] summer the evaporation is the highest with close to $275mm$ on average, while during the dry period - and winter- the average is 192 mm. With a standard deviation of 43 mm the summer shows also the most variability.

Table 3.4: Statistics of evaporation measured at station Alto Achachicala between 2000 and 2016. Std=Standard Deviation.

	Jan (mm)	**Feb** (mm)	**Mar** (mm)	**Apr** (mm)	**May** (mm)	**Jun** (mm)	**Jul** (mm)	**Aug** (mm)	**Sep** (mm)	**Oct** (mm)	**Nov** (mm)	**Dec** (mm)
Average	93	81	82	71	67	58	61	72	79	94	98	100
Maximum	134	126	101	80	75	64	67	85	101	124	144	162
Minimum	80	69	75	61	57	49	48	67	67	85	84	84
Std	15.1	14.4	6.9	5.4	4.3	3.9	4.9	4.7	7.4	8.9	14.1	19.8

Table 3.5: Seasonal and yearly evaporation statistics measured at Alto Achachicala between 2000-2016. Std=Standard Deviation.

	Jun-Sep	**Sep-Dec**	**Dec-Mar**	**Mar-Jun**	**Year**
	(mm)	(mm)	(mm)	(mm)	(mm)
Average	192	271	275	220	958
Maximum	216	368	397	256	1237
Minimum	165	243	240	195	880
Std	12	29	43	15	81

RIVER RUN-OFF

Upon observing the hydrograph it can be seen that the basin responds fast to rain events [TRUE Or???]. Over the period 1981-2016 the maximum, minimum and average observed yearly discharges where respectively 567 mm (2008), 117 mm (1998) and 256 mm per year. The observed discharge in year of the drought was 185 mm (September 2015 - August 2016). As mentioned in Chapter 2 and illustrated in Figure 2.1, there are some concerns with respect to the discharge measurements. Upon inspection of the hydrograph, precipitation data and Q-h rating curves for the period, it has become obvious that the measured discharges in the wet season of 2008 are not much higher than to be expected. Without immediately removing this period from the time-series, the decision has made to exclude this period from calibration. It would be interesting to see if the model can represent this period reasonably.

On a daily basis the discharge is highly variable, The system seems to respond almost instantly to high precipitation events and through a very fast recession goes back to the base-flow within days. Figure 3.9 illustrates a clear example of this fast regression. Sub-figure C shows the hydrograph on a daily scale. On the 2^{nd} of December the mean discharge is almost $5m^3s^{-1}$, and within a day it goes back to less than $1.0m^3s^{-1}$.

RUN-OFF COEFFICIENT AND THE BUDYKO CURVE

Figure 3.10 shows the monthly *RC* values based on 25 years of data. It can be seen that during the first months of the wet-season (November and December) generally the RC are lower than in the months thereafter. January up to and including may show a wide spread, but also high median RCs. One explanation could be that unsaturated zones are depleted during the dry months, that are filled up during the first rains of the season. Another explanation could be an underestimation of the precipitation, or over estimation of high-flows.

[1] Austral refers to the Southern Hemisphere. The austral seasons are exactly opposite to the seasons on the Northern Hemisphere. Such that the austral summer is during the months December to March, while the winter is from June to September

In general, one would expect lower runoff coefficients based on the potential evaporation and precipitation depth. A famous graph relating the energy, precipitation and expected discharge to each-other is the Budyko-curve, by the famous Russian climatologist Mikhail Budyko. The Budyko curve has an iconic status in hydrology science for its accurate representation of the relationship between annual transpiration and water- and energy balances on long temporal- and at the catchment spatial scale. [36] It is often used as indication of transpiration or discharge when data is lacking. [37] This theoretical curve has been validated by empirical data of hundreds of basins all over the world. With a few exceptions, all basins plots within the theoretical lower- and upper-limit. The aridity index of the basin is 1.58 (-) based on the measurements at Alto Achachicala.

$$Aridity_{Index} = \frac{E_p}{P}(-)$$

The location of Choqueyapu on the Budyko-curve could indicate an under-estimation of precipitation. Considering the arid-climate and the empirical lower-bound above which almost every basin in the world plots. When the location is calculated based on the TRMM data ($786mm \cdot a^{-1}$, instead of $P = 585mm \cdot a^{-1}$), the basin plots closer to the theoretical position one would except. This could support this statement.

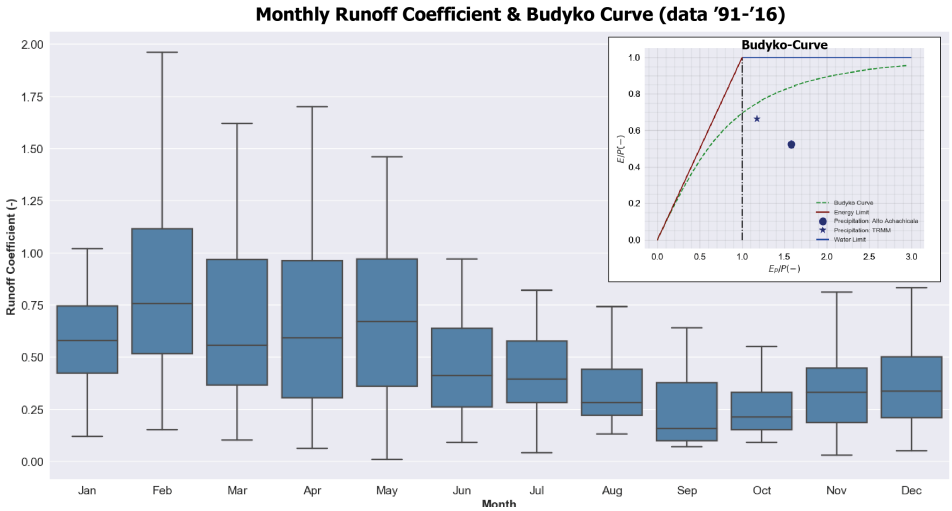


Figure 3.10: Monthly Runoff Coeffiecients based for the Kaluyo basin, based on precipitation measurements at station Alto Achachicala (1991-2016) and discharge data at Achachicala.

However, all precipitation stations in the vicinity of Alto Achachicala measure precipitation depths in the order of Alto Achachicala, and not close to the TRMM data. In addition, the TRMM pixel size is quite large, and extends over the Andes moisture divide. On the other side thereof precipitation is much higher. The high TRMM values could be explained by the average of the two sides of the mountain ridge.

That would mean that the odd location of the Budyko-curve must be explained by something else, for example unusual local conditions; high irradiance and low temperatures, in combination with concentrated rainfall, steep topography, high permeable grounds and only low density vegetation cover.

GLACIERS RETREAT

All across the western Andes glaciers have been retreating over the last century. This regional trend has been true for La Paz water supply catchments as well. Between 1975 and 2006 the glaciers in Tuni-Condoriri and Milluni decreased approximately 50% [38]. In basin Choqueyapu the retreat was even more profound. Once glacier Chacaltaya was the highest ski-piste in the world, and the largest permanent snow mass in the catchment. While covering more than $0.5km^2$ around 1860, totalling an estimated volume of 29.95 hm^3, in 1998 the glacier had been decimated. Just 0.06 km^2 and 0.37 hm^3 remained, a reduction of respectively 88% and 99% [16]. In 2009 the glacier was completely gone [29]. Near the north basin boundary three permanent snow masses remain, but are quite small in size.

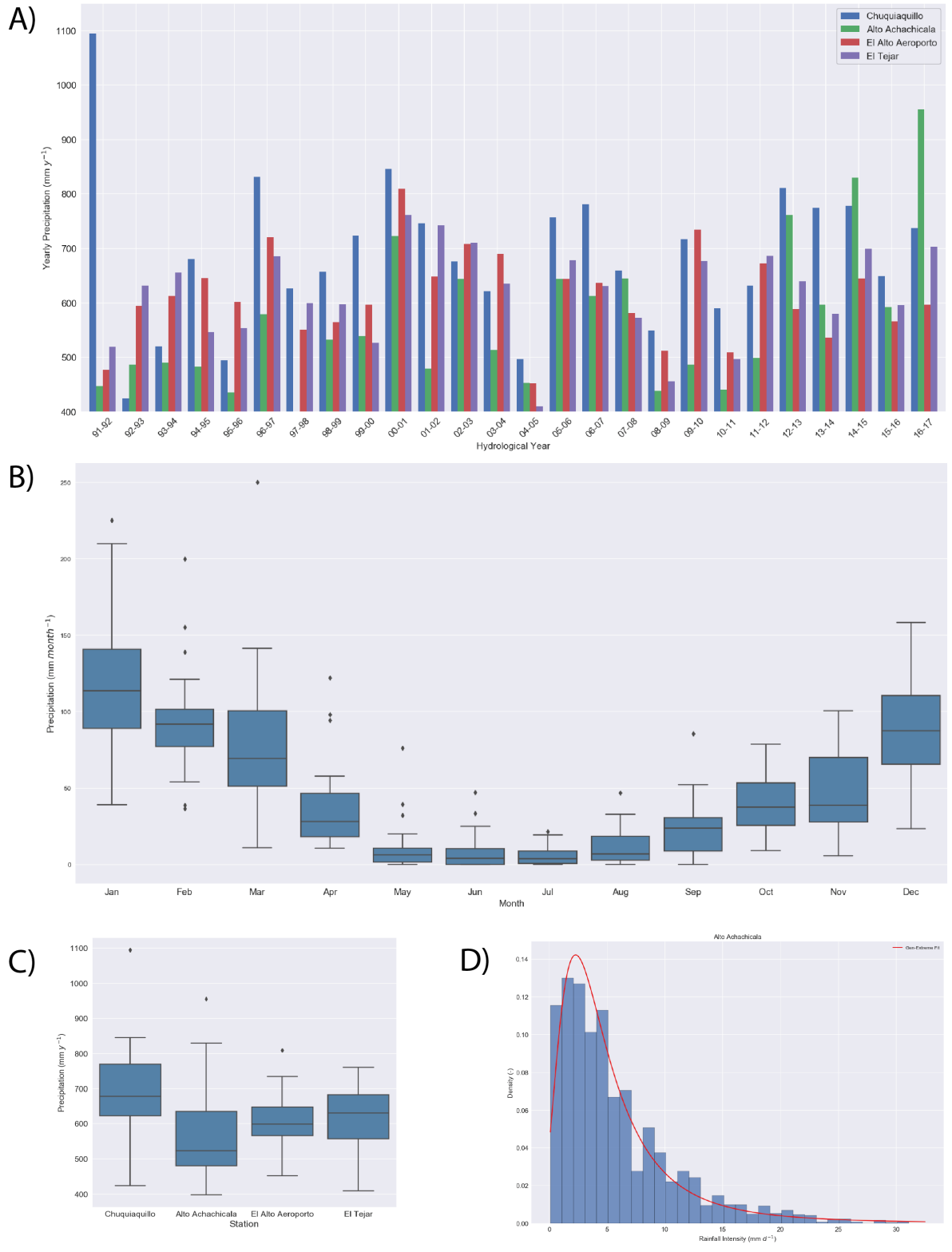


Figure 3.6: **A.** Yearly time-series cumulative precipitation at four stations in and around Basin Kaluyo. **B.** Box-plot of monthly precipitation variability between 1991-2016 at station Alto Achachicala. **C.** Variability of yearly precipitation depth. **D.** Precipitation depth of days on precipitation days, at Alto Achachicala with a generalized Pareto distribution fitted.

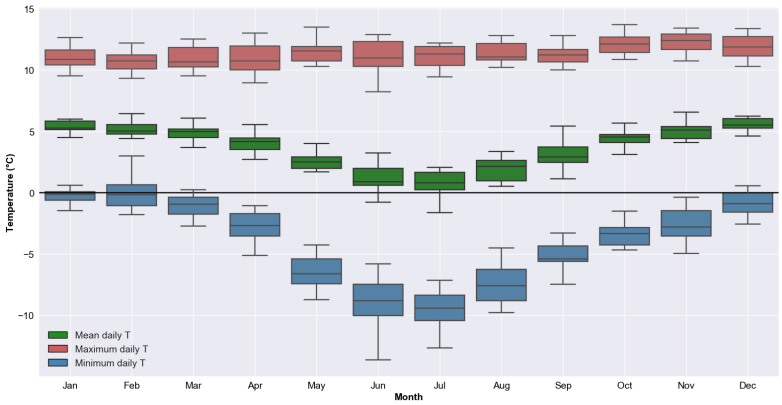


Figure 3.7: Boxplots of the monthly variability of the daily temperatures, measured at Alto Achachicala. It can be seen that the mean daily temperature does not exceed 10 °C on a monthly basis, consistent with the climate classification

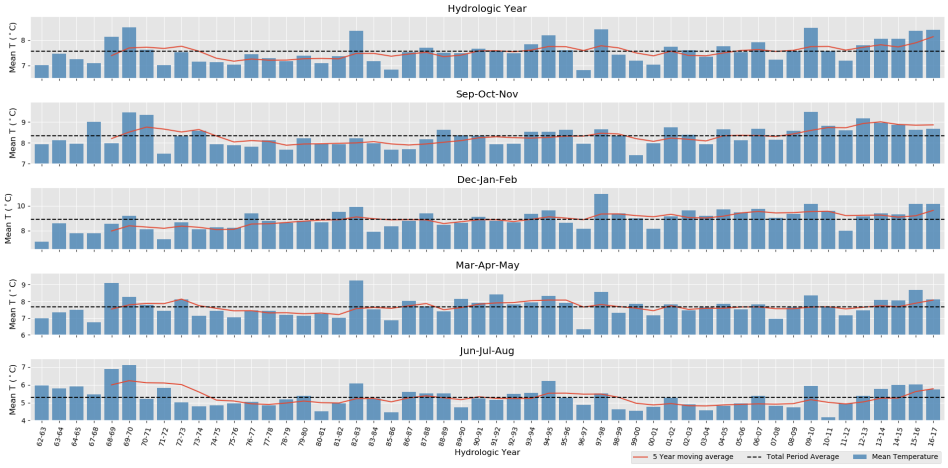


Figure 3.8: Measured Temperature at El Alto Aearopuerto over the years.

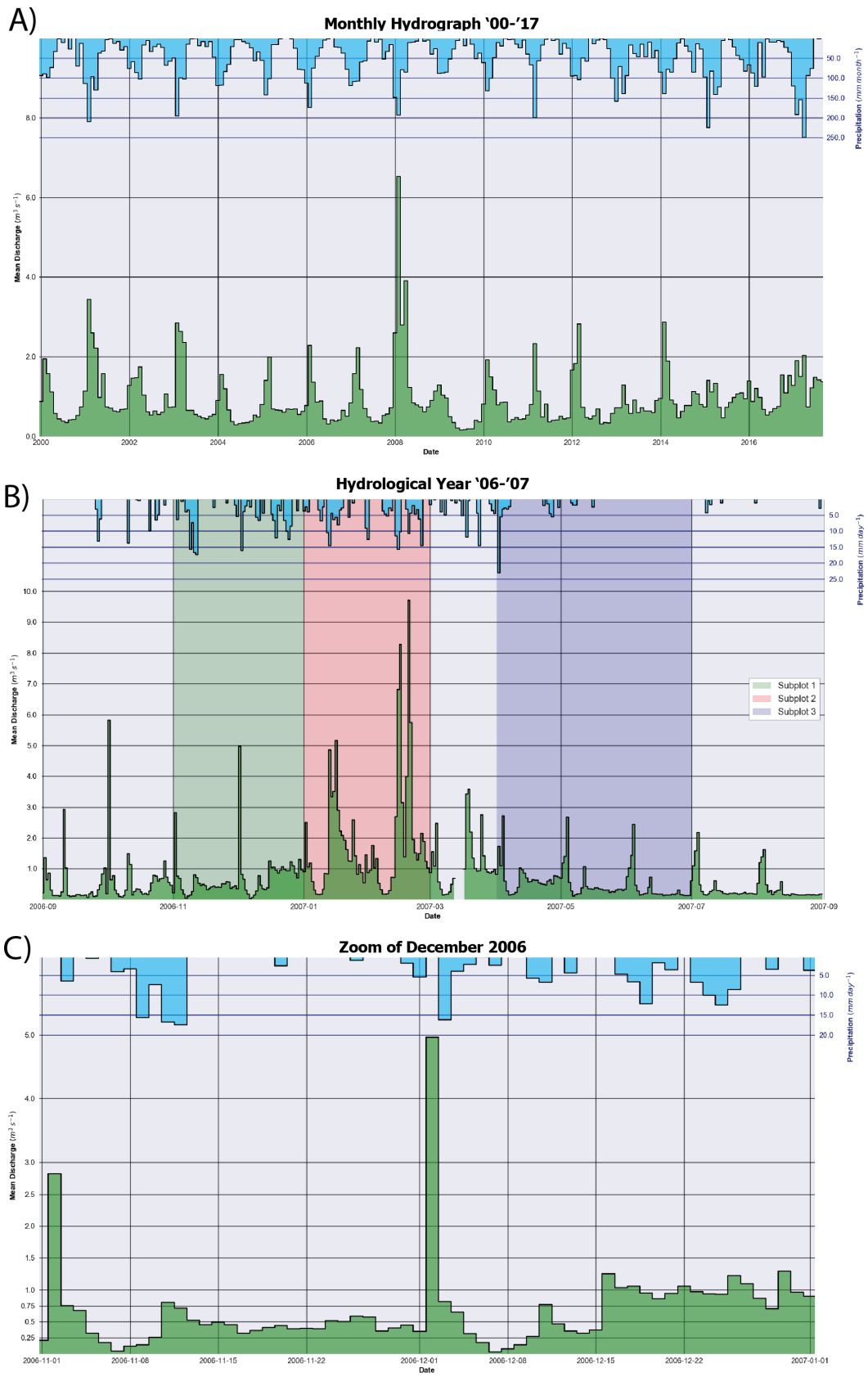


Figure 3.9: **A.**Monthly Discharges and Precipitation for the period from 2000 tot 2017. **B.** Daily discharge of the Hydrological Year 2006. **C.** Close-up of December 2006 to illustrate the fast run-off regression.

4

METHOD

This chapter is divided into four sections; each covering one of the method components leading to the end result. The first section describes in detail how the rainfall-runoff model is developed, calibrated and evaluated. The second part tells how the ENSO hypothesis are tested. The third is about the way the physical elements of the Achachicala Supply system are captured and the last part entails how the individual parts will work together to arrive at the end result.

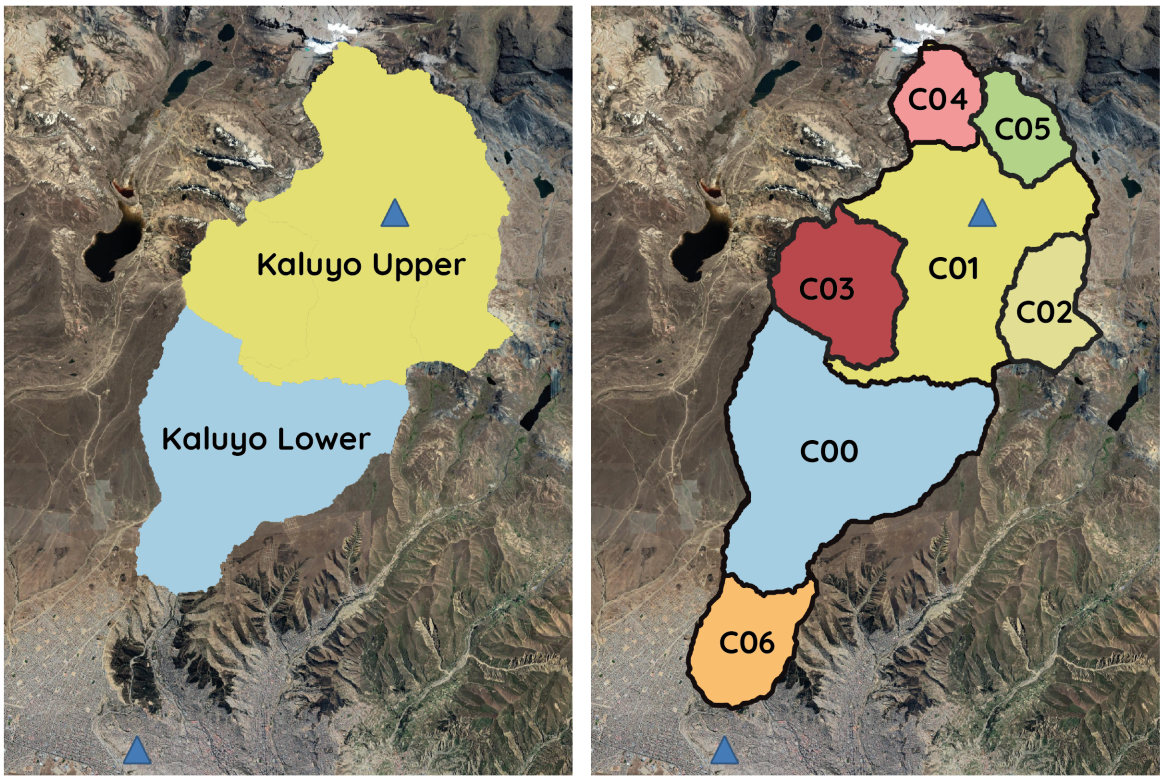


Figure 4.1: Different basin distributions that have been used. On the left the sub-basins that have been applied to the FLEX-Topo model development and calibration phase. On the right the basins that have been used for the final hind-cast.

4.1. HYDROLOGICAL MODEL

The Rainfall-Runoff model forms the backbone of the research and therefore receives a lot of attention. A river catchment is a complex system that could be considered a black box. It is internally heterogeneous and internal fluxes, -states, and properties can often not be observed, or only on small scales. The model choice is a subjective process in which the modelers preferences are an important factor along with data-availability and the modeling goals. A model is a way to capture- and simply reality. Different types of models exist (e.g. scale models or graphical models) but in this report a *Model* refers to a mathematical representation of the perceived reality, captured in computer code. Within this definition, the Hydrological model is a mathematical representation of the movement of water in time and space. Although other classifications are used, three main categories are lumped-, distributed and data-driven models are often considered.

Lumped models treat the whole basin as a homogeneous entity. The existing heterogeneous character is hidden in model-parameters. They are capable to reflect emergent behaviour of a catchment with a relatively simple model set-up and limited data requirements. However, they are less capable of reflecting the heterogeneity in the catchment. Fully distributed model, sometimes referred to as physically based models, can use all available data and capture this heterogeneity. In solving the laws of nature on small scale, they tend have more problem to capture large-scale landscape features and properties (e.g. preferential flow paths) without the danger of over-parametrization. Possibly resulting in parameter equifinality, which is associated with predictive uncertainty [39]. The last category consist of the statistical-, or data driven model concepts. Currently they receive a lot of attention due to increasing computational power, data availability and increasing quality. Novel techniques such as Artificial Neural Networks show promising results and overall good performance. Major weaknesses however are over-fitting, and the large training data requirements [**?**]. Moreover, because they treat the catchment as purely a black-box, and fit a model with its model parameters that do not reflect identify-able physical properties. This research needs to divide the larger basin into smaller- in essence ungauged - basins. This is cannot be done through such methods without the assumption that the response of the basin is scale-able.

Especially in data scarce catchments it is important to utilize all available information: topography, vegetation patterns, landscape and distributed forcing. However it is equally important to reflect the emergent behaviour. Thus one needs a model philosophy that lumps and distributes the model with respect to the appropriate spatial and temporal scales. Savenije [40] argued that a river catchment can be considered as a self-organising system; averaging over the right spatial- and temporal scales, one does not have to know the exact make-up of the system to be able to predict it; as the emergent behaviour of the system is the result of interactions- and symbioses between the local conditions, the observable ecosystem is thus integrated result of the interactions of the local- and boundary conditions. The local conditions are the hydrology, land-use and landscape while the climate and geology are the boundary conditions. It should subsequently be possible to deduce the dominant hydrological processes from observing the ecosystem, land-use, climate, geology and landscape. This let to the development of the FLEX-Topo (FLEX as abbreviation for Flexible, and Topo for Topography) modeling concept. The FLEX-Topo is a topography driven, semi-distributed and conceptual model. [40] [41] [42]. In this research a model according to this philosophy is developed.

In essence a FLEX-Topo model is the combination several separate conceptual rainfall-runoff models. The catchment is divided into classes that are perceived to respond internally consistent to external hydrological forcing: Hydrological Response Units or HRUs. For each HRU the perceived dominant flow processes must be captured in a simple, mathematical representation; of which the main building blocks are fluxes, states and lag-functions. The individual HRU models only *communicate* or interact with each other through the groundwater- or slow reservoir. HRUs can be distributed with respect to forcing data.

4.1.1. DATA REQUIREMENTS AND MANIPULATION

The basin is divided into two parts: *Kaluyo Upper* and *Kaluyo Lower*, see Figure 4.1. The former uses precipitation data measured at station Alto Achachicala (AAC), and the latter of station El Tejar (ELT). Precipitation has been adjusted for each elevation class by multiplication with a lapse rate of 2% increase per 100 meter rise in elevation. For more information refer to Appendix A.

Figure 4.2: The four HRUs illustrated with pictures from the Kaluyo Basin.

4.1.2. MODEL DEVELOPMENT

Five steps of model development can be identified according to Beven [43]. Although one typically starts at the first step and ends at the fifth, the process is not linear but rather circular. Outcomes, and realisations in latter steps may lead to revision of former. The first step is development of The Perceptual Model. *"The summary of our perceptions of how the catchment responds to rainfall under different conditions"*, as Beven describes it. The second step is to translate these perceptions into equations: Conceptual Model. The third step is to program the model into procedural computer code. The last two steps are calibrating and - if possible validating - the model. For a FLEX-Topo model the Perceptual model goes hand in hand with the HRU classification and that is therefore discussed first.

HRU CLASSIFICATION

First basin is analysed as to its physical and ecological make-up, to determine which HRUs are present. Through (satellite) photos, remote-sensed data, and literature (as described in Chapter 3) four HRUs were identified:

1. Riparian: The wetlands - or Bofedahls - that can be found around the river- and streams.
2. Hill-slope: Covered with grass during the wet period. In the dry period the grasses are dormant.
3. Moraine: Non-vegetated mountain-slopes and -peaks. With barren bedrock at the surface or covered by moraine deposits;
4. Glacier: parts of the basin that are permanently covered by ice and snow.

Figure 4.2 illustrates the classified HRUs with photos taken in the Kaluyo basin. It makes clear why it is justified to develop a model structure for each ecosystem separately. Subsequently parts of the basin must be attributed to the HRUs. This classification is performed based on data-sources and thresholds are shown in Figure 4.3.

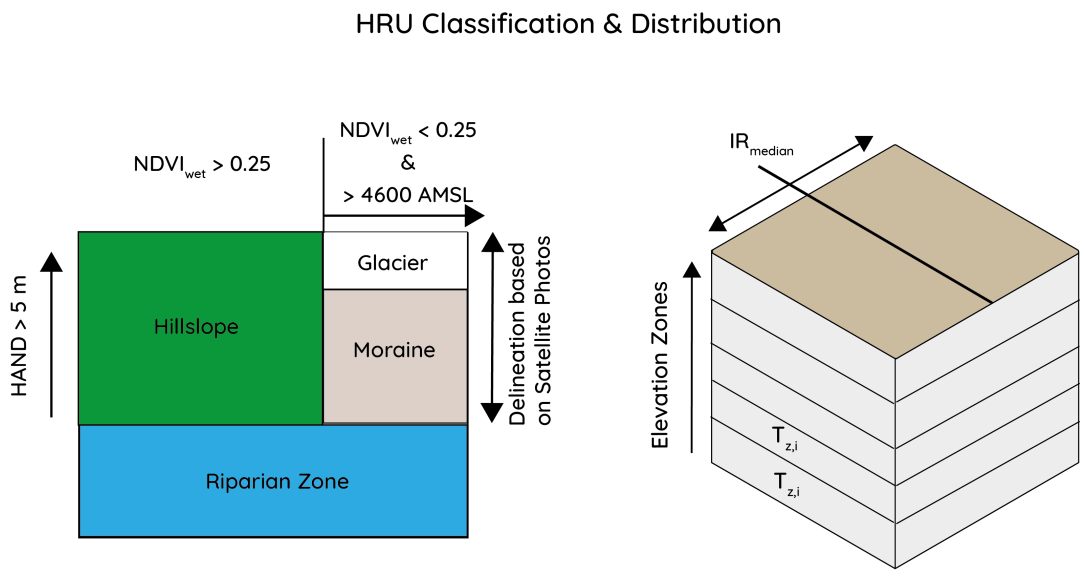


Figure 4.3: HRU classification and internal distribution: classification thresholds

The first classification is made based on the stream-network in the basin. The streams are drawn based on satellite images and subsequently burned into the DEM. With the *TauDEM* tool *Infinity Distance Down* the vertical distance is calculated between every pixel of the DEM and the Kaluyo stream network. If this vertical distance is smaller or equal to 5.0 m, then the pixel is considered to be within the Riparian zone. The vertical distance to the stream is also known as Height Above Nearest Drain (HAND). . The chosen threshold is in accordance with literature; where values between 5 and 10 meters are mentioned. In the Choqueyapu basin the slopes are steep and the difference between a HAND of 5- or 10 meters was small in terms of Riparian HRU size.

The second step is the delineation of the Glaciers based on satellite images. Delineation based on MODIS NDSI data was not opportune due to the relative large MODIS pixel sizes with respect to the glacier size.

All remaining non classified pixels are divided between the Moraine and Hillslope HRU based on the application of two thresholds simultaneously:

1. $\overline{NDVI}_{Wet} = 0.25$; and
2. Elevation = 4600.

The NDVI threshold is based on literature. As mentioned in chapter 2, NDVI values above around 0.25 are considered to correspond to vegetated areas. It is important to consider the characteristic vegetation dynamics of the basin for the demarcation between the grass Hillslope- and the barren Moraine HRU; for a substantial period every year is the vegetation absent the Hill-slopes. In this period both the Moraine and the Hillslope HRU might have a NDVI corresponding to the absence of vegetation. Thus the NDVI threshold must be applied to an observational period in which this is not likely the case. The NDVI observational interval is 16 days. To account for observational error the mean value for the 4 observations in February and March of 2001 is calculated. The assumption is that if a pixel does not exceed a NDVI value of 0.25 at the end of the wet season, then the pixel will not have vegetation in the rest of the year as well.

$$Pixel = \begin{cases} \text{Vegetated}, & \text{if } \overline{NDVI}_{Wet} \geq 0.25 \\ \text{Not Vegetated}, & otherwise. \end{cases}$$

Where $\overline{NDVI}_{Wet}$ is the mean NDVI value of the observations during February and March of 2001. The year 2001 has been chosen because its both the first full wet season observation of MODIS in the catchment, as well as at the start of the simulation period.

The last threshold is the elevation threshold of 4600 meters. This is based on satellite imagery and literature together. As mentioned in the previous chapter, Puna grasses can grow on altitudes up to 4500 to 5000 meter.Houston2009 Because some lonely pixels were labeled as *Not Vegetated* in the lower parts and valley of the basin. Inspection indicated that these were likely man-made structures such as houses or roads. To make sure that these were not assigned to the Moraine HRU the elevation threshold was applied.

INTERNAL HRU DISTRIBUTIONS

The HRU distribution breaks the basin down according to their hydrological functioning. The next step is to internally distribute these lumped sections further. Chosen is to apply two additional distributions, based on their perceived importance:

1. Elevation; and
2. Solar Irradiance.

The reason to internally distribute the HRU is mainly to be able to distribute the forcing data of the model, whereas the HRU distribution is made to distinguish zones based on their functioning independent of the forcing.

Based on the DEM the HRU are divided into seven elevation zones of 250 m. It is common to apply lapse rates on precipitation data and temperature. Often by assuming that temperature linearly decreases with elevation, and precipitation increases with increasing elevation. Measured temperatures are adjusted based on the median elevation of the elevation zone, the height at the measurement location and the temperature lapse rates. Precipitation is also adjusted for each elevation zone.

The solar energy influx variability is considered to be significant as result of the complex geography. The high peaks and ridges result in shading of nearby areas; leading to reduced solar irradiance influx.

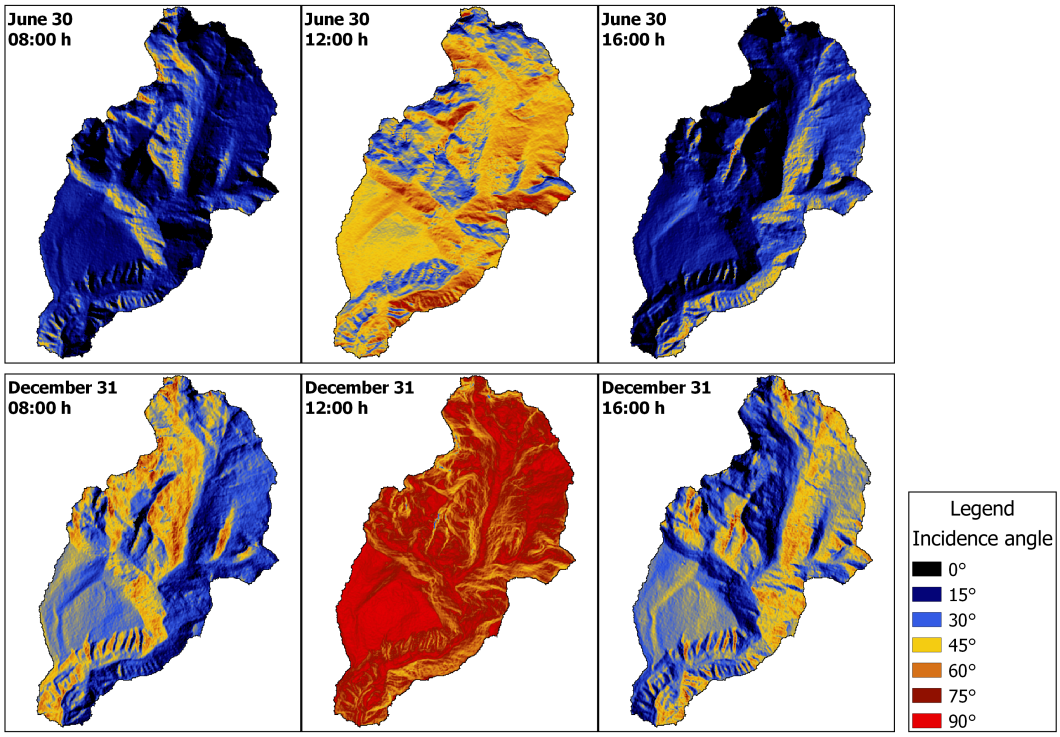


Figure 4.4: Incidence Angle throughout two the day. Comparison June and Decemeber.

With the GRASS GIS tool *r.sun*, can calculate total daily irradiance, in combination with a DEM and information about the year and day. Although more information on the atmospheric conditions can be specified,

the standard values are used. Namely because the calculated irradiance is used to scale the measured evaporation at the weather station, to the rest of the basin. The atmospheric conditions of the measurement location are included in this evaporation measurement. The implicit assumption is thus that the atmospheric conditions in the basin are equal or similar to those at the station. For four normative days within the year (at the end of the seasons in March, June, September and December) the total daily irradiance is calculated. Classes are based on shading and total irradiance on 31st of March. And subsequently the the calulated values are interpolated through the year between the four seasons. Because median value for elevation zones in the HRU is taken, 50% of the zone falls in the low irradiance class (IR_{Low}) and 50 % in the high (IR_{High}).

The factor with which E_p and the sublimation E_s is adjusted, at time-step t, for HRU i and shading class j is calculated by dividing the mean calculated irradiance for every cell in that zone by the calculated irradiance at Alto Achachicala. Such that:

$$f_{IR,t,i,j} = \frac{\overline{IR}_{t,i,j}}{IR_{t,AA}}(-) \tag{4.1}$$

4.1.3. PERCEPTUAL AND CONCEPTUAL MODEL

The result of the identification of the dominant flow processes is captured in Figure 4.5. In this section the governing equations and reasons behind the conceptualization are set out.

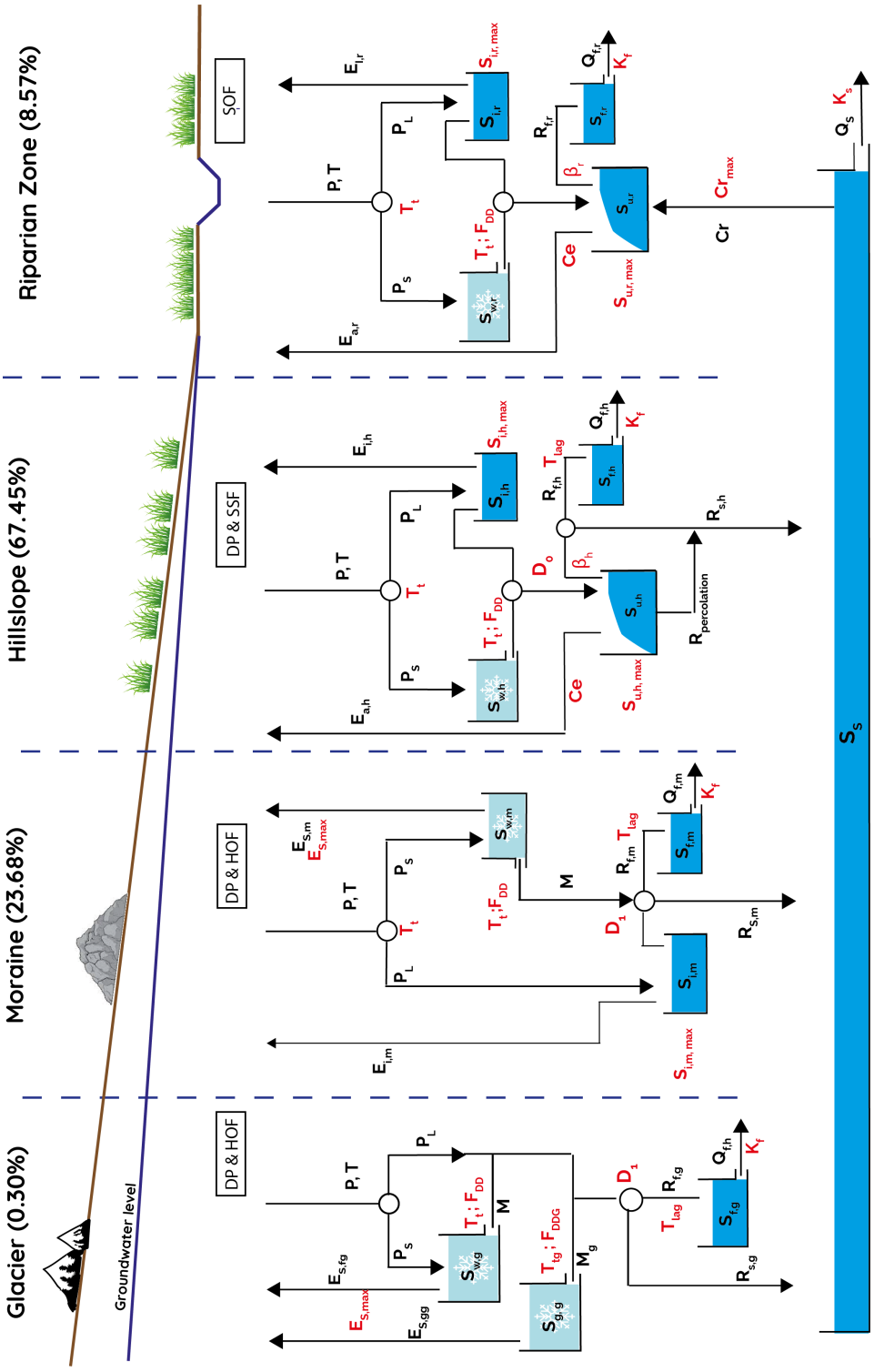


Figure 4.5: Model set-up. Parameters are shown in red, fluxes and states in black.

The model is basically a combination of states, fluxes, transfer functions, closure relations and parameters.[REFERENCE READER HYDRO MOD] States are given by the letter S, with their sub-script referring to which specific state it is. The states of the model are conceptualized as reservoirs, or buckets. Each bucket

drains and fill according to its perceived functioning. Some drain when the volume reaches a certain level, or *threshold*. Others buckets regress based on their relative or absolute volumes c.q. state.

DISCHARGE

As stated the FLEX-Topo model is i.e. the combination of four separate models. The only way they *communicate* is through the groundwater pushing up in the riparian zone. Each HRU has its own fast runoff. In addition all HRUs have a slow runoff component, with the exception of $HRU_{Riparian}$ as this HRU receives water from the slow reservoir. The total modelled discharge the sum of the fast HRU runoff, plus the slow runoff.
Given by the constructive equations:

$$Q_{tot} = Q_f + Q_s \tag{4.2}$$

$$Q_s = \frac{S_s}{K_s} \tag{4.3}$$

$$Q_f = \sum \left(f_{(A,HRU)} \cdot Q_{f,hru} \right) \tag{4.4}$$

$$Q_{f,hru} = \frac{S_{f,hru}}{K_{f,hru}} \tag{4.5}$$

$$\tag{4.6}$$

The slow and fast reservoirs fill up according to the runoff of each elevation, and irradiance zone in their specific HRU. Thus while each Water Balance Equations:

$$\frac{dS_s}{dt} = \sum (A_{hru} * R_{s,hru}) - Q_s \tag{4.7}$$

$$\frac{dS_{f,HRU}}{dt} = \sum (f_A * R_{(f,HRU,zone_{z,IR})}) - Q_{f,hru} \tag{4.8}$$

$$\tag{4.9}$$

Where:

Q_{tot}	= Total modeled discharge	$(L \cdot T^{-1})$
Q_f	= Fast component of the discharge	$(L \cdot T^{-1})$
Q_s	= Slow component of the discharge	$(L \cdot T^{-1})$
S_s	= Slow Storage volume	(L)
K_s	= Slow volume regression coefficient	(T)
$Q_{(f,HRU)}$	= HRU specific fast discharge	$(L \cdot T^{-1})$
$f_{(A,HRU)}$	= Fraction of HRU area to total area	$(-)$
$Q_{(f,HRU)}$	= HRU specific Fast Reservoir storage	(L)
$K_{(f,HRU)}$	= HRU specific fast reservoir regression coefficient	(T)
$S_{(f,HRU)}$	= HRU specfic fast storage volume	(L)
$zone_{z,IR}$	= Distributed zone in elevation class z and irradiance class IR.	
$f_{A,(z,IR)}$	= Relative area of $zone_{z,IR}$in HRU_h	$(-)$

PRECIPITATION PARTITIONING

Within each HRU the precipitation is partitioned into rain (P_l) and snow (P_s), for corrected temperature T^* above and respectively below zero: $T^* = T - T_t$, where T_t is the temperature threshold. Such that:

$$P = P_l + P_s \tag{4.10}$$

$$P_l = \begin{cases} P, & \text{if } T^* \geq 0 \\ 0, & otherwise \end{cases} \tag{4.11}$$

$$P_s = \begin{cases} P, & \text{if } T^* < 0 \\ 0, & otherwise \end{cases} \tag{4.12}$$

Subsequently, P_s is routed into the snow reservoir. P_l in turn to the interception reservoir, except for the Glacier HRU. This HRU does not have an interception reservoir. As the glacier only consists of snow and ice, rain is perceived to flow either through cracks in the surface towards the subsurface, or flow over the surface to leave the HRU. Therefore liquid precipitation is immediately routed towards the runoff partitioning point D.

SNOW RESERVOIR

The time between falling and ending up in the stream-flow, takes longer for solid precipitation compared to liquid precipitation. Considering the high elevation and regular subzero temperatures in the catchment, a snow reservoir S_w is included in the model.

The model has two different types of S_w set-ups, one for the lower - riparian and hillslope - and one for the higher HRUs (glacier and moraine); the difference is the presence of sublimation. In hydrology a lot is uncertain about fluxes in general, and evaporation specifically, for sublimation this is even more the case. It is undoubtedly there, but its significance is not known. The decision to *include* sublimation for the Glacier and the Moraine, and *exclude* is based on the assumption that the fraction $\frac{sublimation}{melt}$, and therefore its significance increases with altitude. This is based on two observations: **1.** Temperature decreases with altitude, thus the ratio $\frac{P_s}{P_l}$ increases, and **2.** a study into sublimation on a Chilean glacier found increasing sublimation to melt ratios with altitude. [14]. Water Balance for the snow reservoir is then given by:

$$\frac{dS_{w,HRU}}{dt} = \begin{cases} P_s - M & \text{,for } HRU_{Riparian} \text{ or } HRU_{Hillslope} \\ P_s - M - E_s & \text{,for } HRU_{Moraine} \text{ or } HRU_{Glacier} \end{cases} \tag{4.13}$$

Where:

$$E_s = \text{Sublimation } (L \cdot T^{-1}) \tag{4.14}$$

$$M = \text{Melt} \qquad (L \cdot T^{-1}) \tag{4.15}$$

$$\tag{4.16}$$

Because sublimation is mostly influenced by the presence and magnitude of solar energy, it is assumed to be zero on days with precipitation (due to cloud-cover). Furthermore is limited by the maximum possible daily sublimation - expressed through parameter ($E_{(s,max)}$) - and scaled by the irradiance factor of the specific elevation and irradiance zone in which it is present. Naturally, melt and sublimation are constrained by the presence of snow in the reservoir. The equations governing the rates of sublimation and melt are:

$$M = \begin{cases} min(F_{dd} \cdot (T - T_t), S_w) & \text{, if } T > T_t \\ 0 & \text{,} otherwise \end{cases} \tag{4.17}$$

$$E_s = \begin{cases} min(E_{s,max} \cdot f_{IR}, S_w) & \text{, if } P = 0 \\ 0 & \text{,} otherwise \end{cases} \tag{4.18}$$

Where T_t is the temperature threshold ($°C$) and F_{dd} is the *Melt degree day factor* ($LT^{-1}\cdot$) that expresses the melt rate per $°C$) above the threshold.

GLACIER RESERVOIR

A special case of the snow reservoir can be found in the Glacier HRU. It expresses the difference between the properties of glacial ice and fresh snow. The latter is less reflecting and more compressed, therefore it is assumed that water stored here melts and sublimates slower. In the model set-up it does not receive fresh snow from precipitation, and is only activated when the S_w reservoir is empty.

$$\frac{dS_g}{dt} = -M_{glacier} - E_{s,glacier} \tag{4.19}$$

$$\tag{4.20}$$

Where:

$$E_{s,glacier} = \text{Sublimation from reservoir } S_g \quad (L)$$
$$M_{glacier} = \text{Melt from reservoir } S_g \quad (L)$$

$$M_{glacier} = \begin{cases} F_{dd,glacier}(T - T_{t,glacier}) & \text{, if } (T > T_{t,glacier}) \text{ and } S_w = 0 \\ 0 & \text{, otherwise} \end{cases} \tag{4.21}$$

$$E_s = \begin{cases} E_{s,max} * f_{IR} & \text{, if } P = 0 \text{ and } S_w = 0 \\ 0 & \text{, otherwise} \end{cases} \tag{4.22}$$

Please note that melt and sublimation from the glacier reservoir are not limited by the presence of glacial ice in the reservoir. The reservoir is assumed to be unlimited. To ensure that the calibrated parameters will reflect reality, i.e. not a glacier decline that is unreasonably large, the related parameters $T_{t,glacier}$ and $F_{dd,glacier}$ are independently calibrated.

INTERCEPTION RESERVOIR

As mentioned earlier an interception reservoir is present for all HRU's, except the Glacier. Interception is the part of rain that is evaporated before it can enter the ground, or run-off over the ground. It is the part of rain that falls on rocks, leaves, grass or man-made structures. The mass-balance and governing equations are the same for every HRU:

$$\frac{dS_{i,hru}}{dt} = P_l - E_i - R_{inter} \tag{4.23}$$

(4.24)

Where E_i is interception evaporation and R_{inter} is the run-off from the interception reservoir (often expressed as effective precipitation P_e or throughfall). They are given by:

$$E_i = \begin{cases} max(S_i, E_p) & \text{, if } S_i > 0 \\ 0 & \text{, otherwise} \end{cases} \tag{4.25}$$

$$R_{inter} = \begin{cases} 0 & \text{, if } S_i < S_{i,max} \\ max(0, P_l - E_i) & \text{, otherwise} \end{cases} \tag{4.26}$$

(4.27)

UNSATURATED RESERVOIR

An unsaturated reservoir (S_u) is present in the two vegetated HRU classes. This reservoir is fed by the combined fluxes M and R_{inter}. S_u is absent in the Glacier for the same reason that the interception reservoir is absent here: the soil is covered by ice and snow, so no evaporation is possible from here.

The surface of the Moraine however, is not necessarily covered by snow. But without the presence of vegetation, the organic content of the soil here is low to non-existent. Ipso facto, the capacity to hold water against gravity is very low. Furthermore, without vegetation there is no transpiration. The only reason to include the reservoir here would be to account for evaporation from the thin top-layer of the soil. Besides the fact that soil evaporation is generally believed to be small, it can also be accounted for by the interception reservoir, when conceptually the physical interpretation of the interception is broadened. Combining these considerations with the knowledge that the soils are highly permeable and the slopes steep, an unsaturated zone is assumed to be insignificant of the Moraine HRU.

Contrariwise the significance and function of the unsaturated in the riparian zone and on the grass covered hillslopes could not be exaggerated. Not only for the internal HRU functioning, but for the catchment as a whole. From this zone (in or modelling terminology: reservoir) most of the evaporation takes place. Moisture that has passed the interception- or snow reservoir, has *entered* the system. It is stored somewhere in

the soil and from here there are only two possible ways out: as evaporation, or as discharge. Which one of those it is, is controlled by the vegetation. In some sense the only living water manager in the basin. Whereas the snow-, interception-, fast-, and slow reservoirs are basically representations of the movement of water through a abiotic medium, encountering the energy of the sun and the gravity of the earth, but unhindered by a living organism that need water to survive. [1].

The perceived importance of the unsaturated zone is underlined by the attention it receives in calibration. The unsaturated reservoirs $S_{u,r}$ and $S_{u,h}$ have a different setup. On the hillslopes the water-balance of the reservoir is given by:

$$\frac{dS_{u,h}}{dt} = R_{i,h} + M_h - E_{t,h} - R_{u,h} - R_{perc} \tag{4.28}$$

$$\tag{4.29}$$

Where R_{perc} is the percolation from the root-zone towards the slow reservoir.

With the constructive equations:

$$E_{t,h} = min\left((E_p - E_i) \cdot min\left(\frac{S_u(t)}{S_{(u,max)} \cdot C_e}, 1\right), S_u(t)\right) \tag{4.30}$$

$$R_{u,h} = (R_{i,h} + M_h) \cdot \left(1 - \left(\frac{S_{u,h}(t)}{S_{u,h,max}}\right)^{\beta}\right) \tag{4.31}$$

$$R_{perc} = \begin{cases} \frac{S_u(t)}{S_{(u,max,h)} * C_e} & ,\text{if } \frac{S_u(t)}{S_{(u,max,h)} * C_e} > 1 \\ 0 & ,otherwise \end{cases} \tag{4.32}$$

For the riparian zone it is different. Here there is no percolation to the groundwater, but instead the unsaturated zone receives water from it:

$$\frac{dS_{u,t}}{dt} = R_{i,r} + M_r - E_{t,r} + C_r - R_{f,r} \tag{4.33}$$

$$\tag{4.34}$$

With the constructive equations:

$$E_{t,r} = min\left((E_p - E_i) \cdot min\left(\frac{S_u(t)}{S_{(u,max)} \cdot C_e}, 1\right), S_u(t)\right) \tag{4.35}$$

$$R_{u,r} = (R_{i,r} + M_r) \cdot \left(1 - \left(\frac{S_{u,r}(t)}{S_{u,r,max}}\right)^{\beta}\right) \tag{4.36}$$

$$C_r = min\left(C_{r,max} \cdot \left(\frac{S_u(t)}{S_{u,max}}\right), S_s \cdot \frac{A_{Total}}{A_{Rip}}\right) \tag{4.37}$$

$$R_{f,r} = (R_{i,r} + M_r) \cdot \left(1 - \left(\frac{S_u(t)}{S_{u,max}}\right)^{\beta}\right) \tag{4.38}$$

$$\tag{4.39}$$

INTERNAL RUN-OFF PARTITIONING

Internal run-off partitioning is different for every HRU. In case of the riparian HRU it is not present at all. The flux $R_{f,r}$ leaving the unsaturated reservoir is directly and fully routed towards the reservoir $S_{(f,r)}$.

Hillslope

The flux $R_{unsat,h}$ leaving the unsaturated reservoir on the hillslopes arrives at partitioning *point* D_0. Here the internal run-off is routed towards the slow groundwater reservoir or the fast run-off reservoir based on the

[1] The author is aware that there are small animals, insects and decomposers living in the soil. Their influence is small compared to the presence or absence of vegetation however. And in the absence of vegetation, their density is also small.

value of D_0. Such that:

$$\begin{aligned} R_{Unsat,h} &= R_{f,h} + R_{s,h} \\ R_{f,h} &= R_{Unsat,h} \cdot (1.0 - D_0) \\ R_{s,h} &= R_{Unsat,h} \cdot (D_0) \end{aligned}$$

Moraine & Glacier

For $HRU_{Moraine}$ and $HRU_{Glacier}$ the partitioning works similar, but because of the absence of vegetation the internal flux is not coming from the unsaturated reservoir but is melting water and interception run-off. Because there is no interception reservoir in $HRU_{Glacier}$ the liquid precipitation, melt from the glacier reservoir and melt from the fresh snow reservoir come together, such that:

$$\begin{aligned} R_{Glacier} &= Melt_g + M + P_L \\ R_{f,Glacier} &= R_{Glacier} \cdot (1.0 - D_1) \\ R_{s,Glacier} &= R_{Glacier} \cdot (D_1) \end{aligned}$$

and for $HRU_{Moraine}$, where there is an interception reservoir:

$$\begin{aligned} R_{Moraine} &= R_{Inter} + M \\ R_{f,Moraine} &= R_{Moraine} \cdot (1.0 - D_1) \\ R_{s,Moraine} &= R_{Moraine} \cdot (D_1) \end{aligned}$$

PROCEDURAL MODEL

The model will be written in Python, in a functional manner (opposed to Object Oriented Programming). Python is relatively slow compared to other e.g. C (#/++) or FORTRAN, however it is widely used, easy to interpret and already known to the author. In addition there are open-source packages available to increase calculation speed.

4.1.4. CALIBRATION

The developed model structure will contain several free parameters that need to be calibrated, in order to give a reasonable model output. Rainfall-Runoff models are traditionally calibrated on observed stream-flow data. However, as is shown in Chapter 2 the available discharge data is only available on one location, and there are issues with respect to the quality. Calibration on the stream-flow only might not achieve the desired result- and confidence in the result. There are several ways to introduce additional ways of calibration and validating a model. One is to combine "hard" hydrological data (i.e. stream flow) with "soft" data. Gao et al. [41] identified four types of soft data:

1. Explicit knowledge of hydrological processes;
2. Expert knowledge on parameter values;
3. Understanding relative magnitude of specif parameters of your model; and
4. Using auxiliary (unconventional) data sources;

One of the advantages of the FLEX-Topo model structure is that it is suited for calibration on soft data. First of all because the building blocks (i.e. fluxes and states) reflect physical- observable processes. Secondly, because these processes are related to specific place through the HRU distribution of the model. Lastly one must notice that the explicit hydrological processes are at the essence of the model development. Therefore a FLEX-Topo model is enables different calibration and validation techniques.

The calibration methodology is an iterative stochastic automatic, multi-objective calibration; in separate steps N parameter-sets are sampled through Monte-Carlo Sampling. The bigger the N, the more likely a combination is found that approaches the global optimum. The constraining factor, is the computational power. $N = 500 \cdot 10^3$ is chosen, which approached the limit of the available computational resources; in terms of time and disk-space. The multiple objectives of the calibration are to reproduce the behaviour of the basin in terms of:

1. Magnitude and timing of stream-flow;
2. Stream-flow in the frequency domain; and
3. Observable behaviour ecosystem other than discharge to increase reflection of reality.

The first two objectives are met through the use of multiple objective functions relating to the discharge regime. The used objective functions are listed in 4.1.4. The last objective is met by evaluation of internal model states separately, on auxiliary data sources. To be able to effectively evaluate internal model states, they must represent independent physical behaviour. For the Kaluyo catchment three additional *internal* variables will be evaluated:

1. Vegetation Dynamics;
2. Snow-Cover;
3. Glacier retreat.

The method for the internal evaluation is described in the section *Internal Calibration* further on in this chapter. The result of the internal evaluation is to limit the parameter-ranges for the dependent model-parameters, or to fixed them to reasonable values. The internal calibration is seen as an intermediate step. First the model is run for initial parameter distributions. The initial values are derived from literature, expert knowledge and hydrograph inspection. The internal calibration is performed hereafter, for the three internal variables separately. Lastly, again $N = 500 \cdot 10^3$ sets are sampled, on a the final parameter ranges. The results are then evaluated on the discharge regime only. The Euclidean Distance is calculated for the objective functions to determine the best performing sets.

To ensure that the final parameter-sets meet the most important goals of the model; the following sequential selection is applied on the parameter-sets:

1. Cut-off of NSE_{log} of the best 10 % on a monthly basis;
2. Only sets that fall within the best 20% rated on the FDC are kept;
3. All sets with an absolute RVE greater than 10 % are thrown out;

The objective functions scores are normalized between [$-\infty$ and 1], where 1 is the perfect score.The smallest ED represents the parameter-set that scores on average best for the chosen objective functions. In essence it reflects the parameter-set that can mimic the hydrograph on all of its elements the best. The Euclidean distance (ED) is calculated by:

$$ED_p = \sqrt{\sum_{i=1}^{N}((1 - OF_i)^2 \cdot W_i)} \tag{4.40}$$

Where:

ED_p = Euclidean Distance for parameter-set $p(-)$
$OF_{p,i}$ = Objective function score for OF i, and for parameter-set $p(-)$
W_i = Weight of OF_i to adjust for its relative importance$(-)$

Because the smaller the error of each OF, the smaller the Euclidean Distance. So, the sets with the smallest values are performing the best, on average, on all aspects of the flow regime. From the calculated ED the best 100 sets selected for comparison. The procedure is followed for the split-calibration/validation procedure as well as calibration over the whole time-series. Based on the results, the best set of either one of the two periods or of the whole period is used for the hind-cast of the reservoir dynamics.

The following section describes the internal calibration procedures, while the last section describes which objective functions are used and why.

INTERNAL CALIBRATION

[] Opposed to the discharge calibration the internal calibration does not focus on the total emergent behaviour as a result from the interactions of all HRUs, but looks at the dynamics *in* the model; the internal states- and fluxes. The main reasons to include an internal calibration are to increase the reflection of reality, to increase confidence in the final parameter values. The model will consists of several clusters of internal states and fluxes. Such a cluster can only be calibrated separately, when having an independent and clear signal that relates to an observable measured variable other than discharge. Especially interesting are variables that relate to import hydrological processes in the catchment; for example vegetation dynamics and snow cover. In the Kaluyo basin three clusters of internal states and - fluxes are evaluated with additional data sources:

1. Vegetation patterns through NDVI measurements are related to the transpiration in the model;
2. Snow presence measurements are coupled to the fresh-snow reservoirs; and
3. The glacial-ice reservoir with literature studies on glacier retreat.The variables

The used auxiliary data sources used refer to observable, physical processes in the catchment. The internal evaluation ensures that the derived parameter values result in sensible basin behaviour.

Vegetation & NDVI

Arguably the most important of which are the vegetation dynamics. Vegetation patterns, and there influence on the hydrograph are essential for the hydrology of the catchment. Transpiration, or E_T , is often one of the most significant out-going fluxes in the water balance, together with the discharge. Without vegetation there is no transpiration, and without water there is no transpiration. With the long dry season, the grass hill-slopes tend to go dormant for long periods of time. The emergent pattern is observable on the hill slopes of Kaluyo: green during and shortly after the rainy season, while arid and fallow during the dry months. This pattern is the result of the persistence absence of water for several months, when the Puna grasses are forced to go dormant.

This pattern is observable in the measured NDVI-Index, and used to constrain the parameters related to the transpiration. The simulated E_T dynamics are compared with the remote sensed NDVI measurements. To be able to effectively compare them, both time-series are normalized and their respective duration-curves are compared on N evaluation points. Similar to the method of the Flow-Duration Curve evaluation on the discharge, as is shown in 4.6. The assumption is that maximum transpiration coincides with maximum NDVI values. For each the vegetated HRUs, and each IR_{zone} the

$$E'_T = \frac{E_T - E_{T,min}}{E_{T,max} - E_{T,min}}$$

$$NDVI' = \frac{NDVI - NDVI_{min}}{NDVI_{max} - NDVI_{min}}$$

$$OF_{NDVI} = \sqrt{\frac{1}{n_{EP}} * \sum_{n=1}^{N} \left(log_{10}(E'_{T,t}) - \log_{10}(NDVI'_t) \right)^2}$$

Where:

E_T	= Transpiration Evaporation	$(L^3 T^{-1})$
E'_T	= Unity-based normalized image of E_T	$(L^3 T^{-1})$
$NDVI$	= MODIS observed NDVI value	$(-)$
$NDVI'$	= Unity-based normalized image of NDVI	$(-)$
OF_{NDVI}	= Objective Function Score	$(-)$

The OF_{NDVI} is basically a Frequency Duration Curve of the observed measured NDVI and the simulated transpiration. It expresses the fraction of the time certain NDVI values or E_T values are exceeded. Therefore, the OF does not give insight into the precise timing of the time-series. The underlying assumption is that the root-zones in the riparian zone and on the grass hill-slopes have evolved, driven by the local arid-climate, to store as much the available water as is provided but not much more. The latter due to the energy it takes to root-deeper. Assumed is that a too small storage lead to an underestimation of the E_T , while a too large storage lead to an overestimation.

Because there are a five free parameters in this internal calibration, a relatively large Monte-Carlo sample is made, simulated and evaluated: $N = 50 \cdot 10^3$.

Fresh-Snow Reservoirs

The snow reservoirs are compared and evaluated on MODIS Snow data. For each time-step t the presence (True/False) of snow above a threshold is assessed for each HRU and elevation zone separately. The thresholds are included to account for very small snow-depth values in the reservoir and MODIS observations, that could cloud the evaluation. Both thresholds are 0.1. The evaluated score is calculated as:

$$OF_{Snow} = \frac{1}{T}\sum_{t=1}^{T}(score_t)^2$$

$With:$

$$score_t = \begin{cases} 0, & \text{if } SP_{t,sim} = SP_{t,MODIS} \\ 0.25, & \text{if } SP_{t,sim} \neq SP_{t,MODIS} \\ & \text{and } SP_{t,sim} = SP_{t\pm1,MODIS} \\ 1, & otherwise \end{cases}$$

$Where:$

OF_{Snow} = Objective function score (-)

SP_t = Presence of Snow (True / False) on day t;

$score_t$ = Score on day t

The reason to include both an observational bandwidth of $t+/-1day$ is to account for the difference in observational time (around 12 a.m. for MODIS) and the model (11.59 p.m), and from the uncertainty in the day on which the precipitation fell.

Because there are only two free parameters in the fresh snow calibration, a small Monte-Carlo sample is made, simulated and evaluated: $N = 2 \cdot 10^3$.

Glacial-Ice Reservoir

Data from a study by Braun [44] indicated an average snow depth change $\frac{dh}{dt} = -0.234mm\ yr^{-1}$ for the snow covered areas in Choqueyapu (2000-2013). This value is taken as the true value and evenly distributed over the Glacier HRU in space and time. Based on the length of the simulation period, the total snow snow and ice depth decline in $HRU_{Glacier}$ decline must be- or close to:

$$dh = -0.234mm \cdot N_{years}$$

All combinations of parameters that result in either too much glacier decline, or too high increase in snow depth are regarded as unlikely and thrown out. There are four free parameters in the $HRU_{Glacier}$ calibration, a Monte-Carlo sample of $N = 4 \cdot 10^3$ was considered appropriate.

DISCHARGE OBJECTIVE FUNCTIONS

The model's performance is evaluated will a number of objective functions. Each reflecting different aspects of the flow regime. The objective functions are listed and described in this section. In summary six OFs are used to evaluate different run-off dynamics:

1. The timing of the flow: the coefficient of determination (R^2);
2. Low flows: the Nash-Sutcliffe efficiency of the log of the flows;
3. High flows: the Nash-Sutcliffe efficiency of the flows;
4. Frequency Distribution of the flows: Flow Duration Curve Exceedence.
5. Run-off and precipitation relationship on longer time-scales; Monthly Run-off coefficient;
6. Error in discharge volumes; Relative Volume Error.

$$NSE = 1 - \frac{\sum_{t=1}^{T}(Q_{s,t} - Q_{o,t})^2}{\sum_{t=1}^{T}(Q_{s,t} - \overline{Q}_o)^2} \tag{4.41}$$

$$NSE_{Log} = 1 - \frac{\sum_{t=1}^{T}\left(log_{10}Q_{s,t} - \log_{10}Q_{o,t}\right)^2}{\sum_{t=1}^{T}(\log_{10}Q_{s,t} - \log_{10}\overline{Q}_o)^2} \tag{4.42}$$

$$RVE = \frac{Q_s - Q_o}{Q_o} * 100 \tag{4.43}$$

$$FDC = \sqrt{\frac{1}{n_{EP}} * \sum_{n=1}^{N}\left(log_{10}(Q_{s,n}) - \log_{10}(Q_{o,n})\right)^2} \tag{4.44}$$

$$R^2 = \frac{\sum_{t=1}^{T}\left(Q_{s,t} - \overline{Q}_{o,t}\right)^2}{\sum_{t=1}^{T}\left(Q_{o,t} - \overline{Q}_{o,t}\right)^2} \tag{4.45}$$

$$RC_{rmse} = \sqrt{\frac{1}{n_{months}}\sum_{i=1}^{N}\left(\frac{Q_{s,i}}{P_i} - \frac{Q_{o,i}}{P_i}\right)^2} \tag{4.46}$$

Where the abbreviations of the objective functions stand for:

NSE = Nash–Sutcliffe efficiency (–)
NSE_{log} = Log Nash–Sutcliffe efficiency (–)
RVE = Relative Volume Error (%)
RC_{rmse} = Root mean squared error of the Runoff Coefficient(–)
FDC_{rmse} = Root mean squared error of the Flow Duration Curves(–)
R^2 = Coefficient of Determination.(–)

With the variables:

T = Total time-steps (d)
t = time-step t (d)
$Q_{s,t}$ = Simulated discharge at time t (mmd^{-1})
$Q_{o,t}$ = Observed discharge at time t (mmd^{-1})
$\overline{Q}_o$ = Average observed discharge (mmd^{-1})
P_t = Precipitation on day t (mmd^{-1})
EP = Evaluation point on the flow duration curve,
n_{EP} = number of evaluation points

NSE and NSE_{Log}
The Nash-Sutcliffe Efficiency as tool for hydrograph evaluation is widely used within hydrological modeling. In this study the NSE and NSE_{Log} are calculated on time-steps of both a day- as well as a month. The NSE_{Log} is more biased towards low-flows, whereas the NSE is more biased towards high flows. In calculation on a monthly time-step, all months with more than 30% missing observations in the discharge time-series are disregarded. In the calculation of the Runoff-coefficient this applies to month with missing precipitation as well.
FDC
The FDC is used to reflect the flow regime in the frequency domain, which disregards the timing of the flow. Because the basin is fast responding, but flow and precipitation data is scarce, evaluating the hydrograph with a FDC is a good way to omit the timing, while evaluating the performance of the flow regime. If a peak flow observed on day t, is simulated on day $t+1$, the NSE or R^2 scores will be low. As on both days the error is large. The FDC, in expressing the time that a certain magnitude of flow is exceeded will give a far better score on those days. Therefore a flow duration curve tells how often certain flows are generated, but not when. The FDC is evaluated on 22 Evaluation Points (EP).

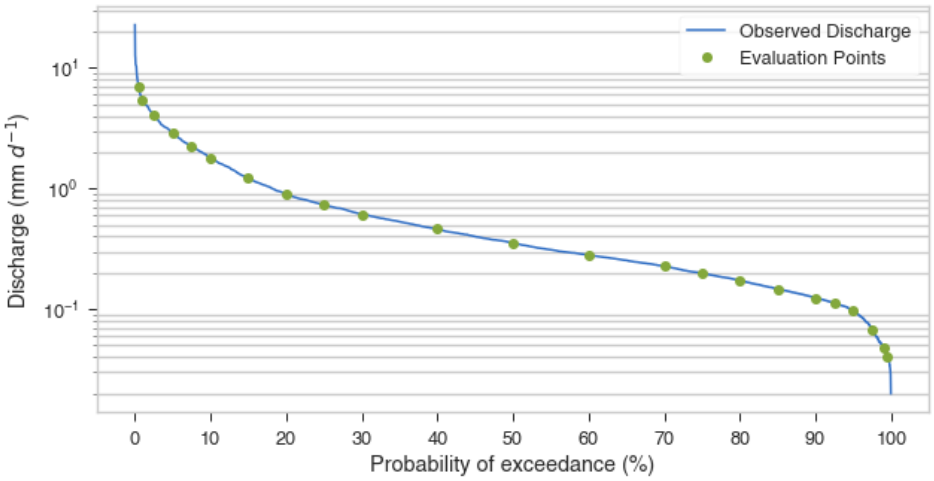


Figure 4.6: Example of the evaluation on the flow duration curve. At every evaluation point the square of the error is calculated on the observed and simulated discharges.

RC_{RMSE}
The reason to evaluate the modelled hydrograph with respect to the RC_{RMSE} is that the run-off coefficient gives insight into the performance with taken the precipitation data into account, as this is main driver of a RRM. The RC is also a climate indicative measure, as for example captured in the Budyko-Curve.

R^2 **and time-lag** The value of the time-lag f_0 is based on an analysis of the parameter-set performance for the objective function R^2 for different f_0 values.
RVE
The relative volume error is included because from an operational point of view the performance of the model in terms of the discharge volume and its error on longer timescales is an important.

4.1.5. TESTING AND EVALUATION
Testing the obtained parameter-sets will be done on the halve of the spitted discharge time-series. The performance is expressed in the same objective function scores, as well as on a visual inspection of the resulting hydrograph.

4.2. TESTING ENSO HYPOTHESIS

[] The hypothesis will be tested by performing a Pearson Correlation analysis, which will be limited to local climate variables precipitation and temperature.

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y}(-)$$

where:

cov = co-variance between X and Y
σ_X = standard deviation of X
σ_Y = standard deviation of Y

The correlation ρ can take on values between and including -1.0 and $+1.0$. Where -1.0 is a perfect negative linear correlation, 0.0 no linear correlation and $+1.0$ a perfect positive linear correlation between variables X and Y.

$$H_0 : \rho = 0$$
$$H_1 : \rho \neq 0$$

The correlation is considered to be significant if the significance level $\alpha = 0.05$ is exceeded.

$$t = \frac{\rho \cdot \sqrt{N-2}}{\sqrt{1-\rho^2}}$$

If the probability value p is smaller than the significance level α then the relationship is significant and H_0 is rejected and H_1 is accepted. In essence the p expresses the chance, based on the sample size and the assumed distribution of the sample that the calculated correlation is a false positive.

The two hypothesis tested are:

$$H_{(T,0)} : \rho_{T,ENSO} = 0 \tag{4.47}$$
$$H_{(T,1)} : \rho_{T,ENSO} \neq 0 \tag{4.48}$$
$$H_{(P,0)} : \rho_{P,ENSO} = 0 \tag{4.49}$$
$$H_{(P,1)} : \rho_{P,ENSO} \neq 0 \tag{4.50}$$

The hypotheses $H_{T,0}$ and $H_{T,1}$ refer to the relationship between the local observed temperatures T and the ENSO-Index, while $H_{P,0}$ and $H_{P,1}$ refer to the measured precipitation. As described, macro-systems in general and the sea surface temperature fluctuations in the pacific specifically influence weather on longer timescales, therefore the correlation will be calculated on a monthly timescale, and the observed ENSO fluctuations of up to twelve months prior to observational time T_0 are considered.

4.3. INTEGRATED SYSTEM MODEL

In order to be able to simulate the reservoir dynamics based on the supply and demand requires another model. This model consists of instances representing the reservoirs, their sub-catchments, the Milluni supply, the demand of the WTP Achachicala and the way in which they work together; in essence a model to capture the total Achachicala WTP system. Chosen is to model this through a sequential mathematical model written in Python, hereafter called the Integrated System Model, or ISM.

The building blocks of the ISM are:

1. Sub-Basins of Kaluyo;
2. Kaluyo Reservoirs;
3. Basin Milluni; and
4. WTP Achachicala.

The ISM consist of three reservoirs in parallel, fed by four sub-basins. The reservoirs have the following physical limitations: reservoir has a maximum volume (L^3) and a dead volume. The Milluni contribution to WTP supply is modelled as a infinite reservoir, with a base level of intake. Short section. Describe that decisions have to made with respect to order of draining reservoirs. Reservoirs have a maximum volume and a dead volume

INPUT, ASSUMPTIONS & SETTINGS

1. The input of the ISM is the simulated run-off for each sub-basin by the developed and calibrated FLEX-Topo model.
2. WTP_{Demand} , chosen is to simulate the 2016 demand.
3. Open water evaporation is not considered
4. Order in which the reservoirs drain is based on their available water level, relative to the maximum volume (%)
5. Maximum river intake is fraction of the streamflow, there in no absolute minimum or maximum.
6. There are no additional losses in the system.

4.3.1. GOVERNING EQUATIONS

On every time-step t (used: $t = 1day$) the governing equations listed in this section are solved. The total WTP_{Demand} is supplied by three supply groups:

1. $S_{FreeFlow}$: The free-flowing part of the Kaluyo Basin;
2. $S_{Reservoirs}$: The three reservoirs, including their sub-catchments;
3. $S_{Milluni}$ The Milluni Supply.

The Milluni supply is modeled as an infinite reservoir, such that the demand is always met. The total $S_{Milluni}$ is given by:

$$
\begin{aligned}
S_{Milluni} &= S_{(Milluni,Base)} + S_{(Milluni,Extra)} \\
S_{(Milluni,Base)} &= D_{WTP} \cdot f_{(Milluni,0)} \\
S_{(Milluni,Extra)} &= \begin{cases} 0, & if\, S_{K,Total} \geq D_{WTP} \\ D_{WTP} - S_{K,Total} \cdot f_{(Intake,max)} - S_{(Milluni,Base)}, & otherwise \end{cases}
\end{aligned}
$$

Only if the Kaluyo basin can not meet the requested demand, then the Milluni basin supplies additional water. As discussed in chapter three, this base supply of Milluni is present in reality as well.

The supply from Kaluyo is a combination of the freely discharging sub-basins, and the sub-basins feeding the reservoirs:

$$S_{K,Total} = S_{(K,Res)} + S_{(K,FF)}$$

Where:

$$S_{K,Res} = \sum S_{Pampalarama} + S_{Chacaltaya} + S_{AlPaquita}$$
$$S_{K,FF} = \sum S_{,FFbasins}$$

The supply of the freely flowing sub-basins is the sum of there daily discharges. The supply of the reservoirs however, depends on the *requested supply*, hereafter $D_{Reservoirs}$. The requested demand is zero if there is enough water in the free flowing river, otherwise the requested demand is the difference.

$$D_{Reservoirs} = \begin{cases} 0, if(S_{K,FF}) \cdot f_{(Intake,max)} \geq D_{WTP} - S_{(Milluni,Base)} \\ D_{WTP} - (S_{K,FF}) \cdot f_{(Intake,max} - S_{(Milluni,Base)}, otherwise \end{cases}$$
$$S_{(K,Res)} = \begin{cases} D_{Reservoirs}, & if \sum V_r eservoirs \geq D_{Reservoirs} \\ \sum V_{Reservoirs}, & otherwise \end{cases}$$

If the reservoirs have enough water in storage to meet the demand the reservoir with the highest reservoir level is drained first. Otherwise the reservoir with the highest level is drained first, untill the demand is met. To determine if there is enough water in storage, the reservoir balance is calculated on every time-step t:

$$\frac{dV}{dt} = I - O$$
$$I = \sum Q$$
$$O = S_{Reservoir} + Spill$$
$$Spill = \begin{cases} 0, & if V_{reservoirs} + I \leq V_{max} \\ (V_{reservoirs} + I) - V_{Max}, & otherwise \end{cases}$$

The model code can be found in Appendix

4.4. APPLICATION OF THE MODELS TO THE NEW SYSTEM

In the final part of the study, the developed ISM and developed and calibrated FLEX-Topo model are applied to the newly formed sub-basins, to provide insight into their Hydrological functioning and integrated response.

4.4.1. APPLICATION OF FLEX-TOPO

In this part the calibrated FLEX-Topo is applied to the new sub-basins, shown in Figure 4.1. The sub-catchment topography, HRU build-up and model settings are listed in the table below. The key assumption here is that the calibrated values for the whole basin, are also applicable on the sub-basins separately. Aim of this part is to identify noticeable differences between the behaviour of the sub-basins and their relative contributions to the observed discharges, and produce the input of the reservoir simulation. **Urban HRU** The urban HRU is added for sub catchment C06. This sub-basin is located after the discharge measurement station, therefore its not included in the model and observed discharge data. It is for a large part urbanized; which suits non of the HRU description. Moreover, there is some Urbanisation within the basin before the measurement station, but this is less dense. Several options have been contemplated to deal with this. In the end, the decision has been made to include a simple HRU_{Urban}, set the parameters to reasonable values: all are equal to the calibrated values, except for the interception reservoir, which is set to $S_{i,max} = 0.5mm$.

A measure to express the contribution of a sub-basin to the total flow, relative to its contribution to the total catchment size is the Relative Contribution factor (RFC). Values above 1.0 signify a higher contribution of the sub-basin to the total flow, then was to be expected based on its area.

$$RCF_i = \frac{f_{i,Area}}{f_{i,Flow}}$$
$$f_{i,Area} = \frac{A_i}{A_{Total}} \cdot 100$$
$$f_{i,Q} = \frac{Q_i}{Q_{Total}} \cdot 100$$

Figure 4.7: Un-calibrated Urban HRU.

Where:

$$RCF = \text{Relative Contribution Factor for sub-catchment } i(-).$$
$$f_{i,Area} = \text{Relative Area Contribution of sub-catchment } i(\%).$$
$$f_{i,Q} = \text{Relative Flow Contribution of sub-catchment } i(\%).$$

Table 4.1: HRU distribution of the new sub-catchments. Note the addition of the Urban HRU.

ID	Riparian (%)	Hillslope (%)	Moraine (%)	Glacier (%)	Urban (%)
C00	5.7	88.4	2.8	0.0	3.1
C01	10.9	67.9	21.2	0.0	0.0
C02	6.1	59.7	34.2	0.0	0.0
C03	7.2	28.0	64.8	0.0	0.0
C04	1.5	41.8	53.7	3.0	0.0
C05	4.7	45.0	49.2	1.1	0.0
C06	0.0	48.7	0.0	0.0	51.3

4.4.2. APPLICATION OF THE ISM

The simulated sub-basin runoff by FLEX-Topo model is the main input of the ISM. In addition to the reservoir properties and intake settings. In order to be able to quantify the supply increase the ISM will also be applied to old system. The period from the first of April for every year chosen as the starting point for the twelve month supply and demand calculations. This starting point is chosen as April is generally seen as the end of the wet period. Table 4.4 list the used reservoir settings, Table 4.2 the settings and partitioning of the sub-basins and Table 4.3 the used intake settings.

Table 4.2: Sub-Catchment information.

Sub-catchment		**Supplies**		**Partitioning**		**Area**	**P Station**
ID	***Name***	***System 1***	***System 2***	***System 1***	***System 2***	***km²***	***ID***
C00	Choqueyapu II	Free Flow	-	1.00	-	38.31	ELT
C01	Kaluyo River	Free Flow	-	1.00	-	33.31	AAC
C02	Al Paquita	Al Paquita	-	1.00	-	8.45	AAC
C03	Chacaltaya	Chacaltaya	-	1.00	-	13.42	AAC
C04	Toma Jumalincu	Free Flow	Pampalarama	**0.05**	**0.95**	6.49	AAC
C05	Pampalarama	Pampalarama	-	1.00	-	6.54	AAC
C06	Choqueyapu II	Free Flow	-	1.00	-	11.94	ELT

Table 4.3: Settings of the three simulations

	Simulation		
	I	**II**	**III**
$\mathbf{f_{Milluni,0}}$	0.1	0.1	0.1
$\mathbf{f_{Intake,max}}$	0.3	0.5	0.7

Table 4.4: Reservoirs volumes.

	Total Volume (hm^3)	**Useful Volume** (hm^3)	**Dead Volume** (hm^3)
Pampalarama	3.41	2.90	0.51
Chacaltaya	2.86	2.70	0.16
Al Paquita	1.42	1.34	0.09

5

RESULTS

5.1. HYDROLOGICAL MODEL

5.1.1. HRU CLASSIFICATION & DISTRIBUTION

The application of the HRU thresholds resulted in the distribution of the four HRUs over the basin as is shown in Figure 5.1. Table 5.1 shows the area fractions for each class. The irradiance classes make up 50% of each elevation and HRU class per design.

Table 5.3 lists the absolute and relative area fractions of the HRU with respect to the whole basin, as well as for the sub-basins $Kaluyo_{Lower}$ and $Kaluyo_{Upper}$. $HRU_{Hillslope}$ is by far the biggest HRU class with an area of 67.5% of the basin. Moreover it makes up almost 91% of $Kaluyo_{Lower}$. The second largest HRU is the $HRU_{Moraine}$, with a relative area contribution just south of 25%. It is to be expected that these HRUs are therefore very influential on the modeled discharge. It is also obvious that $HRU_{Glacier}$ is very small, as it makes up less that a percent of the basin. Omitting this HRU, and attributing this part of the basin to the $Moraine_{HRU}$ could be argued. It would reduce the number of model parameters, ipso facto limit the models complexity and -equifinality. However, the arguments to include it outweigh the drawbacks. The glacier contribution to the dry period run-off is unknown, but assumed to be significant. Also, for the Pampalarama Reservoir the contribution glacial area is close to 1.5 %, while for the other reservoirs it is none. This might result in a small, but not negligible dry-runoff contribution of the glacier to this reservoir. As one main grounds to choose for a rainfall-runoff model concept that accounts for differences in ecological, geographically and geometrical build-up of the sub-basins, as it is hypothesised it would lead to different runoff patterns. Therefore the $HRU_{Glacier}$ is included, independent of its size.

Table 5.1: Relative area of the elevation classes per HRU. Note that due to rounding the total sum of the Riparian elevation percentages is 100.1 %.

Elevation Classes		**HRU**			
Class id	**Elevation range**	**Riparian**	**Hillslope**	**Moraine**	**Glacier**
-	**(m)**	**(%)**	**(%)**	**(%)**	**(%)**
1	3750-4000	1.0	0.5	-	-
2	4000-4250	12.3	9.0	-	-
3	4250-4500	57.7	29.9	-	-
4	4500-4750	28.7	51.5	13.1	-
5	4750-5000	0.4	9.1	63.5	37.4
6	5000-5250	-	-	20.9	62.6
7	5250-5500	-	-	2.5	-

Table 5.2 and Figure 5.2 give information on the measured values by the MODIS satellite for each HRU. It can be seen that the vegetation dynamics are similar to what could be expected; both in yearly patterns

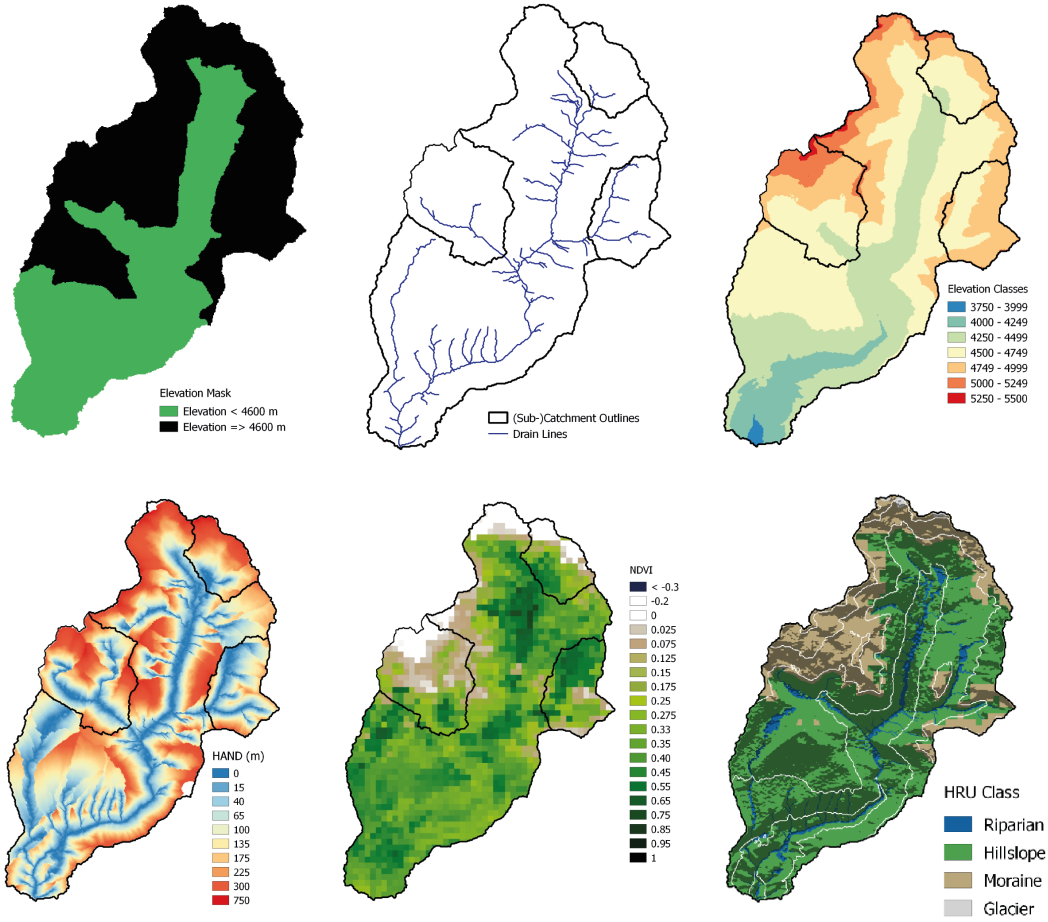


Figure 5.1: Maps to visualize the data that has been used to classify the HRUs and internal HRU distribution. Top left: elevation threshold for the Moraine HRU. Top center: Drainage network. Top Right: Elevation Classes. Bottom Left: map of the HAND. Bottom center: NDVI Map. Bottom Right: HRU Classes with low IR zones slightly shadowed, and High IR zones transparant.

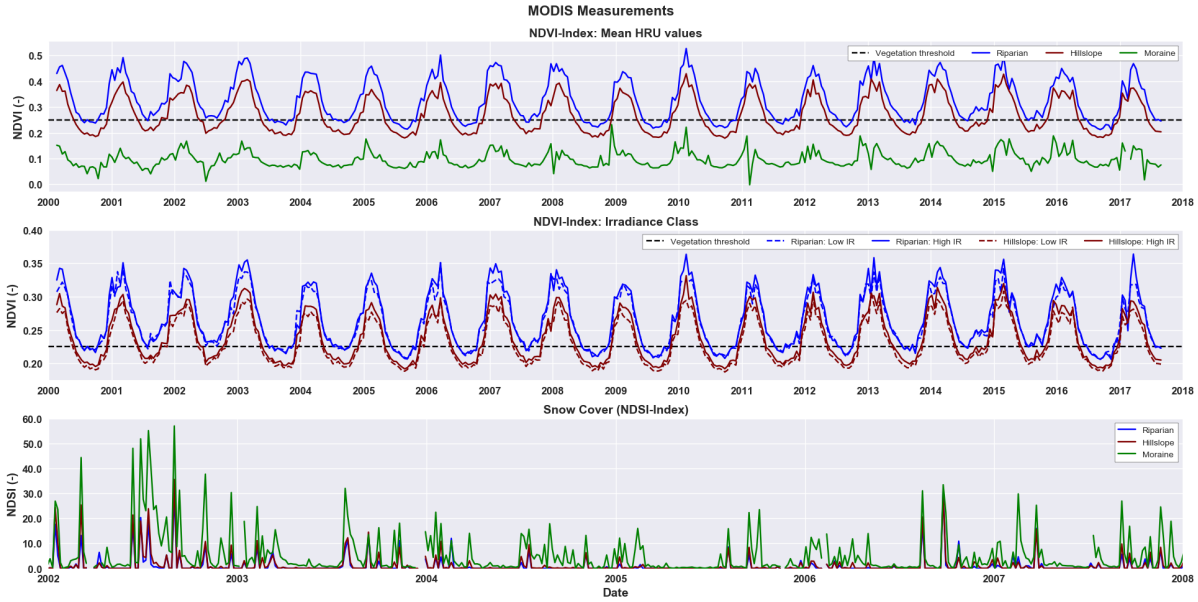


Figure 5.2: MODIS: NDVI- and NDSI-Index time-series. The vegetation threshold is an indicative-, not an absolute threshold.

for all the HRUs, also in differences between the HRUs and lastly in the differences within the HRUs (i.e. the irradiance classes). The mean NDSI-, and NDVI-values for the whole time-series (2000-2017). Expected is that in general:

1. $NDVI_{Riparian} > NDVI_{Hillslope} > NDVI_{Moraine}$
2. $NDSI_{Riparian} < NDSI_{Hillslope} < NDSI_{Moraine}$

Within the HRU one would expect that:

3. $NDVI_{High,IR} > NDVI_{Low,IR}$
4. $NDSI_{High,IR} < NDSI_{Low,IR}$

The HRU distribution between the vegetated and non-vegetated parts of the basin is made based on the NDVI index, thus is is to be expected that the NDVI values of $HRU_{Moraine}$ are lower than those of the riparian- or hillslope zones. The decision whether to attribute a pixel to the $HRU_{Riparian}$ or $HRU_{Hillslope}$ is made based on the HAND value. Because the riparian zone is per definition closer to the groundwater table than the hillslopes, the riparian zone is expected to be more densely vegetated, and stay *green* longer after the end of the wet-season. This behaviour is visible in the figure.

Because the mean elevation of $HRU_{Riparian} < HRU_{Hillslope} < HRU_{Moraine}$, temperatures are assumed to be lower. Therefore, the most snow cover is expected in the moraine, and the least in the riparian zone. Within the HRUs the zone receiving the most irradiance is expected to have less snow cover on average, and more vegetation, as solar energy is a key driver of transpiration. This pattern is also visible in the measured data. The only exception is the riparian zone, in which the NDSI value of the high irradiance class is lower - on average- than the one of the low irradiance class. Overall, these results are indirect evidence of a correctly made classification.

Table 5.2: Mean HRU class values for NDVI-Index and NDSI-Index

	NDVI			**NDSI**		
	Total	***Low IR***	***High IR***	***Total***	***Low IR***	***High IR***
Riparian	0.32	0.32	0.32	0.7	0.6	0.8
Hillslope	0.26	0.26	0.27	0.9	1	0.8
Moraine	0.09	-	-	4.5	5.2	3.7

The basin is divided into two sub-basin to distribute the precipitation data with respect to the two stations; Alto Achachicala and El Tejar. The basin *Kaluyo Upper* receives precipitation from the former, and *Kaluyo Lower* from the latter. The division between these sub-basins is based on a soon to be installed discharge measurement station. Kaluyo Lower has an area of 40.0 km^2 and Kaluyo Upper 68.1 km^2. The sub-basin fractions, and their HRU distributions is shown in Table **??**. For more details about the internal HRU distributions with respect to elevation and IR_{zones} refer to Table A.1 in Appendix A.

Table 5.3: Absolute and relative area fractions per HRU; for the total Kaluyo basin, and of the two simulated sub-basins.

	Kaluyo Lower		**Kaluyo Upper**		**Total Basin**	
	(%)	**(km^2)**	**(%)**	**(km^2)**	**(%)**	**(km^2)**
Riparian	7.3	2.9	9.4	6.4	8.6	9.3
Hillslope	90.9	36.4	53.7	36.6	67.5	72.9
Moraine	1.7	0.7	36.5	24.8	23.6	25.5
Glacier	-	-	0.5	0.3	0.3	0.3
Sum	***100***	***40***	***100***	***68.12***	***100***	***108.12***

The calculated irradiance factors are shown in Figure 5.3, where the monthly mean over all the elevation zones is taken. It is obvious that the spread increases going from the austral summer to the winter. The

pronounced topography, combined with the lowering of the sun in the sky results in more shading in certain parts of the basin. The reason why the highest values of the f_{IR} are during the winter as well, is explained by this too. The factor is calculated with respect to the measurement station Alto Achachicala; which is situated in the upper valley and prone to shading as well. The biggest spread can be found in the HRU glacier, with values of $f_{IR} < 0.30$ on the bottom-end and close to $f_{IR} = 1.20$ in June and July. Remarkably, both the lowest and the highest values here are found in the IR_{Low} class. This might be explained by the small area of the glacier. While the IR_{High} class shows a smaller spread and higher mean. Generally speaking this is the case for all HRUs. Probably due to the fact that the areas in the high class are overall less likely to be shaded. Note that at the end of the year, all factors, converge to 1.0; this coincides with the suns highest, close to perpendicular, position in the skies during this time. Based on these results it is save to say that considering the incoming energy flux in a distributed way, makes sense in the Kaluyo basin.

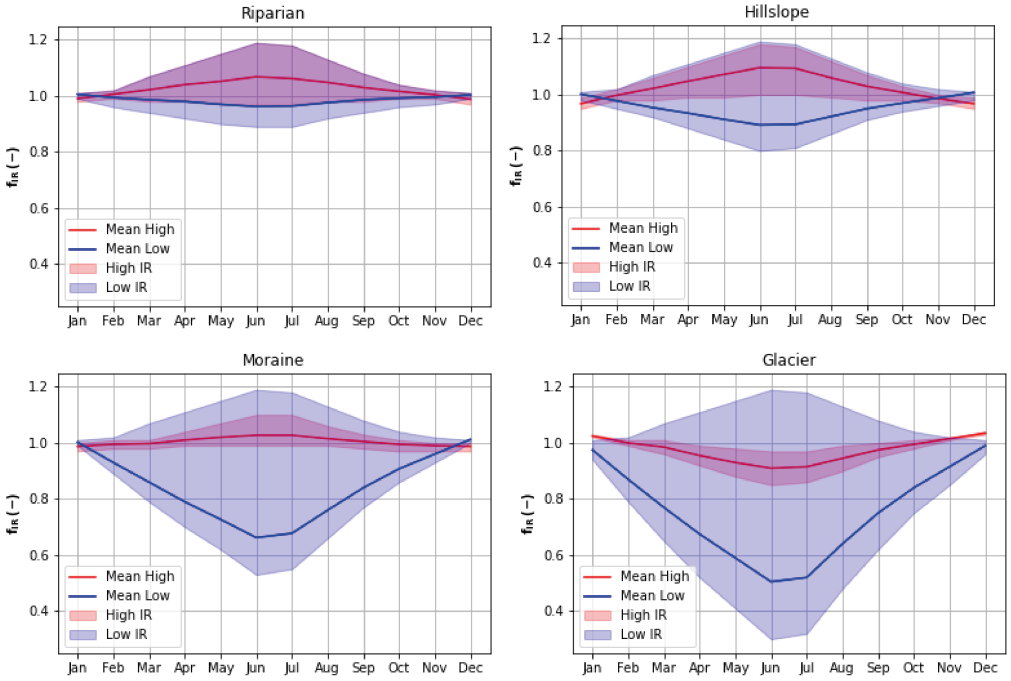


Figure 5.3: Mean Irradiance class factors throughout the year.

5.1.2. CALIBRATION

In accordance with the methodology the calibration is performed in steps. The parameter ranges for the initial calibration are shown in Table 5.4. The limits and steps of the sampling procedure are determined based on physics, literature and estimations. First the initial parameter ranges are discussed.

Parameters D_0, D_1, and C_e are constrained physically; as they represent a fractions; their value must be between 0 and 1. Chosen is to limit the range from 0.1 to All other ranges have been determined based on literature and the perceived function of the basin, or physical interpretation of the parameters.

The temperature thresholds T_t and $T_{t,Glacier}$ as well as the melt degree day factors (F_{DD}, $F_{DD,Glacier}$) are own estimations, as are K_f and K_s; constrained by their perceived likely values. β is an exponential reflecting the non-linearity in friction upon entering and draining of a soil by water. The range $0.5 < \beta < 5.0$ seemed appropriately large. An exponent of close 5.0 is already stretching the limit of the reasonable. For $C_{r,max}$ chosen is for an upper-limit of 5 mm. The main reason that the highest calculated is around 5.0 $mm \cdot d^{-1}$. Higher values of $C_{r,max}$ would not result in more storage in $S_{(u,r)}$ but just run-off. The only difference would be the reservoir through which it discharges. Groundwater pushing up is not limited to the volume of the storage in reality, or the capacity to transpire of vegetation in reality. However, in this modeled reflecting it is assumed that calibrated values of K_s and K_f will be able to compensate for this. The maximum sublimation rate $E_{s,max}$ is determined based on literature [15] [16].

The maximum interception and unsaturated reservoir storage volumes are chosen on expert knowledge; $S_{i,moraine}$, $S_{i,hillslope}$ and $S_{i,riparian}$ can not be large. Interception is the water that is not able to enter the soil because it is physically hindered; it falls on leaves, houses or stones. Every mm interception storage per m^2 is a liter in reality. A mm water depth, evenly distributed is a lot. To be on the safe side values between 0,5 and 2.0 are chosen. The unsaturated storage in the wetlands is assumed to be very small, maximum 50 mm and 150 mm for the hillslopes. The soils on the hillslopes are highly permeable and grasses are assumed to root not very deep. Therefore 150 mm storage seemed the upper-limit.

Table 5.4: Settings for the initial Monte Carlo sampling of N=$500 \cdot 10^3$ parameter-sets.

Parameter	Lower Limit	Upper Limit	Step	Constrained
K_f	2	12	0.1	-
K_s	80	200	1	-
T_t	-3	3	0.1	-
F_{dd}	1	10	0.1	-
D_0	0.1	1	0.025	-
D_1	0.1	1	0.025	-
β	0.5	3	0.1	-
$Si_{max,Riparian}$	0.5	2	0.1	-
$Si_{max,Hillslope}$	0.5	2	0.1	$<Si_{max,riparian}$
$Si_{max,Moraine}$	0.1	1	0.1	$<Si_{max,hillslope}$
$Su_{max,Riparian}$	1	50	1	-
$Su_{max,Hillslope}$	10	150	1	-
Cr	1	5	0.1	-
$E_{s,max}$	1	3	0.1	-
$Tt,glacier$	-3	3	0.1	-
$FDD,glacier$	1	10	0.1	-
Ce	0.1	1	0.1	-
$K_{f,Moraine}$	1	10	0.1	$<K_f$
$time-lag$	-0.3	0.3	0.1	-

The initial calibration results have been rated on the objective functions and inspected on the hydrograph. Sometimes the performance trends are clear; and sometimes they are not. Figure 5.4 show for example that it is clear that on a monthly time-scale, the chosen K_s range was not sufficiently large, as the trend is still increasing upon hitting the upper-limit. Chosen is to increase both the lower- and upper-limit. Remarkably, for the K_f values the model seemed to perform better for higher values. Inspection of the hydrograph however resulted in the opposite conclusion: The hydrograph was way worse for values of $K_f = 10d$, then for $K_f = 3d$.

The performance increase is attributed to averaging out.

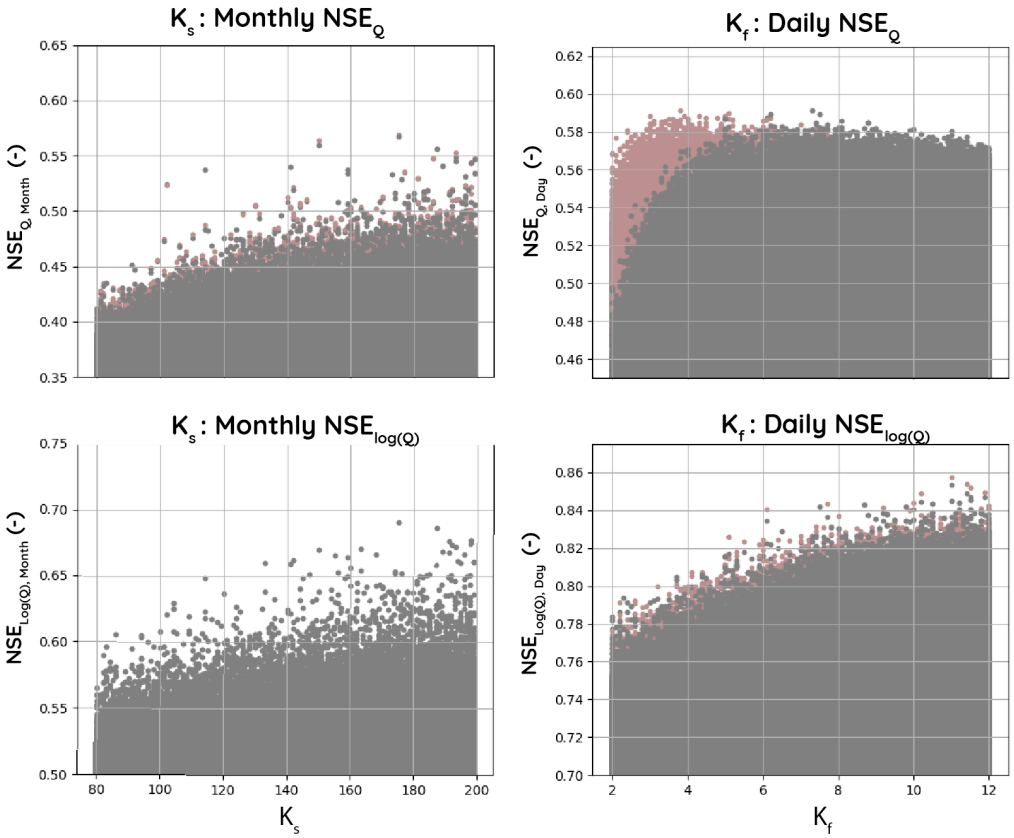


Figure 5.4: Performance of K_s and K_f on initial calibration

INTERNAL CALIBRATION: NDVI

The results of the internal calibration NDVI calibration are discussed here. Table 5.5 shows the parameter settings that have been used. Note that all parameters are fixed, except the parameters that directly influence the transpiration in the model. $50 \cdot 10^3$ parameter-sets are sampled, simulated and analysed. This is done for the two vegetated HRUs. This calibration shows i.e. how close the NDVI patterns (as illustrated in Figure 5.2) are matched by the modeled transpiration in the frequency domain. Thus not considering the exact timing. First the results of the $HRU_{Hillslope}$ are presented and afterwards those of $HRU_{Riparian}$.

Figure 5.5 show interesting patterns of the NDVI evaluation. With respect to the transpiration in the $HRU_{Hillslope}$ three parameters play a role: β, C_e and S_u. The was not a signal observable in the evaluation for the first parameter. For C_e and S_u however there was. When evaluated separately they both show a pattern (see bottom half of Figure 5.5), but its not very well defined. But this changes when they are evaluated together: then the performance measure is very well defined and clear. The limit under which the model does not perform very well is clearly visible. This was to be expected as they together determine fully the functioning of the unsaturated reservoirs. The red arrows point towards "*bumps*" in the performance curve that can not be really explained. For both calibration periods the performance converges towards the optimal value, but than falls back only to rise with rising values again. For the second calibration period two of these bumps are noticeable. The highlighted area (in green) shows the parameter range that has been chosen to limit the parameters to. The main reason is that for both the irradiance zones, and both calibration periods the model performs well above these thresholds.

For the $HRU_{Riparian}$ C_r introduces an extra free-dimension. This could be the reason that the performance curve of $C_e \cdot S_u$ is not as well defined as for $HRU_{Hillslope}$, as Figure 5.6 illustrates. There is still information that can be used. The red arrows point towards the lowest performance region for both $C_e \cdot S_u$ at

Table 5.5: Parameter-settings used in the internal calibration of the Unsaturated Reservoirs.

Parameter	Free / Fixed	Fixed Value	Lower Limit	Upper Limit	Step
K_f	Fixed	3	-	-	-
K_s	Fixed	240	-	-	-
T_t	Fixed	-3	-	-	-
F_{dd}	Fixed	7	-	-	-
D_0	Fixed	0.4	-	-	-
D_1	Fixed	0.7	-	-	-
β	Free	-	0.5	3	0.1
$Si_{max,Riparian}$	Fixed	-	-	-	-
$Si_{max,Hillslope}$	Fixed	-	-	-	-
$Si_{max,Moraine}$	Fixed	-	-	-	-
$Su_{max,Riparian}$	Free	-	5	50	1
$Su_{max,Hillslope}$	Free	-	20	150	1
Cr	Free	-	1	4	0.1
$E_{s,max}$	Fixed	2.5	-	-	-
$T_{t,glacier}$	Fixed	1	-	-	-
$F_{dd,glacier}$	Fixed	2	-	-	-
Ce	Free	-	0.1	0.9	0.025
$K_{f,moraine}$	Fixed	2.5	-	-	-
$Time-lag$	Fixed	0.1	-	-	-

top of the figure, as well as for C_r at the bottom. For the combination of low C_r values (maximum amount of ground-water pushing up per unit area), in combination with low effective storage in the unsaturated reservoir, the NDVI pattern can not be matched by the transpiration in the model. Not enough water can be stored, and not enough water can be pushed up. Based upon these graphs the minimum value of C_r for the final calibration was set to $2.5 mm \cdot d^{-1}$, above the model seems to match the NDVI patterns reasonably well.

In combination with the results of the Hillslope the minimum value for C_e was set to 0.7. $Su_{max,Riparian} > 15mm$ and $Su_{max,Hillslope} > 90mm$

INTERNAL CALIBRATION: SNOW COVER

For the snow calibration all model parameters are fixed except: F_{DD} and T_t. The result of the 2000 simulated parameter combinations is presented in Figure 5.7. What implicitly is shown is that performs worse with increasing altitude: the Riparian zone has the lowest mean elevation and the Moraine the highest. This might be explained by the temperature lapse rate, or the precipitation data. As the higher zones are hypothesised to have more precipitation. A precipitation lapse rate is chosen that increase precipitation depth on days with rain, but does not have an effect on days that are dry at the station. Nonetheless, the shift of the optimum values for F_{DD} and T_t to the left (negative T_t) indicates that the temperature on higher altitudes is lowered to much by the lapse rate, and therefore needs to be adjusted by the temperature threshold. Based on this evaluation seperate values for T_t and F_{DD} for the $HRU_{Riparian}$ and $HRU_{Hillslope}$ on the one hand, and $HRU_{Moraine}$ on the other are determined. For the former $T_t = -4.0$ and $F_{DD} = 8.0$ and for the latter $T_t = -1.8$ and $F_{DD} = 10.0$.

INTERNAL CALIBRATION: GLACIER

The result of the glacier retreat evaluation is shown in Figure 5.8. This calibration has led to determining the $F_{DD,Glacier}$ and $T_{t,Glacier}$ on respectively $1.1 mm \cdot d^{-1}$ and $0.9°C$. Although other combinations could have been chosen as well. With these values, in combination with the F_{DD} and T_t values determined on the MODIS snow-cover evaluation resulted in the correct glacier retreat.

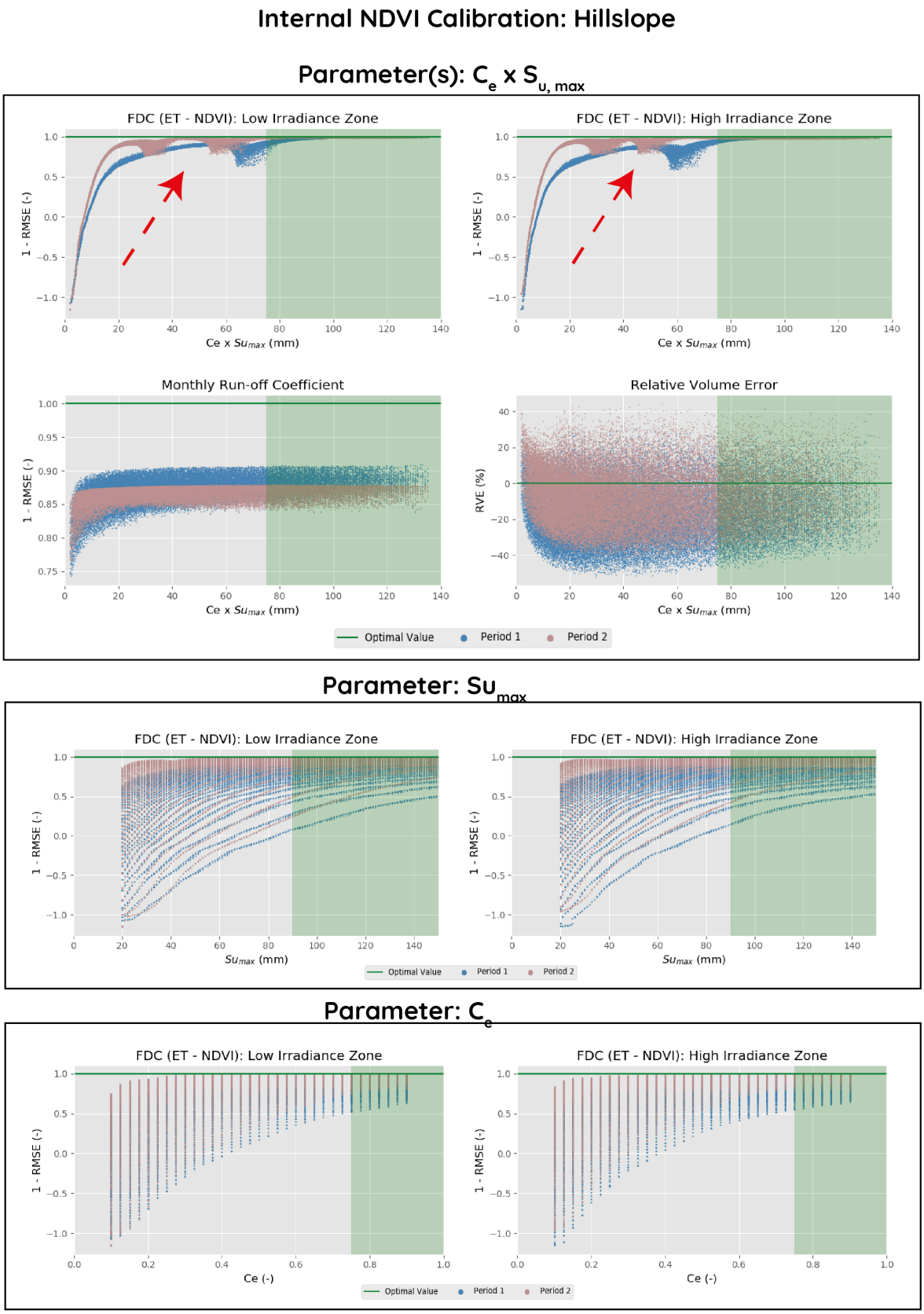


Figure 5.5: NDVI Internal Calibration: Hillslope

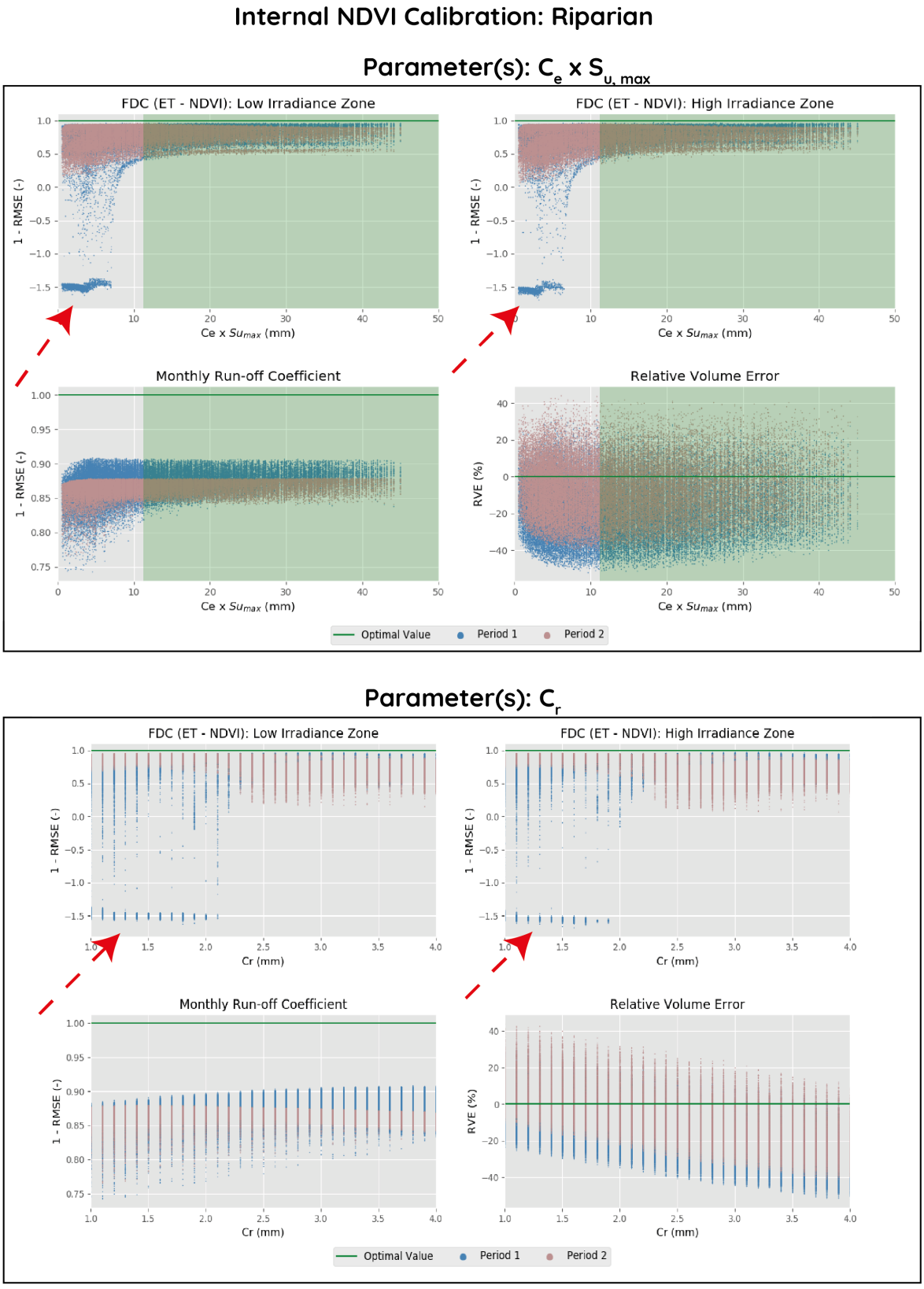


Figure 5.6: NDVI Internal Calibration: Riparian

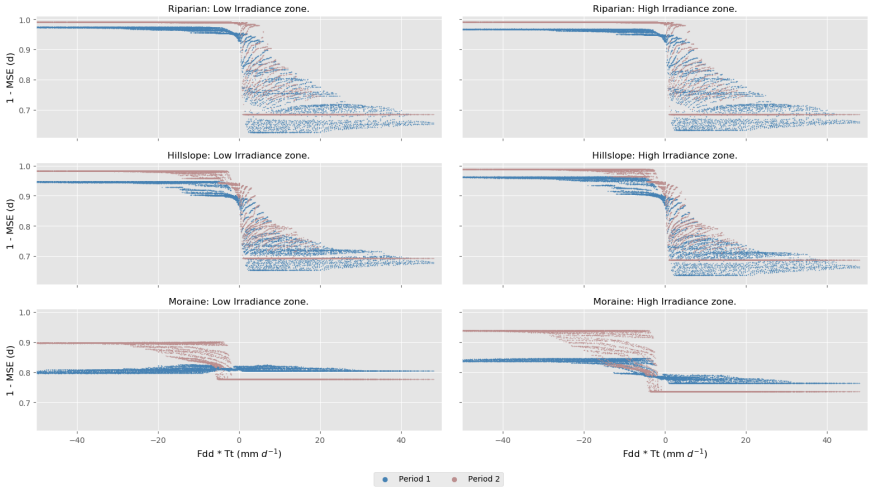


Figure 5.7: The performance of combined parameters T_t and Fdd against observed MODIS Snow data. Score of 1.0 is perfect.

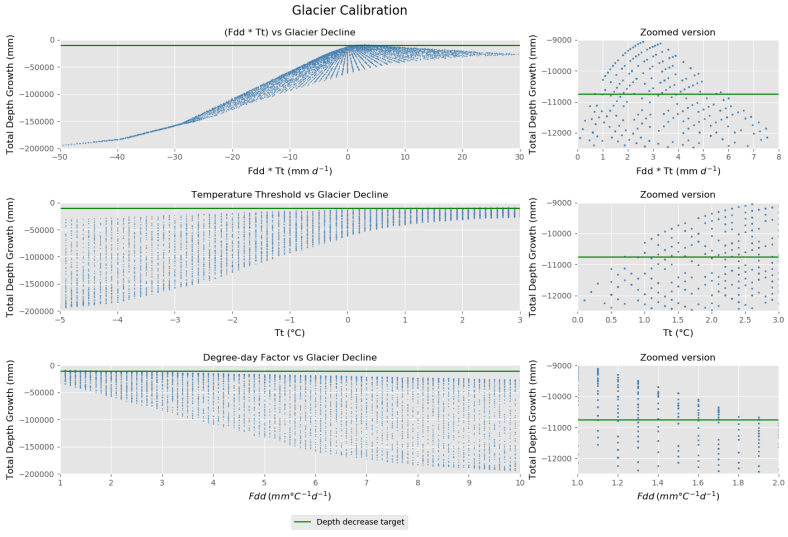


Figure 5.8: Relationship between the parameters $F_{dd,glacier}$ and $T_{t,glacier}$ and the snow decline on the glacier. The horizontal lines show value-bands based on literature research.

FINAL CALIBRATION RESULTS, PERFORMANCE AND TESTING

In this section the results of the final calibration are presented.

The analysis on timing, with the coefficient of determination did not yield very identify-able results, as can be seen in Table 5.6. One must note that the values of $f_0 = 0.0$ are the worst, while the highest values can be found for the minimum and maximum value for the time-lag. This is could be explained by averaging out of peaks. The basin is fast-responding, and the precipitation and discharge data uncertain. Peaks in flows are flattened for $f_0 \neq 0.0$, and are linearly more flattened for increasing $\mid f_0 \mid$ values. Because the purpose of the developed FLEX-Topo model is not to time and predict peak-flows, the f_0 is not considered of the highest importance. Thus because it's influence and signal is not clearly identify-able, a value of $f_0 = 0.4$ is chosen, and the following presented results are for this value.

Table 5.6: Influence on the R^2 for different time-lag f_0 values. $N = 500 \cdot 10^3$. Max = Maximum, Stdv = Standard Deviation

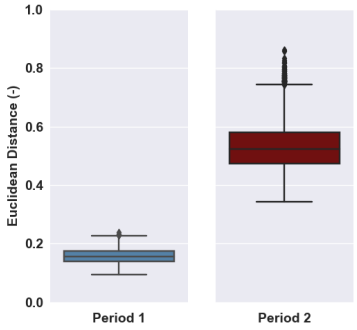
	Calibration Period 1				**Calibration Period 2**			
f_0	**Median**	**Mean**	**Max**	**Stdv**	**Median**	**Mean**	**Max**	**Stdv**
-0.4	0.32	0.30	0.74	0.19	0.37	0.35	0.77	0.18
-0.3	0.32	0.30	0.74	0.19	0.37	0.35	0.77	0.18
-0.2	0.31	0.29	0.74	0.20	0.36	0.34	0.77	0.19
-0.1	0.30	0.28	0.73	0.20	0.35	0.33	0.77	0.19
0	0.29	0.27	0.73	0.21	0.34	0.32	0.77	0.20
0.1	0.30	0.28	0.73	0.20	0.35	0.33	0.77	0.19
0.2	0.31	0.29	0.74	0.20	0.36	0.34	0.77	0.19
0.3	0.32	0.30	0.74	0.19	0.37	0.35	0.77	0.18
0.4	0.32	0.30	0.74	0.19	0.37	0.35	0.77	0.18

Figure 5.11 shows the performance and its spread for the three periods: period 1, period 2 and the total time-series. Noticeable is that the performance on the second period is a lot worse for the $NSE_{Log(Q)}$ on a daily and monthly time-scale, while being slightly better for the NSE_Q.

Figure 5.10 shows the performance of the model on the y-axis relating to individual model parameters. Within the ranges used for the final calibration, not all parameters are particularly well-defined. Note that opposed to the pattern of the initial calibration (Figure 5.4), there does not seem to be a clear trend in the performance of the K_f and K_s values. For all values, K_f and K_s plot within the full range of performance (good to very bad). This was to be expected however, and which is here actually a good thing. The parameter ranges are very constrained based on the initial steps. If a pattern was still observable (extending beyond the limits, and physically possible to exceed these limits) than the range would have been poorly chosen. Within the small range, it seems that either one of the values performs quite well.

Opposed to this are D_0 and D_1, which show a clear trend with respect to their performance. Considering D_1 the results show opposing patterns for different performance measures. Consider the FDC, although the model *can* perform well the FDC measure for lower D_1 values, it shows a clear trend to perform better for increasing one. In approaching the physical limit of $D_1 = 1.0$ the performs is generally the best. D_0 shows the same pattern in increasing performance nearing the upper-limit. However, here the upper-limit is constrained by a perceived upper-limit. This might be an indication that the limit is poorly chosen. On the other hand, the pattern reverses in evaluation of the NSE objective functions. Here a converging pattern approaching the lower, physical limit is present; similar to D_1.

Calculation of the euclidean distance resulted in the selection of the best parameter-sets. Figure 5.9 shows the ED spread for the two periods. It is clear that the model generally performs better on the first; both in spread as well as absolute values. The selected parameter-sets are given in 5.7, their performance in calibration and validation in Table 5.8. The biggest take-away from the performance in calibration and validation, is that the pattern is again the same: good performance on the first period, worse on the second. This is fully supported by the fact that the calibrated values for the second period perform better in validation, then in calibration. Which is remarkable. Especially given the significance of the improvement: except from the



[H]

Figure 5.9: Euclidean distances.

monthly NSE_Q values, all three NSE objective functions improve close to 0.20 in validation. In the case of the daily NSE_Q the score is more than doubled. Not surprisingly, the calibration on the first period, and validation on the second shows the opposite: a significant reduced performance in validation. If the two parameter-sets are compared on the same period, than the values are almost identical. This could support the confidence of the model set-up and calibration technique. That the simulation results are quite similar is supported upon inspection of the hydrographs.

Table 5.7: Final parameter values

Parameter	**Total Period**	**Calibration 1**	**Calibration 2**
kf	3.9	3.8	3.3
kf, Moraine	2.3	3.3	2.6
ks	235	234	239
Tt	-4	-4	-4
Tt, Moraine	-1.8	-1.8	-1.8
Fdd	8	8	8
Fdd, Moraine	10	10	10
d0	0.16	0.1	0.13
d1	0.75	0.79	0.86
β	1.7	1.4	1.8
Simax, Rip	1.1	1.25	1.7
Simax, Hill	0.55	0.5	0.6
Simax, Mor	0.1	0.35	0.2
Sumax, Riparian	15	38	15
Sumax, Hill	125	129	95
Cr	2.6	2.5	2.9
$E_{S,max}$	2.5	2.5	2.5
$Tt_{glacier}$	0.9	0.9	0.9
$Fdd_{glacier}$	1.1	1.1	1.1
Ce	0.9	0.9	0.9
f_0	0.4	0.4	0.4

Figures 5.13 and 5.14 show the hydrographs of the simulated discharge based on the parameters of the first period. It would be safe to conclude that the dry period run-off in the first period is very well simulated. On the second period the dry run-off is far off, especially in the years 2010, 2011 and 2012. In the other years it matches far better. Contrariwise the wet-season runoff in the second period matches better than on the first. The differences between the two parameter-sets is not that big, as the comparison on monthly time-steps in Figure 5.12 shows. What is also shows is that the different parameter-sets, calibrated on different periods result in a small difference in simulated discharges. In the next section, the discussion moves away from objective functions and performance measures , and looks closer into the hydrographs and simulated hydrology.

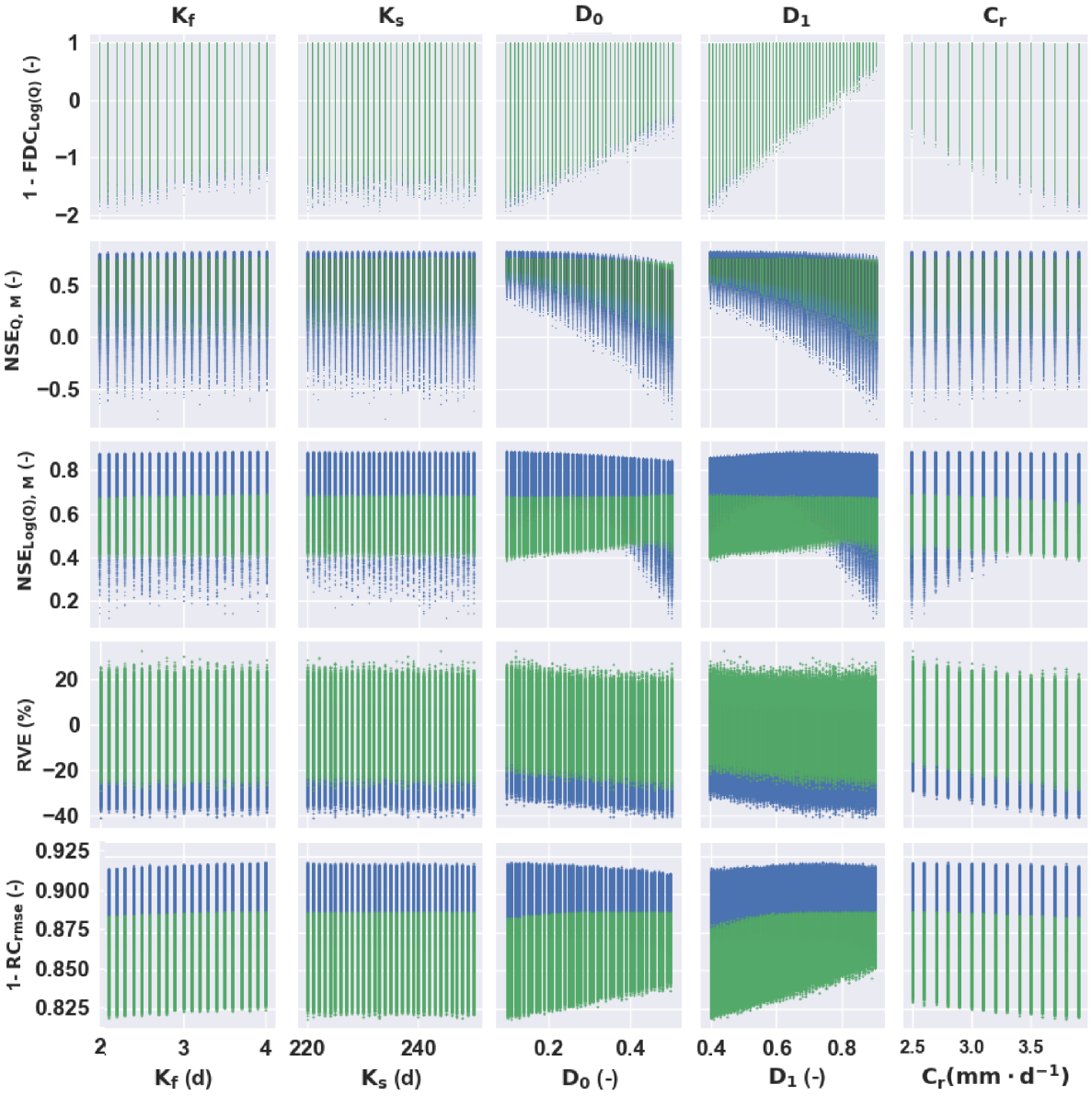


Figure 5.10: Scatter-plot relating model performance to several important model parameter-ranges. Period 1=Blue, Period 2=Green.

Table 5.8: Performance of the calibrated parameter-sets.

		Total Period	**Period 1**			**Period 2**		
		Cal	**Cal**	**Val**	δ	**Cal**	**Val**	δ
$\mathbf{NSE_{(Q,Month)}}$	(-)	0.78	0.78	0.75	-0.03	0.76	0.74	-0.02
$\mathbf{NSE_{(Log(Q),Month)}}$	(-)	0.76	0.88	0.67	-0.21	0.67	0.85	**0.18**
RVE	(%)	5.56	-1.58	13.22	14.80	9.64	-6.14	15.78
$\mathbf{FDC_{log(Q)}}$	(-)	0.91	0.95	0.91	-0.04	0.92	0.94	**0.01**
$\mathbf{NSE_{Q,day}}$	(-)	0.33	0.51	0.20	-0.31	0.22	0.48	**0.26**
$\mathbf{NSE_{log(Q),day}}$	(-)	0.59	0.74	0.49	-0.25	0.48	0.71	**0.23**
RC	(-)	0.91	0.90	0.86	-0.04	0.91	0.91	0.00

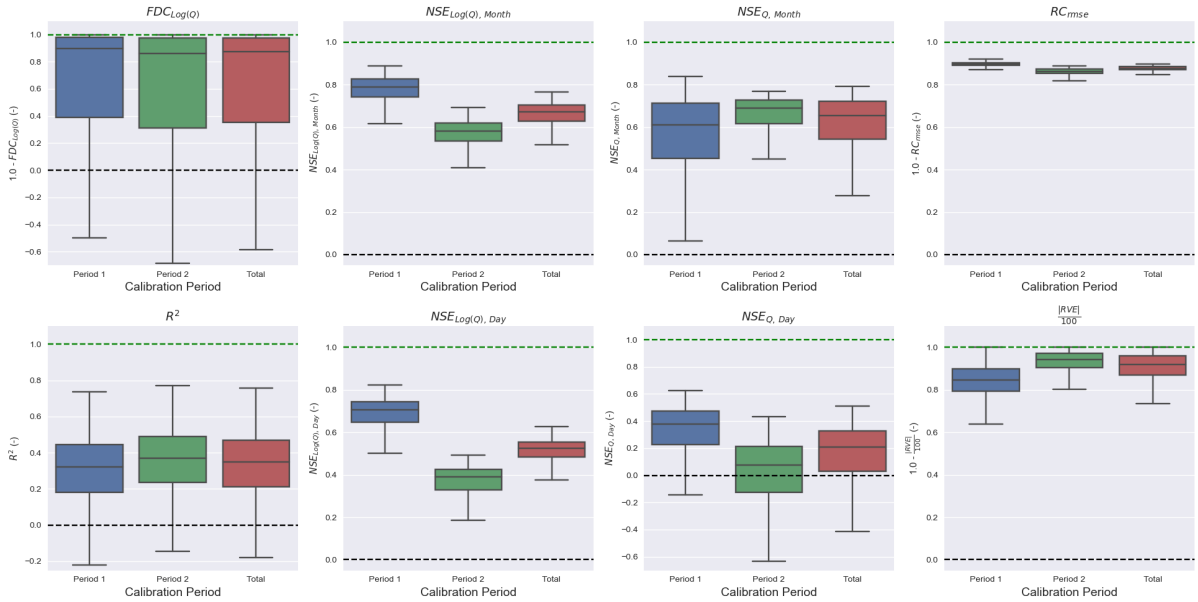


Figure 5.11: Final calibration performance for the different objective functions, captured in boxplots.

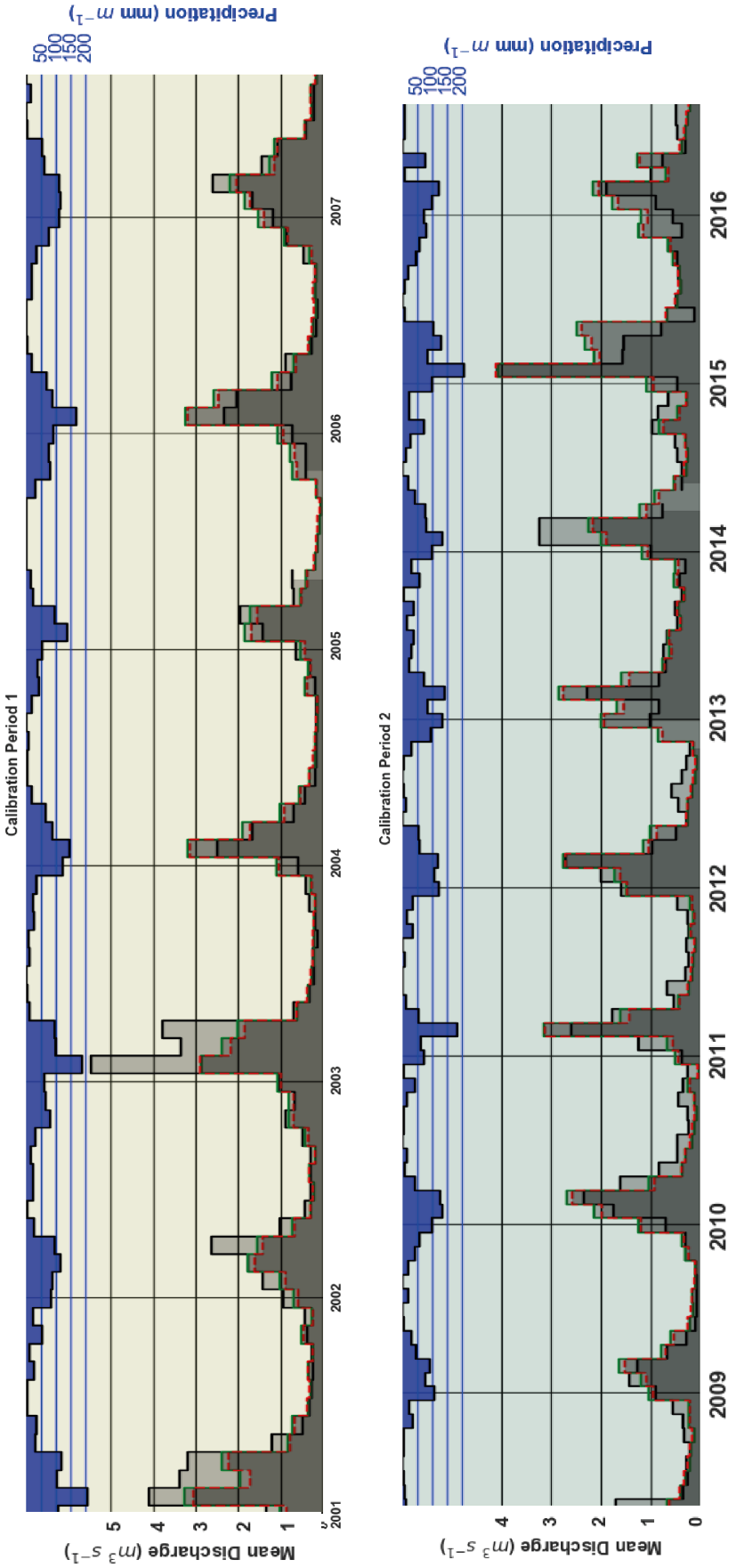


Figure 5.12: Comparison of the simulated monthly discharges, based on the calibration period 1 and period 2. Period 1 is the Green line. Period 2 is the red line and the black line is the observed discharge.

5.1.3. RESULTING HYDROGRAPH & HYDROLOGY DESCRIPTION

This section discusses the result with respect to why the model is developed in the first place: what can be learned about the basins' hydrology from the simulation results. The hydrographs are more closely inspected, both on monthly- and on daily time-steps. In addition special attention is given to the origin of the simulated flows. The results of the *Calibration Period 1* parameter-set is shown.

Over the period from January 1ST 2001 to 31 December 2015, the observed average yearly discharge $249 mm \cdot a^{-1}$, and the simulated $259 mm \cdot a^{-1}$. Divided in a fast and slow run-off $189.4 mm \cdot a^{-1}$ respectively $69.2 mm \cdot a^{-1}$. Table 5.9 further lists the fluxes further distributed with respect to their origin HRU. Figures 5.13 and 5.14 show the accompanied HRU contributions in a stacked manner on monthly time scales.

Table 5.9: Summary of the mean annual fluxes per HRU over the whole modeling period. The water balance for the Riparian zone and Glacier are facto negative. This is explained for the Riparian zone by the ground-water flux pushing up ($670\ mm \cdot a^{-1}$), and for the Glacier in the glacier depth decline ($234 mm \cdot a^{-1}$).

	Riparian ($mm \cdot a^{-1}$)	**Hillslope** ($mm \cdot a^{-1}$)	**Moraine** ($mm \cdot a^{-1}$)	**Glacier** ($mm \cdot a^{-1}$)
P	639	654	684	662
E_i	115	49	30	-
E_t	1115	355	-	-
E_s	-	-	13	306
Q_f	88	225	125	178
Q_s	0	25	516	424

From the figures it is immediately obvious that most of the peak flows originate from the Hillslopes: both absolute as well as relative. The aforementioned table shows the water balance per unit area. Almost $\frac{1}{3}$ of all the precipitation that the hillslopes receives attributes to the fast run-off. The biggest flux is however the transpiration, with 355 mm per year on average. Percolation towards the groundwater reservoir on the $HRU_{Hillslopes}$ is a order of magnitude smaller than the fast run-off (respectively $25 mm \cdot a^{-1}$ and $225 mm \cdot a^{-1}$) , with values as well as interception evaporation.

Opposed to the hillslopes, the contribution of the Glacier and Moraine to the run-off is mainly through the groundwater reservoir. This is expected given the very low values of D_1 (0.79), compared to the values of D_0 (0.10). While the contribution of melting glaciers is assumed to be significant (up to 30%) in some river catchments in the La Paz area, the contribution is negligible in the Kaluyo basin according to the model. Even when looking closely at the daily hydrograph, the glacier contribution is hardly visible. But this is not the complete story; a part is hidden within the slow reservoir discharge. The total average contribution of $HRU_{glacier} = 1.8 mm \cdot a^{-1}$ to the total discharge of $259 mm \cdot a^{-1}$, or 0.7% . Which is approximately two times its fraction of the area. Sublimation is 306 mm per year.

The Moraine is the largest contributor to the slow run-off. 75% of the water it receives in routed towards the groundwater. In combination with its relative large area fraction of almost 24% it is almost solely responsible for the flux towards the groundwater. Sublimation is very low ($13\ mm \cdot a^{-1}$) as well is interception ($30\ mm \cdot a^{-1}$). Note the difference between the sublimation on the Glacier and Moraine.

Considering the evaporation than the contribution of riparian zone stands out. With less than 9% of the total area, the riparian zone is responsible for 27% of the evaporation on a yearly basis. This is possible by groundwater pushing upwards constantly. The Riparian zone receives on average $639 mm \cdot a^{-1}$ precipitation, while evaporating $1230 mm \cdot a^{-1}$ in total and also contributing to the fast run-off . The difference can be explained by $679 mm \cdot a^{-1}$ of groundwater pushing upwards. Which is more than the total volume of precipitation it receives. The evaporation approaches the potential evaporation based on the adjusted Hargreaves evaporation. Indicating that there is almost always transpiration going on, or to say it more clearly: the wetlands stay wet.

Dry season run-off is generally made up of only ground-water discharge. It seems reasonable that the

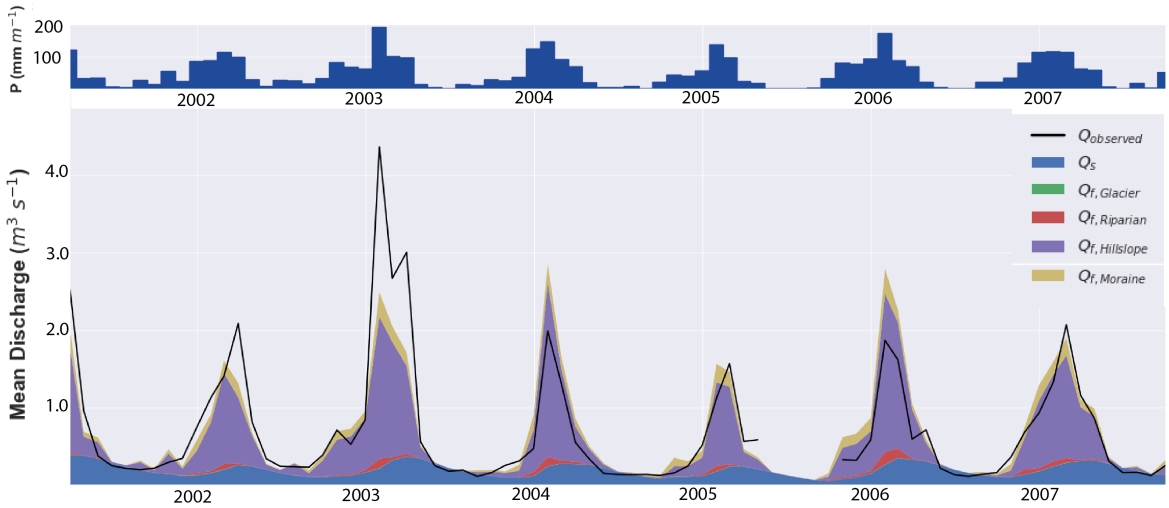


Figure 5.13: Monthly Hydrograph of the first calibration period.

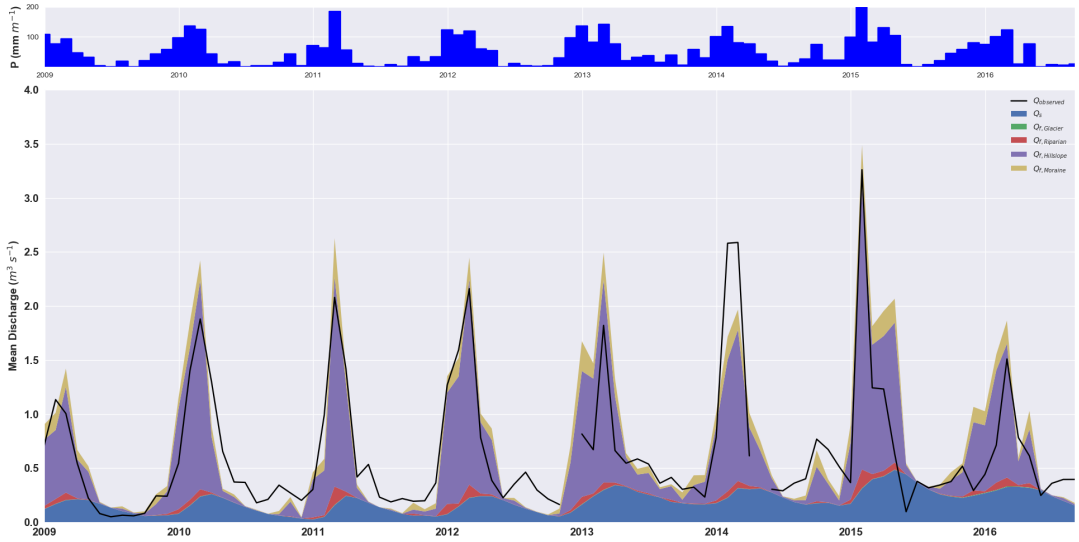


Figure 5.14: Monthly Hydrograph of the second calibration period. Modeled discharge is the discharge simulated with the parameters on the first period.

model represents in this case the reality quite closely. The observed hydrograph shows the same consistent pattern. Even during wet-season it can regress quickly towards what the model considers to be base-flow. The higher dry run-off in the second period is explained by a higher base-flow, but also by a continued fast run-off during the dry months. Especially in the years 2013, and 2015. Inspection of the monthly hydrograph shows that the model performs very well on the dry season run-off in especially calibration period 1.

On daily timescales one must look more at the patterns than the exact timing and magnitude. For example in terms of the speed of the rising and falling limps or precipitation peak absorption. Then the 2006 daily hydrograph could be considered a reasonable good match, see Figure 5.15. Especially the dry run-off is spot on. Note the complete absorption of the precipitation peak by the internal-states of the model; a significant rain event (close to 15 mm) leads to a very small discharge peak peak. A second rain event of only a third of the former depth results in an almost similar peak. Investigation of January to July 2006 shows that the regression seems to match quite well. Both the regression of the peak flows in the early months (K_f) as well as the falling limb towards July. One must note that there is a over-estimation of run-off however in this period.

It is obvious that the system is very fast responding. With its short flow-paths and pronounced topography

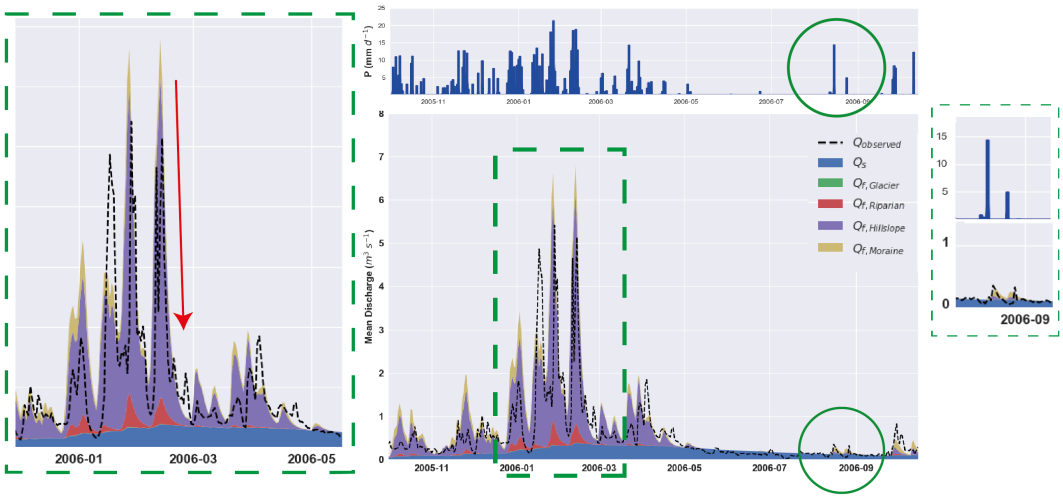


Figure 5.15: Daily Hydrograph 2006. Interesting features are the wet season regression. The simulated values follow closely the observed ones. In September 2006 a precipitation depth is almost completely absorbed by the basin.

and highly permeable soils the basin reacts almost instantly to peaks in precipitation. This also means that errors in precipitation quickly lead to errors in discharge. This is illustrated by Figure 5.16. The red arrows point to two moments in which this is very obvious. The data does not show any precipitation at all, while the measured discharge shows a peak of more than 1 $m^3 s^{-1}$ for a few days in the dry period of 2011. In July the next year the same situation occurs. Based on the hydrograph it seems likely that either the precipitation data is off, or the discharge data, or both.

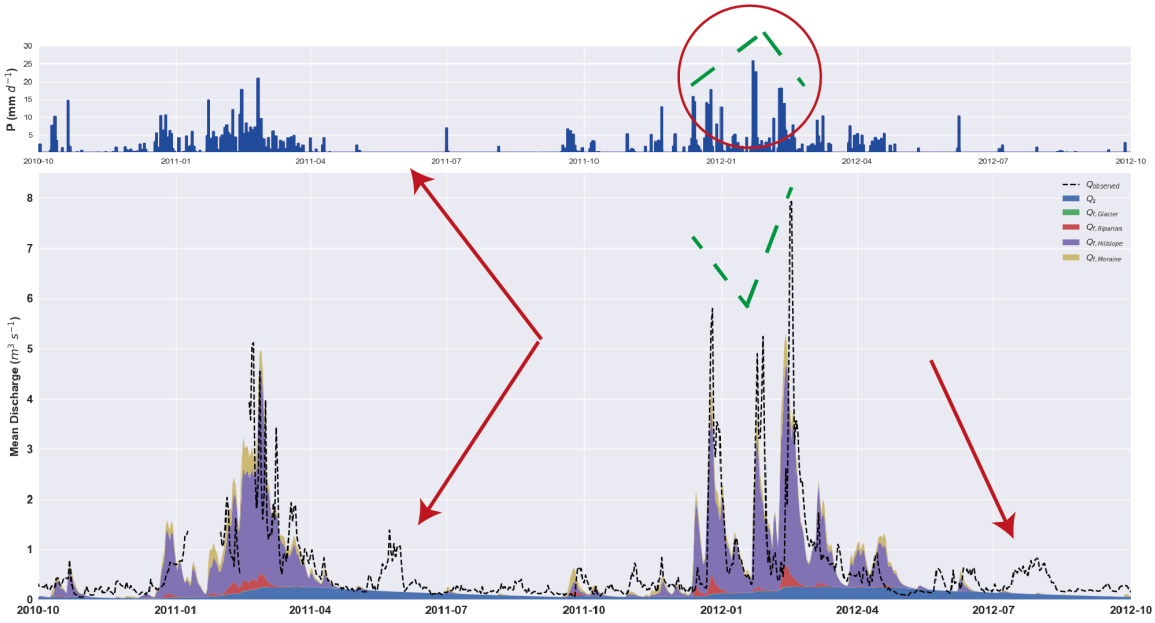


Figure 5.16: Daily Hydrograph 2011-2012. Modeled vs. Observed. The modeled hydrograph is split with respect to the discharge origin.

Upon closely observing the three major discharge peaks in December 2011, and January and February 2011 in Figure 5.16 a pattern is visible in the observed-, that is also visible in the modeled data; indicated by the green lines. Although the modeled values are significantly less then those observed, the pattern is the same: the highest precipitation peaks does not lead to the highest discharge peaks, but the other way around. This might be an indication that the model captures the reality here quite alright, even though the values are a bit off.

5.2. ENSO HYPOTHESIS

In this section the results of the Pearson correlation analysis are presented. In the first section the results of temperature analysis are addressed and in the second second those of the precipitation analysis.

5.2.1. TEMPERATURE

The Pearson correlation analysis is performed on the temperature data at station El Alto-Aeropuerto. This station has been chosen because it has a much longer observational coverage period, opposed to Alto Achachicala; 1962-2017 instead of 1999-2017. The assumption is that the variability in temperature at station used station is similar to the variability at Alto Achachicala, due to its proximity.

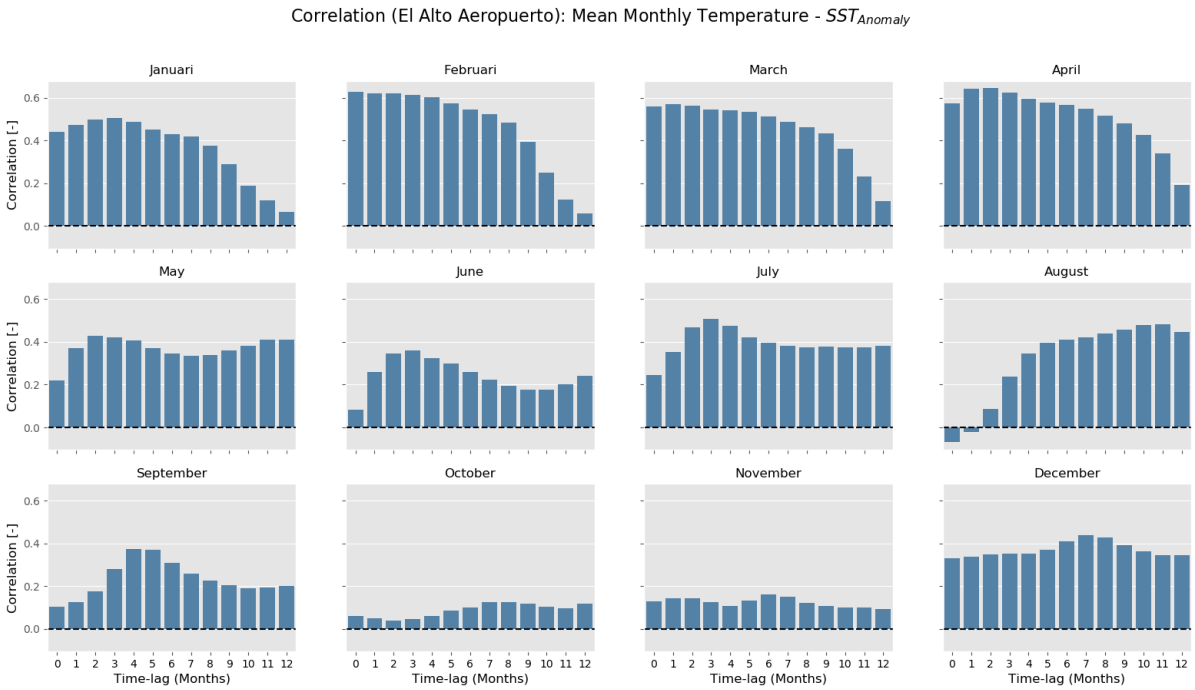


Figure 5.17: Pearson correlation between the local observed Temperature and ENSO-Index. Time-lag signifies the month of which index is taken. Temperature measured at station El Alto-Aeropuerto between 1962 and 2017. N=56 for each month.

It can be seen from Figure 5.17 the local temperature and the ENSO-Index. Generally speaking the correlation is positive, except for the first two values (for time-lags of 0 and 1 month) for August, which are negatively correlated. Table 5.10 that there is a significant relationship observable in 10 out of 12 months. This relationship is not significant for all values for October and November, and for the most time-lags of September. For the other months the relationship is significant, and positively correlated.

5.2.2. PRECIPITATION

The Pearson correlation analysis is performed on the temperature data at station Alto Achachicala. This station has been chosen because it the observational period is considered sufficiently long (1991-2016), in addition its the only station in the basin. Moreover and opposed to that of the temperature, the precipitation variability is considered to be more variable even on short distances. Mainly due to the influence of topography on orographic precipitation.

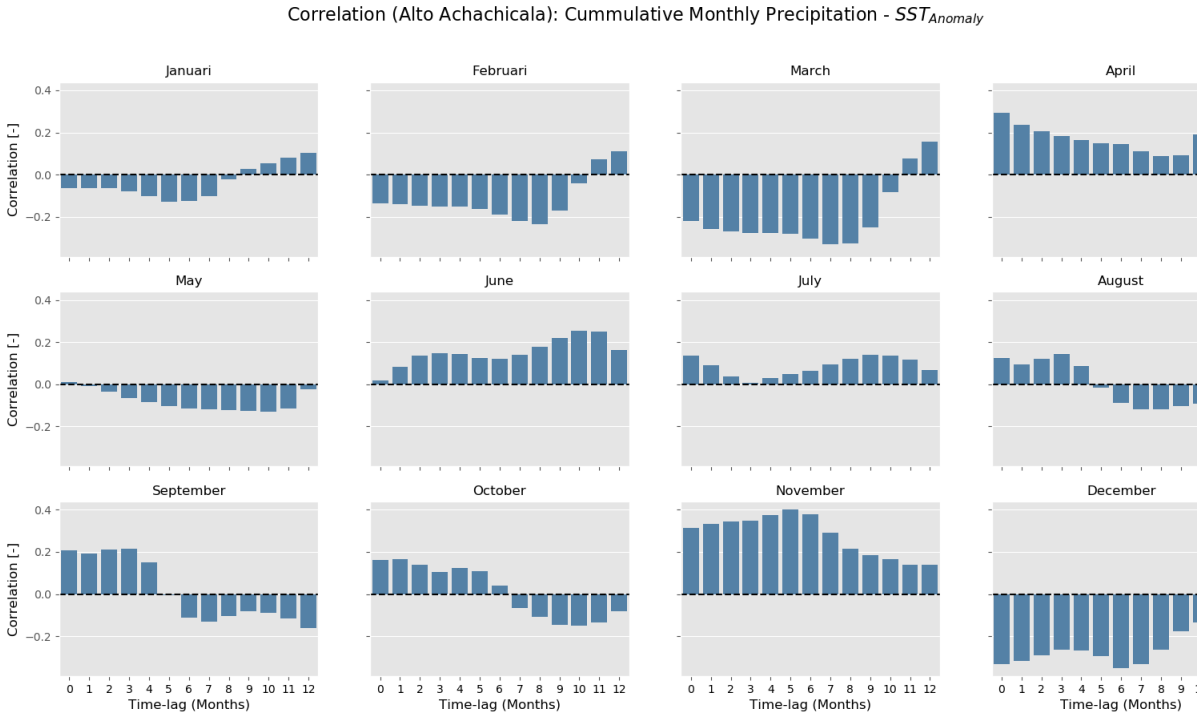


Figure 5.18: Precipitation correlation w.r.t. ENSO-Index. Time-lag signifies the month of which ENSO-index is taken to calculate the correlation. N=27, except for September(N=25), October (N=25), November (N=26) and December (N=26).

Looking at Figure 5.18 it is immediately obvious that the correlation is less strong than the temperature correlation. In most months it is absent or very weak. Another difference is that whereas temperature is mostly positively correlated, the ENSO-Precipitation correlation is often negative. This was to be expected as the warm phase (El Niño) episodes are associated with warm but dry years.
The correlation is weak however, and looking at Table 5.11 shows that the relationship is not significant in any of the months, for any of the considered time-lags at a significance level of $\alpha = 0.05$.

The fact that none of the calculated precipitation correlation can pass the significance test, leads to the rejection of $H_{1,P} \neq 0$ and the acceptance of the null Hypothesis $H_{0,P} = 0$.

Table 5.10: Significance test of the temperature correlation, at a significance level of $\alpha = 0.05$. Based on monthly mean temperatures measured at El Alto Aeropuerto ($N = 56$). Red cells indicate the correlation is not significant, while green cells are significant p-values.

	p-value											
Time-lag	**Jan**	**Feb**	**Mar**	**Apr**	**May**	**Jun**	**Jul**	**Aug**	**Sep**	**Oct**	**Nov**	**Dec**
0	0.00	0.00	0.00	0.00	0.10	0.54	0.07	0.61	0.45	0.66	0.35	0.01
1	0.00	0.00	0.00	0.00	0.00	0.05	0.01	0.88	0.36	0.72	0.29	0.01
2	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.52	0.20	0.78	0.30	0.01
3	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.08	0.04	0.73	0.36	0.01
4	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.01	0.00	0.66	0.44	0.01
5	0.00	0.00	0.00	0.00	0.01	0.02	0.00	0.00	0.01	0.54	0.34	0.01
6	0.00	0.00	0.00	0.00	0.01	0.05	0.00	0.00	0.02	0.47	0.23	0.00
7	0.00	0.00	0.00	0.00	0.01	0.10	0.00	0.00	0.05	0.36	0.26	0.00
8	0.00	0.00	0.00	0.00	0.01	0.15	0.00	0.00	0.09	0.36	0.37	0.00
9	0.03	0.00	0.00	0.00	0.01	0.19	0.00	0.00	0.13	0.39	0.43	0.00
10	0.16	0.06	0.01	0.00	0.00	0.19	0.00	0.00	0.16	0.45	0.46	0.01
11	0.38	0.36	0.09	0.01	0.00	0.14	0.00	0.00	0.15	0.48	0.46	0.01
12	0.63	0.67	0.40	0.15	0.00	0.07	0.00	0.00	0.14	0.39	0.50	0.01

Table 5.11: Significance test of the precipitation correlation, at a significance level of $\alpha = 0.05$. Red cells indicate the correlation is not significant. N=27, except for September ($N = 25$), October ($N = 25$), November ($N = 25$) and December ($N = 25$)

	p-value											
Time-lag	**Jan**	**Feb**	**Mar**	**Apr**	**May**	**Jun**	**Jul**	**Aug**	**Sep**	**Oct**	**Nov**	**Dec**
0	0.44	0.36	0.44	0.34	0.84	0.55	0.74	0.94	0.74	0.73	0.17	0.34
1	0.41	0.46	0.65	0.35	0.98	0.54	0.78	1.00	0.83	0.77	0.20	0.34
2	0.38	0.43	0.77	0.36	0.85	0.60	0.90	0.89	0.93	0.83	0.25	0.28
3	0.36	0.49	0.81	0.35	0.84	0.77	0.94	0.73	0.86	0.85	0.29	0.23
4	0.37	0.47	0.84	0.41	0.82	0.96	0.75	0.51	0.66	0.94	0.29	0.21
5	0.37	0.47	0.84	0.41	0.79	0.93	0.55	0.34	0.51	0.69	0.31	0.17
6	0.38	0.51	0.85	0.39	0.74	0.77	0.47	0.29	0.50	0.47	0.45	0.11
7	0.39	0.61	0.94	0.38	0.69	0.76	0.35	0.30	0.56	0.46	0.65	0.09
8	0.43	0.61	0.91	0.39	0.66	0.80	0.39	0.34	0.69	0.48	0.79	0.11
9	0.51	0.51	0.76	0.41	0.73	0.87	0.47	0.35	0.95	0.52	0.79	0.17
10	0.72	0.52	0.72	0.31	0.89	0.89	0.59	0.32	0.95	0.43	0.80	0.23
11	0.93	0.67	0.60	0.24	0.95	0.82	0.66	0.30	0.96	0.44	0.77	0.26
12	0.90	0.84	0.54	0.26	0.91	0.74	0.80	0.29	0.92	0.49	0.72	0.24

5.3. APPLICATION ON NEW KALUYO SYSTEM

In this final section the behaviour of the new Kaluyo system is presented. First the modeled hydrology of the sub-catchments with the focus is on the noticeable differences between the sub-catchments. Thereafter, the results of the ISM is presented.

5.3.1. HYDROLOGY INVESTIGATION THROUGH FLEX-TOPO

Figure 5.20 shows the average monthly total discharge per unit area, for each of the sub-basins. It can be seen that the differences in flow contribution are significant. When considering the RCF for each basin (Table 5.12 this is even clearer. Sub-basin Jumalincu, or C04 has the highest contribution to the flow, relative to its area. With a $RCF = 1.9$ (period: 2001-2015), the basin contributes almost two times more than expected based on its area. On the other side of the spectrum there is the lower-area of Kaluyo - the basin C00 - with a $RCF = 0.6$.

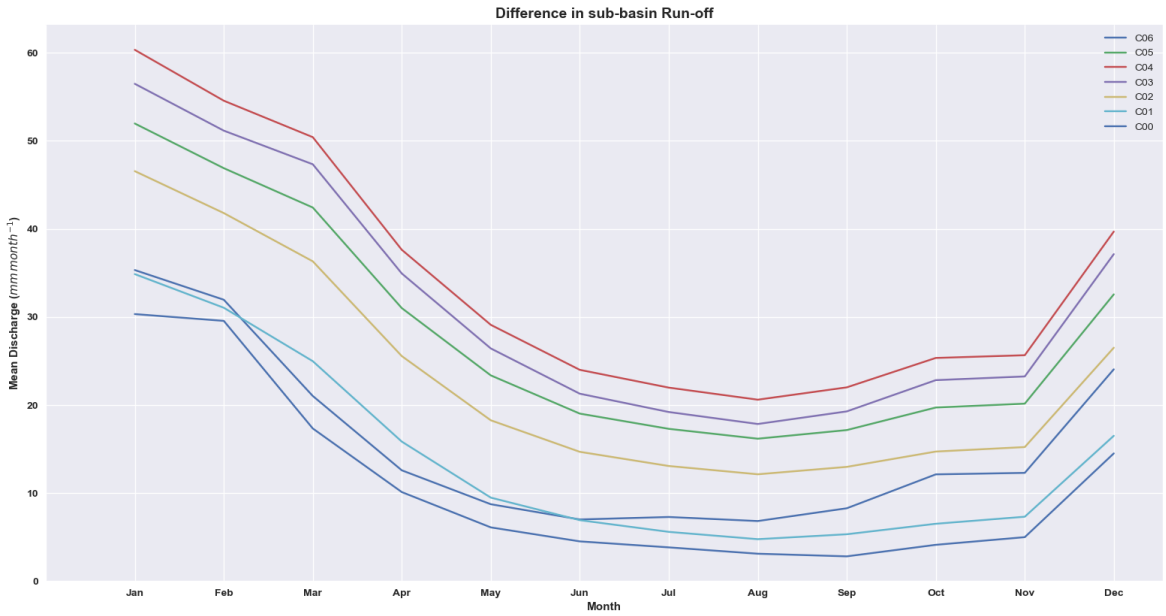


Figure 5.19: Modelled mean monthly runoff of the new sub-basins, per unit area.

Modelled mean runoff per calendar month of the new sub-basins, per unit area. (2001-2015)

Table 5.12: Average discharge contribution of each sub-basin. Both in terms of relative contribution per unit area, as well as absolute contribution to the discharge. RCF = Relative Contribution Factor.

	Mean Discharge				Area	RCF
Basin	**(mm d-1)**	**(mm a-1)**	**(m3s^{-1})**	**(%)**	**(%)**	**(-)**
C00	0.36	131	0.160	20.0	32.3	**0.6**
C01	0.46	169	0.179	22.4	28.1	0.8
C02	0.76	278	0.074	9.3	7.1	1.3
C03	1.03	377	0.160	20.1	11.3	1.8
C04	1.13	411	0.085	10.6	5.5	**1.9**
C05	0.92	338	0.070	8.8	5.5	1.6
C06	0.51	187	0.071	8.9	10.1	0.9

When looking closer and consider the monthly RFC for each sub-basin, it is clear that the higher situated basins contribute more to the flow than the lower ones. A possible explanation is the presence of less vegetation, relatively more Hill-slope and Moraine HRUs. In the case of C04 (basin Jumalincu) the relative high Glacier contribution might also be a factor for the high, dry period RFC values. For the basins with a low contribution to the flow, especially C00 and C01 one explanation might be found in the higher presence of

riparian zones leading to more transpiration. Especially in the dry periods. With the groundwater pushing upwards, these areas receive almost a constant flux of water; water that will be evaporated.

Table 5.13: Mean RFC values on a monthly time-scale. It can be seen that the higher situated basins contribute more to the flow than the lower ones. A possible explanation is the presence of less vegetation, relatively more Hill-slope and Moraine HRUs. In the case of C04 (basin Jumalincu) the relative high Glacier contribution might also be a factor for the high, dry period RFC values.

	$RFC_{Monthly}$: **2001-2015**						
	C00 (-)	**C01** (-)	**C02** (-)	**C03** (-)	**C04** (-)	**C05** (-)	**C06** (-)
Jan	0.8	0.9	1.2	1.4	1.5	1.3	0.9
Feb	0.8	0.9	1.2	1.4	1.5	1.3	0.9
Mar	0.6	0.9	1.3	1.7	1.8	1.5	0.8
Apr	0.5	0.9	1.4	1.9	2.0	1.7	0.7
May	**0.5**	0.7	1.4	2.1	2.3	1.8	0.7
Jun	**0.5**	0.7	1.5	2.1	**2.4**	1.9	0.7
Jul	**0.4**	0.6	1.5	2.2	**2.5**	2.0	0.8
Aug	**0.4**	0.6	1.5	2.2	**2.6**	2.0	0.9
Sep	**0.3**	0.6	1.5	2.3	**2.6**	2.0	1.0
Oct	0.4	0.6	1.4	2.2	2.4	1.9	1.2
Nov	0.4	0.7	1.4	2.1	2.3	1.8	1.1
Dec	0.7	0.8	1.2	1.7	1.8	1.5	1.1

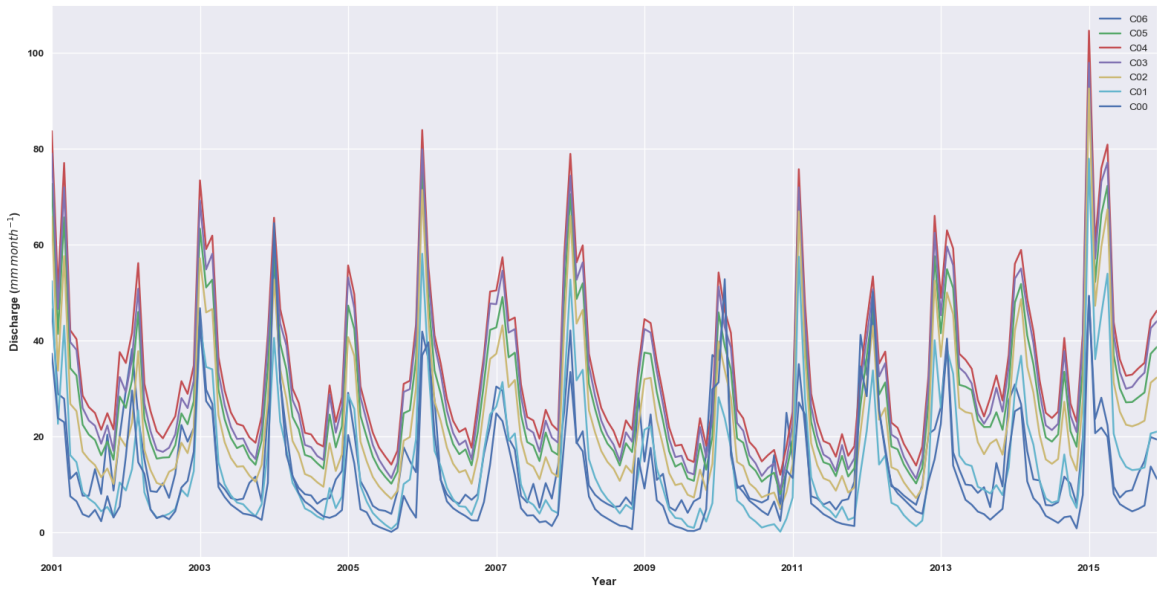


Figure 5.20: Time-series of the monthly accumulated discharge of the new sub-basins, per unit area.

5.3.2. HINDCASTING THROUGH THE ISM

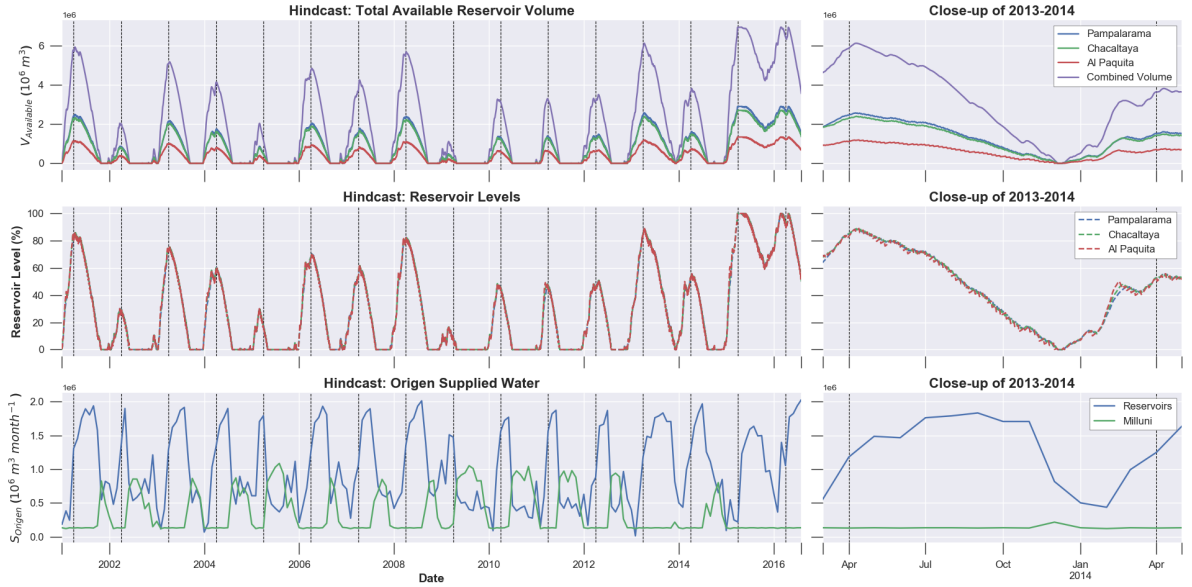


Figure 5.21: Overview of reservoir dynamics of the hindcast, for demand the 2016 demand scenario and a maximum river intake of 50% of the streamflow.

It can be seen that the maximum fraction of stream-flow that can be abstracted is determining to a great extent the available volumes over time. For high (here 0.7) values of $f_{intake,max}$ the reservoirs fill up to their maximum capacity at the end of 10 out of 16 seasons (not in 2006, 2010, 2011 and 2012), while this is only the case in two cases for a maximum intake of 50% and in none of the years for '$f_{intake,max} = 0.3$.

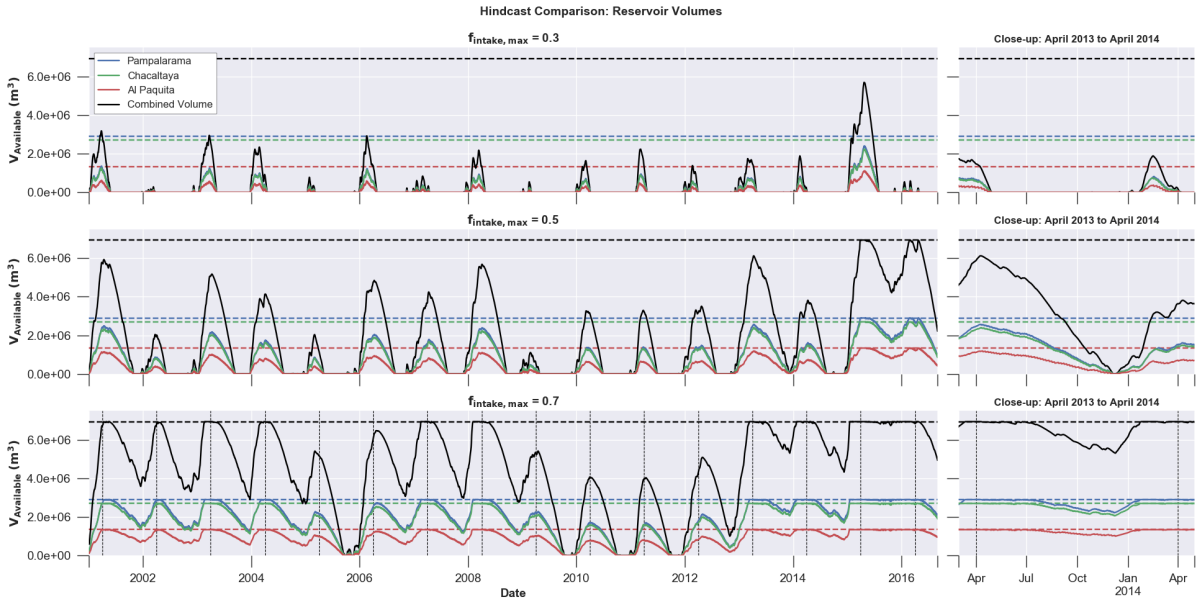


Figure 5.22: Comparison of reservoir volume dynamics for different values of $f_{intake,max}$. Demand scenario 2016 is shown.

Table 5.14 shows supplied Kaluyo volumes to the WTP, for the three intake settings and compares them to the old system. Depending on the maximum intake, the system provides between the 5% and 45% more than it used to do. For the a maximum river intake of 70% for 10 out-of 14 full simulated periods the Milluni supply was equal to the base supply of 10%. It can be seen that the effect of the reservoirs is more pronounced for increasing values of $f_{intake,max}$. For intake settings of 0.7, the extra required Milluni supply is zero except for the years mentioned above.

Although the reservoirs have some effect at $f_{(intake,max)} = 0.3$, it is clear from the figure that the reservoirs are not fully utilized. Because the required base-flow is so high, the reservoirs are almost continuously required to supply and not able to build up a buffer. One could argue that with low maximum river intake values the reservoir do not serve their purpose at all; their purpuse being to distribtue the river discharge more evenly over the year. Even for the situation that 50% of the discharge may be taken in by Achachicala, only in two out of the 14 years the reservoirs fill up to their maximum volume. In non of the years, even for the highest intake value, do the reservoirs fill-up from being empty to being completely full.

Table 5.14: Effect of changes in maximum river intake settings, and presence of reservoirs on the total Kaluyo supply to the WTP Achachicala. All values are in $10^7\,m^3$. *New* represents the situation with- and *Old* without reservoirs present.

		$f_{(Intake,max)} = 0.3$			$f_{(Intake,max)} = 0.5$			$f_{(Intake,max)} = 0.7$		
Start	**Demand**	**New**	**Old**	**dV**	**New**	**Old**	**dV**	**New**	**Old**	**dV**
Apr '02	1.54	0.74	0.63	0.11	1.26	0.92	0.34	1.39	1.12	0.26
Apr '03	1.55	0.70	0.67	0.03	1.05	0.91	0.14	1.39	1.08	0.31
Apr '04	1.54	0.78	0.64	0.14	1.18	0.89	0.28	1.39	1.10	0.29
Apr '05	1.54	0.61	0.55	0.06	1.09	0.82	0.27	1.39	1.03	0.35
Apr '06	1.54	0.64	0.58	0.06	0.88	0.79	0.09	**1.33**	0.93	0.40
Apr '07	1.55	0.78	0.68	0.10	1.24	0.96	0.28	1.39	1.13	0.26
Apr '08	1.54	0.76	0.69	0.07	1.16	0.93	0.23	1.39	1.11	0.29
Apr '09	1.54	0.63	0.56	0.06	1.20	0.85	0.35	1.39	1.06	0.32
Apr '10	1.54	0.61	0.55	0.06	0.80	0.75	0.05	**1.29**	0.89	0.40
Apr '11	1.55	0.54	0.50	0.03	0.89	0.72	0.17	**1.20**	0.89	0.31
Apr '12	1.54	0.67	0.56	0.11	0.97	0.79	0.19	**1.24**	0.95	0.29
Apr '13	1.54	0.76	0.72	0.04	1.14	0.95	0.19	1.39	1.09	0.30
Apr '14	1.54	0.83	0.73	0.10	1.38	1.04	0.34	1.39	1.23	0.15
Apr '15	1.55	0.76	0.73	0.03	1.20	0.99	0.20	1.39	1.16	0.22
	Mean	0.70	0.63	0.07	1.10	0.88	0.22	1.35	1.06	0.30
	Maximum	0.83	0.73	0.14	1.38	1.04	0.35	1.39	1.23	0.40
	Minimum	0.54	0.50	0.03	0.80	0.72	0.05	1.20	0.89	0.15
	Std	0.09	0.08	0.04	0.17	0.10	0.09	0.06	0.10	0.07

6

DISCUSSION

6.1. FLEX-TOPO MODEL

DATA AND DATA MANIPULATION

The lapse-rates - both for precipitation and temperature - are highly debatable. But the alternative is to use no lapse rates at all, or to make them a free calibration parameter. Especially with respect to precipitation the local consensus was that there was more precipitation on higher altitudes then lower.

Another topic of discussion is the arbitrary division of the basin into two parts: *Kaluyo Upper* and *Kaluyo Lower*. First of all, the station El Tejar is located fully outside of the basin. Secondly, the division line between the two parts has been made based the consideration that a discharge gauge will be placed on this location. Thus in doing so, the modeling results can be compared in the future to discharge measurements. An other, arguably better way to cut the basin be to do it through a geo-statistical method such as Kriging.

The results of NDVI per HRU and Irradiance zone show the expected results. This is a clear indication that the NDVI threshold in the classification of $HRU_{Hillslope}$ and $HRU_{Moraine}$ was appropriate; and that using this data source has helped to reflect reality with respect to the HRU distribution. NDSI observations were ineffective in the Glacier classification due to the large pixel-size, and small glacier area.

CALIBRATION

With the aforementioned data quality issues it would have been better to apply a k-fold cross validation, with for example five folds. Application of this method would have randomly sampled the original time-series into k parts. In calibrating the model based on these k-folds, and subsequently validation thereon would have provided more insight into the uncertainties. On the other hand, even with only a split sample, the calibrated model parameters did converge. Moreover, although it performed quite poorly on the second period it did so in both calibration- and validation on the second period. The derived parameters are very close to those of the first period; the model just was not able to simulate the observed data. Remarkably calibration on the second period led to validation performance measures that were very close to those calibrated on the first. This could be seen as a very good result. It is not unlikely that the data quality in the second period is just below par. Remembering the first rule about modelling: *Garbage in is Garbage out.* This two faced performance pattern can also indicate that the model set-up is more suited to simulate the behaviour of the first period.

Time-lag: The signal of the time-lag is not clear. Therefore, with hindsight it might have proven more useful and insightful to have chosen an value based on analysing the observational times P and Q data, or to just omit the time-lag all together, given the small catchment size and therefore short flow paths.

Internal NDVI-Transpiration calibration: the derived method of evaluation of the observed NDVI values to the transpiration proved insightful. The results showed that the performance difference of parameter combinations were clearly identify-able. It was informative especially on the parameter combination that resulted in low availability of water. Approaching larger effective unsaturated storage values the objective

function lost its distinguishing value. Distinct bad performance for too small values of Ce and Su_{max}, however after threshold all values perform equally good. That could be explained because the objective function evaluated the performance in the frequency domain, and not in the sequential time domain we live in. In order to also be able to constrain the values on the upper-limit, a performance measure that considers the exact timing of the NDVI versus the Transpiration dynamics is required. Because the observational interval of MODIS NDVI data is 16 days this proved to be quite difficult for this basin. Because the basin is small, precipitation data scarce and the basin very fast responding; it proved to be difficult to re-sampling of the data and transpiration in such a manner that the evaluation made sense.

With hindsight, the author also debates whether the right cut-off values have been chosen. First of all the assumption is made the the C_e for both HRU was the same. Secondly, again with hindsight, it could be argued that especially the Riparian zone saturated zone is too large; seeing that the water is at the surface. The internal evaluation is less well defined for the Riparian zone. Probably due to the additional parameter C_r. This introduces an extra degree of freedom. And due to the constant groundwater pushing upwards a very influencing for the total amount of evaporation.
All things considered, for the Kaluyo basin it makes sense to include an internal evaluation based on NDVI, mainly because the vegetation patterns are so pronounced. Comparing the transpiration and NDVI in the frequency domain makes sense, but it is just a start. More information lies in the exact timing. With more time and dedication one could explore this path into much more depth.
Fresh Snow: It does not make much difference if the T_t is -1 or -10, for all HRUs. This indicates that either the used method is not very good, or there is another explanation. One is provided by observations of local inhabitants; they told the author that generally snow disappears within the day. Also the MODIS measurements are evaluated per HRU. a better way would be to perform the evaluation per elevation zone; and in doing so make the T_t a correction might better be made for each elevation zone, with a Fdd compensating for the HRU specific effects of the local conditions (e.g. presence- or absence of vegetation or exposure to wind) on the snow melt.

Glacier calibration: constraining the glacier retreat to one study that used remote sensing techniques to evaluate the glacier masses of a large part of the Andes is very shaky at least. However, the only other way would be solely based on literature. There is little known about the glaciers in Kaluyo, other than Glacier Chacaltaya disappearing. Besides, due to the small area the contribution to the discharge is not large enough to assess this separately.

HYDROLOGY & REALISM

A large part of the hydrology has already been discussed in the results chapter. The simulated hydrology makes sense overall; the model combined with the calibrated parameters simulates the observed discharge patterns quite well. One of the interesting findings is that the location on the Budyko Curve, which is outside of the theoretical bandwidth based on the data, is closely simulated. On beforehand the assumption was made that either the measured discharge, or the measured precipitation was far off. But with an total RMSE of just 0.09 between to the modelled run-off coefficient and the observed RC this assumption must be rejected.

One of the goals of the hydrological model development was to make a *realistic* model. Realistic in the sense that it represents the actually occurring processes reasonably well. Whether the model actually succeeds in this goal is quite difficult to determine. Either one declares victory upon acceptable model outputs, or tries to verify with in situ measurements or comparative studies.

The first topic of discussion in model realism traces back to maybe the single most influential model assumption: the assumption that precipitation that falls on the Glacier or Moraine bypasses the hillslope all together. To recap, this is made on perception that water either drains quickly through fast (sub-)surface run-off processes, or percolates to the groundwater reservoir.

A study by Cooper investigated ten very similar bofedahl wetlands in Bolivia and Peru. Actually, two of the ten sites fall are also source regions for the water supply of La Paz: the basins *Lower Tuni* and *Outcrop Tuni*. The study analysed $\delta^{18}O$ and δD stable isotopes ratios. Using this method its possible to identift the *age* of the water in the samples. The study concluded that *"Topographic and groundwater elevations in the peatlands demonstrated that the water source of all 10 peatlands was hillslope groundwater flowing from lateral*

moraines, talus, colluvium, or bedrock aquifers into the peatlands. There was little to no input from streams, whether derived from glacier melt or other sources, and glacier melt could not have recharged the hillslope aquifers supporting peatlands ". [45] In essence this fully supports the assumption. As described, the HRUs only communicate through the groundwater reservoir, and not via streams. Fast HRU runoff is directly routed towards the *total* stream, whereas groundwater pushing up in the riparian zone is the only other influx next to rain. This could also be part of an explanation as to why the difference between the Hillslope and Riparian vegetation evaporation is so large (Riparian evaporation is about three times larger), the next topic of model realism discussion.

The 3 : 1 ratio of riparian vs. hillslope evaporation is quite large. Roughly speaking the wetland evaporation approaches the potential (Hargreaves) evaporation. On first glance this is not fully supported by the NDVI data. However, as described in Chapter 2, NDVI values are indicative. One possible explanation can be found in difference in soil types. Dry soils or bare rock are associated with NDVI values between 0.5 and 0.15, wet peat soils and open water bodies however have negative NDVI values. Water can approach $NDVI = -1$. Due to large presence of moisture in peat, these soils can have NDVI of around -0.20 to -0.10. With MODIS NDVI pixels of 250 by 250 meters, average NDVI values might be lower than the evaporation indicates. This potential explanation is supported by one of the three study site used in a study by White[46]. Where ephemeral springs had higher NDVI values for the same vegetation cover compared to perennial sites. A possible way to improve the method would be to extent the NDVI with a NDWI (Normalized Difference Water Index) analysis. This measure is more sensitive to the presence of moisture.

A method was found to link NDVI values to evaporation estimations, derived by Groeneveld. This method adjusts the NDVI to $NDVI^*$ based on the $NDVI$ at zero vegetation cover ($NDVI_0$) and NDVI at saturation ($NDVI_s$). [47] No full analysis was made but, when applying the assumptions above ratios between the dry hillslopes and bofedahls were achieved that were comparable to the modeled values. This would be a topic for further research.

The last model *realism* assessment is with respect to the glacier sublimation. With 306 mm, sublimation is around two times larger than the $166 mm \cdot a^{-1}$ estimated by Ramirez [16] for the Chacaltaya glacier. But with a daily rate of $0.84 mm \cdot d^{-1}$ slightly below the lower-limit ($1.0 mm \cdot d^{-1}$) of high glacier surface sublimation indicated by MacDonald [34]. The latter is a very detailed study, whereas the former is based on a simple water-balance. Therefore the simulated rate is not considered unreasonable at worst, and reasonable at best.

6.2. COMPONENT 2: ENSO & LOCAL CLIMATE

To draw conclusion on the ENSO variability in relation to the local climate, the Pearson-Correlation analysis is arguably both too thin or simple-, as well as extensive enough at the same time. The temperature relationship is significant, as could be expected as this is generally more stable and predictable than precipitation. However the temperature is a less impact-full on operations of water supply systems. With respect to the precipitation relationship. Looking at the precipitation relationship, the outcomes are also what was to be expected. Extensive research done towards this topic shows that even for sufficiently large areas it hard to find observable, significant and reliable relationships. [48][21][22][23] Even products are developed by large teams, with a lot of funding that forecast based on the ENSO-Index note that results are highly uncertain, and can only be used on areas in the order of at a minimum $100 \cdot 10^3 km^2$.[18] The Kaluyo basin is a factor 1000 smaller. In addition it is essentially situated on the moisture divide between the arid Altiplano and humid Amazon basin. The topography is complex, with mountain peaks over 6000 m. It is save to assume that thus atmospheric conditions are at least as complex. Moreover, the precipitation-and weather station network is thin, with large observational temporal intervals and known issues with respect to the quality [10].

Although it is tempting to relate El Niño and La Niña to dry respectively wet years, this can not be safely used to base operational decision for a Water Supply System on. Except for conservative decisions aiming to prepare for dry-spells. Because although the relationship is not significant, conservative action in water management must always be encouraged.

6.3. COMPONENT 3: UNGAUGED RESERVOIR HYDROLOGY & DYNAMICS

As every other model; the integrated system model is a simplification of reality. Most notably is that open water evaporation is not included in the model. One could argue, that as the evaporation in the riparian zone approaches the potential evaporation this might not make a large difference. On the other hand the surface area of the reservoirs extent into the hillslopes, where evaporation is significantly less.

In addition the introduction of the $Urban_{HRU}$ is problematic, as it is a completely un-calibrated model set-up, and because its large contribution to the sub-basin **C06**, with more than 51 %. This is could also be seen as a argument for including it. It is clear that the an urban environment functions completely different than hillslopes or the riparian zone: a lot of paved surface and sparse vegetation. Therefore, in combination with the simple model set-up and fixed parameters the decision was made to include it.

As can be seen from Table 5.12 its run-off contribution is not remarkably different than the other lower lying basins. With *RCF* value it close to one on the long-term mean one could argue that the model set-up is at least producing out of the ordinary values. Relative to basin *C00* and *C01* its relative contribution is slightly higher, what is to be expected without the presence of vegetation. Note that the highest RCF values are found on higher elevations. This is partly explained by increasing precipitation.

Sub-basins Jumalincu (C04) has the highest relative contribution to the discharge of all basins. This can be explained by the small riparian (1.5%) zone, relatively large glacier area (3.0%) large moraine (53.7%). In addition it has the second lowest relative hillslope fraction. Taken together, this sub-basin has a low capacity for transpiration, and high capacity for run-off generation. Moreover because both the Moraine zone is very large, it generates a lot of slow run-off as well. Therefore it produces run-off the whole year round, and consistently more then the other basins. The Toma Jumalincu bifurcation towards reservoir Pampalarama makes thus a lot of sense. All sub-basins contributing to a reservoir have RCF of more than 1.0, with Al Paquita being the lowest with 1.3. The dry period run-off is for basins Chacaltaya, Jumalincu and Pampalarama all significantly higher than the other basins. It makes therefore sense that when determining which reservoir must be drained first, to choose for on either Chacaltaya or Pampalarama in stead of Al Paquita. Although the modeled dynamics first must be validated by actual observations to be sure.

From the results it is obvious that the effect of the reservoir to the total Kaluyo supply is very much dependent on $f_{Intake,max}$ values. In four year the total available volume hits zero, for intake settings of 0.7. In the cases that reservoirs are at maximum capacity at the end of the wet season, not once the available volume because zero during the next dry season. In 10 out of 14 years, the additional Milluni supply is zero. Comparing supplied $Millui_{Base}$ volume to the remaining reservoir volume, leads to the conclusion that in 9 of those 10 years the WTP Achachicala supply system was completely independent from basin Milluni for its supply, which was never the case in the old system.

7

Conclusions and Recommendations

Component 1

The research question relating to the first component are:

1. What are the dominant hydrological processes in the basin, and where are flows created?

2. Can auxiliary data measurements be used in model development and calibration to increase the reflection of reality, and decrease the dependence on discharge data in calibration?

Based on the outcomes of the FLEX-Topo model, the following conclusions with respect to the hydrology of the Kaluyo basins, and the dominant flow-processes are drawn. Fast runoff is the most dominant flow process, accounting for almost 75% of the total discharge. The grass hillslopes have the largest contribution to the fast run-off both in relative and absolute terms. Where the fast runoff is mostly generated on the grass hillslopes, the flux towards the groundwater is mainly generated on the barren, moraine mountain peaks. Here interception evaporation is low, and through the highly permeable soils water drains towards the groundwater reservoir. Transpiration occurs on the hillslopes during the wet period, and in the riparian zone all year round. In the latter zone the transpiration is close to the potential evaporation. The contribution of the glacier is less than a percent, and thus insignificant.Total glacier disappearance would not result in a significant stream-flow reduction for the Kaluyo basin. On the glaciers, sublimation is estimated to account for one-third of the outgoing water flux. Dry period run-off is almost exclusively contributed to the groundwater reservoir. All modeling sensitive to the precipitation input and due to the scarcity of precipitation locations it is advised to treat the results of this research carefully;
The NDVI observations have proved to help distribute the basin with respect to its ecosystems, and thus in increasing the reflection of reality in the model. Comparative literature studies indicate that the main assumptions underlying the model reflect reality. NDVI is sensitive to moisture content, therefore it is recommended to elaborate the analysis further with for example NDWI data. NDSI observations were not effective in the Glacier classification, due to the small glacier size. NDVI observations have also proved useful in calibration. It provided insight into lower limits of model parameters associated with transpiration. Upper limits were not able to be identified, due to the evaluation technique only assessing the transpiration dynamics in the frequency domain.

Snow and glacier related model parameters were constrained using NDSI observations and literature studies with respect to glacier retreat. The latter was useful to prevent unrealistically high glacier retreat. Discharge calibration was not possible due to the small glacierized area, this auxiliary source helped to increase the reflection of perceived reality as well, but without actual glacier measurements it is not certain the perceived reality is correct.

With respect to the NDSI observations the results show that the best performance was achieved when minimizing snow presence over the basin. Based on local observations this is in line with reality as thus must be concluded that the NDSI as auxiliary data source helped to reflect reality. Because of the limited relevance of snow cover and melt in the basin to the hydrology the NDSI observations did not reduce the dependence on

discharge data for calibration.

This was the case for the NDVI measurements however. They helped to identify values for $C_{r,max}$ and combinations of unsaturated reservoir storage. It is recommended however to improve calibration on the internal transpiring states with a additional performance measures on the NDVI dynamics in the sequential time domain to fully utilize the information.
This research offers a deterministic model, while more information on uncertainty and confidence bounds is achieved with a probabilistic model result. Such as an ensemble of possible model parametersets. It is recommended - in order to reduce uncertainty - to further expand this research in a probabilistic manner.

COMPONENT 2

"Can a statistically significant relationship between the local climate variability and fluctuations in the ENSO-Index be found?"
With research question of **Component 2** and based on the Pearson-Correlation analysis at a significance level of $\alpha = 0.05$, the following conclusions are drawn. There is a significant relationship between the ENSO-Index and the mean monthly Temperature measured at station El Alto Aeropuerto. This relationship is present in 10 out of 12 months, depending on the considered time-lag. The temperature is positively correlated, in all cases that the relationship is significant.

Based on the Pearson-Correlation analysis for precipitation it is clear that the there is no significant linear relationship between the monthly ENSO-Index and the monthly precipitation measured at station Alto Achachicala. Therefore, one must conclude that the ENSO-Index can not be used to base operational decisions on. It is recommended to still consider it and raise alertness if the system is already in a critical state, as there is a negative correlation present in most months; it is just not significant. Therefore it can be used to support decisions that have a conservative goal, in other words are to increase drought preparedness, not the other way around.

COMPONENT 3

1. What is the relative contribution of the sub-basins to the total flow?
2. What is the impact of the reservoirs, in terms of water supply increase and dependence on water from the Milluni basin?

Based on the simulated maximum intake settings the construction of the reservoirs has led to an relative increase of between 5% and 45% of yearly Kaluyo Supply, and absolute supply increase between 0.3 and $4.0 \cdot 10^6 m^3 \cdot a^{-1}$. Where the biggest increase is achieved for the highest intake values. In order to make the most use of the reservoirs the river intake must be 50% or more. The hind-cast shows that otherwise the reservoirs do not fill up sufficiently during the wet period to perform their function; distribution of available supply during the dry period. It is therefore recommended to do use a minimum intake percentage of 50%. The hind-cast also clearly shows that the dependence on Milluni supply is reduced. In the case of a maximum river intake values of 70% the WTP Achachicala is completely independent from the Milluni supply in 10 out of 14 years.

The contribution of each sub-basins ranges between 8.8% for basin C05 (Pampalarama) and 22.4 % for basin C01. Most noticable are the contributions of basin C03 and C04, accounting for 20.1 and 10.6 % of the runoff, while their area is only 11.3 and 5.5 of the basin. Resulting in the RCF values of 1.8 and 1.9. During dry periods their relative contribution is even higher, up to $RCF = 2.6$. The lowest relative contributions are found in the basins C00, C01 and C06 in contributing only RCF=0.6, RFC=0.8 and RCF=0.9. The RCF is highest for the sub-basins contributing to the reservoirs. This can be attributed to their HRU distribution and high elevation.

FINAL REMARKS

It is shown that it makes sense to use a process-based, ecosystem driven model-structure to investigate the hydrology of ungauged sub-basins based on the hydrology of the whole. The introduced realism by accounting for different HRU classes makes it possible to calibrate on additional data-sources and internal states in

a distributed manner. Furthermore, the reflection of reality allows to evaluate model outputs with respect to real-life, measurable phenomena . This can lead to increasing confidence in the model. The results show great differences between sub-basins that must be accounted for upon making operational water management decisions. The developed FLEX-Topo model is a appropriate tool to do so. The FLEX-Topo modeling concept provides more information on the distribution of flows within the sub-basin. It is recommended to further investigate the reservoir dynamics and sub-basin hydrology as more data is currently gathered. The model can easily be adjusted and re-calibrated if necessary. One of the major advantages of this flexible model is that the HRU distribution can be changed and adjusted. Investigative studies with respect to the effects of land-use due to e.g. climate change can be facilitated.

A

INPUT DATA: FLEX-TOPO

[]

A.1. HRU DISTRIBUTIONS

The internal HRU distributions are given in Table A.1

Table A.1: Internal HRU Distribution with respect to elevation, and irradiance classes.

		Sub-Basin: Kaluyo Lower							
Elevation Class		Riparian		Hillslope		Moraine		Glacier	
ID	Elevation *(m)*	(IR_{Low}) *(%)*	(IR_{High}) *(%)*	(IR_{Low}) *(%)*	(IR_{High}) *(%)*	(IR_{Low}) *(%)*	(IR_{High}) *(%)*	(IR_{Low}) *(%)*	(IR_{High}) *(%)*
1	**3750-4000**	1.5	1.6	0.5	0.5	-	-	-	-
2	**4000-4250**	19.8	19.8	9.1	9.1	-	-	-	-
3	**4250-4500**	17.8	16.7	20.6	23.0	-	-	-	-
4	**4500-4750**	13.3	9.6	21.8	15.2	25.0	24.4	-	-
5	**4750-5000**	-	-	0.2	0.0	12.3	38.3	-	-
6	**5000-5250**	-	-	-	-	-	-	-	-
7	**5250-5500**	-	-	-	-	-	-	-	-

		Sub-Basin: Kaluyo Upper							
Elevation Class		Riparian		Hillslope		Moraine		Glacier	
ID	Elevation *(m)*	(IR_{Low}) *(%)*	(IR_{High}) *(%)*	(IR_{Low}) *(%)*	(IR_{High}) *(%)*	(IR_{Low}) *(%)*	(IR_{High}) *(%)*	(IR_{Low}) *(%)*	(IR_{High}) *(%)*
1	**3750-4000**	-	-	-	-	-	-	-	-
2	**4000-4250**	-	-	-	-	-	-	-	-
3	**4250-4500**	30.2	31.6	9.3	7.1	-	-	-	-
4	**4500-4750**	19.9	17.7	31.2	36.6	6.3	6.8	0.0	0.0
5	**4750-5000**	0.3	0.3	7.9	7.9	31.9	34.7	18.7	18.7
6	**5000-5250**	-	-	-	-	9.7	8.9	31.3	31.3
7	**5250-5500**	-	-	-	-	0.9	0.8	-	-

A.2. EVAPORATION

Based on the Hargreaves equation the provided evaporation measurements can be adjusted for each elevation class, using the lapse-rate adjusted temperatures.

$$E_{P,Hg} = 0.0023 * R_A * \Delta_T^{0.5} * (T_{mean} + 17.8)$$

By calculating the Δ_T based on the lapse rates($\gamma_{T,mean}$, $\gamma_{T,mim}$, and $\gamma_{T,max}$) mentioned in section 2.1.2, values for elevation class adjusted T_{min} and T_{max} , and the T_{mean} are derived for each day. These values are combined with the elevation and calculated irradiance factor f_G to give an approximation of the class specific potential evaporation.

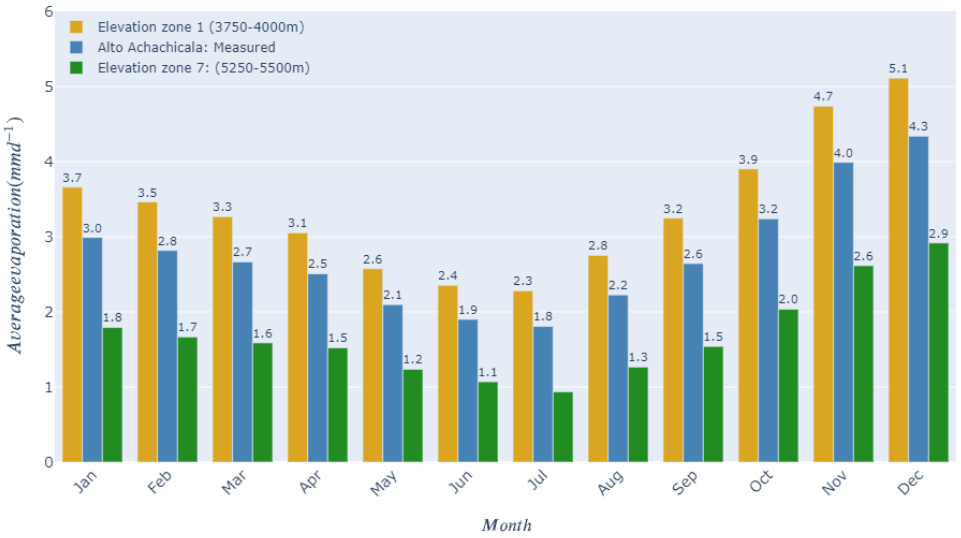


Figure A.1: The calculated values for evaporation are shown for elevation class 1 and 7. Due to the method, other class values lie in between.

Because the values for E_p, Δ_T the evaporation for each elevation class can be adjusted, using the lapse rates mention in chapter two and the calculated irradiance factor. This implicitly assumes evenly distributed atmospheric conditions over the basin.

B

MODIS

B.1. NDVI

Table B.1: Pixel Reliability classification [1] and their presence in the dataset.

Rank Key	Summary QA	Description	Present in Data
-1	Fill/No Data	Not Processed	No
0	Good Data	Use with confidence	Yes
1	Marginal Data	Useful, but look at other QA information	No
2	Snow/Ice	Target covered with snow/ice	No
3	Cloudy	Target not visible, covered with cloud	Yes

B.2. SNOWCOVER - NDSI

Table B.2: NDVI Pixel quality description

Parameter	Bit	Description
VI Quality	00	VI produced with good quality
	01	VI produced, but check other QA
	10	Pixel produced, but most probably cloudy
	11	Pixel not produced due to other reasons than clouds
VI Usefulness	0000	0 - Highest quality
	0001	1 -Lower quality
	0010	2 -Decreasing quality
	0011	3 -Decreasing quality
	0100	4 -Decreasing quality
	0101	5 -Decreasing quality
	0110	6- Decreasing quality
	0111	7 -Decreasing quality
	1000	8- Decreasing quality
	1001	9 - Decreasing quality
	1010	10 - Decreasing quality
	1011	11- Decreasing quality
	1100	12 - Lowest quality
	1101	13 - Quality so low that it is not useful
	1110	14 - L1B data faulty
	1111	15 - Not useful for any other reason/not processed
Aerosol Quantity	00	Climatology (best)
	01	Low (2nd best)
	10	Intermediate (2nd to worst)
	11	High (worst)
Adjacent cloud detected	0	No
	1	Yes
Atmosphere BRDF Correction	0	No
	1	Yes
Mixed Clouds	0	No
	1	Yes
Land/Water Mask	000	Shallow ocean
	001	Land (Nothing else but land)
	010	Ocean coastlines and lake shorelines
	011	Shallow inland water
	100	Ephemeral water
	101	Deep inland water
	110	Moderate or continental ocean
	111	Deep ocean
Possible snow/ice	0	No
	1	Yes
Possible shadow	0	No
	1	Yes

Table B.3: Quality assessment of MODIS Snow Cover data and decision to remove or keep the data for each value. Information on bands and their value interpretation from [2].

Band	Value	Description	Decision	Count
NDSI_Snow_Cover_Basic_QA	0	Best	Keep	2887807
	1	Good	Keep	5568
	2	Ok	Keep	0
	3	Poor	Remove	0
	211	Night	Remove	0
	293	Ocean	Remove	0
	NaN	Couds	Remove	1264981
NDSI_Snow_Cover_Class	200	Missing Data	Remove	71
	201	No Decision	Keep	26950
	211	Night	Remove	0
	237	Inland Water	Keep	230
	239	Ocean	Remove	0
	250	Cloud	Remove	1264981
	254	Detector Saturated	Remove	264
	NaN	NaN if no snow is detected	Keep	1841744

C

MODIS - Data Processing scripts

C.1. NDVI

First pixel coordinates are calculated and stored through Google Earth Engine.

```
var images = ee.ImageCollection('MODIS/006/MOD10A1')
                          .filterDate('2000-06-01','2001-08-31')
                          .filterBounds(AOI).select('NDSI_Snow_Cover');

print(images)

// get a single MODIS image
var image = ee.Image(images.first())

// Paint all the polygon edges with the same number and width, display.
var empty = ee.Image().byte();
var outline = empty.paint({
  featureCollection: AOI,
  color: 1,
  width: 3
});
Map.addLayer(outline, {palette: 'FF0000'}, 'edges')
//Map.addLayer(AOI)
Map.addLayer(image)
print(image)

// generate a new image containing lat/lon of the pixel and reproject it to MODIS projection
var coordsImage = ee.Image.pixelLonLat().reproject(image.projection())

// extract lat/lon coordinates as a list
var coordsList = coordsImage.reduceRegion({
  reducer: ee.Reducer.toList(2),
  geometry: AOI
}).values().get(0)

coordsList = ee.List(coordsList)

// convert coordinates to points and add to map
var coordsPoints = coordsList.map(function(xy) {
  var geom = ee.Algorithms.GeometryConstructors.Point(xy)

  return ee.Feature(geom, {})
```

```
})

coordsPoints = ee.FeatureCollection(coordsPoints)

Map.addLayer(coordsPoints)

Export.table.toDrive(coordsPoints,
"pixel-coordinates",
"MODIS/Snow/Coordinates",
"Snow_coordinates",
"CSV");
```

C.1.1. NDVI DATA DOWNLOAD

```
import ee, datetime
import pandas as pd
import geopandas as gpd
import matplotlib.dates as mdates
from matplotlib import dates
from shapely.geometry import shape
import numpy as np
import skimage

ee.Initialize()
###
start = ee.Date('1998-09-01')
finish = ee.Date('2017-08-31')

#### Make Points for every pixel in region
df = pd.read_csv('Data/NDVI/NDVI_coordinates.csv', delimiter=';', index_col=0, header=0)
pixels = pd.DataFrame(df.Coords.str.split(',',1).tolist(),
                                    columns = ['lat','lon'])
pnts = []
print(pixels)
for index, row in pixels.iterrows():
    lat = np.float(row['lat'])
    lon = np.float(row['lon'])
    pnt = ee.Feature(ee.Geometry.Point(lat, lon))
    pnts.append(pnt)

points = ee.FeatureCollection(pnts)

#### Load Raster
collection = ee.ImageCollection('MODIS/006/MOD13Q1')\
    .filterDate(start, finish)\
    .select('NDVI')

# ===============================================================================
# Adapted from: https://events.hpc.grnet.gr/event/47/material/1/12.py

def fc2df(fc):
    # Convert a FeatureCollection into a pandas DataFrame
    # Features is a evaluator of dict with the output
    features = fc.getInfo()['features']
    dictarr = []

    for f in features:
```

```
        # Store all attributes in a dict
        attr = f['properties']
        # and treat geometry separately
        attr['geometry'] = f['geometry']  # GeoJSON Feature!
        # attr['geometrytype'] = f['geometry']['type']
        dictarr.append(attr)

    df = gpd.GeoDataFrame(dictarr)
    # Convert GeoJSON features to shape
    df['geometry'] = map(lambda s: shape(s), df.geometry)
    return df

# ================================================================================
# Function to iterate over image collection, returning a pandas dataframe
def extract_point_values(img_id, pts):
    image = ee.Image(img_id)

    fc_image_red = image.reduceRegions(collection=pts,
                                       reducer=ee.Reducer.mean(),
                                       scale=30)

    # Convert to Pandas Dataframe
    df_image_red = fc2df(fc_image_red)

    # Add Date as Variable
    df_image_red['date'] = image.getInfo()['properties']['system:index']

    return df_image_red

# ================================================================================
#### Make evaluator of image IDs
clip_id = []
for f in collection.getInfo()['features']:
  image_id = f['properties']['system:index'].encode('ascii', 'ignore')
  image_id = image_id.decode('utf-8')
  image_id = 'MODIS/006/MOD13Q1/' + image_id
  clip_id.append(image_id)

#### Create Initial Pandas Dataframe
df_all = extract_point_values(clip_id[0], points)
df_all = df_all.drop([0,1])

#### Iterate over all impages
for i in clip_id:
    df_all = df_all.append(extract_point_values(i, points))
    # print(df_all)

#### Display Results
# print(df_all)
df_all.to_csv('Output/CSV/NDVI')
```

C.1.2. NDVI Data Processing & Quality Control

```
import numpy as np
import pandas as pd
from datetime import date
import matplotlib.pyplot as plt
```

```
import copy

# ====================== Load MODIS Data ======================
df = pd.read_csv('Data/NDVI/NDVI_timeseries_per_pixel.csv', delimiter=',', header=0)
df.rename_axis('Index', inplace=True)

# ========= convert date to datetime ========================
df['date'] = df['date'].astype('|S')
df['date'] = pd.to_datetime(df['date'].str.decode("utf-8"), format='%Y_%m_%d')

# ================ get pixel coordinates ====================
pix = pd.read_csv('Data/NDVI/NDVI_coordinates.csv', delimiter=';', index_col=0, header=0)
pixels = pd.DataFrame(pix.Coords.str.split(',',1).tolist(),
                                    columns = ['lat','lon'])
pixels = pixels.astype(dtype='float32')

# ================== remove unnecessary columns, rename index =============
columns = ['Pixel', 'DayOfYear', 'DetailedQA', 'EVI', 'NDVI', 'RelativeAzimuth',
       'SolarZenith', 'SummaryQA', 'ViewZenith', 'sur_refl_b01',
       'sur_refl_b02', 'sur_refl_b03', 'sur_refl_b07', 'geometry', 'date']

df.drop(columns=['DayOfYear', 'EVI', 'RelativeAzimuth',
                 'SolarZenith', 'ViewZenith', 'sur_refl_b01',
                 'sur_refl_b02', 'sur_refl_b03', 'sur_refl_b07',
                 'geometry'], inplace=True)

df.set_index('date', inplace=True)
# ================== Quality Selection
# based on MODIS Vegetation Index User's Guide(MOD13 Series)

# Remove Clouds ======== (only value 3 present in data out of -1, 2, 3)
df = df[df.SummaryQA != 3]

# Remove Values Based on detailed QA --> see excel file for reason
remove = [2172, 2175, 2237, 4220, 4223, 4285, 4287, 4349, 4413, 5246, 5247, \
          5311, 6205, 6206, 6207, 6268, 6271, 6333, 6335, 6397, 6461, 6463, \
          7230, 7231, 7294, 7295, 7358, 7359, 7487, 37181, 37183, 38975, 39036, \
          39039, 39101, 39229, 39231]

for value in remove:
    df = df[df.DetailedQA != value]

# ================ Normalize NDVI ======================
df['NDVI'] = df['NDVI'].divide(10000)
df.to_csv('Output/CSV/Cleaned Data/NDVI.csv')
```

D

FLEX-TOPO: BASE

```
import numpy as np
from numba import jit, float64, prange, int64
from glacier import glacier
from hillslope import hillslope
from moraine import moraine
from riparian import riparian
from urban import urban

from data_FLEX import Flex_data
from parameter_FLEX import Flex_parameter_index


@jit((Flex_data.class_type.instance_type,
      Flex_parameter_index.class_type.instance_type,
      float64[::1]), nopython=True, cache=True)
def run_flex_numba(data, parameter_index, parameters):

    # parameters_indices
    idx_ks = parameter_index.ks
    idx_kf = parameter_index.kf
    idx_kf_moraine = parameter_index.kf_moraine
    idx_sublimax = parameter_index.sublimax
    idx_lapse_rate = parameter_index.lapse_rate
    idx_timelag = parameter_index.time_lag

    # parameters
    ks = parameters[idx_ks]
    kf = parameters[idx_kf]
    kf_moraine = parameters[idx_kf_moraine]
    sublimax = parameters[idx_sublimax]
    time_lag = parameters[idx_timelag]
    lapse_rate_base = parameters[idx_lapse_rate]


    # time_lag
    f0 = time_lag
    f1 = 1 - f0
    f2 = 0.0

    # lapse_rate
```

```
reference_elevation = data.reference_elevation
reference_elevation = reference_elevation.astype(np.int64)
base_elevation = data.base_elevation

dz_ = reference_elevation - base_elevation
dz_ = dz_ / 100
dz = np.zeros((7, 2))
dz[:, 0], dz[:, 1] = dz_, dz_.copy()
lapse_rate_ = np.ones((7, 2)) + lapse_rate_base / 100
lapse_rate = np.power(lapse_rate_, dz)

lapse_rate_urban = lapse_rate[1, 0]

n_hrus = data.n_hrus
urban_present = data.urban_present
glacier_present = data.glacier_present

temperature = data.temperature
evaporation = data.evaporation
precipitation = data.precipitation
hru_perc_4hru = data.hru_percentage_4hrus
topo_4hru = data.topo_4hrus
hru_perc_5hru = data.hru_percentage_5hrus
topo_5hru = data.topo_5hrus

# define number of distributed zones per HRU. Take the most classes!
irradiance_zones = 2

# elevation zones
min_max_dem = data.min_max_nzones
el_min = data.min_max_nzones[:, 0]
el_max = data.min_max_nzones[:, 1]
el_max = el_max + 1  # plus 1 to include when slicing arrays
total_elevation_zones = np.max(el_max) - np.min(el_min)


# elevation per HRU
rip_dmin = el_min[0] - np.min(el_min[el_min >= 0])
hill_dmin = el_min[1] - np.min(el_min[el_min >= 0])
mor_dmin = el_min[2] - np.min(el_min[el_min >= 0])

rip_dmax = el_max[0] - np.max(el_max)
hill_dmax = el_max[1] - np.max(el_max)
mor_dmax = el_max[2] - np.max(el_max)

elevation_riparian = min_max_dem[0]
elevation_hillslope = min_max_dem[1]
elevation_moraine = min_max_dem[2]

if glacier_present == True:
    glac_dmin = el_min[3] - np.min(el_min)
    glac_dmax = el_max[3] - np.max(el_max)
    elevation_glacier = min_max_dem[3]

ir_r = data.irradiance_riparian[:, el_min[0]: el_max[0]]
ir_r = np.ascontiguousarray(ir_r)
```

```
ir_h = data.irradiance_hillslope[:, el_min[1]: el_max[1]]
ir_h = np.ascontiguousarray(ir_h)
ir_m = data.irradiance_moraine[:, el_min[2]: el_max[2]]
ir_m = np.ascontiguousarray(ir_m)

if glacier_present == True:
    ir_g = data.irradiance_glacier[:, el_min[3]: el_max[3]]
    ir_g = np.ascontiguousarray(ir_g)
    n_states = 10
    n_evap = 7
else:
    n_states = 8
    n_evap = 6

# time-variables
tmax = int64(len(precipitation))
dt = np.int64(1)

if urban_present == True:
    t_hru5 = 3288  # moment that HRU5 kicks in
    first_urban = np.int64(0)
    states_urban = np.zeros((tmax, 2), dtype=np.float64)


# Forcing -->
temp = temperature * dt
epdt = evaporation * dt
pdt = precipitation * dt

# create sublimation array, based on submax & irradiance factor
subl_m = ir_m * sublimax

if glacier_present == True:
    topo_glac = topo_4hru[3, el_min[3]: el_max[3]]
    subl_g = ir_g * sublimax

# output states and fluxes

# total discharge of system q_tot, qs, qf
q_dt = np.zeros((tmax, 3), dtype=np.float64)

# ea_rip, ei_rip, ea_hil, ei_hil, es_mor, ei_mor, es_gla, es_glac
e_dt = np.zeros((tmax, n_evap,
                 total_elevation_zones,
                 irradiance_zones),
                 dtype=np.float64)
states_dt = np.zeros((tmax,
                      n_states,
                      total_elevation_zones,
                      irradiance_zones),
                      dtype=np.float64)

ss = np.zeros(tmax, dtype=np.float64)
sf = np.zeros((tmax, n_hrus), dtype=np.float64)

# distributed fluxes HRUs (fi_hru = flux, internal)
```

```
fi_r = np.zeros((tmax, 5, elevation_riparian[2], irradiance_zones),
                dtype=np.float64)
fi_h = np.zeros((tmax, 5, elevation_hillslope[2], irradiance_zones),
                dtype=np.float64)
fi_m = np.zeros((tmax, 5, elevation_moraine[2], irradiance_zones),
                dtype=np.float64)


if glacier_present is True:
    fi_g = np.zeros((tmax, 4, elevation_glacier[2], irradiance_zones),
                    dtype=np.float64)

# distributed states HRUs
states_rip = np.zeros((tmax, 3, elevation_riparian[2], irradiance_zones),
                      dtype=np.float64)
states_hill = np.zeros((tmax, 3, elevation_hillslope[2], irradiance_zones),
                      dtype=np.float64)
states_mor = np.zeros((tmax, 2, elevation_moraine[2], irradiance_zones),
                      dtype=np.float64)
states_urban = np.zeros((tmax, 2),
                      dtype=np.float64)

if glacier_present == True:
    states_glac = np.zeros((tmax, 2, elevation_glacier[2], irradiance_zones),
                           dtype=np.float64)

    # give the glacier states an initial snowdepth
    states_glac[0, 1, :, :] = np.float64(200000)
    states_glac[0, 1, :, :] = states_glac[0, 1, :, :] * np.where(topo_glac > 0, 1, 0)

# lumped states & external fluxes
fe_r = np.zeros((tmax, 2), dtype=np.float64)  # ss
fe_h = np.zeros((tmax, 2), dtype=np.float64)  # qsdt, qfdt
fe_m = np.zeros((tmax, 2), dtype=np.float64)

if glacier_present == True:
    fe_g = np.zeros((tmax, 2), dtype=np.float64)

if urban_present == True:
    fe_urban = np.zeros((tmax, 2), dtype=np.float64)

for t in range(0, tmax - 1):
    epdt_r = epdt[t, el_min[0]: el_max[0]] * ir_r[t]
    epdt_h = epdt[t, el_min[1]: el_max[1]] * ir_h[t]
    epdt_m = epdt[t, el_min[2]: el_max[2]] * ir_m[t]

    lapse_rate_r = lapse_rate[el_min[0]: el_max[0]]
    lapse_rate_h = lapse_rate[el_min[1]: el_max[1]]
    lapse_rate_m = lapse_rate[el_min[2]: el_max[2]]


    if urban_present == True:
        if t < t_hru5:
            # convert topo_hru
            topo_rip = topo_4hru[0, el_min[0]: el_max[0]]
            topo_hill = topo_4hru[1, el_min[1]: el_max[1]]
```

```
            topo_mor = topo_4hru[2, el_min[2]: el_max[2]]

            # define relative areas of HRUS (sum = 1)
            hru_percentages = hru_perc_4hru
            riparian_area = hru_percentages[0]
            hillslope_area = hru_percentages[1]
            moraine_area = hru_percentages[2]
            glacier_area = 0.0
            urban_area = 0.0

            if glacier_present == True:
                topo_glac = topo_4hru[3, el_min[3]: el_max[3]]
                lapse_rate_g = lapse_rate[el_min[3]: el_max[3]]
                glacier_area = hru_percentages[3]

        else:
            topo_rip = topo_5hru[0, el_min[0]: el_max[0]]
            topo_hill = topo_5hru[1, el_min[1]: el_max[1]]
            topo_mor = topo_5hru[2, el_min[2]: el_max[2]]

            hru_percentages = hru_perc_5hru
            riparian_area = hru_percentages[0]
            hillslope_area = hru_percentages[1]
            moraine_area = hru_percentages[2]
            glacier_area = 0.0
            urban_area = hru_perc_5hru[4]

    else:
        # convert topo_hru
        topo_rip = topo_4hru[0, el_min[0]: el_max[0]]
        topo_hill = topo_4hru[1, el_min[1]: el_max[1]]
        topo_mor = topo_4hru[2, el_min[2]: el_max[2]]

        # define relative areas of HRUS (sum = 1)
        hru_percentages = hru_perc_4hru
        riparian_area = hru_percentages[0]
        hillslope_area = hru_percentages[1]
        moraine_area = hru_percentages[2]

        if glacier_present == True:
            lapse_rate_g = lapse_rate[el_min[3]: el_max[3]]
            topo_glac = topo_4hru[3, el_min[3]: el_max[3]]
            glacier_area = hru_percentages[3]


    # Riparian rfdt, eidt, eadt, mdt, rcdt pldt, psdt, states, exflux
    fi_r[t], states_rip[t + 1], fe_r[t] = riparian(parameter_index,
                                                   temp[t, el_min[0]: el_max[0]],
                                                   epdt_r, pdt[t],
                                                   parameters,
                                                   states_rip[t],
                                                   ss[t],
                                                   hru_percentages,
                                                   topo_rip,
                                                   lapse_rate_r,
                                                   ir_r[t])
```

```
        # hillslope
        fi_h[t], states_hill[t + 1], fe_h[t] = hillslope(param_index=parameter_index,
                                                         temp=temp[t, el_min[1]: el_max[1]],
                                                         epdt=epdt_h,
                                                         pdt=pdt[t],
                                                         parameters=parameters,
                                                         states=states_hill[t],
                                                         hillslope_topo=topo_hill,
                                                         lapse_rate=lapse_rate_h,
                                                         irradiance_hillslope=ir_h[t])

        # Moraine
        fi_m[t], states_mor[t + 1], fe_m[t] = moraine(parameter_index,
                                                      temp[t, el_min[2]: el_max[2]],
                                                      epdt_m,
                                                      pdt[t],
                                                      subl_m[t],
                                                      parameters,
                                                      states_mor[t],
                                                      topo_mor,
                                                      lapse_rate_m,
                                                      ir_m[t])

        # Glacier
        if glacier_present == True:
            ir_g = data.irradiance_glacier[:, el_min[3]: el_max[3]]
            ir_g_ = np.ascontiguousarray(ir_g)
            fi_g[t], states_glac[t + 1],
                fe_g[t] = glacier(parameter_index,
                                  temp=temp[t, el_min[3]: el_max[3]],
                                  pdt=pdt[t],
                                  sublimation=subl_g[t],
                                  parameters=parameters,
                                  states=states_glac[t],
                                  glacier_topo=topo_glac,
                                  lapse_rate=lapse_rate_g,
                                  ir_g=ir_g_[t])
        # urban

        if urban_present == True:
            if t >= t_hru5:
                if first_urban == np.int64(0):
                    states_urban = np.zeros((tmax, 2), dtype=np.float64)
                    sf[:, 3] = 0.0
                    first_urban = 1

                states_urban[t + 1], fe_urban[t] = urban(parameter_index,
                                                         temp=temp[t, 1],
                                                         epdt=epdt_r[1],
                                                         pdt=pdt[t],
                                                         parameters=parameters,
                                                         states=states_urban[t],
                                                         lapse_rate=lapse_rate_urban)
```

```
# ================================================
# ==============  t = 0 ==========================
# ================================================
if t == 0:
    sf_r = sf[t, 0] + fe_r[t, 1] * f0
    sf_h = sf[t, 1] + fe_h[t, 1] * f0
    sf_m = sf[t, 2] + fe_m[t, 1] * f0

    if glacier_present == True:
        sf_g = sf[t, 3] + fe_g[t, 1] * f0

        # ======= Slow Reservoir
        ss__ = fe_r[t, 0] + fe_h[t, 0] * hillslope_area + \
               fe_m[t, 0] * moraine_area + fe_g[t, 0] * glacier_area

    else:
        ss__ = fe_r[t, 0] + fe_h[t, 0] * hillslope_area + \
               fe_m[t, 0] * moraine_area

# ================================================
# ==============  t = 1 ==========================
# ================================================

elif t == 1:
    sf_r = sf[t, 0] + fe_r[t, 1] * f0 + fe_r[t - 1, 1] * f1
    sf_h = sf[t, 1] + fe_h[t, 1] * f0 + fe_h[t - 1, 1] * f1
    sf_m = sf[t, 2] + fe_m[t, 1] * f0 + fe_m[t - 1, 1] * f1

    if glacier_present == True:
        sf_g = sf[t, 3] + fe_g[t, 1] * f0 + fe_g[t - 1, 1] * f1

        # ======= Slow Reservoir
        ss__ = fe_r[t, 0] + \
               fe_h[t, 0] * hillslope_area + \
               fe_m[t, 0] * moraine_area + \
               fe_g[t, 0] * glacier_area

    else:
        # ======= Slow Reservoir
        ss__ = fe_r[t, 0] + \
               fe_h[t, 0] * hillslope_area + \
               fe_m[t, 0] * moraine_area


# ==============  t > 1 ==========================
# ================================================
# =============== NO URBAN ========================


elif urban_present == False:
    sf_r = sf[t, 0] + fe_r[t, 1] * f0 + fe_r[t - 1, 1] * f1 + fe_r[t - 2, 1] * f2
    sf_h = sf[t, 1] + fe_h[t, 1] * f0 + fe_h[t - 1, 1] * f1 + fe_h[t - 2, 1] * f2
    sf_m = sf[t, 2] + fe_m[t, 1] * f0 + fe_m[t - 1, 1] * f1 + fe_m[t - 2, 1] * f2

    if glacier_present == True:
        sf_g = sf[t, 3] + \
```

```
                fe_g[t, 1] * f0 + \
                fe_g[t - 1, 1] * f1 + \
                fe_g[t - 2, 1] * f2

            ss__ = fe_r[t, 0] + \
                fe_h[t, 0] * hillslope_area + \
                fe_m[t, 0] * moraine_area + \
                fe_g[t, 0] * glacier_area

        else:
            ss__ = fe_r[t, 0] + \
                fe_h[t, 0] * hillslope_area + \
                fe_m[t, 0] * moraine_area


    # ================================================
    # ==============  t > 1 ==========================
    # ================================================
    # ===============  URBAN TRUE =====================
    # ================================================

    elif urban_present == True:
        if t < t_hru5:
            sf_r = sf[t, 0] + fe_r[t, 1] * f0 + \
                fe_r[t - 1, 1] * f1 + \
                fe_r[t - 2, 1] * f2

            sf_h = sf[t, 1] + \
                fe_h[t, 1] * f0 + \
                fe_h[t - 1, 1] * f1 + \
                fe_h[t - 2, 1] * f2

            sf_m = sf[t, 2] + \
                fe_m[t, 1] * f0 + \
                fe_m[t - 1, 1] * f1 + \
                fe_m[t - 2, 1] * f2

            if glacier_present == True:
                sf_g = sf[t, 3] + \
                    fe_g[t, 1] * f0 + \
                    fe_g[t - 1, 1] * f1 + \
                    fe_g[t - 2, 1] * f2

                ss__ = fe_r[t, 0] + \
                    fe_h[t, 0] * hillslope_area + \
                    fe_m[t, 0] * moraine_area + \
                    fe_g[t, 0] * glacier_area

            else:
                ss__ = fe_r[t, 0] + \
                    fe_h[t, 0] * hillslope_area + \
                    fe_m[t, 0] * moraine_area
        else:
            sf_r = sf[t, 0] + \
                fe_r[t, 1] * f0 + \
                fe_r[t - 1, 1] * f1 + \
```

```
                fe_r[t - 2, 1] * f2

            sf_h = sf[t, 1] + \
                   fe_h[t, 1] * f0 + \
                   fe_h[t - 1, 1] * f1 + \
                   fe_h[t - 2, 1] * f2

            sf_m = sf[t, 2] + \
                   fe_m[t, 1] * f0 + \
                   fe_m[t - 1, 1] * f1 + \
                   fe_m[t - 2, 1] * f2

            if glacier_present == True:
                sf_g = sf[t, 3] + \
                       fe_g[t, 1] * f0 + \
                       fe_g[t - 1, 1] * f1 + \
                       fe_g[t - 2, 1] * f2

                sf_urban = sf[t, 4] + fe_urban[t, 1]

                ss__ = fe_r[t, 0] + \
                       fe_h[t, 0] * hillslope_area + \
                       fe_m[t, 0] * moraine_area + \
                       fe_g[t, 0] * glacier_area + \
                       fe_urban[t, 0] * urban_area

            else:
                sf_urban = sf[t, 3] + fe_urban[t, 1]
                ss__ = fe_r[t, 0] + \
                       fe_h[t, 0] * hillslope_area + \
                       fe_m[t, 0] * moraine_area + \
                       fe_urban[t, 0] * urban_area

    # ========================================
    # ======= Fast Reservoir fluxes
    qf_r = max(0, sf_r / kf)
    qf_h = max(0, sf_h / kf)
    qf_m = max(0, sf_m / kf_moraine)

    sf[t, 0] = max(0, sf_r - qf_r)
    sf[t, 1] = max(0, sf_h - qf_h)
    sf[t, 2] = max(0, sf_m - qf_m)

    if glacier_present == True:
        qf_g = max(0, sf_g / kf)
        sf[t, 3] = max(0, sf_g - qf_g)

        if urban_present == True:
            if t >= t_hru5:
                qf_urban = max(0, sf_urban / kf)
                sf[t, 4] = max(0, sf_urban - qf_urban)

                q_dt[t, 2] = qf_r * riparian_area + \
                             qf_h * hillslope_area + \
                             qf_m * moraine_area + \
                             qf_g * glacier_area + \
```

```
                            qf_urban * urban_area
        else:
            q_dt[t, 2] = qf_r * riparian_area + \
                         qf_h * hillslope_area + \
                         qf_m * moraine_area + \
                         qf_g * glacier_area

    elif urban_present == True:
        if t >= t_hru5:
            qf_urban = max(0, sf_urban / kf)
            sf[t, 3] = max(0, sf_urban - qf_urban)

            q_dt[t, 2] = qf_r * riparian_area + \
                         qf_h * hillslope_area + \
                         qf_m * moraine_area + \
                         qf_urban * urban_area

    else:
        q_dt[t, 2] = qf_r * riparian_area + \
                     qf_h * hillslope_area + \
                     qf_m * moraine_area

    # ======= Slow Reservoir fluxes
    ss__ = max(0, ss__)
    qs = ss__ / (dt * ks)
    qs = min(qs, ss__)
    qs = max(0, qs)
    ss[t] = ss__ - qs
    q_dt[t, 1] = qs

    if t < tmax - 1:
        ss[t + 1] = ss[t]
        sf[t + 1] = sf[t]

# Total Flow - calculated after whole loop
q_dt[:, 0] = q_dt[:, 1] + q_dt[:, 2]

# reduce calculation on empty zones, to save RAM time
e_end = len(e_dt[0, 0])
e_dt[:, 0, rip_dmin: rip_dmax + e_end] = fi_r[:, 1]
e_dt[:, 1, rip_dmin: rip_dmax + e_end] = fi_r[:, 2]
e_dt[:, 2, hill_dmin: hill_dmax + e_end] = fi_h[:, 2]
e_dt[:, 3, hill_dmin: hill_dmax + e_end] = fi_h[:, 3]
e_dt[:, 4, mor_dmin: mor_dmax + e_end] = fi_m[:, 2]
e_dt[:, 5, mor_dmin: mor_dmax + e_end] = fi_m[:, 3]

if glacier_present == True:
    e_dt[:, 6, glac_dmin: glac_dmax + e_end] = fi_g[:, 2]


s_end = len(states_dt[0, 0])

states_dt[:, 0, rip_dmin: rip_dmax + s_end] = states_rip[:, 0]
states_dt[:, 1, rip_dmin: rip_dmax + s_end] = states_rip[:, 1]
states_dt[:, 2, rip_dmin: rip_dmax + s_end] = states_rip[:, 2]
```

```
    states_dt[:, 3, hill_dmin: hill_dmax + s_end] = states_hill[:, 0]
    states_dt[:, 4, hill_dmin: hill_dmax + s_end] = states_hill[:, 1]
    states_dt[:, 5, hill_dmin: hill_dmax + s_end] = states_hill[:, 2]

    states_dt[:, 6, mor_dmin: mor_dmax + s_end] = states_mor[:, 0]
    states_dt[:, 7, mor_dmin: mor_dmax + s_end] = states_mor[:, 1]

    if glacier_present == True:
        states_dt[:, 8, glac_dmin: glac_dmax + e_end] = states_glac[:, 0]
        states_dt[:, 9, glac_dmin: glac_dmax + e_end] = states_glac[:, 1]

    # ===== Slow Reservoir Fluxes for each HRU (except Riparian)
    ss_hru = np.zeros((tmax, n_hrus - 1))
    ss_hru[:, 0] = fe_h[:, 0]
    ss_hru[:, 1] = fe_m[:, 0]

    if glacier_present == True:
        ss_hru[:, 2] = fe_g[:, 0]
        if urban_present == True:
            ss_hru[:, 3] = fe_urban[:, 0]
    elif urban_present == True:
        ss_hru[:, 2] = fe_urban[:, 0]

    return q_dt.astype(np.float32),
           states_dt.astype(np.float32),
           e_dt.astype(np.float32),
           sf.astype(np.float32),
           ss.astype(np.float32),
           ss_hru.astype(np.float32)

##########################################
##########################################
# Function to run FLEX-Topo model in parallel

@jit((Flex_data.class_type.instance_type,
      Flex_parameter_index.class_type.instance_type,
      float64[:, ::1]),
      debug=False,
      nopython=True,
      cache=True,
      parallel=True)
def flex_parallel_numba(data_,
                        parameter_index,
                        parameterset):

    n = np.int64(len(parameterset))
    precipitation = data_.precipitation

    n_hrus = data_.n_hrus
    glacier_present = data_.glacier_present
    elevation_zones = data_.min_max_nzones
    elevation_zones = np.max(elevation_zones[:, 1] + 1) - np.min(elevation_zones[:, 0])

    if glacier_present == True:
        n_states = 10
        n_evap = 7
```

```
    else:
        n_states = 8
        n_evap = 6

    tmax = np.int64(len(precipitation))
    irradiance_zones = 2

    # output: full output, reduced output, fully reduced
    # Fully reduced: returns only total discharge
    # Reduced output: returns all non distributed fluxes and states
    # full output: returns everything

    if data_.output == 0:
        q = np.zeros((n, tmax, 3),
                     dtype=np.float32)

        e_dt = np.zeros((n,
                         tmax,
                         n_evap,
                         elevation_zones,
                         irradiance_zones),
                         dtype=np.float32)

        states_dt = np.zeros((n,
                              tmax,
                              n_states,
                              elevation_zones,
                              irradiance_zones),
                              dtype=np.float32)

        sf_dt = np.zeros((n, tmax, n_hrus), dtype=np.float32)
        ss_dt = np.zeros((n, tmax), dtype=np.float32)
        ss_hru = np.zeros((n, tmax, n_hrus - 1), dtype=np.float32)

        for i in prange(n):
            q[i], states_dt[i],
                e_dt[i], sf_dt[i],
                    ss_dt[i], ss_hru[i] = run_flex_numba(data_,
                                                         parameter_index,
                                                         parameterset[i])

    elif data_.output == 1:
        q = np.zeros((n, tmax, 3), dtype=np.float32)
        sf_dt = np.zeros((n, tmax, n_hrus), dtype=np.float32)
        ss_dt = np.zeros((n, tmax), dtype=np.float32)
        ss_hru = np.zeros((n, tmax, n_hrus - 1), dtype=np.float32)

        for i in prange(n):
            q[i], x, y, sf_dt[i],
                ss_dt[i], ss_hru[i] = run_flex_numba(data_,
                                                     parameter_index,
                                                     parameterset[i])

        # assign after loop to save memory during loop.
        # Must be assigned b.c. numba can't return different types/sets
        # within one function
```

```
        e_dt = np.zeros((n, tmax, n_evap, elevation_zones, irradiance_zones),
                        dtype=np.float32)
        states_dt = np.zeros((n, tmax, n_states, elevation_zones, irradiance_zones),
                             dtype=np.float32)

    elif data_.output == 2:
        q = np.zeros((n, tmax, 3), dtype=np.float32)

        for i in prange(n):
            q[i], x, y, z, x1, y1 = run_flex_numba(data_,
                                                   parameter_index,
                                                   parameterset[i])

        # assign after loop to save memory during loop.
        # Must be assigned b.c. numba can't return
        # different types/sets within one function
        e_dt = np.zeros((n, tmax, n_evap, elevation_zones, irradiance_zones),
                        dtype=np.float32)

        states_dt = np.zeros((n, tmax, n_states, elevation_zones, irradiance_zones),
                             dtype=np.float32)

        sf_dt = np.zeros((n, tmax, n_hrus), dtype=np.float32)
        ss_dt = np.zeros((n, tmax), dtype=np.float32)
        ss_hru = np.zeros((n, tmax, n_hrus - 1), dtype=np.float32)

    return q, sf_dt, ss_dt, states_dt, e_dt, ss_hru
```

E

FLEX-Topo: HRU Scripts

E.1. Riparian

```
import numpy as np
from numba import jit, float64
from parameter_FLEX import Flex_parameter_index

#
@jit((Flex_parameter_index.class_type.instance_type,
      float64[:, ::1],
      float64[:, ::1],
      float64,
      float64[::1],
      float64[:, :, ::1],
      float64,
      float64[::1],
      float64[:, ::1],
      float64[:, ::1],
      float64[:, ::1]), nopython=True, cache=True)
def riparian(param_index,
             temp: np.ndarray,
             epdt: np.ndarray,
             pdt: np.float64,
             parameters: np.ndarray,
             states: np.ndarray,
             ss: np.float64,
             hru_perc: np.ndarray,
             riparian_topo: np.ndarray,
             lapse_rate: np.ndarray,
             irradiance_riparian: np.ndarray):
    """Info
    update function information
    """
    # ================================================================
    # ========== Initialize Parameters, States and Fluxes ==========
    # parameter_indices
    idx_tt = param_index.tt
    idx_fdd = param_index.fdd
    idx_beta = param_index.beta
    idx_simax_riparian = param_index.simax_riparian
    idx_sumax_riparian = param_index.sumax_riparian
```

```
idx_cr = param_index.cr
idx_ce = param_index.ce

# ==== Parameters ====
tt = parameters[idx_tt]
fdd = parameters[idx_fdd]
beta = parameters[idx_beta]

simax_riparian = parameters[idx_simax_riparian]
sumax_riparian = parameters[idx_sumax_riparian]

capillary_max = parameters[idx_cr]
ce = parameters[idx_ce]

# ==== States ===============
sw = states[0]
si = states[1]
su = states[2]

# === HRU Topography and Characteristics ===
riparian_hru_perc = hru_perc[0]
elevation_mask = np.where(riparian_topo > 0, 1, 0)

# mask forcing
epdt = epdt * elevation_mask

# ==== Precipitation Separation into Rain and snow =========
pldt = np.where((temp - tt) >= 0, pdt, 0) *\
       elevation_mask * lapse_rate     # liquid precipitation: i.e. rain

psdt = np.where((temp - tt) < 0, pdt, 0) *\
       elevation_mask * lapse_rate      # solid precipitation: i.e. snow

# ====== Snow Reservoir ================================================
swdt = sw + psdt

# ===== Melt ===============
# potential melt
melt_p = fdd * (temp - tt) * elevation_mask * irradiance_riparian
melt_p = np.where(temp - tt > 0, melt_p, 0)      # melt, (so only positive values)
mdt = np.where(swdt - melt_p > 0, melt_p, swdt)

# ==== Update Snow Reservoir ===============
sw_ = swdt - mdt

# ====== Interception Reservoir ========================================
sidt = si + pldt

# effective precipitation
pedt = np.where(sidt - simax_riparian > 0,
                sidt - simax_riparian, 0)
sidt = sidt - pedt

# === interception Run-off   ============
r_inter = pedt + mdt
```

```
# ====== Unsaturated Reservoir ============================

# Inflow of r_inter into Unsaturated zone(effective precipitation & melt)
# rho expresses the increasing friction for water to enter
su_ = su
rho = (su_ / sumax_riparian) ** beta

# separation between run-off and filling up storage, limited to Su_max.
su_ = np.where(su_ + (1 - rho) * r_inter < sumax_riparian,
               su_ + (1 - rho) * r_inter,
               sumax_riparian)

sudt = su_ - su          # Volume difference after entering of r_inter
rfdt = r_inter - sudt     # generated run-off from Su_reservoir

# evaporation from interception reservoir
eidt = np.where(epdt - sidt >= 0, sidt, pedt)

# risidual potential evaporation (Epdt - Eidt)
epdt = epdt - eidt

# Ea = evaporation from rootzone, limited by field capacity,
ea_ = np.where(su_ / (sumax_riparian * ce) < 1,
               epdt * (su_ / (sumax_riparian * ce)), epdt)

# limited Ea by available moisture in unsaturated zone
eadt = np.where(ea_ - sudt > 0, su_, ea_)

# update sates of Unsaturated- and Interception Reservoir
su_ = su_ - eadt
su_ = np.where(su_ < 0, 0, su_)
si_ = sidt - eidt
si_ = np.where(si_ < 0, 0, si_)

#  ======= Cr--> Ground Water Pushing up============================

# rc proportional to moisture content in elevation class.
ss_ = np.zeros_like(riparian_topo) + ss
rcdt = (1 - su_ / sumax_riparian) * capillary_max * elevation_mask

# check if the groundwater has enough water to facilitate capillary rise.
# If not, uptake is limited to volume of groundwater,
# relative to elevation/shade class area.
rcdt = np.where(rcdt - (ss_ / riparian_hru_perc) < 0,
                rcdt,
                ss_ / riparian_hru_perc) * elevation_mask

rcdt = np.where(rcdt - (sumax_riparian - su_) < 0,
                rcdt, sumax_riparian - su_)     # Su cannot > Sumax

rcdt = np.where(rcdt < 0, 0, rcdt)

# update volumes of unsaturated & groundwater reservoir after Capillary Rise
su_ = su_ + rcdt
rc_total = np.sum(rcdt * riparian_topo * riparian_hru_perc)
ss = np.float64(ss - rc_total)
```

```
    ss = max(np.float64(0), ss)

    # ====== Fast Reservoir ========================================
    qfdt = np.sum(rfdt * riparian_topo)

    # ===== Save Output ============================================
    states[0] = sw_
    states[1] = si_
    states[2] = su_

    fluxes = np.zeros((5, len(riparian_topo), len(riparian_topo[0])),
                      dtype=np.float64)
    fluxes[0] = rfdt
    fluxes[1] = eidt
    fluxes[2] = eadt
    fluxes[3] = mdt
    fluxes[4] = rcdt

    external_fluxes = np.array([ss, qfdt])
    return fluxes, states, external_fluxes
```

E.2. HILLSLOPE

```
import numpy as np
from numba import jit, float64
from parameter_FLEX import Flex_parameter_index


@jit((Flex_parameter_index.class_type.instance_type,
      float64[:, ::1],
      float64[:, ::1],
      float64,
      float64[::1],
      float64[:, :, ::1],
      float64[:, ::1],
      float64[:, ::1],
      float64[:, ::1]),
      nopython=True, cache=True)
def hillslope(param_index,
              temp: np.ndarray,
              epdt: np.ndarray,
              pdt: np.float64,
              parameters: np.ndarray,
              states: np.ndarray,
              hillslope_topo: np.ndarray,
              lapse_rate: np.ndarray,
              irradiance_hillslope: np.ndarray):

    # ========================================================
    # ========== Initialize Parameters, States and Fluxes ============
    # parameter_indices

    idx_tt = param_index.tt
    idx_fdd = param_index.fdd
    idx_d0 = param_index.d0
    idx_beta = param_index.beta
    idx_simax_hillslope = param_index.simax_hillslope
```

```
idx_sumax_hillslope = param_index.sumax_hillslope

idx_ce = param_index.ce

# ==== Parameters ====
tt = parameters[idx_tt]
fdd = parameters[idx_fdd]
d0 = parameters[idx_d0]
beta = parameters[idx_beta]

simax_hillslope = parameters[idx_simax_hillslope]
sumax_hillslope = parameters[idx_sumax_hillslope]
ce = parameters[idx_ce]

# ==== States ===============
sw = states[0]
si = states[1]
su = states[2]

# === HRU Topography and Characteristics ===
elevation_mask = np.where(hillslope_topo > 0, 1, 0)

# ================================================
# mask forcing (and apply lapse rate for rain)
epdt = epdt * elevation_mask
pdt = pdt

# ==== Precipitation Separation into Rain and snow =========
# liquid precipitation: i.e. rain
pldt = np.where((temp - tt) >= 0, pdt, 0) * \
       elevation_mask * lapse_rate

# solid precipitation: i.e. snow
psdt = np.where((temp - tt) < 0, pdt, 0) * \
       elevation_mask * lapse_rate

# ====== Snow Reservoir ========================================
sw_ = sw + psdt

# ===== Melt ===============
# potential melt
melt_p = fdd * (temp - tt) * elevation_mask * irradiance_hillslope
melt_p = np.where(temp - tt > 0, melt_p, 0)  # melt, (so only positive values)

# potential melt limited to available snow
mdt = np.where(sw_ - melt_p > 0, melt_p, sw_)

# ==== Update Snow Reservoir ===============
sw = sw_ - mdt

# ====== Interception Reservoir ============================
sidt = si + pldt
pedt = np.where(sidt - simax_hillslope > 0,
                sidt - simax_hillslope, 0)   # effective precipitation
sidt = sidt - pedt
```

```
# interception evaporation
eidt = np.where(epdt - sidt >= 0, sidt, epdt)
epdt = epdt - eidt  # risidual potential evaporation (Epdt - Eidt)

# === interception Run-off   ============
r_inter = pedt + mdt  # outflow is pedt (from Si) and melt combined

# === Update Interception Reservoir
si = sidt - eidt

# ====== Unsaturated Reservoir ======================
# Inflow of rInter into Unsat (eff p & melt)
# create variable for intermediate state of su
su_ = su

# rho expresses the increasing friction for water
# to enter the Su_reservoir with increasing saturation
rho = (su_ / sumax_hillslope) ** beta

# separation run-off & filling up Su,
# limited to Su_max. and dependent in moisture-content
su_ = np.where(su_ + (1 - rho) * r_inter < sumax_hillslope,
               su_ + (1 - rho) * r_inter, sumax_hillslope)

sudt = su_ - su                 # Volume difference after entering of rInter
r_unsat = r_inter - sudt          # Runoff from Unsaturated zone

# Ea = evap from rootzone, limited between field capacity and wilting_point
ea_ = np.where(su_ / (sumax_hillslope * ce) < 1,
               epdt * (su_ / (sumax_hillslope * ce)), epdt)

# limited Ea by available moisture in Su
eadt = np.where(ea_ - su_ > 0, su_, ea_)
su_ = su_ - eadt

# percolation through the day- only when soil
# moisture content is above field capacity.
f_percolation = su_/(sumax_hillslope * ce)
f_percolation = np.where(f_percolation < 1, 0, f_percolation)

# percolation is relative to soil moisture content above field capacity.
percolation = np.where(su_ - f_percolation > sumax_hillslope * ce,
                       f_percolation, su_-sumax_hillslope * ce)

percolation = np.where(percolation > 0, percolation, 0)
percolation = np.where(su_ - percolation > 0, percolation, su_)

# update reservoirs
su = su_ - percolation

# ====== Qufdt Flow seperation  ======================
rsdt = d0 * r_unsat + percolation
rfdt = (1 - d0) * r_unsat

# ====== Fast/Slow Runoff ===========1
qsdt = np.sum(rsdt * hillslope_topo)
```

```
    qfdt = np.sum(rfdt * hillslope_topo)

    # ==== Save Output ===============
    states[0] = sw
    states[1] = si
    states[2] = su

    exflux = np.array([qsdt, qfdt])

    fluxes = np.zeros((5, len(hillslope_topo),
                       len(hillslope_topo[0])),
                        dtype=np.float64)
    fluxes[0] = rsdt
    fluxes[1] = rfdt
    fluxes[2] = eidt
    fluxes[3] = eadt
    fluxes[4] = mdt
    return fluxes, states, exflux
```

E.3. MORAINE

```
import numpy as np
from numba import jit, float64
from parameter_FLEX import Flex_parameter_index
#
#
@jit((Flex_parameter_index.class_type.instance_type,
      float64[:, ::1], float64[:, ::1], float64,
      float64[:, ::1], float64[::1], float64[:, :, ::1],
      float64[:, ::1], float64[:, ::1], float64[:, ::1]),
      nopython=True, cache=True)
def moraine(param_index,
            temp: np.ndarray,
            epdt: np.ndarray,
            pdt: np.float64,
            sublimation: np.ndarray,
            parameters: np.ndarray,
            states: np.ndarray,
            moraine_topo: np.ndarray,
            lapse_rate: np.ndarray,
            irradiance_m: np.ndarray):

    # parameter_indices
    idx_tt = param_index.tt
    idx_fdd = param_index.fdd
    idx_d1 = param_index.d1
    idx_simax_moraine = param_index.simax_moraine

    # ==== Parameters ====
    tt = parameters[idx_tt]
    fdd = parameters[idx_fdd]

    tt = -1.8
    fdd = 10

    d1 = parameters[idx_d1]
    simax_moraine = parameters[idx_simax_moraine]
```

```
# ========== Initialize Parameters, States and Fluxes ============
superfast_moraine = False
threshold_superfast = 10

# correction ir_factor for albedo of fresh snow
albedo_fresh = np.float64(0.25 / 0.60)

# ==== States ===============
sw = states[0]
si = states[1]

# === HRU Topography and Characteristics ===
elevation_mask = np.where(moraine_topo > 0, 1, 0)

# Forcing
# mask forcing (and apply lapse rate for rain)
epdt = epdt * elevation_mask
subldt = sublimation * albedo_fresh

# ==== Precipitation Separation into Rain and snow =========
pldt = np.where((temp - tt) >= 0, pdt, 0) *\
       elevation_mask * lapse_rate  # liquid precipitation: i.e. rain

psdt = np.where((temp - tt) < 0, pdt, 0) *\
       elevation_mask * lapse_rate  # solid precipitation: i.e. snow

swdt = sw + psdt

# ====== Melt & Sublimation ================
# snow routine starts with evaluating if any
# snow reservoir is not empty
if np.any(swdt > 0):

    # only sublimation if it is not raining
    if pdt > 0:
        subldt[:, :] = np.float64(0)
        melt_mask = (temp - tt) * elevation_mask

        # this is correct
        if np.any(melt_mask > 0):
            mdt_ = np.where(melt_mask > 0,
                            fdd * melt_mask * irradiance_m, 0)

            melt_left = np.where(swdt - mdt_ < 0, mdt_ - swdt, 0)
            mdt = mdt_ - melt_left
            swdt = swdt - mdt

        else:
            mdt = np.zeros_like(subldt, dtype=np.float64)

    # not raining, thus:
    else:
        subldt = subldt
        melt_mask = (temp - tt) * elevation_mask
```

```
        if np.any(melt_mask > 0):  # check if there is melt
            mdt_ = np.where(melt_mask > 0,
                            fdd * melt_mask * irradiance_m, 0)

            # ablimation is intermediate variable
            # for the combined melt and sublimation
            # created to account for situation that:
                # melt + sublimation > available snow.
            # In this case melt is given priority.

            ablimation_ = mdt_ + subldt

            ablimation_left = np.where(swdt - ablimation_ < 0,
                                       ablimation_ - swdt, 0)

            ablimation = ablimation_ - ablimation_left
            swdt = swdt - ablimation
            mdt = np.where(mdt_ - ablimation >= 0,
                           ablimation, mdt_)
            subldt = ablimation - mdt

        else:
            mdt = np.zeros_like(subldt, dtype=np.float64)
            subldt = np.where(subldt - swdt >= 0, swdt, subldt)
            swdt = swdt - subldt
else:
    mdt = np.zeros_like(subldt, dtype=np.float64)
    subldt = np.zeros_like(subldt, dtype=np.float64)

# ==== Update Snow Reservoir ===============
sw = swdt
esdt = subldt

# ====== Interception Reservoir ======================

sidt = si + pldt
pedt = np.where(sidt - simax_moraine > 0, sidt - simax_moraine, 0)
si_ = sidt - pedt
si_ = np.where(si_ < 0, 0, si_)

# ===== interception evaporation ==================================
eidt = np.where(epdt - si_ > 0, si_, epdt)
si_ = np.where(si_ - eidt > 0, si_ - eidt, 0)
si = si_

# after interception reservoir, melt is routed to the seperation point
pedt = pedt + mdt
# ====== Flow Partitioning ============================
# === Run-off from interception

if superfast_moraine:
    superfast = np.where(pedt > threshold_superfast,
                         pedt - threshold_superfast, 0)
    superfast = np.where(superfast < 0.0, 0.0,
                         superfast)
    regular = pedt - superfast
```

```
        regular = np.where(regular < 0.0, 0.0, regular)
        rsdt = d1 * regular
        rfdt = (1.0 - d1) * regular + superfast

    else:
        rsdt = d1 * pedt
        rfdt = (1.0 - d1) * pedt

    # ====== Fast/ Slow Runoff ===========
    qsdt = np.sum(rsdt * moraine_topo)
    qfdt = np.sum(rfdt * moraine_topo)

    # ==== Save Output ===============
    states[0] = sw
    states[1] = si

    fluxes = np.zeros((5, len(moraine_topo), len(moraine_topo[0])),
                      dtype=np.float64)
    fluxes[0] = rsdt
    fluxes[1] = rfdt
    fluxes[2] = eidt
    fluxes[3] = esdt
    fluxes[4] = mdt

    external_fluxes = np.array([qsdt, qfdt])
    return fluxes, states, external_fluxes
```

E.4. GLACIER

```
import numpy as np
from numba import jit, float64
from parameter_FLEX import Flex_parameter_index


@jit((Flex_parameter_index.class_type.instance_type,
      float64[:, ::1],
      float64,
      float64[:, ::1],
      float64[::1],
      float64[:, :, ::1],
      float64[:, ::1],
      float64[:, ::1],
      float64[:, ::1]), nopython=True, cache=True)
def glacier(param_index,
            temp: np.ndarray,
            pdt: np.float64,
            sublimation: np.ndarray,
            parameters: np.ndarray,
            states: np.ndarray,
            glacier_topo: np.ndarray,
            lapse_rate: np.ndarray,
            ir_g: np.ndarray):
    """Info

    """
    # ============================================================
    # ========== Initialize Parameters, States and Fluxes ============
```

```
# parameter_indices
idx_tt = param_index.tt
idx_fdd = param_index.fdd
idx_d1 = param_index.d1
# idx_sublimax = param_index.sublimax
# irradiance_glacier is adjusted in FLEX_TOPO.py

idx_tt_glacier = param_index.tt_glacier
idx_fdd_glacier = param_index.fdd_glacier

# ==== Parameters ====
tt = parameters[idx_tt]
fdd = parameters[idx_fdd]
d1 = parameters[idx_d1]

tt_glacier = parameters[idx_tt_glacier]
fdd_glacier = parameters[idx_fdd_glacier]

# ==== Parameters ====
# correction of calculated irradiance factor for albedo of glacial ice
albedo_glacier = np.float64(0.25 / 0.35)
# correction of calculated irradiance factor for albedo of fresh snow
albedo_fresh = np.float64(0.25 / 0.60)

# ==== States ====
sw = states[0]  # Snow Reservoir
sg = states[1]  # glacier reservoir

# === HRU Topography and Characteristics ===
elevation_mask = np.where(glacier_topo > 0, 1, 0)

# ==== Forcing ===============
subldt = sublimation
pdt = pdt

# ==== Precipitation Separation into Rain and snow =========
# liquid precipitation: i.e. rain
pldt = np.where((temp - tt) >= 0,
                pdt,
                0) * lapse_rate * elevation_mask
# solid precipitation: i.e. snow
psdt = np.where((temp - tt) < 0,
                pdt,
                0) * lapse_rate * elevation_mask

# ====== Snow Reservoir ======================
swdt = sw + psdt        # state of snow reservoir + fresh_snow
sgdt = sg               # state of glacier reservoir

# ====== Melt & Sublimation ==============================
# snow routine starts with evaluating if any snow reservoir is not empty
if np.any(swdt > 0):

    if pdt > 0:  # only sublimation if it is not raining
        subldt = np.zeros_like(swdt)
        melt_mask = (temp - tt) * elevation_mask
```

```
            if np.any(melt_mask > 0):
                melt = np.where(melt_mask > 0, fdd * melt_mask * ir_g, 0)
                melt_left = np.where(swdt - melt < 0, melt - swdt, 0)
                # first melt all snow in fresh-snow reservoir, then from glacier
                melt_glacier = fdd_glacier / fdd * melt_left

                swdt = swdt - (melt - melt_left)
                sgdt = sgdt - melt_glacier
                melt = melt + melt_glacier - melt_left
            else:
                melt = np.zeros_like(swdt)

        else:
            subldt = subldt * albedo_fresh
            melt_mask = (temp - tt) * elevation_mask

            if np.any(melt_mask > 0):  # check if there is melt
                melt = np.where(melt_mask > 0, fdd * melt_mask * ir_g, 0)
                subldt_glacier = np.where(swdt - subldt < 0, subldt - swdt, 0)
                swdt = swdt - (subldt - subldt_glacier)
                subldt = subldt - subldt_glacier
                subldt_glacier = subldt_glacier * (albedo_glacier / albedo_fresh)
                subldt = subldt + subldt_glacier

                melt_left = np.where(swdt - melt < 0, melt - swdt, 0)
                melt_glacier = fdd_glacier / fdd * melt_left

                swdt = swdt - (melt - melt_left)
                sgdt = sgdt - (subldt_glacier + melt_glacier)
                melt = melt + melt_glacier - melt_left

            else:
                melt = np.zeros_like(swdt)
                subldt_glacier = np.where(swdt - subldt < 0, subldt - swdt, 0)
                swdt = swdt - (subldt - subldt_glacier)
                subldt = subldt - subldt_glacier
                subldt_glacier = subldt_glacier * (albedo_glacier / albedo_fresh)
                subldt = subldt + subldt_glacier
                sgdt = sgdt - subldt_glacier

    else:  # no snow in fresh snow reservoir

        if pdt > 0:  # only sublimation if it is not raining
            subldt = np.zeros_like(swdt)
            melt_mask = (temp - tt_glacier) * elevation_mask

            if np.any(melt_mask > 0):  # check if there is melt
                melt = np.where(melt_mask > 0, fdd_glacier * melt_mask * ir_g, 0)
                sgdt = sgdt - melt

            else:
                melt = np.zeros_like(swdt)
        else:
            subldt = subldt * albedo_glacier
            melt_mask = (temp - tt_glacier) * elevation_mask
```

```
        if np.any(melt_mask > 0):
            melt = np.where(melt_mask > 0, fdd_glacier * melt_mask * ir_g, 0)
            sgdt = sgdt - (melt + subldt)

        else:
            melt = np.zeros_like(swdt)
            sgdt = sgdt - subldt

# ==== Update Snow Reservoir ===============
sw = np.where(swdt > 0, swdt, 0)
sg = np.where(sgdt > 0, sgdt, 0)
esdt = subldt
# ===== Internal Fluxes & flow-seperation ============================
rsdt = d1 * (melt + pldt)
rfdt = (1 - d1) * (melt + pldt)

# ===== Run-off Partitioning =======================================
qfdt = np.sum(rfdt * glacier_topo)
qsdt = np.sum(rsdt * glacier_topo)

# ==== Save Output ===============
states[0, :, :] = sw
states[1, :, :] = sg

exflux = np.array([qsdt, qfdt])

fluxes = np.zeros((4, len(glacier_topo), len(glacier_topo[0])), dtype=np.float64)
fluxes[0, :, :] = rsdt
fluxes[1, :, :] = rfdt
fluxes[2, :, :] = esdt
fluxes[3, :, :] = melt
return fluxes, states, exflux
```

F

Integrated System Code

```
import numpy as np
import pandas as pd
# from heapq import nsmallest
from heapq import nlargest
import matplotlib.pyplot as plt
import datetime
from dateutil.relativedelta import relativedelta
import seaborn as sns
import os
plt.style.use('seaborn')
def loader(name, path, prediction_year):
    """Reading data from keys"""
    dir = path + name + '/' + str(prediction_year)
    with open(dir + "/keys.txt", "r") as f:
        sc_keys = eval(f.read())

    dictex = {}

    for sc_key in sc_keys:
        dictex[sc_key] = {}
        sub_dir = dir + '/' + sc_key
        with open(sub_dir + "/keys.txt", "r") as f:
            keys = eval(f.read())

        for key in keys:
            dictex[sc_key][key] = pd.read_csv(sub_dir + "/data_{}.csv".format(str(key)),
                                              header=[0, 1],
                                              index_col=0,
                                              parse_dates=True)
    return dictex


def saver(name, path, prediction_year=None, observed=None, dictex=None):
    import os

    if prediction_year is None:
        dir = path + name
        if observed is not None:
            if os.path.exists(dir):
                observed.to_csv(dir + '/observed.csv')
```

```
            else:
                os.mkdir(dir)
                observed.to_csv(dir + '/observed.csv')
        else:
            return

    else:
        if isinstance(prediction_year, int):
            prediction_year = str(prediction_year)

        dir = path + name + '/' + prediction_year

        if not os.path.exists(dir):
            dir0 = path + name
            if not os.path(dir0):
                os.mkdir(dir0)

            os.mkdir(dir)

        if observed is not None:
            if isinstance(observed, pd.DataFrame):
                observed.to_csv(dir + '/observed.csv')

        if dictex is not None:

            for subdir_name, dictionary in dictex.items():
                sub_dir = dir + '/' + subdir_name
                os.mkdir(sub_dir)
                with open(dir + "/keys.txt", "w") as f:  # saving keys to file
                    f.write(str(list(dictex.keys())))

                for key, val in dictionary.items():
                    val.to_csv(sub_dir + "/data_{}.csv".format(str(key)))

                    with open(sub_dir + "/keys.txt", "w") as f:  # saving keys to file
                        f.write(str(list(dictionary.keys())))
        return


def load_system_data(flow_scenarios=None,
                     demand_scenario=None,
                     intake_max=None,
                     intake_milluni=None,
                     intake_jumalincu=None,
                     historical=None,
                     prediction_year=None):
    """

    :param flow_scenarios:
    :param demand_scenario:
    :param intake_max:
    :param intake_milluni:
    :param intake_jumalincu:
    :param historical:
    :param prediction_year: string or interger
```

```
    :return:
    """

    reservoir_info = pd.read_excel(
        r'system_data/system_information.xlsx',
        sheet_name='Reservoirs',
        skiprows=0,
        index_col=0,
        header=0)

    demand = pd.read_excel(
        r'system_data/system_information.xlsx',
        sheet_name='Demand',
        skiprows=0,
        index_col=0,
        header=0)

    va_curves = pd.read_excel(
        r'system_data/system_information.xlsx',
        sheet_name='volume_area_curves',
        skiprows=0,
        header=[0, 1])

    subcatchments = pd.read_excel(
        r'system_data/system_information.xlsx',
        sheet_name='Subcatchments',
        skiprows=0,
        header=0)

    systems = subcatchments['Sub-System0'].unique().tolist()
    sub_systems = {}

    for system in systems:
        sub = subcatchments.copy()
        sub0 = sub[sub['Sub-System0'] == system].copy()
        sub1 = sub[sub['Sub-System1'] == system].copy()

        # id_ = [(id + ' [m3/d]', x) for id, x in zip(sub0['ID'].values, sub0['Partioning SS0'])]
        # if len(sub1) > 0:
        #     id_part = [(id + ' [m3/d]', x) for id, x in zip(sub1['ID'].values, sub1['Partioning SS1']
        id_ = [(id_sub0, x) for id_sub0, x in
               zip(sub0['ID'].values, sub0['Partioning SS0'])]
        if len(sub1) > 0:
            id_part = [(id_sub1, x) for id_sub1, x in
                       zip(sub1['ID'].values, sub1['Partioning SS1'])]

            if len(sub1) == 1:
                id_.append(id_part[0])
            else:
                for id in id_part:
                    id_.append(id.copy())

        sub_systems[system] = id_.copy()

    # set-intake settings in case None are given
    if intake_jumalincu is None:
```

```
        intake_jumalincu = 0.95

    if intake_milluni is None:
        intake_milluni = 0.1

    if intake_max is None:
        intake_max = 0.5

    # load scenarios
    scenarios_dict = {}

    if historical is True:
        xlsx = pd.ExcelFile('E:\eOutput\Modular_flex\output/'
                            'forecasting\historical_scenarios_final/'
                            'scenarios.xlsx')

        if prediction_year is None:
            year_str = xlsx.sheet_names
            year_str.remove('observed')
            for idx, name in enumerate(year_str):
                print(f'Reading sheet #{idx}: {name}')
                sheet = xlsx.parse(sheet_name=name,
                                   header=[0, 1, 2],
                                   index_col=0,
                                   parse_dates=True)
                basins = sheet.columns.levels[0].to_list()
                for basin in basins:
                    area = subcatchments[subcatchments['ID'] == basin]['Area [m2]'].values[0]
                    sheet[basin] = sheet[basin] * (area / 1000)

                scenarios_dict[name] = sheet.copy()
        else:
            if isinstance(prediction_year, int):
                prediction_year = str(prediction_year)

            print(f'Reading sheet: {prediction_year}')
            sheet = xlsx.parse(sheet_name=prediction_year,
                               header=[0, 1, 2],
                               index_col=0,
                               parse_dates=True)
            scenarios_dict[prediction_year] = sheet.copy()
            basins = sheet.columns.levels[0].to_list()

            for basin in basins:
                area = subcatchments[subcatchments['ID'] == basin]['Area [m2]'].values[0]
                sheet[basin] = sheet[basin] * (area / 1000)

            scenarios_dict[prediction_year] = sheet.copy()

    else:
        if not (flow_scenarios, str):
            return print('specify flow scenarios if historical is not True')

        # specify directory to load flow_scenarios
        print('None Historical Flow Scenarios Not Working')
```

```
    # Set Demand scenario
    if demand_scenario is None:
        demand_scenario = 2016
    print('Demand Scenario: ', demand_scenario)

    demand = demand.loc[demand_scenario]['Demand [m3/d]']

    # load observed
    q_obs = xlsx.parse(sheet_name='observed',
                       header=0,
                       index_col=0,
                       parse_dates=True)
    for basin in basins:
        area = subcatchments[subcatchments['ID'] == basin]['Area [m2]'].values[0]
        q_obs[basin] = q_obs[basin] * area / 1000

    return_dict = {'observed': q_obs,
                   'scenarios': scenarios_dict,
                   'reservoir_info': reservoir_info,
                   'sub-systems': sub_systems,
                   'demand': demand,
                   'va_curves': va_curves,
                   'intake_max': intake_max,
                   'intake_milluni': intake_milluni,
                   'intake_jumalincu': intake_jumalincu}
    return return_dict

def run_system(data, observed=False,
               area_calculation=None,
               reservoirs_present=True):
    """
    run this
    :param inflows:
    :param demand:
    :param subcatchment_info:
    :param area_calculation:
    :return:
    """

    # TO DO: milluni demand is still including max intake perc river

    def create_result_df(data):
        reservoirs = data['reservoir_info'].index.values.tolist()
        reservoir_info = data['reservoir_info']
        demand = data['demand']

        if observed is True:
            flows = data['observed']

        else:
            flows = data['run_scenario']

        intake_milluni = data['intake_milluni']
        max_intake = data['intake_max']

        res_ = ['State [m3]',
```

```
            'Available [m3]',
            'Reservoir Level [%]',
            'Inflow [m3]',
            'Outflow [m3]',
            'Spill [m3]',
            'Area Approx [m2]']
    res_columns = [(i, j) for i in reservoirs for j in res_]

    free_col = [('Free Flowing', i) for i in ['Total [m3]', 'Used [m3]', 'Un-used [m3]']]

    demand_col = [('Demand', i) for i in ['Net Demand [m3]',
                                          'Total Kaluyo [m3]',
                                          'Free Flowing [m3]',
                                          'Reservoirs [m3]',
                                          'Milluni [m3]']]

    col_index = []
    for list in [demand_col, free_col, res_columns]:
        for tuple in list:
            col_index.append(tuple)

    col_index = pd.MultiIndex.from_tuples(col_index,
                                          names=['System', 'Flux'])
    df_ = pd.DataFrame(index=flows.index,
                       columns=col_index)

    for system in data['sub-systems']:
        contribution = data['sub-systems'][system]

        for ind, ss in enumerate(contribution):
            if system == 'Free Flowing':
                if ind == 0:
                    df_.loc[:, ('Free Flowing', 'Total [m3]')] = flows[ss[0]].copy() * ss[1]
                else:
                    df_.loc[:, ('Free Flowing', 'Total [m3]')] = df_.loc[:, ('Free Flowing',
                                                                             flows[ss[0]].copy() * ss[1]
            else:
                if ind == 0:
                    df_.loc[:, (system, 'Inflow [m3]')] = flows[ss[0]].copy() * ss[1]

                else:
                    df_.loc[:, (system, 'Inflow [m3]')] = df_.loc[:, (system, 'Inflow [m3]')]
                                                                  (flows[ss[0]].copy() * ss[1])

                # set initial state to sate
                df_.loc[df_.first_valid_index(), (system, 'State [m3]')] = reservoir_info.loc

    base_milluni = 0
    df_.loc[:, ('Demand', 'Net Demand [m3]')] = demand

    if intake_milluni is not None:
        base_milluni = demand * intake_milluni
        demand = demand - base_milluni

    df_.loc[:, ('Demand', 'Total Kaluyo [m3]')] = demand / max_intake
    df_.loc[:, ('Demand', 'Milluni [m3]')] = base_milluni
```

```
    if reservoirs_present is not True:
        df_.loc[:, ('Free Flowing', 'Total [m3]')] = df_.loc[:, ('Free Flowing', 'Total [m3]')] + \
                                                     df_.loc[:, ('Pampalarama', 'Inflow [m3]')] + \
                                                     df_.loc[:, ('Al Paquita', 'Inflow [m3]')] + \
                                                     df_.loc[:, ('Chacaltaya', 'Inflow [m3]')]

    return df_.copy()

df = create_result_df(data=data)
reservoirs = data['reservoir_info'].index.values.tolist()
reservoir_info = data['reservoir_info']
va_curves = data['va_curves']
max_intake = data['intake_max']

if reservoirs_present is True:

    for t, date in enumerate(df.index):
        remaining_demand = 0.0

        # Assess if river stream is sufficient to cover demand
        demand = df.loc[date, ('Demand', 'Total Kaluyo [m3]')]
        free_flow = df.loc[date, ('Free Flowing', 'Total [m3]')]

        # if stream flow is enough:
        if demand <= free_flow:
            df.loc[date, ('Demand', 'Reservoirs [m3]')] = 0
            df.loc[date, ('Demand', 'Free Flowing [m3]')] = demand

            df.loc[date, ('Free Flowing', 'Used [m3]')] = demand
            df.loc[date, ('Free Flowing', 'Un-used [m3]')] = np.round(free_flow - demand, 1)

        else:
            df.loc[date, ('Demand', 'Free Flowing [m3]')] = free_flow
            df.loc[date, ('Free Flowing', 'Used [m3]')] = free_flow
            df.loc[date, ('Free Flowing', 'Un-used [m3]')] = 0.0
            remaining_demand = demand - free_flow

        # Assess available water in Reservoirs
        for dam in reservoirs:
            # get Volume from t-1, add influx [m3]
            if t == 0:
                state = df.loc[date, (dam, 'State [m3]')]
            else:
                state = df.iloc[t - 1][dam]['State [m3]']

            total_volume = reservoir_info.loc[dam]['Total Volume [m3]']
            dead_volume = reservoir_info.loc[dam]['Dead Volume [m3]']
            inflow = df.loc[date, (dam, 'Inflow [m3]')]
            state = state + inflow

            # if Volume is greater that max volume --> spill (later reduce spill with demand)
            if state > total_volume:
                spill = state - total_volume
                state = total_volume
                df.loc[date, (dam, 'State [m3]')] = state
```

```
                df.loc[date, (dam, 'Available [m3]')] = max(state - dead_volume, 0.0)
                df.loc[date, (dam, 'Spill [m3]')] = spill
                df.loc[date, (dam, 'Outflow [m3]')] = 0.0
                df.loc[date, (dam, 'Reservoir Level [%]')] = 100.0

            # No Spill
            else:
                df.loc[date, (dam, 'State [m3]')] = state
                df.loc[date, (dam, 'Available [m3]')] = max(state - dead_volume, 0.0)
                df.loc[date, (dam, 'Spill [m3]')] = 0.0
                df.loc[date, (dam, 'Outflow [m3]')] = 0.0
                df.loc[date, (dam, 'Reservoir Level [%]')] = df.loc[date, (dam, 'Available [m
                                                             / reservoir_info.loc[dam, 'Usefu


            if area_calculation is True:
                id = va_curves[dam].iloc[(va_curves[dam]['Volume [m3]'] - (state - dead_volum
                v = id['Volume [m3]'].values
                x = (state - v[1]) / (v[0] - v[1])
                area = id['Area [m2]'].values[1] + (id['Area [m2]'].values[0] - id['Area [m2]
                df.loc[date, (dam, 'Area Approx [m2]')] = area

        # if stream-flow is sufficient, continue
        if remaining_demand == 0.0:
            continue

        # Demand from reservoirs + Milluni
        else:
            res_available = np.array([df.loc[date, (dam, 'Available [m3]')] for dam in reserv
            res_available = np.where(res_available < 0.0, 0.0, res_available)
            res_level = np.array([df.loc[date, (dam, 'Reservoir Level [%]')] for dam in reser
            reservoir_demand = np.zeros(len(res_available))

            # if any reservoirs have enough water to provide demand, all water is taken from
            # with highest water level
            if np.all(res_available > remaining_demand):
                reservoir_demand[np.argmax(res_level)] = remaining_demand
                milluni_demand = 0.0

            elif np.any(res_available > remaining_demand):
                res_id = np.argmax(res_level)
                if res_available[res_id] >= remaining_demand:
                    reservoir_demand[np.argmax(res_level)] = remaining_demand
                else:
                    second_id = np.argwhere(res_level == nlargest(2, res_level)[-1])[0][0]
                    third_id = np.argwhere(res_level == nlargest(3, res_level)[-1])[0][0]
                    second = res_available[second_id]

                    if second >= remaining_demand:
                        reservoir_demand[second_id] = remaining_demand
                    else:
                        reservoir_demand[third_id] = remaining_demand

                milluni_demand = 0.0

            # if any reservoir has more than 0 m3 available
```

```
        elif np.any(res_available > 0):
            # if totla sum is enough for totla demand
            if np.sum(res_available) >= remaining_demand:
                remaining_demand_loop = remaining_demand
                i = 0
                while (remaining_demand_loop > 0.0) and (i < len(reservoir_demand)):
                        n_fullest = nlargest(i + 1, res_level)[-1]
                        id_ = np.argwhere(res_level == n_fullest)[0][0]
                        subtract = res_available[id_] if res_available[id_] < remaining_demand_
                            else remaining_demand_loop
                        reservoir_demand = np.where(res_level == n_fullest,
                                                    subtract,
                                                    reservoir_demand)
                        remaining_demand_loop = remaining_demand_loop - subtract
                        i = i + 1
                milluni_demand = 0.0

            else:
                reservoir_demand = res_available
                milluni_demand = (remaining_demand - np.sum(res_available)) * max_intake

        # no available water in reservoirs.
        else:
            reservoir_demand = np.zeros(len(reservoirs))
            milluni_demand = remaining_demand * max_intake


        df.loc[date, ('Demand', 'Milluni [m3]')] = df.loc[date, ('Demand', 'Milluni [m3]')] + m
        df.loc[date, ('Demand', 'Reservoirs [m3]')] = np.sum(reservoir_demand)
        df.loc[date, ('Demand', 'Total Kaluyo [m3]')] = (df.loc[date, ('Demand', 'Net Demand [m
                                                         df.loc[date, ('Demand', 'Milluni [m3]')]

        # Recalculate Reservoir level, Spill (adjust with outflow)
        for ind, dam in enumerate(reservoirs):
            spill = df.loc[date, (dam, 'Spill [m3]')]

            if df.loc[date, (dam, 'Available [m3]')] < 0.0:
                test = 0

            if spill == 0.0:
                df.loc[date, (dam, 'State [m3]')] = df.loc[date, (dam, 'State [m3]')] - reservo
                available = df.loc[date, (dam, 'Available [m3]')] - reservoir_demand[ind]
                state = df.loc[date, (dam, 'State [m3]')]
                df.loc[date, (dam, 'Available [m3]')] = available

            elif spill >= reservoir_demand[ind]:
                df.loc[date, (dam, 'Spill [m3]')] = spill - reservoir_demand[ind]
                df.loc[date, (dam, 'Outflow [m3]')] = reservoir_demand[ind]
                df.loc[date, (dam, 'Reservoir Level [%]')] = 100.0

                if area_calculation is True:
                    df.loc[date, (dam, 'Area Approx [m2]')] = va_curves[dam]['Area [m2]'].value

                continue

            else:
```

```
                    df.loc[date, (dam, 'State [m3]')] = df.loc[date, (dam, 'State [m3]')] \
                                                        - (reservoir_demand[ind] - spill)
                    available = df.loc[date, (dam, 'Available [m3]')] \
                                                          - (reservoir_demand[ind] - spill)
                    df.loc[date, (dam, 'Available [m3]')] = available
                    state = df.loc[date, (dam, 'State [m3]')]
                    df.loc[date, (dam, 'Spill [m3]')] = 0


                df.loc[date, (dam, 'Outflow [m3]')] = reservoir_demand[ind]
                df.loc[date, (dam, 'Reservoir Level [%]')] = df.loc[date, (dam, 'Available [m
                                                             / reservoir_info.loc[dam, 'Usefu
                if area_calculation is True:
                    idx = va_curves[dam].iloc[(va_curves[dam]['Volume [m3]'] - state).abs().a
                    v = idx['Volume [m3]'].values
                    x = (state - v[1]) / (v[0] - v[1])
                    area = idx['Area [m2]'].values[1] + (idx['Area [m2]'].values[0] - idx['Ar
                    df.loc[date, (dam, 'Area Approx [m2]')] = area

    else:

        for t, date in enumerate(df.index):
            remaining_demand = 0.0

            # Assess if river stream is sufficient to cover demand
            demand = df.loc[date, ('Demand', 'Total Kaluyo [m3]')]
            free_flow = df.loc[date, ('Free Flowing', 'Total [m3]')]

            # if stream flow is enough:
            if demand <= free_flow:
                df.loc[date, ('Demand', 'Reservoirs [m3]')] = 0
                df.loc[date, ('Demand', 'Free Flowing [m3]')] = demand

                df.loc[date, ('Free Flowing', 'Used [m3]')] = demand
                df.loc[date, ('Free Flowing', 'Un-used [m3]')] = np.round(free_flow - demand, 1)

            else:
                df.loc[date, ('Demand', 'Free Flowing [m3]')] = free_flow
                df.loc[date, ('Free Flowing', 'Used [m3]')] = free_flow
                df.loc[date, ('Free Flowing', 'Un-used [m3]')] = 0.0
                remaining_demand = demand - free_flow
                milluni_demand = remaining_demand * max_intake
                df.loc[date, ('Demand', 'Milluni [m3]')] = df.loc[date, ('Demand', 'Milluni [m3]'

    return df
```

BIBLIOGRAPHY

[1] K. Didan, A. B. Munoz, R. Solano, and A. Huete, *MODIS Vegetation Index User 's Guide (Collection 6)*, **2015**, 31 (2015).

[2] G. A. Riggs, D. K. Hall, and V. V. Salomonson, *Earth Science*, Tech. Rep. August (2006).

[3] The Guardian, *Bolivia declares state of emergency over worst drought in 25 years - World news - The Guardian,* (2016).

[4] C. Valdivia, J. Thibeault, J. L. Gilles, M. García, and A. Seth, *Climate trends and projections for the Andean Altiplano and strategies for adaptation,* Advances in Geosciences **33**, 66 (2013).

[5] H. Kats, *Minimizing water shortages and operational costs of a water supply system by providing decision support on real-time control : a case study in La Paz, Bolivia,* (2019).

[6] EPSAS, *Plan Maestro Metropolitano de Agua Potable y Saneamiento La Paz - El Alto , Bolivia La Informe Final Resumen Ejecutivo*, Tech. Rep. Tomo 1 (EPSAS, La Paz, 2014).

[7] H. Nomden, R. Huting, and A. J. Kerk, *RoyalHaskoningDHV*, Tech. Rep. October (Amersfoort, 2018).

[8] UNISDR and United Nations secretariat of the International Strategy for Disaster Reduction (UNISDR), *Drought Risk Reduction Framework and Practices: Contributing to the Implementation of the Hyogo Framework for Action.* , 2013 (2009).

[9] .

[10] S. Hunziker, S. Gubler, J. Calle, I. Moreno, M. Andrade, F. Velarde, L. Ticona, G. Carrasco, Y. Castellón, C. Oria, M. Croci-Maspoli, T. Konzelmann, M. Rohrer, and S. Brönnimann, *Identifying, attributing, and overcoming common data quality issues of manned station observations,* International Journal of Climatology **37**, 4131 (2017).

[11] M. Córdova, R. Célleri, C. J. Shellito, J. Orellana-Alvear, A. Abril, and G. Carrillo-Rojas, *Near-surface air temperature lapse rate over complex terrain in the Southern Ecuadorian Andes: Implications for temperature mapping,* Arctic, Antarctic, and Alpine Research **48**, 673 (2016).

[12] G. H. Hargreaves and R. G. Allen, *History and evaluation of hargreaves evapotranspiration equation,* Journal of Irrigation and Drainage Engineering **129**, 53 (2003).

[13] T. G. Farr, P. A. Rosen, E. Caro, R. Crippen, R. Duren, S. Hensley, M. Kobrick, M. Paller, E. Rodriguez, L. Roth, D. Seal, S. Shaffer, J. Shimada, J. Umland, M. Werner, M. Oskin, D. Burbank, and D. Alsdorf, *The Shuttle Radar Topography Mission,* Reviews of Geophysics **45**, RG2004 (2007).

[14] A. Ayala, F. Pellicciotti, N. Peleg, and P. Burlando, *Melt and surface sublimation across a glacier in a dry environment: Distributed energy-balance modelling of Juncal Norte Glacier, Chile,* Journal of Glaciology **63**, 803 (2017).

[15] S. Macdonell, C. Kinnard, T. Mölg, L. Nicholson, and J. Abermann, *Meteorological drivers of ablation processes on a cold glacier in the semi-arid Andes of Chile,* Cryosphere **7**, 1513 (2013).

[16] E. Ramírez, B. Francou, P. Ribstein, M. Descloitres, R. Guérin, J. Mendoza, R. Gallaire, B. Pouyaud, and E. Jordan, *Small glaciers disappearing in the tropical Andes: A case-study in Bolivia: Glaciar Chacaltaya (16°S),* Journal of Glaciology **47**, 187 (2001).

[17] .

[18] European Centre for Medium-Range Weather Forecasts, *SEAS5 User Guide*, Tech. Rep. (European Centre for Medium-Range Weather Forecasts, Reading, United Kingdom, 2017).

[19] S. Kellert, *In the wake of chaos: Unpredictable order in dynamical systems,* Bibliovault OAI Repository, the University of Chicago Press **19** (1995), 10.1016/0160-9327(95)90056-X.

[20] M. Vuille, R. S. Bradley, and F. Keimig, *Climate variability in the Andes of Ecuador and its relation to tropical Pacific and Atlantic Sea Surface temperature anomalies,* Journal of Climate **13**, 2520 (2000).

[21] J. Ronchail and R. Gallaire, *ENSO and rainfall along the Zongo valley (Bolivia) from the Altiplano to the Amazon basin,* International Journal of Climatology **26**, 1223 (2006).

[22] M. Vuille, *Atmospheric circulation over the Bolivian Altiplano during dry and wet periods and extreme phases of the southern oscillation,* International Journal of Climatology **19**, 1579 (1999).

[23] R. Garreaud, M. Vuille, and A. C. Clement, *The climate of the Altiplano: Observed current conditions and mechanisms of past changes,* Palaeogeography, Palaeoclimatology, Palaeoecology **194**, 5 (2003).

[24] Y. G. Ham, J. S. Kug, and J. Y. Park, *Two distinct roles of Atlantic SSTs in ENSO variability: North Tropical Atlantic SST and Atlantic Niño,* Geophysical Research Letters **40**, 4012 (2013).

[25] V. B. S. Silva and V. E. Kousky, *The South American Monsoon System: Climatology and Variability,* in *Modern Climatology*, edited by S.-Y. Wang (InTech China, Shanghai, 2012) 1st ed., Chap. 5, pp. 124–151.

[26] B. Liebmann and C. R. Mechoso, *South American Monsoon System,* The Global Monsoon System , 137 (2011).

[27] H. E. Beck, N. E. Zimmermann, T. R. McVicar, N. Vergopolan, A. Berg, and E. F. Wood, *Present and future Köppen-Geiger climate classification maps at 1-km resolution,* Scientific Data **5** (2018), 10.1038/sdata.2018.214.

[28] R. D. Garreaud, *The Andes climate and weather,* Advances in Geosciences **22**, 3 (2009).

[29] A. Soruco, C. Vincent, A. Rabatel, B. Francou, E. Thibert, J. E. Sicart, and T. Condom, *Contribution of glacier runoff to water resources of La Paz city, Bolivia (16° S),* Annals of Glaciology **56**, 147 (2015).

[30] M. Vuille, M. Carey, C. Huggel, W. Buytaert, A. Rabatel, D. Jacobsen, A. Soruco, M. Villacis, C. Yarleque, O. Elison Timm, T. Condom, N. Salzmann, and J. E. Sicart, *Rapid decline of snow and ice in the tropical Andes – Impacts, uncertainties and challenges ahead,* Earth-Science Reviews **176**, 195 (2018).

[31] EPSAS, *Plan Maestro Metropolitano de Agua Potable y Saneamiento La Paz - El Alto , Bolivia Informe especial N ° 1 Aprovechamiento sostenible de las fuentes de abastecimiento superficiales y subterráneas*, Tech. Rep. (La Paz, 2014).

[32] J. Houston, *A recharge model for high altitude, arid, Andean aquifers,* Hydrological Processes **23**, 2383 (2009).

[33] F. Maier, I. van Meerveld, K. Greinwald, T. Gebauer, F. Lustenberger, A. Hartmann, and A. Musso, *Effects of soil and vegetation development on surface hydrological properties of moraines in the Swiss Alps,* Catena - in Review , 104353 (2019).

[34] A. M. MacDonald, L. Maurice, M. R. Dobbs, H. J. Reeves, and C. A. Auton, *Relating in situ hydraulic conductivity, particle size and relative density of superficial deposits in a heterogeneous catchment,* Journal of Hydrology **434-435**, 130 (2012).

[35] S. Barontini, A. Clerici, R. Ranzi, and B. Bacchi, *Saturated Hydraulic Conductivity and Water Retention Relationships for Alpine Mountain Soils,* in *Climate and Hydrology in Mountain Areas* (John Wiley & Sons, Ltd, 2006) pp. 101–121.

[36] G. Sposito, *Understanding the budyko equation,* Water (Switzerland) **9**, 1 (2017).

[37] A. Mianabadi, M. Coenders-Gerrits, P. Shirazi, B. Ghahraman, and A. Alizadeh, *A global Budyko model to partition evaporation into interception and transpiration,* Hydrology and Earth System Sciences Discussions , 1 (2017).

[38] A. Soruco, C. Vincent, B. Francou, and J. F. Gonzalez, *Glacier decline between 1963 and 2006 in the Cordillera Real, Bolivia,* Geophysical Research Letters **36**, 2 (2009).

[39] M. Hrachowitz, H. H. Savenije, G. Blöschl, J. J. McDonnell, M. Sivapalan, J. W. Pomeroy, B. Arheimer, T. Blume, M. P. Clark, U. Ehret, F. Fenicia, J. E. Freer, A. Gelfan, H. V. Gupta, D. A. Hughes, R. W. Hut, A. Montanari, S. Pande, D. Tetzlaff, P. A. Troch, S. Uhlenbrook, T. Wagener, H. C. Winsemius, R. A. Woods, E. Zehe, and C. Cudennec, *A decade of Predictions in Ungauged Basins (PUB)-a review,* Hydrological Sciences Journal **58**, 1198 (2013).

[40] H. H. Savenije, *HESS opinions "topography driven conceptual modelling (FLEX-Topo)",* Hydrology and Earth System Sciences **14**, 2681 (2010).

[41] H. Gao, M. Hrachowitz, F. Fenicia, S. Gharari, and H. H. Savenije, *Testing the realism of a topography-driven model (FLEX-Topo) in the nested catchments of the Upper Heihe, China,* Hydrology and Earth System Sciences **18**, 1895 (2014).

[42] F. Fenicia, H. H. Savenije, and H. C. Winsemius, *Moving from model calibration towards process understanding,* Physics and Chemistry of the Earth **33**, 1057 (2008).

[43] K. Beven, *Rainfall-Runoff Modelling*, 2nd ed. (John Wiley & Sons, Ltd, West Sussex, 2012) p. 4.

[44] M. H. Braun, P. Malz, C. Sommer, D. Farías-Barahona, T. Sauter, G. Casassa, A. Soruco, P. Skvarca, and T. C. Seehaus, *Constraining glacier elevation and mass changes in South America,* Nature Climate Change **9**, 130 (2019).

[45] D. J. Cooper, J. Sueltenfuss, E. Oyague, K. Yager, D. Slayback, E. M. C. Caballero, J. Argollo, and B. G. Mark, *Drivers of peatland water table dynamics in the central Andes, Bolivia and Peru,* Hydrological Processes **33**, 1913 (2019).

[46] D. C. White, M. M. Lewis, G. Green, and T. B. Gotch, *A generalizable NDVI-based wetland delineation indicator for remote monitoring of groundwater flows in the Australian Great Artesian Basin,* Ecological Indicators **60**, 1309 (2016).

[47] D. P. Groeneveld, W. M. Baugh, J. S. Sanderson, and D. J. Cooper, *Annual groundwater evapotranspiration mapped from single satellite scenes,* Journal of Hydrology **344**, 146 (2007).

[48] M. Vuille, *Atmospheric circulation over the Bolivian Altiplano during dry and wet periods and extreme phases of the southern oscillation,* International Journal of Climatology **19**, 1579 (1999).