

## 0.1 Abstract

The **daWUAP** team at the University of Montana has created a hydrological rainfall-runoff model that predicts streamflows across the state of Montana. **daWUAPhydroengine** is informed by a variety of smaller models, including a groundwater model, snow water equivalent (swe) model, and agricultural component model. This paper focuses on the calibration of **daWUAPhydroengine** via a Dual State Parameter Estimation Ensemble Kalman Filter. Uniquely, this filter operates on multiple high dimensional parameters represented as raster images. These raster images are both ingested and outputted by the **daWUAPhydroengine** and were previously set at a constant value. The filter aims to both calibrate and distribute these variables geospatially.



# Contents

0.1	Abstract . . . . .	1
<b>1</b>	<b>Introduction</b>	<b>5</b>
<b>2</b>	<b>Methods</b>	<b>9</b>
2.1	The Hierarchical Dual Ensemble Kalman Filter . . . . .	9
2.1.1	Prediction Phase . . . . .	10
2.1.2	Parameter Correction Phase . . . . .	11
2.1.3	State Correction Phase . . . . .	12
<b>3</b>	<b>Results</b>	<b>15</b>
3.1	Implementation for daWUAPhydroengine . . . . .	15
<b>4</b>	<b>Discussion</b>	<b>21</b>
4.1	Hierarchical Kalman Filter . . . . .	21
<b>5</b>	<b>Conclusion</b>	<b>23</b>
<b>A</b>	<b>Appendix A</b>	<b>25</b>
A.1	The Dual Ensemble Kalman Filter . . . . .	25
A.1.1	Prediction Phase . . . . .	25
A.1.2	Parameter Correction Phase . . . . .	26
A.1.3	State Correction Phase . . . . .	27



# Chapter 1

## Introduction

Utilizing sequential data assimilation techniques to filter hydrologic models is an efficient way to correct and calibrate them both before and after implementation in the field. Many observations such as SWE, streamflow, and precipitation are collected on a daily basis across various geographic regions, allowing the previous day's information to be dynamically ingested by the hydrologic model and inform present and future predictions. More accurate models allow hydrologists to better understand the past and predict the future.

Models that ingest data sequentially can be efficiently corrected by a Kalman Filter, a sequential data assimilation algorithm. Kalman Filters only need the previous timestep's state estimate, parameter estimate, and co-variance matrices to update the current timestep's state estimate, parameter estimate, and co-variance matrices. The original Kalman filter[7] was created to solve linear problems and more complicated implementations must be used to solve non-linear problems. The extended Kalman Filter[5] works for mildly non-linear systems but does not function optimally on heavily non-linear systems[9]. The Unscented Kalman Filter[6] is an all-around improvement on the Extended Kalman Filter that allows for the filtering of highly non-linear systems. The Ensemble Kalman Filter[4], a predecessor to the Unscented Kalman Filter, filters non-linear systems by generating an 'ensemble' of model instances and adding unique noise to each model's forcing data. The main advantage of this ensemble based approach is the substitution of the original Kalman Filter's

error covariance matrix with an ensemble covariance matrix, which allows for the efficient computation of the covariance of high dimensional state vectors.

In order to calibrate parameters within a hydrologic model a Dual State Kalman Filter may be used as demonstrated by Moradkhani et. al in 2005 [10]. Dual state Kalman filters add a small perturbation to a series of parameters that the user wishes to calibrate. These perturbed parameters vectors are then corrected in a similar fashion to the state vectors. After this happens a 'second' filter runs and corrects the state vectors in the normal fashion. The Dual State Ensemble Kalman Filter implemented by Moradkhani et. al [10] extends the Ensemble Kalman Filter into a dual state configuration and successfully predicts a set of parameters.

In this paper hierarchical modeling techniques are combined with a Dual State Ensemble Kalman Filter's parameter perturbation equation to create a Hierarchical Dual State Ensemble Kalman Filter. A hierarchical parameter perturbation framework allows the model to account for parameters that are spatially similar. To examine the Hierarchical Dual State Ensemble Kalman Filter's application to high dimensional geospatially distributed raster data Professor Marko Maneta's **daWUA-Phydroengine**, a hydrological rainfall-runoff model that predicts streamflows across the state of Montana, is used. **daWUA-Phydroengine** is informed by a variety of smaller models, including a groundwater model, snow water equivalent (swe) model, and agricultural component model. Each of these smaller models are influenced by a set of uncalibrated high-dimensional parameters that are stored as 2D raster data. Despite this complexity, **daWUA-Phydroengine** is designed to be a quick and efficient model. Accordingly, a Hierarchical Dual State Ensemble Kalman Filter is an optimal calibration algorithm to calibrate this raster data because 1) the HDEnKF does not have to compute the high dimensional state covariance matrix and 2) **daWUA-Phydroengine** is a sequential model.

Chapter 2 covers the methods behind the Hierarchical Dual Ensemble Kalman Filtering algorithm. Chapter 3 discusses the application of a Hierarchical Dual Ensemble Kalman Filter to **daWUA-Phydroengine**. Chapter 4 discusses results and compares those results with calibrated parameters from a Dual State Ensemble Kalman Filter

as implemented by Moradkhani et al.





# Chapter 2

## Methods

### 2.1 The Hierarchical Dual Ensemble Kalman Filter

A generic dynamic model can be defined as one more more discrete nonlinear stochastic processes[3]:

$$x_{t+1} = f(x_t, u_t, \theta_t) + \varepsilon_t \quad (2.1)$$

where  $x_t$  is an  $n$  dimensional vector representing the state variables of the model at time step  $t$ ,  $u_t$  is a vector of forcing data (e.g temperature or precipitation) at time step  $t$ , and  $\theta_t$  is a vector of model parameters which may or may not change per time step (e.g *soil beta* or *DDF*). The non-linear function  $f$  takes these variables as inputs. The noise variable  $\varepsilon_t$  accounts for both model structural error and for any uncertainty in the forcing data.

A state's observation vector  $z_t$  can be defined as

$$z_t = h(x_t, \theta_t) + \delta_t \quad (2.2)$$

Where the  $x_t$  vector represents the true state,  $\theta_t$  represents the true parameters,  $h(.)$  is a function that determines the relationship between observation and state vectors, and  $\delta_t$  represents observation error.  $\delta_t$  is Gaussian and independent of  $\varepsilon_t$ .

The Dual Hierarchical State Ensemble Kalman Filter can be split into three sub-

sections: The prediction phase, the parameter correction phase, and the state correction phase.

### 2.1.1 Prediction Phase

Just as in a standard Dual Ensemble Kalman filter (see [Appendix A](#) for methods), each ensemble member  $i$  is represented by a stochastic model similar to (2.1). The modified equation is as follows:

$$x_{t+1}^{i-} = f(x_t^{i+}, u_t^i, \theta_t^{i-}) + \omega_t, \quad i = 1, \dots, n \quad (2.3)$$

Where  $n$  is the total number of ensembles. The  $-/+$  superscripts denote corrected (+) and uncorrected (−) values. Note that  $\theta_t^{i-}$ 's  $t$  subscript does not necessarily denote that  $\theta$  is time variant but rather indicates that parameter values change as they are filtered over time. The noise term  $\omega_t$  accounts for model error and will hereafter be excluded from the state equation.

Errors in the forcing data are accounted for through the perturbation the forcing data vector  $u_t$  with random noise  $\zeta_t^i$  to generate a unique variable  $u_t^i$  for each ensemble.  $\zeta_t^i$  is drawn from a normal distribution with a covariance matrix  $Q_t^i$ .

$$u_{t+1}^i = u_t + \zeta_t^i, \quad \zeta_t^i \sim N(0, Q_t^i) \quad (2.4)$$

### Hierarchical Parameter Perturbation

To generate the priori parameters  $\theta_{t+1}^{i-}$  an evolution of the parameters similar to the evolution of the state variables must be implemented. Legacy implementations of parameter evolution added a small perturbation sampled from  $N(0, \Sigma_t^\theta)$ , where  $\Sigma_t^\theta$  represents the covariance matrix of  $\theta$  at timestep  $t$ . This legacy method of evolution resulted in overly disposed parameter samples and the loss of continuity between two consecutive points in time [8] [3]. To overcome this the kernel smoothing technique developed by West [12] and implemented by Liu [8] has been used effectively in previous Dual Ensemble Kalman filter implementations [10] and similar models [3].

$$\theta_{t+1}^{i-} = a\theta_t^{i+} + (1-a)\bar{\theta}_t^+ + \tau_t^i \quad (2.5)$$

$$\tau_t^i = N(0, h^2 V_t) \quad (2.6)$$

Where  $\bar{\theta}_t^+$  is the mean of the parameters with respect to the ensembles,  $V_t = \text{var}(\theta_t^{i+})$ ,  $a$  is a shrinkage factor between (0,1) of the kernel location, and  $h$  is a smoothing factor.  $h$  is defined by  $\sqrt{1-a}/2$ , while  $a$  is generally between (.45,.49). Note that  $h$  and  $a$  tend to vary per model and optimal values for these parameters are generally found via experimentation [10] [1] [2] [3].

In a Hierarchical Duel Ensemble Kalman Filter, parameter perturbation is accomplished using a hierarchical algorithm. For a simple overview of hierarchical models refer to Osborne [11]. First, ensembles are placed into a series of groups  $G_g$  based on shared characteristics (spatial or otherwise.) Algorithms (2.5) and (2.6) are then updated to conform to the hierarchical model structure:

$$\theta_{t+1}^{i-,g} = a\bar{\theta}_{t,g}^+ + (1-a)\langle\theta\rangle_{t,g}^{i+} + \tau_{t,g}^i \quad (2.7)$$

$$\tau_{t,g}^i = N(0, h^2 V_{t,g}^i) \quad (2.8)$$

$$V_t^i = a * \text{var}(\theta_{t,g}) + (1-a)\text{var}(\theta_t^{i-}) \quad (2.9)$$

Where  $a\bar{\theta}_{t,g}^+$  is the mean over all ensembles for all members of group  $g$ ,  $\langle\theta\rangle_{t,g}$  is the mean of all members in ensemble  $i$ , group  $g$ ,  $\text{var}(\theta_{t,g})$  is the variance over the ensembles for all members of group  $g$ , and  $\text{var}(\theta_t^{i-})$  is the variance of all members in ensemble  $i$ , group  $g$ .

### 2.1.2 Parameter Correction Phase

In an Ensemble Kalman Filter, observations are perturbed to reflect model error. To accomplish this  $n$  unique perturbations are created. Therefore, the variable  $z_{t+1}^i$  is

defined as follows:

$$z_{t+1}^i = z_{t+1} + \eta_{t+1}^i, \quad \eta_{t+1}^i = N(0, R_{t+1}) \quad (2.10)$$

Where  $z_{t+1}$  is an observation vector defined by (2.2) and  $\eta_{t+1}^i$  is a random perturbation drawn from a normal distribution with covariance matrix  $R_{t+1}$ . A set of state predictions that can be related to the observations are generated by running the priori state vector through the function  $h(\cdot)$ :

$$\hat{y}_{t+1}^i = h(x_{t+1}^{i-}, \theta_{t+1}^{i-}) \quad (2.11)$$

The parameter update equation is similar to the update equation of the linear Kalman filter  $\hat{x}_t^+ = \hat{x}_t^- + K_t(z_t - H\hat{x}_t^-)$ . Notably, parameters are corrected in lieu of the states:

$$\theta_{t+1}^{i+} = \theta_{t+1}^{i-} + K_{t+1}^\theta(z_{t+1}^i - \hat{y}_{t+1}^i) \quad (2.12)$$

To facilitate this,  $K_{t+1}^\theta$  is defined as

$$K_{t+1}^\theta = \frac{\Sigma_{t+1}^{\theta, \hat{y}}}{\Sigma_{t+1}^{\hat{y}, \hat{y}} + R_{t+1}} \quad (2.13)$$

where  $\Sigma_{t+1}^{\theta, \hat{y}}$  is the cross covariance of  $\theta_{t+1}$  and  $\hat{y}_{t+1}$ ,  $\Sigma_{t+1}^{\hat{y}, \hat{y}}$  is the covariance of  $\hat{y}_{t+1}$ , and  $R_{t+1}$  is the observation error matrix from (2.10).

### 2.1.3 State Correction Phase

After  $\theta_{t+1}^{i+}$  has been calculated the model is run again (2.3) with the  $\theta_{t+1}^{i+}$  replacing  $\theta_{t+1}^{i-}$ .

$$x_{t+1}^{i-} = f(x_t^{i+}, u_t^i, \theta_t^{i+}), \quad i = 1, \dots, n \quad (2.14)$$

After a new state vector is generated it is re-run through (2.11) with the new parameter vector:

$$\hat{y}_{t+1}^i = h(x_{t+1}^{i-}, \theta_{t+1}^{i+}) \quad (2.15)$$

The corrected state vector is then run through the state update equation

$$x_{t+1}^{i+} = x_{t+1}^{i-} + K_{t+1}^x (z_{t+1}^i - \hat{y}_{t+1}^i) \quad (2.16)$$

$$K_{t+1}^x = \frac{\Sigma_{t+1}^{x,\hat{y}}}{\Sigma_{t+1}^{\hat{y},\hat{y}} + R_{t+1}} \quad (2.17)$$

where  $\Sigma_{t+1}^{x,\hat{y}}$  is the cross covariance of  $x_{t+1}$  and  $\hat{y}_{t+1}$ .



# Chapter 3

## Results

### 3.1 Implementation for daWUAPhydroengine

The **daWUAPhydroengine** hydrologic model is designed to be implemented in any geographic location. For this study it was utilized to model streamflows throughout the state of Montana. **daWUAPhydroengine** takes precipitation, minimum temperatures, and maximum temperatures as inputs and outputs streamflow data along with some additional states such as snow water equivalent. Since Montana is such a large geographical area sections of the landscape differ in various ways (soil composition, forestation, etc.) To account for this, a series of geospatially distributed raster parameters are calibrated by the filter. Since each parameter is corrected by model state predictions and observations raster images were separated into distinct theissen raster regions, each with an observation point at its center.

To apply the filter to **daWUAPhydroengine** minimum temperature, maximum temperature, and precipitation were treated as forcing data.

Table 3.1: Calibrated Parameters

Parameter ( $\theta$ )	Purpose	Dimensions
ddf	Controls Rate of Snowfall	45012
aet_lp	Controls AET	45012
soil_beta	Controls portion of ponded water that goes into soil storage	45012
soil_max_wat	Controls soil maximum water capacity	45012

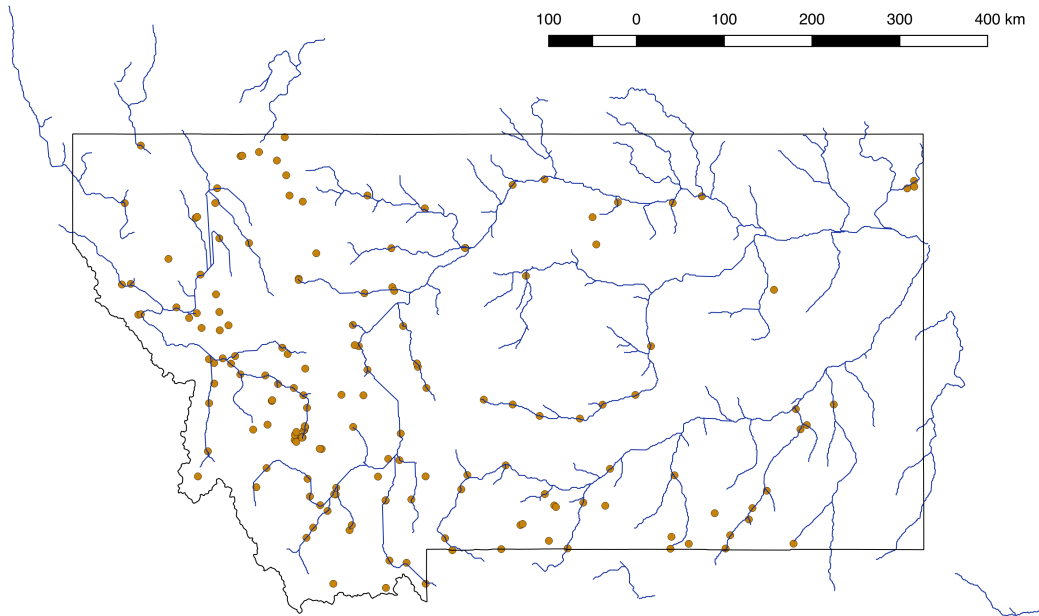


Figure 3-1: all SWE stations plotted against modeled streamflows

Table 3.2: Forcing Data

Forcing Data ( $u$ )	Purpose	Dimensions
tempmin	Lowest temperature for timestep	45012
tempmax	Highest temperature for timestep	45012
precipitation	Amount of rainfall for timestep	45012

To implement the Dual Ensemble Kalman Filter with these parameters and forcing data it was necessary to set a minimum and maximum cutoff point for random perturbation (temperatures under 0 Kelvin, for example, break the filter.) Although this method worked well for the forcing data the maximum and minimum cutoffs broke the calibrated parameters. This is because the filter would sometimes correct above or below the parameter values, which resulted in a collapse of the ensemble variance. Since ensemble variance is used to determine the variance of the next timestep's random perturbations the filter could not recover. Two solutions were implemented to keep this from happening: 1) A 'minimum variance' was specified and 2) When one or more values would be corrected past a minimum or maximum cutoff the whole



Table 3.3: States

State ( $x$ )	Purpose	Dimensions
streamflow	Streamflow (in cumeecs)	330
swe	Snow Water Equivalent (in $mm^3$ )	45012

ensemble would simply not correct for that iteration.

Each run involved 100-200 ensembles and took anywhere from 20 hours to 2 days to run. The filter initially did not converge correctly when the kernel smoothing  $a$  and  $h$  values suggested by Moradkhani et al. were used. When the stronger set of values suggested by Chen et al. [3] were used the parameters converged nicely. Although it was initially worried that stronger kernel smoothing would not allow the ensembles to converge to the correct value values seem constant and consistent with other examples of the literature.

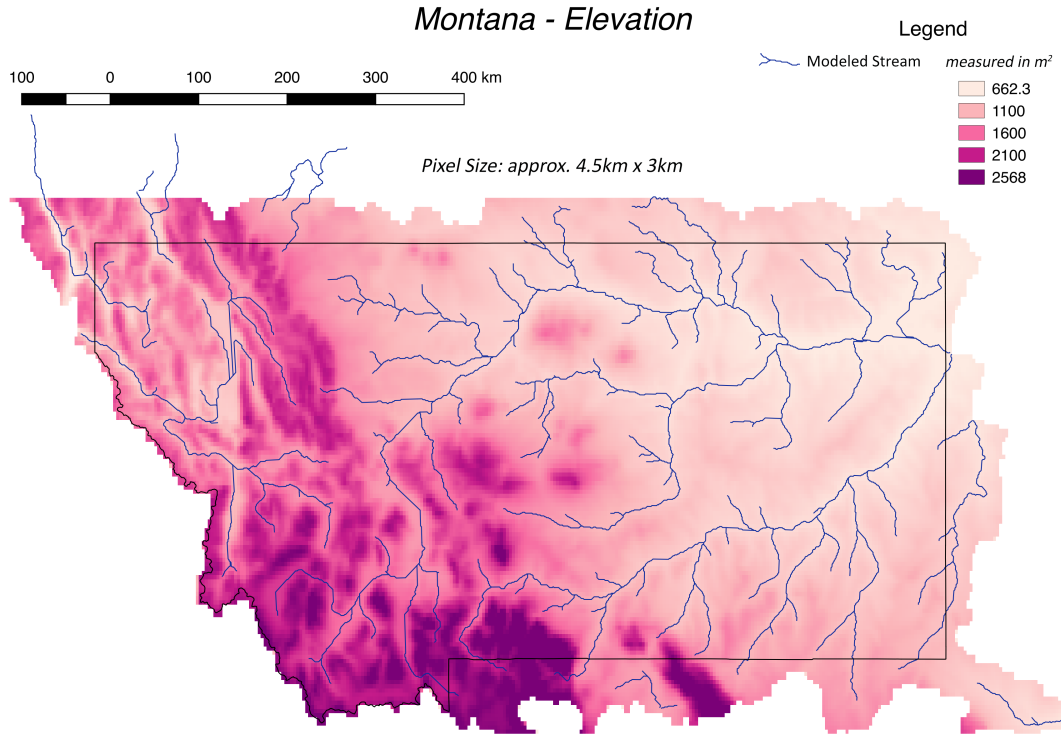


Figure 3-2: Elevation throughout Montana

[3-3](#) is a visualization of the resulting raster of the calibrated soil max water content parameter. [3-4](#) is a visualization of the resulting raster raster of the soil beta

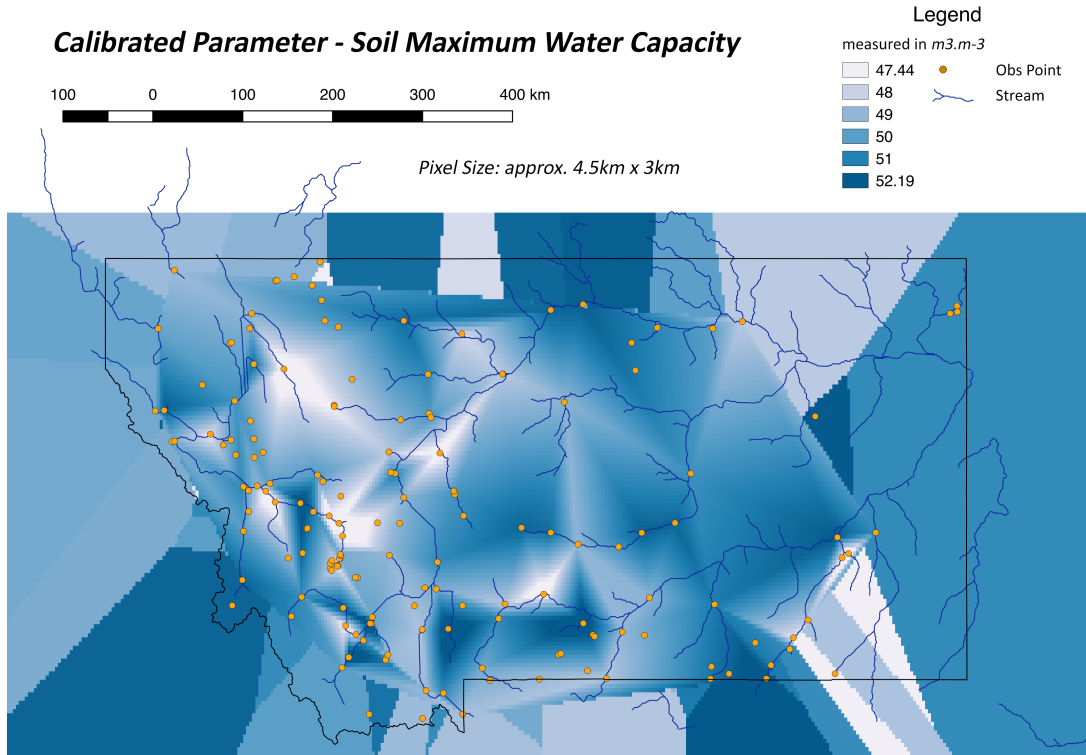


Figure 3-3: A calibrated raster - maximum water capacity

parameter. When contrasting these with 3-2 it becomes clear that these parameters are not correlated with elevation. If this was the case, it would point to a flaw in **daWUAPhydroengine**'s algorithms. Instead, parameter values are spaced evenly and do not form a trivial pattern. This is optimal, as these parameters make up for real world factors that **daWUAPhydroengine**'s methods do not take into account.

3-5 shows the convergence process of three streamflow parameters: AET LP, Soil Beta, and Soil Max Water. The confidence interval is substituted for the ensemble cloud since the amount of perturbation per time step is derived from the variance over all ensembles. Each ensemble converges very nicely.

3-6 display stream observations compared to two instances of **daWUAPhydro-engine**, one using the calibrated parameters and one using the default parameters. Both models were started without spun up parameters (e.g both models started with 0 for each streamflow and have to converge with the real world.) Note how much more successful the calibrated model behaves.

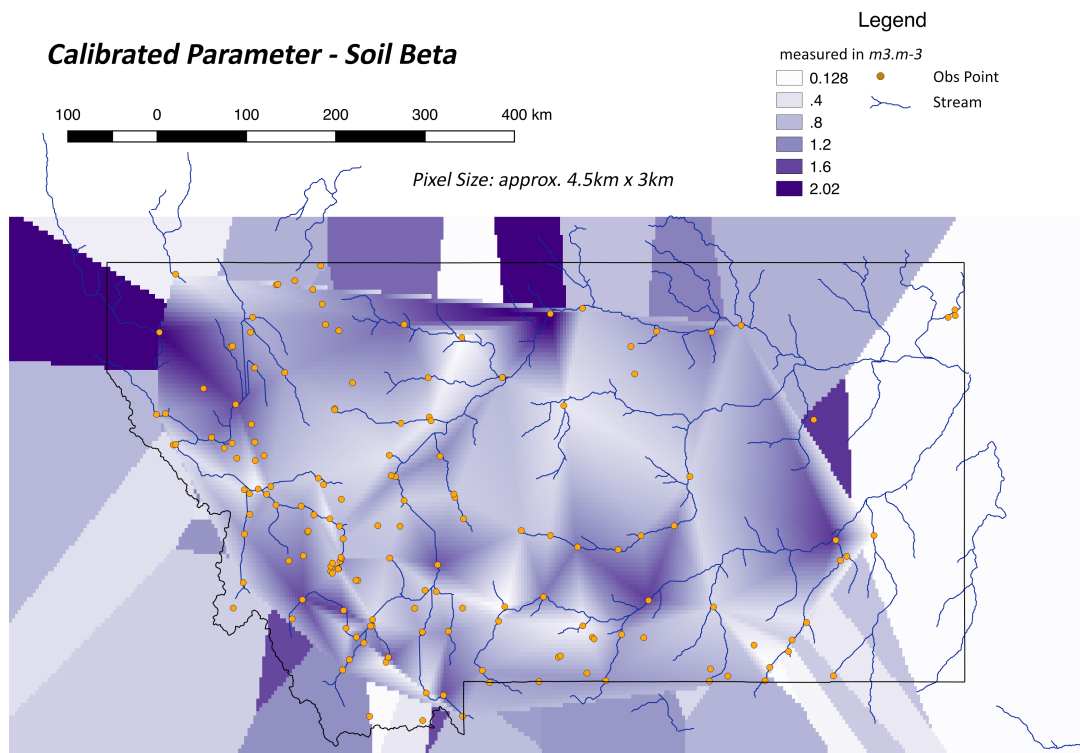


Figure 3-4: A calibrated raster - maximum water capacity

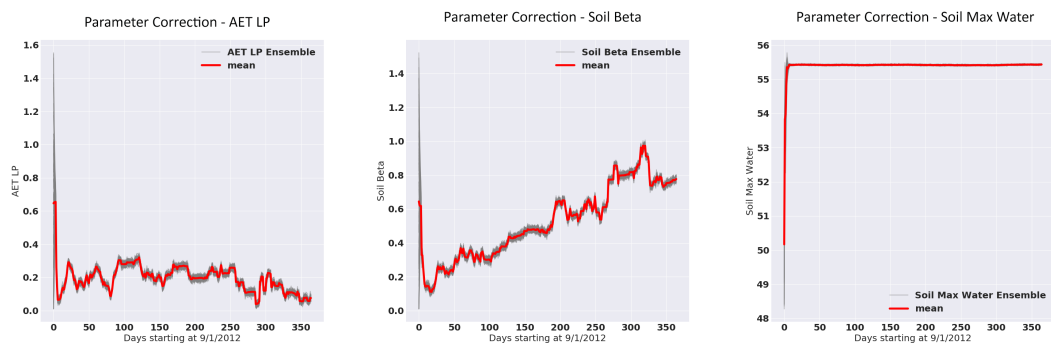


Figure 3-5: Parameter convergences over time

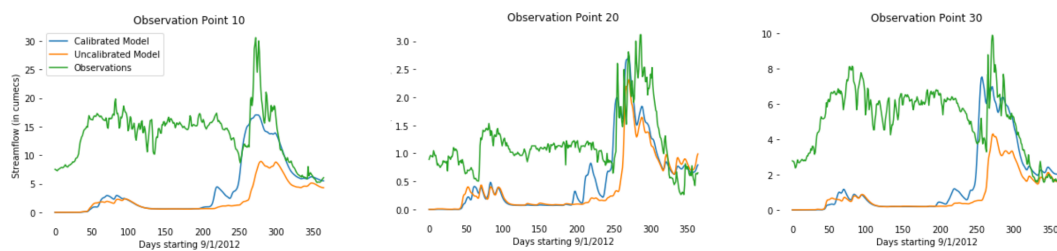


Figure 3-6: A calibrated raster - maximum water capacity



# Chapter 4

## Discussion

Currently, Prof. Johnson and Prof. Maneta and I are working on a way to combine a Hierarchical regression technique with this Kalman filter. We believe that this will allow us to switch from the interpolation scheme shown earlier in this paper to a scheme where catchments are used as parameter zones.

### 4.1 Hierarchical Kalman Filter

The old equation for the evolution of priori parameters is:

$$\theta_{t+1}^{i-} = a\theta_t^{i+} + (1-a)\bar{\theta}_t^+ + \tau_t^i \quad (4.1)$$

$$\tau_t^i = N(0, h^2 V_t) \quad (4.2)$$

$$V_t = var(\theta_{t+1}) \quad (4.3)$$

Here is a draft of the hierarchical evolution:

$$\theta_{t+1}^{i-} = \alpha_t^{i-} C_t^i + (1 - \alpha_t^{i-}) G_t^i + \tau_t^i \quad (4.4)$$

$$C_t^i = a\theta_t^{i+} + (1-a)\bar{\theta}_t^+ \quad (4.5)$$

$$G_t^i = a\bar{\theta}_t^{i+} + (1-a)\bar{\bar{\theta}}_t^+ \quad (4.6)$$

$$\tau_t^i = N(0, h^2 V_t) \quad (4.7)$$

$$V_t = \alpha_t^{i-} \text{var}(h^2 \theta_{t+1}) + (1 - \alpha_t^{i-}) \text{var}(h^2 G_t^i) \quad (4.8)$$

$$\alpha_{t+1}^{i-} = a \alpha_t^{i+} + (1 - a) \bar{\alpha}_t^+ + \delta_t^i \quad (4.9)$$

$$\delta_t = N(0, h^2 {}_{\alpha} V_t) \quad (4.10)$$

$${}_{\alpha} V_t = \text{var}(\alpha_t) \quad (4.11)$$

Where initial  $\alpha$ ,  $m$ , and  $n$  are variables choosable by the user. This is a work in progress.

# Chapter 5

## Conclusion

In conclusion, this Dual Ensemble Kalman Filter was successfully implemented on a hydrologic model. The resulting parameters do lead to better estimation of hydrologic components. Although these results are good, we hope to eventually implement an innovative new Hierarchical Kalman filter that will calibrate parameters at the catchment level.





# Appendix A

# Appendix A

## A.1 The Dual Ensemble Kalman Filter

### A.1.1 Prediction Phase

In a Dual Ensemble Kalman filter, each ensemble member  $i$  is represented by a stochastic model similar to (2.1). The modified equation is as follows:

$$x_{t+1}^{i-} = f(x_t^{i+}, u_t^i, \theta_t^{i-}) + \omega_t, \quad i = 1, \dots, n \quad (\text{A.1})$$

Where  $n$  is the total number of ensembles. The  $-/+$  superscripts denote corrected (+) and uncorrected (−) values. Note that  $\theta_t^{i-}$ 's  $t$  superscript does not necessarily denote that  $\theta$  is time variant but rather indicates that parameter values change as they are filtered over time. The noise term  $\omega_t$  accounts for model error and will hereafter be excluded from the state equation.

Errors in the forcing data are accounted for through the perturbation the forcing data vector  $u_t$  with random noise  $\zeta_t^i$  to generate a unique variable  $u_t^i$  for each ensemble.  $\zeta_t^i$  is drawn from a normal distribution with a covariance matrix  $Q_t^i$ .

$$u_{t+1}^i = u_t + \zeta_t^i, \quad \zeta_t^i \sim N(0, Q_t^i) \quad (\text{A.2})$$

To generate the priori parameters  $\theta_{t+1}^{i-}$  an evolution of the parameters similar to

the evolution of the state variables must be implemented. To accomplish this the kernel smoothing technique developed by West [12] and implemented by Liu [8] is used. Legacy implementations of parameter evolution added a small perturbation sampled from  $N(0, \Sigma_t^\theta)$ , where  $\Sigma_t^\theta$  represents the covariance matrix of  $\theta$  at timestep  $t$ . This legacy method of evolution resulted in overly disposed parameter samples and the loss of continuity between two consecutive points in time [8] [3]. Kernel smoothing has been used effectively to solve this problem in previous Dual Ensemble Kalman filter implementations [10] and similar models [3].

$$\theta_{t+1}^{i-} = a\theta_t^{i+} + (1-a)\bar{\theta}_t^+ + \tau_t^i \quad (\text{A.3})$$

$$\tau_t^i = N(0, h^2 V_t) \quad (\text{A.4})$$

Where  $\bar{\theta}_t^+$  is the mean of the parameters with respect to the ensembles,  $V_t = \text{var}(\theta_t^{i+})$ ,  $a$  is a shrinkage factor between (0,1) of the kernel location, and  $h$  is a smoothing factor.  $h$  is defined by  $\sqrt{1-a}/2$ , while  $a$  is generally between (.45,.49). Note that  $h$  and  $a$  tend to vary per model and optimal values for these parameters are generally found via experimentation [10] [1] [2] [3].

### A.1.2 Parameter Correction Phase

In an Ensemble Kalman Filter, observations are perturbed to reflect model error. Therefore, the variable  $z_{t+1}^i$  is defined as follows:

$$z_{t+1}^i = z_{t+1} + \eta_{t+1}^i, \quad \eta_{t+1}^i = N(0, R_{t+1}) \quad (\text{A.5})$$

Where  $z_{t+1}$  is an observation vector defined by (2.2) and  $\eta_{t+1}^i$  is a random perturbation drawn from a normal distribution with covariance matrix  $R_{t+1}$ . A set of state predictions that can be related to the observations are generated by running the priori state vector through the function  $h(\cdot)$ :

$$\hat{y}_{t+1}^i = h(x_{t+1}^{i-}, \theta_{t+1}^{i-}) \quad (\text{A.6})$$

The parameter update equation is similar to the update equation of the linear Kalman filter  $\hat{x}_t^+ = \hat{x}_t^- + K_t(z_t - H\hat{x}_t)$ . Notably, parameters are corrected in lieu of the states:

$$\theta_{t+1}^{i+} = \theta_{t+1}^{i-} + K_{t+1}^\theta(z_{t+1}^i - \hat{y}_{t+1}^i) \quad (\text{A.7})$$

To facilitate this,  $K_{t+1}^\theta$  is defined as

$$K_{t+1}^\theta = \frac{\Sigma_{t+1}^{\theta, \hat{y}}}{\Sigma_{t+1}^{\hat{y}, \hat{y}} + R_{t+1}} \quad (\text{A.8})$$

where  $\Sigma_{t+1}^{\theta, \hat{y}}$  is the cross covariance of  $\theta_{t+1}$  and  $\hat{y}_{t+1}$ ,  $\Sigma_{t+1}^{\hat{y}, \hat{y}}$  is the covariance of  $\hat{y}_{t+1}$ , and  $R_{t+1}$  is the observation error matrix from (A.5).

### A.1.3 State Correction Phase

After  $\theta_{t+1}^{i+}$  has been calculated the model is run again (A.1) with the  $\theta_{t+1}^{i+}$  replacing  $\theta_{t+1}^{i-}$ .

$$x_{t+1}^{i-} = f(x_t^{i+}, u_t^i, \theta_t^{i+}), \quad i = 1, \dots, n \quad (\text{A.9})$$

After a new state vector is generated it is re-run through (A.6) with the new parameter vector:

$$\hat{y}_{t+1}^i = h(x_{t+1}^{i-}, \theta_{t+1}^{i+}) \quad (\text{A.10})$$

The corrected state vector is then run through the state update equation

$$x_{t+1}^{i+} = x_{t+1}^{i-} + K_{t+1}^x(z_{t+1}^i - \hat{y}_{t+1}^i) \quad (\text{A.11})$$

$$K_{t+1}^x = \frac{\Sigma_{t+1}^{x, \hat{y}}}{\Sigma_{t+1}^{\hat{y}, \hat{y}} + R_{t+1}} \quad (\text{A.12})$$

where  $\Sigma_{t+1}^{x, \hat{y}}$  is the cross covariance of  $x_{t+1}$  and  $\hat{y}_{t+1}$ .



# Bibliography

- [1] Jeffrey L. Anderson, Stephen L. Anderson, Jeffrey L. Anderson, and Stephen L. Anderson. A Monte Carlo Implementation of the Nonlinear Filtering Problem to Produce Ensemble Assimilations and Forecasts. *Monthly Weather Review*, 127(12):2741–2758, dec 1999.
- [2] J. D. Annan, D. J. Lunt, J. C. Hargreaves, and P. J. Valdes. Parameter estimation in an atmospheric GCM using the Ensemble Kalman Filter. *Nonlinear Processes in Geophysics*, 12(3):363–371, feb 2005.
- [3] M. Chen, S. Liu, L.L. Tieszen, and D.Y. Hollinger. An improved state-parameter analysis of ecosystem models using data assimilation. *Ecological Modelling*, 219(3-4):317–326, dec 2008.
- [4] Geir Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *Journal of Geophysical Research*, 99(C5):10143, may 1994.
- [5] Andrew H. Jazwinski. Stochastic Processes and Filtering. *Mathematics in Science and Engineering*, 1970.
- [6] Simon J. Julier and Jeffrey K. Uhlmann. New extension of the Kalman filter to nonlinear systems. volume 3068, page 182. International Society for Optics and Photonics, jul 1997.
- [7] R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1):35, mar 1960.
- [8] Fang Liu and Fang Liu. Bayesian Time Series: Analysis Methods Using Simulation-Based Computation. 2000.
- [9] Robert N. Miller, Michael Ghil, and François Gauthiez. Advanced Data Assimilation in Strongly Nonlinear Dynamical Systems. *Journal of the Atmospheric Sciences*, 51(8):1037–1056, apr 1994.
- [10] Hamid Moradkhani, Soroosh Sorooshian, Hoshin V. Gupta, and Paul R. Houser. Dual state-parameter estimation of hydrological models using ensemble Kalman filter. *Advances in Water Resources*, 28(2):135–147, feb 2005.

- [11] Jason W. Osborne. Advantages of hierarchical linear modeling. Osborne, Jason W. *Research & Evaluation*, 7(1), 2000.
- [12] Mike West. Mixture models, Monte Carlo, Bayesian updating, and dynamic models. *Computing Science and Statistics*, 1993.