Project Final Report

Project Title: (**Amendment to**) Integrating Data Assimilation and Multi-modeling Within CHPS for Improved Seasonal Drought Prediction

Award Number: NA13OAR4310093

PI Name: Hamid Moradkhani

Email: hamidm@pdx.edu

Total Support: $17,500

Final Report: 2/27/2015

Hamid Moradkhani, PhD, PE, D.WRE, F.ASCE
Associate Professor

Department of Civil & Environmental Engineering

Portland State University

1930 S.W. 4th Avenue, Suite 200

Portland, OR 97201

Phone: (503) 725-2436

Fax: (503) 725-5950

Summary:

This project has been the amendment to the main project entitled above. According to the earlier planning Dr. Pedro Restrepo (previously at OHD and now at the NCRFC, Chanhassen, Minn.) was involved in this project to mainly provide assistance with the development of the Community Hydrologic Prediction System (CHPS) adapters within Flood Early Warning System (FEWS) framework developed at Deltares. Given his job relocation to North Central River Forecast Center (NCRFC) as a hydrologist in charge, more effort by Portland State University (PSU) was needed to meet the project objectives, hence more personnel support in the amount of $17,500 was requested.

The modified research tasks for the 3rd year of the project were as follows:

- Develop the model adapters for CHPS and integrate the data assimilation framework developed at PSU to the standalone CHPS system.
- Incorporate the framework within the newly developed operational CHPS.

1. Introduction-Data Assimilation

Bringing ensemble data assimilation (DA) into the Community Hydrologic Prediction System (CHPS) is of high interest due to its applicability to hydrologic forecast initialization. Hydrologic forecasts are highly sensitive to initial conditions (i.e. soil moisture and snow water equivalent), and therefore accurate estimation of these states, and their uncertainty is vital to forecasts. DA refers to a range of techniques that may be used to ingest observed information into a model simulation to reduce uncertainty, and therefore improve the accuracy, of simulations. The DA adapter developed for FEWS/CHPS currently utilizes the Particle Filter (PF) developed by the PI and his research group.

2. Particle Filtering

Application of the PF requires viewing a hydrologic model through the state space framework, and applying the model in an ensemble framework. Within hydrologic models, states may be physically based (e.g. soil moisture or snow water equivalent) or conceptual (i.e. conceptual reservoir water content). For example, the Snow-17 model estimates the snow water equivalent, but the Sacramento Soil Moisture Accounting model estimates soil moisture as a series of conceptual reservoirs. Via the state-space framework, an ensemble simulation of a hydrologic model may be viewed according to equation (1).

$$x_{i,t} = f(x_{i,t-1}, u_{i,t}, \theta) + \omega_{i,t} \tag{1}$$

In equation (1), $x_{i,t}$ represents the state vector at time *t*, which is the sum of the model (f(.)) estimate and the model error $\omega_{i,t}$. This model requires the true states at the previous time ($x_{i,t-1}$), the true forcing data at time *t* ($u_{i,t}$), and model parameters (θ) to characterize the land surface condition. It is often the case in hydrology that a subsequent model must be used to translate these model states into the observation space. A typical example is applying a hydrologic routing model to translate land surface water fluxes to flow at a watershed outlet, allowing for simple comparison of simulated and observed runoff. This model is referred to as an observational operator and is represented in equation (2).

$$y_{i,t} = h(x_{i,t}, \Psi) + \nu_{i,t} \tag{2}$$

where $y_{i,t}$ represents the forecast value, which is the sum of the observational operator (h(.)) estimate and the observational operator error $\nu_{i,t}$. The observational operator requires the true state value and true parameters (Ψ). Though θ and Ψ are identified as independent values/vectors in the above notation, in hydrologic model analysis, these may be examined in a combined vector, which can be represented as θ for simplicity. From this framework, the posterior distribution of the states, may be estimated through sequential Bayes Law.

Sequential Bayes Law is provided in equation (3).

$$p(x_{1:N,t}, \theta_t \mid \tilde{y}_{1:t}) = \frac{p(\tilde{y}_t \mid x_{1:N,t}, \theta_t)\, p(x_{1:N,t}, \theta_t \mid \tilde{y}_{1:t-1})}{p(\tilde{y}_t \mid \tilde{y}_{1:t-1})} \tag{3}$$

In equation (3), *N* is the ensemble size, $\tilde{y}_t$ is the observation at time *t*, $p(x_{1:N,t}, \theta_t \mid \tilde{y}_{1:t-1})$ represents the prior information (estimated with the model), $p(\tilde{y}_t \mid x_{1:N,t}, \theta_t)$ represents the likelihood and $p(\tilde{y}_t \mid \tilde{y}_{1:t-1})$ is the normalizing constant.

In the PF, the posterior distribution is estimated as an ensemble, where each ensemble member is weighted after each observation becomes available. In its most basic form, a PF performs Sequential Importance Sampling (SIS). SIS propagates a Monte Carlo sample of potential states and parameters over a number of time steps. The posterior at each time step is represented by SIS through equation (4). At each time that an observation is available, the weight of each point in the sample is updated. In accordance with sequential Bayes Law, this posterior weight is the normalized product of the likelihood and the prior probability, as shown in equations (5) and (6).

$$p\left(x_t, \theta_t \mid \hat{y}_t\right) \approx \sum_{i=1}^{N_{ens}} w_{i,t}^{+} \delta\left(x_t - \hat{x}_{i,t}^{-}, \theta_t - \theta_{i,t}^{-}\right) \tag{4}$$

$$p\left(y_{i,t} \mid \hat{x}_{i,t}^{-}, \theta_{i,t}^{-}\right) = \frac{L\left(y_{i,t} \mid \hat{x}_{i,t}^{-}, \theta_{i,t}^{-}\right)}{\sum_{i=1}^{N_{ens}} L\left(y_{i,t} \mid \hat{x}_{i,t}^{-}, \theta_{i,t}^{-}\right)} = p\left(y_{i,t} - \hat{y}_{i.t} \mid R_k\right) \tag{5}$$

$$w_{i,t}^{+} = \frac{w_{i,t}^{-} \cdot p\left(y_{i,t} \mid \hat{x}_{i,t}^{-}, \theta_{i,t}^{-}\right)}{\sum_{i=1}^{N_{ens}} w_{i,t}^{-} \cdot p\left(y_{i,t} \mid \hat{x}_{i,t}^{-}, \theta_{i,t}^{-}\right)} \tag{6}$$

Though SIS can theoretically estimate the posterior distribution at each time step in a hydrologic model, practically the sample will develop a few highly weight particles with many low weighted particles. This is referred to as weight degeneracy, and leads to a poorly representative sample. In order to avoid this scenario, resampling is typically performed, which replicates particles of high weights, and discards particles of low weights. Through resampling, all particles are kept within meaningful portions of the posterior, which leads to a more accurate representation of predictive uncertainty.

3. FEWS/CHPS

The Flood Early Warning System (FEWS) provides a framework for manipulating and passing time series information, intended for use in models used for flood forecasting. Within this framework, FEWS passes information across a "Published Interface", via xml files, to allow the end-user to adapt any model to their system. This has been performed for the National Weather Service to create CHPS with the OHDFewsAdapter (documented at ftp://hydrology.nws.noaa.gov/pub/CHPS/For_Software_Developers/). This adapter is the insertion point in CHPS, translating the Published Interface files into a framework that is more conducive to running Office of Hydrologic Development (OHD) models. As shown in Figure 1, the OHDFewsAdapter takes the files passed through the Published Interface, "instantiates" the model driver, and then the driver passes any other necessary information through xml or text files to the models itself. This allows any model, whether or not it is written in Java, to be adapted to the CHPS framework.

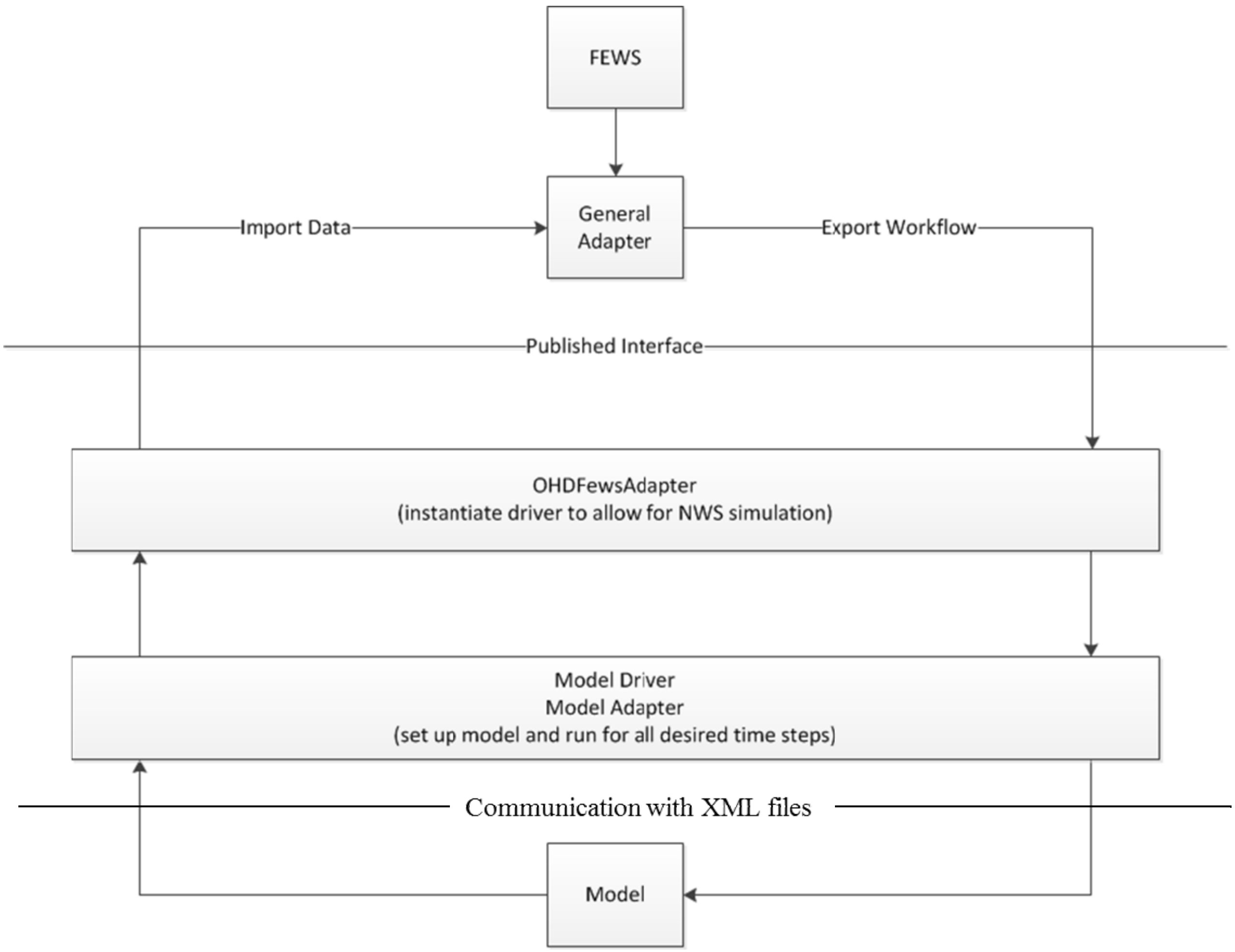

Figure 1. Flow chart of FEWS and CHPS

The DADriver is a driver developed to perform data assimilation based on a user configuration, which is capable of performing assimilation on any model adapted to CHPS. Configuration of the DADriver for an experiment is similar to setting up any other model that has a driver already developed for CHPS, but has some additional complications. A primary point is that a workflow within a data assimilation system will be treated as a singular module instance. For example, assimilation of snow into a basin with two elevation bands will only have 1 DADriver module instance for both snow17 model runs, but the DADriver will be directed to run both snow DA experiments.

4. Challenges of Bringing Data Assimilation into CHPS

FEWS and CHPS are developed to run a single model simulation at a time. In order to make this system as modular, and therefore flexible, as possible, FEWS/CHPS pass much of their information through xml files. By passing information through xml files, via the FEWS "published interface", a subsequent model can be adapted to an existing system, without any additional software development to FEWS or CHPS. The flow of data in the FEWS/CHPS system is shown in Figure 1. In this figure, xml files are used to pass information across the published interface, and between the OHDFewsAdapter and the actual model drivers.

Data flow within a DA system differs from that of FEWS. As shown in Figure 2, the model simulations are performed in an ensemble loop, during each time-step. This requires the model simulations to be stopped at each time-step, and subsequent models run before moving on to a further time-step. Since FEWS ships whole time-series' to a model, one at a time, updates in a sequential manner become quite challenging. If a model is to be run one step at a time, in an ensemble fashion, the computational demand will become excessive. Due to the number of xml files that need to be passed at each model run, for each ensemble member, it is infeasible to run ensemble simulations without significant software development on the CHPS side. This software development is described in following sections.

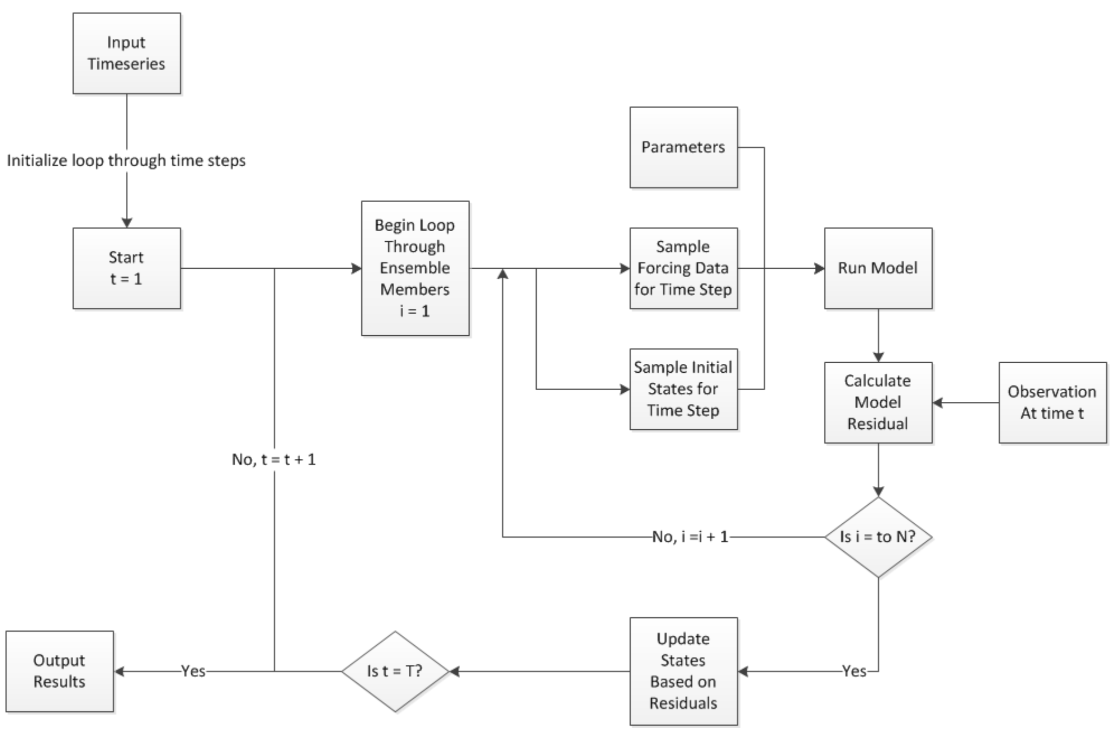

Figure 2. DA flowchart

5. Development of Data Assimilation within CHPS

The challenges mentioned above required the development of a new model driver. This is called the DADriver, and is a separate java class that takes on many functions of the OHDFewsAdapter. Utilizing the functionalities of the OHDFewsAdapter to pass files for multiple models, which was developed for the flash flood guidance (FFG) system, the DADriver can run multiple models simultaneously. This is advantageous, as multiple models may be required to perform the data assimilation. For example, streamflow assimilation will require, at the very least, a SAC-SMA and Unit-HG model. The DADriver instantiates each of these models, and runs them one time-step at a time, in an ensemble framework. This allows for assimilation of the observation at any time when one becomes available. With the input files passed to the DADriver, the DADriver instantiates and ensemble of model drivers, which it then provides information to and executes as necessary. Several differences between standard CHPS simulations and with the DADriver are listed in Table 1.

Table 1. Comparison of standard simulations and DA simulations

Standard Simulations	DA Simulations
OHDFewsAdapter ships data for model runs one at a time	OHDFewsAdapter ships data for multiple model runs (similar to ffh)
Each model's driver performs simulation over the whole timeseries	DADriver only gives data required to run driver to the next observation
OHDFewsAdapter imports output data from single model	OHDFewsAdapter imports data from all models involved in DA simultaneously

In addition to the DADriver, a class for running an ensemble model was developed, which is called the StochasticModel. This class houses all of the instantiated models, and will run or manipulate the drivers when told to do so. Within this class, a sampling utility, named samplingUtils, will perform all of the error sampling necessary to run an ensemble simulation. It also applies the DA algorithm when an observation is provided, and performs the multi-model averaging in the event that multi-modeling is being performed.

Computational expense becomes an issue with very large ensemble sizes, which may be required based on the project setup. In order to reduce this demand, the DA system utilizes parallel computing to run ensemble members on separate processors. In order to achieve this, the ParallelOHDModelAdapter and ParallelOHDModel classes were developed, which relies on the Java ExecutorService utility. In addition, these classes skip some of the file writing required by the models themselves, further reducing computational demand. The ParallelOHDModelAdapter class sets up and starts the parallel computation, and the ParallelOHDModel class executes the specific class. More specifically, the ParallelOHDModel runs the model, ensures the states are properly set after the update, and then extracts the results from the driver.

The final component of the DA system is the TransformationModelDriver. Since some of the workflows in the NWS forecast system require transformations (i.e. aggregation from hourly to daily flows) between models, and these transformations are only available within FEWS, these transformations are not available to the DADriver. Therefore, a TransformationModelDriver was added as a component to the DA system, to allow for transformations between model simulations. Only a small set of transformations are currently available in the TranformationModelDriver, and therefore may require further development if other transformations are desired.

6. Features of Data Assimilation within CHPS

DA with the DADriver allows for assimilation of any observation into any model that has been adapted to CHPS. Along with this report, several documents have been provided to describe the features of the DADriver, and its subsequent tools. Few specific experiments are explained below. These include creating a synthetic observation, basic streamflow DA (single model with single observation), and a complex data assimilation (multiple models and multiple observations) and how to load ensemble states into the DADriver for forecasting. Each of these features is controlled with input files, primarily the parameter file.

7. Case Studies

Some case studies have been performed to verify the utility of the CHPS Data Assimilation framework. These case studies have been performed in the Johnson Creek and Clackamas River basins in northwestern Oregon, using observed precipitation and temperature data, estimated potential evapotranspiration data and model parameters provided by the Northwest River Forecast Center (NWRFC). Also, each of these case studies include a synthetic data assimilation experiment. Synthetic experiments serve as a proof of concept, to demonstrate the ability of the method with known error characteristics, and the real data experiments demonstrate the ability of the methods to improve forecasts in a real forecasting environment.

Johnson Creek was selected for the first case study because it is rain dominated, and therefore does not require estimation of snow for accurate streamflow forecasting. In this basin, the NWRFC routinely provides forecasts of flow at the Sycamore gaging station (SYCO3). Since snow is not a major component, only the SAC-SMA model and the Unit hydrograph (Unit-HG) are required. In this case study, synthetic streamflow observations generated by running the model with over a three year period with the perturbed observed data, to simulate data errors, then perturbing the streamflow estimates to simulated model error. Then the particle filter (PF) is applied to the model over the same time period, in order to assimilate the observations, to retrieve the true (simulated with unperturbed data) streamflow. In this scenario, a one-timestep-ahead (6-hourly) forecasting experiment is performed, where the synthetic observation is assimilated at each timestep, and the forecast at the following timestep is prepared. The comparison of the open loop (simulated with perturbed data), DA and true streamflow is presented in Figure 3. From this figure, it may be observed that the open loop simulation is biased low, in comparison to the true streamflow. After applying the PF, the forecasts are shifted towards the truth, indicating that the PF reduces error. Further evidence of this reduction in error is provided in Table 2, where the Mean Square Error (MSE) and bias are lower in the DA case than the open loop simulation. This suggests that the DA system is effectively improves short-term streamflow forecasting when the error characteristics are known.

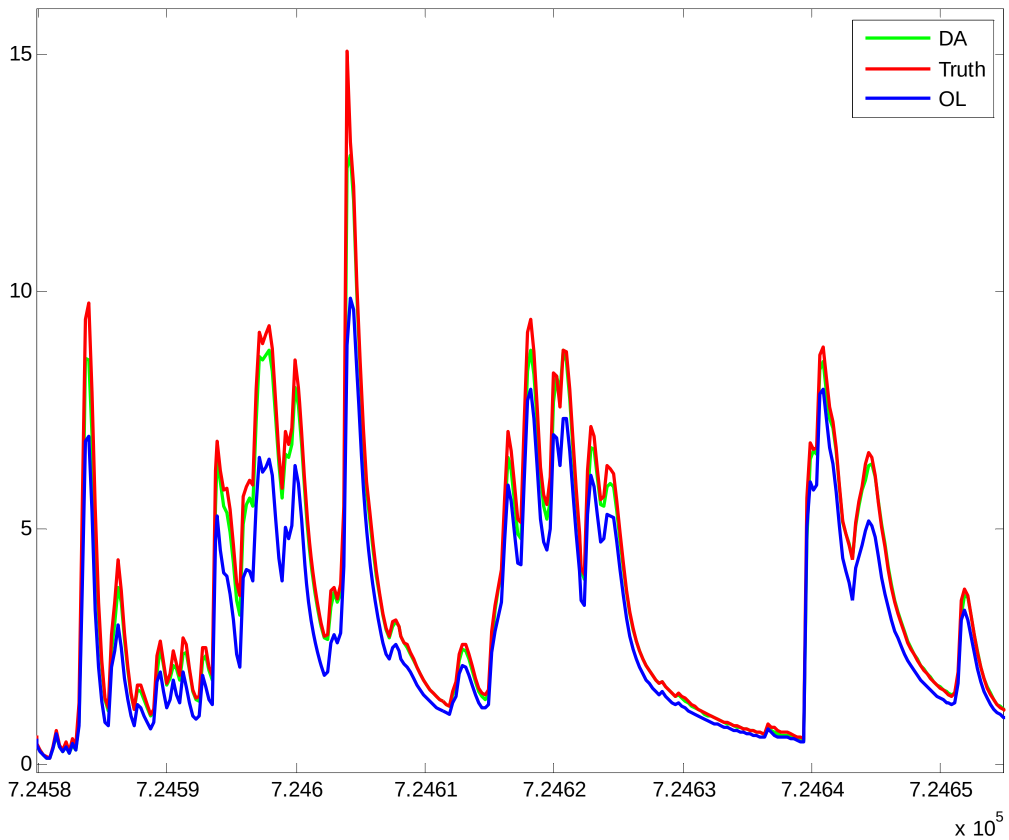

Figure 3. Synthetic streamflow forecasting experiments for Johnson Creek at Sycamore (SYCO3)

In order to demonstrate the effectiveness of the DA system in improving real forecasts, a real data experiment is shown in Figure 4. In this figure, the open loop is the model simulation without perturbed data, DA with the USGS gage measured streamflow observations, and the streamflow observations are compared. In this figure, the improvements due to DA are more modest than those of the synthetic DA experiment. This is to be expected as the model error is a greater factor, the observations are less frequent (daily) and the true error characteristics of the model are unknown. Although the improvements are not as great as the synthetic experiment, the forecast from DA is an improvement over the open loop simulations. This is further evidenced in Table 2, where the MSE and bias from DA is less than the MSE and bias from the open loop simulations. Therefore DA is capable of improving streamflow forecasts in a rainfall dominated basin.

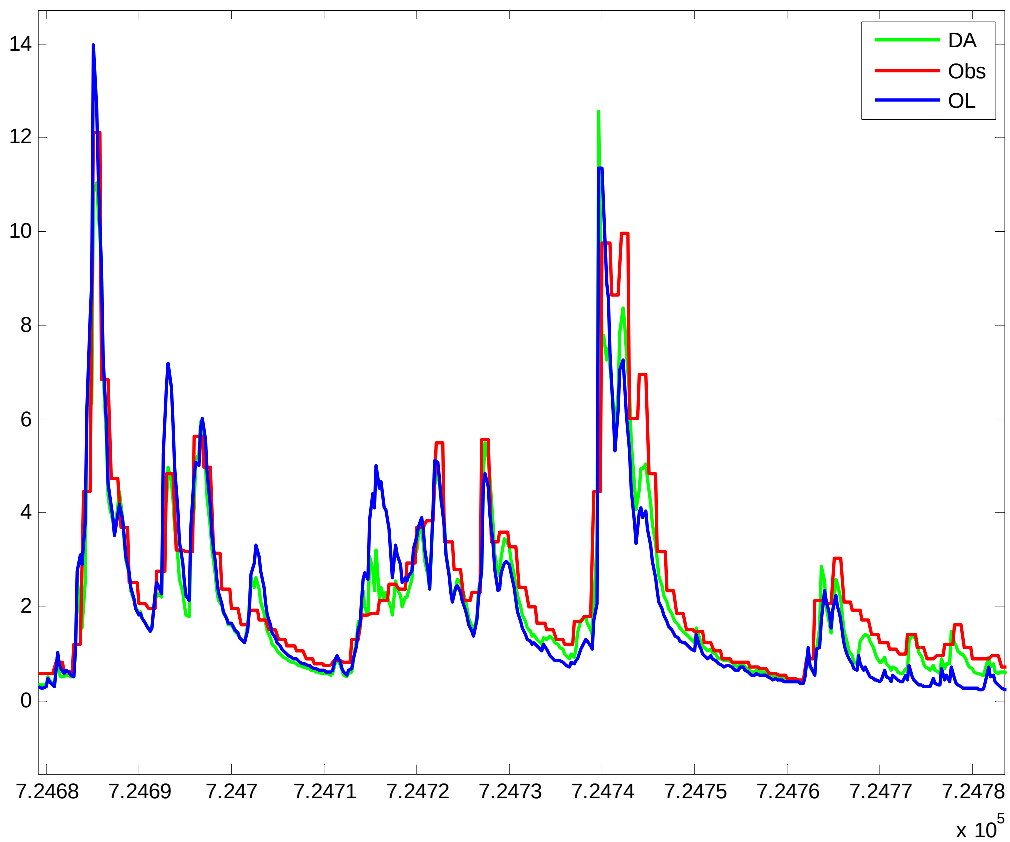

Figure 4. Real streamflow forecasting experiments for Johnson Creek at Sycamore (SYCO3).

Table 4. Comparison of RMSE and bias from both the synthetic and real data experiments

SYCO3 Synthetic Experiments

	MSE	Bias
Open Loop	0.653	-0.3092
Data Assimilation	0.2418	-0.1082

SYCO3 Real Experiments

	MSE	Bias
Open Loop	1.5678	-0.0417
Data Assimilation	1.1233	-0.1607

This project amendment was meant to facilitate the main project which is still ongoing. More case studies with detailed results and elaboration of the adaptors and real data assimilation with inclusion of multi-modeling will be reported.