Data Assimilation and Climate
AIMS Rwanda, March, 2020

Lecture and Lab Activities
Kalman Filter

1. Assume we have a non-stationary data model

$$\mathbf{Y}(i) \quad = \quad \mathbf{H}_i\mathbf{X}(i) + \boldsymbol{\epsilon}_i, \quad i = 1, \ldots, N$$

with n state variables so that $\mathbf{X}(i) \in \mathbb{R}^n$, for all i.

(a) If every variable in the state vector $\mathbf{X}(i)$ is observed at every time step, find $\mathbf{H}_i$.

Solution: $\mathbf{H}_i = \mathbf{I}_n$, i.e. the $n \times n$ identity matrix.

(b) Let $n = 4$ and assume only the first two states are observed at the time step represented by $i = 10$. Find $\mathbf{H}_{10}$.
Solution:

$$\mathbf{H}_{10} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

.

2. Assume we have the one dimensional process model

$$x(i) = \alpha x(i-1) + \delta_i, \quad i = 1, \ldots, N, \quad \delta_i \sim \mathcal{N}(0, q^2),$$

and $x(0)$ is from a standard normal distribution. Assume the data model is

$$y(i) = h_i x(i) + \epsilon_i, \quad i = 1, \ldots, N, \quad \epsilon_i \sim \mathcal{N}(0, r^2), \quad h_i = \begin{cases} 1, & i = 1, \ldots, N_1 \\ 0.1, & i = N_1 + 1, \ldots, N_2 \\ 1 & i = N_2 + 1, \ldots, N \end{cases}$$

(a) Describe somewhat qualitatively, e.g. in words, how this choice of h_i is effecting the data.

Solution: Between N_1 and N_2 the state is not observed as well as it is other times.

(b) Use Python to program and plot the process model using $\alpha = 0.8$, $q = 0.4$ and $N = 30$.

Solution:

```
%matplotlib notebook
%pylab
def process(alpha,x0,q,N):
    # One dimensional process model i.e. the dimension of X is 1
    # N+1 is the number of elements in the time series, including the background
    x=np.zeros(N+1)
    x[0]=x0 # x0 is one dimension
    for i in range(1,N+1):
```

```
        delta=np.random.normal(0,q)
        x[i]=alpha*x[i-1]+delta
    return x
#Find and plot states
N=30
q=0.1 # standard deviation on process
alpha=0.8
x0=np.random.normal(0,1)
x=process(alpha,x0,q,N)
figure(1)
clf()
plot(x,markersize=4)
ylabel('Process variables x_i', fontsize=12)
xlabel('i', fontsize=12)
show()
```

(c) Use Python to program and plot the data on the same graph as the process model. Use $r = 0.1$, $N = 30$, $N_1 = 10$, and $N_2 = 20$.

Solution:

```
def data(x,r,N,N1,N2):
    # Data has period of low observability represented between N1 and N2
    y=np.zeros(N+1) # There is no data y[0]
    for i in range(1,N+1):
        eps=np.random.normal(0,r)
        h=1
        if np.logical_and(N1 +1 <= i, i <= N2):
            h=0.1
        y[i]=h*x[i]+eps
    return y
# Find and plot data
r=0.03 # standard deviation on data
N1=10
N2=20
y=data(x,r,N,N1,N2)
figure(1)
plot(y[1:N],'rx',markersize=4)
show()
```

(d) Run your code multiple times and describe how the graph changes as you sample different values for δ_i and ϵ_i. If you have time change the values of α, q and r and describe your observations.

Solution: For all values we see that the data between $i = 10$ and $i = 20$ do not reflect the state as well as for other values of i. As we sample different values of δ_i and ϵ_i, the shape of the graph changes significantly.

(e) Using the general formulas given for the Kalman filter to find the one dimensional formulas for: $\mu_{1|0}, \sigma_{1|0}, k_1, \mu_{1|1}, \sigma_{1|1}$. Leave your answer in terms of $y(1)$.

Solution:

$$\begin{aligned}
\mu_{1|0} &= \alpha \times 0 = 0, \\
\sigma_{1|0} &= q^2 + \alpha \times 1 \times \alpha = q^2 + \alpha^2, \\
k_1 &= (q^2+\alpha^2) \times 1 \left(1 \times (q^2+\alpha^2) \times 1 + r^2\right)^{-1} = \frac{q^2+\alpha^2}{q^2+\alpha^2+r^2} \\
\mu_{1|1} &= 0 + \frac{q^2+\alpha^2}{q^2+\alpha^2+r^2}(1 \times y(1) - 0) = \frac{q^2+\alpha^2}{q^2+\alpha^2+r^2} y(1), \\
\sigma_{1|1} &= (1 - \frac{q^2+\alpha^2}{q^2+\alpha^2+r^2})(q^2+\alpha^2) = \frac{r^2(q^2+\alpha^2)}{q^2+\alpha^2+r^2}
\end{aligned}$$

(f) Write pseudo code to calculate $\mu_{i|i-1}, \sigma_{i|i-1}, k_i, \mu_{i|i}, \sigma_{i|i}$ for $i = 1, \ldots, N$. Be sure to initialize the loop over i. *Hint: I suggest using mu_old[i] for* $\mu_{i|i-1}$, *sigma_old[i] for* $\sigma_{i|i-1}$, *mu_new[i] for* $\mu_{i|i}$, *sigma_new[i] for* $\sigma_{i|i}$.
Solution:

```
mu_new[0]=mu
sig_new[0]=sig
for i in range(1,N+1):
    h=1
    if np.logical_and(N1 +1 <= i, i <= N2):
        h=0.1
    mu_old[i]=alpha*mu_new[i-1]
    sig_old[i]=q2+alpha*sig_new[i-1]*alpha
    gain[i]=sig_old[i]*h*(h*sig_old[i]*h+r2)**(-1)
    mu_new[i]=mu_old[i]+gain[i]*(y[i]-h*mu_old[i])
    sig_new[i]=(1-gain[i]*h)*sig_old[i]
```

(g) Identify the variables in your pseudo code that give (i) a filtered estimate of the state and its uncertainty, and (ii) a forecast of the state and its uncertainty.

Solution: The filtered estimates are in mu_new[1:N] and standard deviation is $\sqrt{\text{sig\_new[1:N]}}$. The forcasted estimates are in mu_old[1:N] and standard deviation is $\sqrt{\text{sig\_old[1:N]}}$.

(h) Code the Kalman filter in Python for this example. Plot the states x_i and data y_i together with the filtered and forecasted estimates on the same graph. Label the graph with a legend and discuss your results.

Solution: The forecasted estimates are similar to the filtered estimates. The estimates are closer to the true state x_i when $h_i = 1$.

(i) Plot the standard deviation of the filtered and forecasted estimates together with their errors on the same graph. Label the graph with a legend and discuss your results.

Solution: During the period of low visibility, i.e. $h_i = 0.1$, the standard deviations are larger. Forecasting standard deviations are larger than filtering standard deviations. The errors some times lie outside the standard deviation bounds.

3. Consider the nonlinear process model

$$\begin{aligned}\mathbf{X}(i) &= \mathcal{M}_i(\mathbf{X}(i-1)) \\ &= \begin{pmatrix} 4x_1(1-x_1) \\ 4x_2(1-x_2) \\ \vdots \\ 4x_n(1-x_n) \end{pmatrix}(i-1)\end{aligned}$$

for $i = 1, \ldots, N$. Find the tangent linearization of $\mathcal{M}_i$ evaluated at

$$\mathbf{X} = \begin{pmatrix} 1 \\ 2 \\ \vdots \\ 2 \end{pmatrix}$$

Solution:

$$\frac{\partial\left(\mathcal{M}_i\right)}{\partial x} = \begin{pmatrix} 4-8x_1 & 0 & \ldots & \ldots \\ 0 & 4-8x_2 & 0 & \ldots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \ldots & \ldots & 4-8x_n \end{pmatrix}$$

$$\mathbf{M}_i = \begin{pmatrix} -4 & 0 & \ldots & \ldots \\ 0 & -12 & 0 & \ldots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \ldots & \ldots & -12 \end{pmatrix}$$