Journal of Summer 2014

Leah Dickstein

November 9, 2014

Contents

1 2014-06-03 Channel Estimation

1.1 Problem Statement:

$$X + Z = Y$$

$$X \sim N(0, \sigma^2)$$
$$Z \sim N(0, 1)$$

$$\mathbb{E}[X] = 0$$
$$Var[X] = 0$$

1.2 Solution:

$$f_{X|Y=y}(X) = \frac{f_X(X) * f_{Y=y|X}(Y)}{f_{Y=y}(Y=y)}$$
$$Pdf = \frac{1}{\sigma\sqrt{2\pi}} * e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

$$Var[Y|X] = 1$$
$$Var[Y] = \sigma^2 + 1$$
$$Var[X] = \sigma^2$$
$$Var[Z] = 1$$

$$f_{X|Y=y}(X) = \frac{\sqrt{\sigma^2+1}}{\sigma\sqrt{2\pi}} e^{\frac{-(y-x)^2}{2} - \frac{x^2}{2\sigma^2} + \frac{y^2}{2(\sigma^2+1)}}$$
$$\frac{d}{dx} f_{X|Y=y}(X) = \frac{\sigma^2}{\sigma^2+1} Y$$

1.3 Multiple Copies of Y

for the Same Realization of X:
It's more effective because there is an averaging effect in the variance of the noise:

$$Var[Y|X] = \frac{1}{n}$$
$$Var[Y] = \sigma^2 + \frac{1}{n}$$
$$Var[X] = \sigma^2$$
$$Var[Z] = \frac{1}{n}$$

This assumes we are working with Gaussians.

2 Another L[X|Y]

$$min\mathbb{E}[(\alpha Y + \beta - X)^2]$$

$$\begin{aligned}
\frac{d}{d\alpha}\mathbb{E}[(\alpha Y + \beta - X)^2] &= \mathbb{E}[2(\alpha Y + \beta - X)(Y)] = 0 \\
&= \mathbb{E}[\alpha Y^2 + \beta Y - XY] = 0 \\
&= \mathbb{E}[\alpha Y^2] = \mathbb{E}[XY - \beta Y] \\
&= \alpha = \frac{\mathbb{E}[XY - \beta Y]}{\mathbb{E}[Y^2]} \\
&= \alpha = \frac{Cov[XY]}{Var[Y]}
\end{aligned}$$

This assumes either $\beta = 0$ or $\mathbb{E}[Y] = 0$. The former we will soon show, and the latter is a result of Y being 0-mean, which is a result of X and Z being 0-mean.

$$\begin{aligned}
\frac{d}{d\beta}\mathbb{E}[(\alpha Y + \beta - X)^2] &= \mathbb{E}[2(\alpha Y + \beta - X)] = 0 \\
&= \mathbb{E}[\beta] = \mathbb{E}[X - \alpha Y] \\
&= \beta = \mathbb{E}[X] - \alpha\mathbb{E}[Y] \\
&= \beta = 0
\end{aligned}$$

$$L[X|Y] = \alpha Y + \beta = \frac{Cov[XY]}{Var[Y]} Y$$

This assumes we are using a linear estimator for random variables that aren't necessarily Gaussian, thus showing that for Gaussians the optimal estimator **is** the linear the estimator.

3 Notes from Kalman Filter Wikipedia

4 2014-06-05

4.1 Q1

$$x[n] = a * x[n-1]$$
$$x[0] \sim N(0, a^{2n})$$

The variance increases so that the probability of being larger remains the same.

4.2 Q2

$$\begin{aligned}
x[n] &= a * x[n-1] + w[n-1] \\
w[n-1] &\sim N(0, \sigma_w^2) \\
x[n] &\sim N(0, \sigma_w^2 \Sigma_{i=0}^{n-1} a^{2i} + a^{2n})
\end{aligned}$$

4.3 Q3

$$x[n] = a * x[n-1]$$
$$y[n] = c * x[n]$$
$$\hat{x}[n] = 1/c * y[n]$$

In this case, the observation is noiseless.

4.4 Q4

$$
\begin{aligned}
x[n] &= a * x[n-1] + w[n-1] \\
y[n] &= c * x[n] \\
\hat{x}[n] &= 1^{*}_{,}y[n]
\end{aligned}
$$

W[n] doesn't matter because it's incorporated into the state, and we're trying to guess the state.

4.5 Q5

Estimate x[n] using memory.
If the observations are noiseless, then memory doesn't matter since we have perfect observation anyway. If observations are noisy, over time the noise → 0.

If you add noise to the observation, use the $L[X|Y]$ shown above.

5 Notes from EE126 Appendix A

6 Proofs about L[X|Y]

6.1 L[X|Y,Z] = L[X|Y] + L[X|Z]

6.2 L[X|Y,Z] = L[X|Y] + L[X|Z-L[Z|Y]]

7 2014-06-09 2014-06-15 Kalman Filter

7.1 Problem Setup:

$$
\begin{aligned}
X[n] &= AX[n-1] + W[n-1] \\
Y[n] &= CX[n] + V[n]
\end{aligned}
$$

$$
\begin{aligned}
X &\sim N(0, A^{2n} + \sigma_W^2 \Sigma_{i=0}^{n-1} A^{2i} \\
Y &\sim N(0, C^2(A^{2n} + \sigma_W^2 \Sigma_{i=0}^{n-1} A^{2i}) + \sigma_V^2
\end{aligned}
$$

$$
\begin{aligned}
X(0) &\sim N(0,1) \\
W[n] &\sim N(0, \Sigma_W) \\
V[n] &\sim N(0, \Sigma_V)
\end{aligned}
$$

7.2 Goal:

$$
\begin{aligned}
\mathbb{E}[X[n+1]|Y^n] &= \hat{X}[n+1] \\
Y^n &= (Y[0] \dots Y[n]) \\
\mathbb{E}[X[n+1]|Y^n] &= \sqcup \mathbb{E}[X[n]|Y^{n-1}] + \sqcup(Y[n] - \mathbb{E}[Y[n]|Y^{n-1}]
\end{aligned}
$$

7.3 Equations:

$$
\begin{aligned}
&(1) & L[X|Y] &= \mathbb{E}[X] + \frac{cov(X,Y)}{cov(Y)}(Y - \mathbb{E}[Y]) \\
&(2) & L[X|Y,Z] &= L[X|Y] + L[X|Z - L[Z|Y]] \\
&(3) & cov(AX, CY) &= Acov(X,Y)C' \\
&(4) & if V, W \perp cov(V+W) &= cov(V) + cov(W)
\end{aligned}
$$

$$
\mathbb{E}[X[n+1]|Y^n] = \mathbb{E}[X[n+1]|Y^{n-1}] + \mathbb{E}[X[n+1]|Y[n] - \mathbb{E}[Y[n]|Y^{n-1}]
$$

7.4 $\mathbb{E}[X[n+1]|Y^{n-1}]$

$$
\begin{aligned}
&(1) \\
&\mathbb{E}[AX[n] + W[n]|Y^{n-1}] = \mathbb{E}[AX[n]|Y^{n-1}] + \mathbb{E}[W[n]|Y^{n-1}] \\
&(2) \qquad = A\hat{X}[n] + \mathbb{E}[W[n]] \\
&(3) \qquad = A\hat{X}[n]
\end{aligned}
$$

7.5 $\mathbb{E}[Y[n]|Y^{n-1}]$

$$
\begin{aligned}
&(4) \\
&\mathbb{E}[CX[n] + V[n]|Y^{n-1}] = C\mathbb{E}[X[n]|Y^{n-1}] + \mathbb{E}[V[n]|Y^{n-1}] \\
&(5) \qquad = C\hat{X}[n]
\end{aligned}
$$

7.6 $\mathbb{E}[X[n+1]|Y[n] - C\hat{X}[n]]$

$$
\begin{aligned}
&(6) \\
&\mathbb{E}[X[n+1]|Y[n] - C\hat{X}[n]] = \mathbb{E}[AX[n] + W[n]|Y[n] - C\hat{X}[n]] \\
&(7) \qquad = \mathbb{E}[AX[n]|Y[n] - C\hat{X}[n]] \\
&(8) \qquad = \mathbb{E}[AX[n] - A\hat{X}[n]|Y[n] - C\hat{X}[n]]
\end{aligned}
$$

Lemma: $Y^{n-1} \perp Y[n] - \mathbb{E}[Y[n]|Y^{n-1}]$

Strong Induction!
Base Case: $cov(Y[0], Y[1] - \mathbb{E}[Y[1]|Y[0]]) = 0$

$$
\begin{aligned}
&(1) & &\mathbb{E}[y[0](cax[0] + cw[0] + v[1] - \frac{ac^2\Sigma_{x[0]}}{\Sigma_{y[0]}}y[0])] \\
&(2) & &= \mathbb{E}[y[0](cax[0] - \frac{ac^2\Sigma_{x[0]}}{\Sigma_{y[0]}}y[0])] \\
&(3) & &= \mathbb{E}[(cx[0] + v[0])cax[0] - \frac{ac^2\Sigma_{x[0]}}{\Sigma_{y[0]}}y^2[0]] \\
&(4) & &= \mathbb{E}[c^2ax^2[0] - \frac{ac^2\Sigma_{x[0]}}{\Sigma_{y[0]}}y^2[0]] \\
&(5) & &= c^2a\Sigma_{x[0]} - \frac{ac^2\Sigma_{x[0]}}{\Sigma_{y[0]}}\Sigma_{y[0]} \\
&(6) & &= c^2a\Sigma_{x[0]} - ac^2\Sigma_{x[0]} = 0
\end{aligned}
$$

Inductive Hypothesis: $cov(Y[n-1], Y[n] - \mathbb{E}[Y[n]|Y[n-1]]) = 0 \wedge \dots \wedge cov(Y[0], Y[n] - \mathbb{E}[Y[n]|Y[0]]) = 0$

Inductive Step: $cov(Y[n], Y[n+1] - \mathbb{E}[Y[n+1]|Y[n]]) = 0$

$$\mathbb{E}[y[n](cax[n] - \frac{c^2 a\Sigma_{x[n]}}{\Sigma_{y[n]}} y[n])] = c^2 a\Sigma_{x[n]} - \frac{c^2 a\Sigma_{x[n]}}{\Sigma_{y[n]}}\Sigma_{y[n]} = 0$$

In addition,

$$\begin{aligned}
&(7) \quad \forall t \leq n, \quad cov(y[t], y[n+1] - \mathbb{E}[y[n+1]|y[t]]) = 0 \\
&(8) \qquad = \mathbb{E}[y[t](ca^{n+1-t}x[t] - \frac{c^2 a^{n+1-t}\Sigma_{x[t]}}{\Sigma_{y[t]}} y[t])] \\
&(9) \qquad = \mathbb{E}[c^2 a^{n+1-t}x^2[t] - \frac{c^2 a^{n+1-t}\Sigma_{x[t]}}{\Sigma_{y[t]}} y^2[t]] \\
&(10) \qquad = c^2 a^{n+1-t}\Sigma_{x[t]} - \frac{c^2 a^{n+1-t}\Sigma_{x[t]}}{\Sigma_{y[t]}}\Sigma_{y[t]} = 0
\end{aligned}$$

If $t = n+1$,$cov(y[n+1], y[n+1] - \mathbb{E}[y[n+1]|y[n+1]]) = cov(y[n+1], y[n+1] - y[n+1]) = cov(y[n+1], 0) = 0$.
The answer is trivial.

$$cov(Y^{n-1}, Y[n] - \mathbb{E}[Y[n]|Y^{n-1}])$$

$$= \mathbb{E}\left[[Y[0] \ldots Y[n-1]]\left[Y[n] - \frac{cov(Y[n], Y^{n-1})}{cov(Y^{n-1})}\begin{bmatrix} Y[0] \\ \vdots \\ Y[n-1] \end{bmatrix}\right]\right]$$

We have proved for $\forall t < n$ this $= 0$, therefore the answer is the 0 vector and we prove the Lemma. Since $\hat{X}[n] = \mathbb{E}[X[n]|Y^{n-1}]$, it is the projection of X onto Y^{n-1}. If $Y^{n-1} \perp \tilde{Y}, \hat{X}[n] \perp \tilde{Y}$. We can add if inside the cov() since it's equivalent to adding 0.

$$cov(AX[n] - A\hat{X}[n], CX[n] - C\hat{X}[n]) = Acov(X[n] - \hat{X}[n])C'$$

$$S_n = cov(X[n] - \hat{X}[n])$$

$$cov(Y[n] - C\hat{X}[n]) = cov(CX[n] + V[n] - C\hat{X}[n])$$

$$= cov(C(X[n] - \hat{X}[n])) + cov(V[n]) = CS_nC' + \sigma_v^2$$

$$K_n = \frac{AS_nC'}{CS_nC' + \sigma_v^2}$$

$$\hat{X}[n+1] = \mathbb{E}[X[n+1]|Y^n]$$

$$\boxed{= A\hat{X}[n] + \frac{Acov(X[n] - \hat{X}[n])C'}{Ccov(X[n] - \hat{X}[n])C' + \sigma_v^2}\left(Y[n] - C\hat{X}[n]\right)}$$

8 2014-06-16 Underlying X1 and X2

8.1 Problem Setup

$$\begin{bmatrix} x_1(n+1) \\ x_2(n+1) \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix} * \begin{bmatrix} x_1(n) \\ x_2(n) \end{bmatrix}$$

Y(n) = $[1 \quad 1] * \begin{bmatrix} x_1(n) \\ x_2(n) \end{bmatrix}$

Calculate $x_1(n), x_2(n)$ from y(n)
Calculate $x_1(n), x_2(n)$ from y(n), y(n-1)

Variations: Y(n) = $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} x_1(n) \\ x_2(n) \end{bmatrix}$

Y(n) = $[0 \quad 1] * \begin{bmatrix} x_1(n) \\ x_2(n) \end{bmatrix}$

8.2 Solution

$$X[n] = \begin{bmatrix} 2 & 0 \\ 0 & 3 \end{bmatrix} * \begin{bmatrix} 1 & 1 \\ 2 & 3 \end{bmatrix}^{-1} * \begin{bmatrix} y(n-1) \\ y(n) \end{bmatrix}$$

$$X[n] = \begin{bmatrix} 6 & -2 \\ -6 & 3 \end{bmatrix}\begin{bmatrix} y(n-1) \\ y(n) \end{bmatrix}$$

In this case we showed that when dealing with X_1 and X_2, system error matters. This is because instead of trying to estimate the fluctuating line, we are now trying to estimate the "straight line" where the system "should" go.

9 2014-06-18 Conditions of Observability

$$\begin{bmatrix} C \\ \vdots \\ CA^{\lfloor \frac{m}{n} \rfloor} \end{bmatrix} * \begin{bmatrix} x_1(n - \lfloor \frac{m}{n} \rfloor) \\ \vdots \\ x_n(n - \lfloor \frac{m}{n} \rfloor) \end{bmatrix} = \begin{bmatrix} y(n - \lfloor \frac{m}{n} \rfloor) \\ \vdots \\ y(n) \end{bmatrix}$$

Check that $\begin{bmatrix} C \\ \vdots \\ CA^{\lfloor \frac{m}{n} \rfloor} \end{bmatrix}$ is full rank, then delete $\lceil \frac{m}{n} \rceil * n - m = -mmodn$ lines and solve for x.

$\begin{bmatrix} C \\ \vdots \\ CA^{\lfloor \frac{m}{n} \rfloor} \end{bmatrix}$ must be full rank, or span X.

10 2014-06-19 Various Proofs

10.1 Best Control

$$\begin{aligned}
&min\mathbb{E}[||x(n+1)||^2] \\
=&min\mathbb{E}[||ax(n) + w(n) + u(n)||^2] \\
=&min\mathbb{E}[||ax(n) + w(n) + \alpha y(n) + \beta||^2] \\
=&min\mathbb{E}[||ax(n) + w(n) + \alpha cx(n) + \alpha v(n) + \beta||^2]
\end{aligned}$$

$$\begin{aligned}
\frac{d}{d\alpha}\mathbb{E}[] &= \mathbb{E}[2(ax(n) + w(n) + \alpha cx(n) + \alpha v(n) + \beta)(cx(n) + v(n))] = 0 \\
&= \mathbb{E}[acx^2(n) + \alpha(c^2x^2(n) + v^2(n))] = 0 \\
&= \alpha\mathbb{E}[c^2x^2(n) + v^2(n)] = -\mathbb{E}[acx^2(n)] \\
&= \alpha = \frac{-acVar(x)}{Var(Y)}
\end{aligned}$$

$$\begin{aligned}
\frac{d}{d\beta}\mathbb{E}[] &= \mathbb{E}[2(ax(n) + w(n) + \alpha cx(n) + \alpha v(n) + \beta)] = 0 \\
&= \mathbb{E}[\beta] = 0
\end{aligned}$$

$$u(n) = \frac{-acVar(X)}{Var(Y)}Y(n) = -L[X|Y]$$

10.2 Error of Control Problem

$$\begin{aligned}
&\mathbb{E}[||x(n) + u(n)||^2] \\
&= \mathbb{E}[||x(n) - \frac{ac\Sigma_X}{\Sigma_Y}(cx(n) + v(n))||^2] \\
&= \mathbb{E}[||(1 - \frac{ac\Sigma_X}{\Sigma_Y})x(n) - \frac{ac\Sigma_X}{\Sigma_Y}v(n)||^2] \\
&\rightarrow \boxed{(1 - \frac{ac\Sigma_X}{\Sigma_Y})^2\Sigma_X + (\frac{ac\Sigma_X}{\Sigma_Y})^2\Sigma_V}
\end{aligned}$$

10.3 Without System Error, Estimation Error = 0

In this system, we assume $a \leq 1$.

$$\begin{aligned}
\Sigma_n &= cov(x(n) - \hat{x}(n)) \\
S_n &= A^2\Sigma_{n-1} + \Sigma_W = A^2\Sigma_{n-1} \\
\Sigma_n &= (1 - KnC)Sn = A^2(1 - KnC)\Sigma_{n-1} \\
&\rightarrow 0 < \left(1 - \frac{c^2S_n}{c^2S_n + \Sigma_V}\right) < 1 \\
\left(1 - \frac{c^2S_n}{c^2S_n + \Sigma_V}\right) &< 1 \rightarrow \frac{c^2S_n}{c^2S_n + \Sigma_V} > 0
\end{aligned}$$

It is easy to see the error coefficient is greater than 0, because 1 - fraction > 0. By definition $c \neq 0$, since the signal must have *some* power. By definition Σ_n starts out > 0, thus $S_n > 0$, thus $\left(1 - \frac{c^2S_n}{c^2S_n+\Sigma_V}\right) < 1$ and $n \rightarrow \infty \implies \Sigma_n \rightarrow 0$.

11 2014-06-29 Kalman Filter with Multiplicative Noise

11.1 Problem Setup:

$$\begin{aligned}
X[n] &= AX[n-1] + BW[n-1] \\
Y[n] &= U[n]CX[n] + V[n] \\
\\
X(0) &\sim N(0, S(0)) \\
W[n] &\sim N(0, \Sigma_W) = N(0, Q(n)) \\
V[n] &\sim N(0, \Sigma_V) = N(0, R(n)) \\
U(n) &\sim N(\mu, \sigma^2) = N(M(n), N(n))
\end{aligned}$$

11.2 Goal:

$$\begin{aligned}
\hat{X}[n] &= \mathbb{E}[X[n]|Y^n] \\
Y^n &= (Y[0] \dots Y[n]) \\
\mathbb{E}[X[n]|Y^n] &= \sqcup\mathbb{E}[X[n]|Y^{n-1}] + \sqcup(Y[n] - \mathbb{E}[Y[n]|Y^{n-1}]
\end{aligned}$$

It is necessary and sufficient: $\tilde{X}[n] = X[n] - \hat{X}[n] \perp Y^n$ or $Y[n] - Y^{n-1}$.

11.3 Equations:

$$L[X|Y] = \mathbb{E}[X] + \frac{cov(X,Y)}{cov(Y)}(Y - \mathbb{E}[Y]) \tag{11}$$

$$L[X|Y,Z] = L[X|Y] + L[X|Z - L[Z|Y]] \tag{12}$$

$$cov(AX, CY) = Acov(X,Y)C' \tag{13}$$

(14)

$$if V, W \perp cov(V + W) = cov(V) + cov(W)$$

$$\mathbb{E}[X[n]|Y^n] = \mathbb{E}[X[n]|Y^{n-1}] + \mathbb{E}[X[n]|Y[n] - \mathbb{E}[Y[n]|Y^{n-1}]$$

11.4 $\mathbb{E}[X[n]|Y^{n-1}]$

(1)

$$\mathbb{E}[AX[n-1] + BW[n-1]|Y^{n-1}] = \mathbb{E}[AX[n-1]|Y^{n-1}] + \mathbb{E}[BW[n -$$

$$= A\hat{X}[n-1] + B\mathbb{E}[W[n-1]] \tag{2}$$

$$= A\hat{X}[n-1] \tag{3}$$

11.5 $\mathbb{E}[Y[n]|Y^{n-1}]$

(4)

$$\mathbb{E}[U[n]CX[n] + V[n]|Y^{n-1}] = \mathbb{E}[U[n]CX[n]|Y^{n-1}]$$

$$= C\mathbb{E}[U[n]|Y^{n-1}]\mathbb{E}[X[n]|Y^{n-1}] \tag{5}$$

$$= C\mathbb{E}[U[n]] * A\hat{X}[n-1] \qquad = CM(\tag{6}$$

11.6 $\mathbb{E}[X[n]|Y[n] - CM(n)A\hat{X}[n-1]]$

Note: $P(n) = cov(X[n] - A\hat{X}[n-1]) = \mathbb{E}[X^2[n] - 2X[n]A\hat{X}[n-1] + A^2\hat{X}^2[n-1]]$

Numerator:

(7)

$$cov(X[n], U[n]CX[n] + V[n] - CM[n]A\hat{X}[n-1])$$

(8)

$$= cov(X[n], C(U[n]X[n] - M[n]A\hat{X}[n-1]))$$

(9)

$$= cov(X[n] - A\hat{X}[n-1], C(U[n]X[n] - M[n]A\hat{X}[n-1]))$$

(10)

$$= C\mathbb{E}[U[n]X^2[n] - U[n]AX[n]\hat{X}[n-1] - M[n]X[n]A\hat{X}[n-1] + M[$$

(11)

$$= C(\mathbb{E}[U[n]]\mathbb{E}[X^2[n]] - \mathbb{E}[U[n]]\mathbb{E}[AX[n]\hat{X}[n-1]] - M[n]\mathbb{E}[X[n]A\hat{X}[$$

(12)

$$= C(M[n]\mathbb{E}[X^2[n]] - M[n]\mathbb{E}[AX[n]\hat{X}[n-1]] - M[n]\mathbb{E}[X[n]A\hat{X}[n -$$

(13)

$$= CM[n](\mathbb{E}[X^2[n]] - \mathbb{E}[AX[n]\hat{X}[n-1]] - \mathbb{E}[X[n]A\hat{X}[n-1]] + \mathbb{E}[A^2$$

(14)

$$= CM[n]P[n]$$

Denominator:

(15)
$cov(U[n]CX[n] + V[n] - M[n]CA\hat{X}[n-1])$

(16)
$= cov(C(U[n]X[n] - M[n]A\hat{X}[n-1])) + cov(V[n])$

(17)
$= C^2 cov(U[n]X[n] - M[n]X[n] + M[n]X[n] - M[n]A\hat{X}[n-1]) + R[n]$

(18)
$= C^2 cov((U[n] - M[n])X[n] + M[n](X[n] - A\hat{X}[n-1])) + R[n]$

(19)
$= C^2 \mathbb{E}[||(U[n] - M[n])X[n] + M[n](X[n] - A\hat{X}[n-1])||^2] + R[n]$

(20)
$= C^2 \mathbb{E}[(U[n] - M[n])^2 X^2[n] + 2(U[n] - M[n])X[n]M[n](X[n] - A\hat{X}[n-1]) + M^2[n](X[n] - A\hat{X}[n-1])^2] + R[n]$

(21)
$= C^2 \mathbb{E}[(U[n] - M[n])^2 X^2[n] + M^2[n](X[n] - A\hat{X}[n-1])^2] + R[n]$

(22)
$= C^2(\mathbb{E}[(U^2[n] - 2U[n]M[n] + M^2[n])X^2[n]] + M^2[n]\mathbb{E}[(X[n] - A\hat{X}[n-1])^2]) + R[n]$

(23)
$= C^2(\mathbb{E}[(U^2[n] - M^2[n])X^2[n]] + M^2[n]P[n]) + R[n]$

(24)
$= C^2(\mathbb{E}[N(n) * X^2[n]] + M^2[n]P[n]) + R[n]$

(25)
$= C^2(N[n]S[n] + M^2[n]P[n]) + R[n]$

$$\boxed{Kf = \frac{M[n]P[n]C}{C^2 N[n]S[n] + C^2 M^2[n]P[n] + R[n]}}$$

$$\hat{X}[n] = \mathbb{E}[X[n]|Y^n] = A\hat{X}[n-1] + Kf\left(Y[n] - CM(n)A\hat{X}[n-1]\right)$$

12 Notes from A Mathematical Theory of Communication

12.0.1 Introduction

- Similar to the coding learned in CS70 (Hamming, RSA, Error Correcting Codes), an important aspect of communication is being able to *distinguish* messages. Meaning of messages doesn't matter, but the code for the messages must be a distance apart
  - "The significant aspect is that the actual message is one *selected from a set* of possible messages"
- Log function used to measure information produced when a message is chosen, 3 reasons
  1. Useful: Time, bandwith, etc. vary linearly with log(number of possibilities)
  2. Intuitive: Doubling the possibilities also doubles the amount of information when using the log function
  3. Mathematically suitable: it helps the math work out
- Converting to bits is easy: $log_2 M = log_{10} M / log_{10} 2 = 3.32 log_{10} M$
- Five parts to the communication system:
  1. Information Source
  2. Transmitter
  3. Channel
  4. Receiver
  5. Destination
- Three general types of communication systems: Discrete, continuous and mixed

12.1 Discrete Noiseless Systems

12.1.1 The Discrete Noiseless Channel

- The capacity C of a discrete channel $C = \underset{T \to \infty}{Lim} \frac{logN(T)}{T}$
- **Theorem 1**: Let $b_{ij}^{(s)}$ be the duration of the s^{th} symbol which is allowable in state i and leads to state j.
  C = log W where W is the largest real root of:
  $|\sum_s W^{-b_{ij}^{(s)}} - \delta_{ij}| = 0$

12.1.2 The Discrete Source of Information

- We want to give the shortest codes to the letters/words/phrases with the highest probability, to minimize bits that need to be sent across
- This system that produces a sequence of symbols based on probability = stochastic process; a stochastic process that does this = discrete source
- Simple to Complicated: Choosing letters independently, choosing letters based on probability, choosing letters based on transition probability (based on what the previous letter was), choosing letters based on previous two letters (trigram), choosing letters n-gram, choosing words independently, etc.

12.1.3 The Series of Approximations to English

- Zero-order approximation = Symbols independent and equiprobable
- First-order = Symbols independent but with frequencies of English text
- Second-order = Digram structure
- Third-order = Trigram
- First-order word approximation
- Second-order word approximation
- "Note that these samples have reasonably good structure out to about twice the range that is taken into account in their construction" e.g. If the process ensures reasonable text for two-letter sequences, what actually results is usually reasonable four-letter sequences

12.1.4 Graphical Representation of a Markoff (Markov?) Process

Markoff process = information source if a letter is produced for each transition between states
?? "The states will correspond to the 'residue of influence' from preceding letters"

12.1.5 Ergodic and Mixed Sources

- Ergodic process = statistical homogeneity = as the lengths of the sequences increase, the probabilities approach limits independent of the sequence

- Two properties necessary and sufficient:

  1. The graph doesn't have two isolatd parts A and B that can't reach each other
  2. A closed series of lines in the graph = "circuit"; number of lines = "length" of the "circuit". The GCD of the lengths of all circuits = 1 $\implies$ no periodicity

- If a condition is violated, separate the graph into subgraphs that satisfy the conditions. The source = "mixed" made of pure components $L = p_1L_1 + p_2L_2 + p_3L_3 + \cdots$, with p_i = the probability you start in any of the subgraphs (since isolated, you would never leave the subgraph)

- "Except when the contrary is stated we shall assume a source to be ergodic. This assumption enables one to idenitfy averages along a sequence with averages over the ensemble of possible sequences."

- ?? P_i = probability of state i, $p_i(j)$ = probability of transition from i to j; for process to be stationary equilibrium conditions must be satisfied: $P_j = \sum_i P_i p_i(j)$

12.1.6 Choice, Uncertainty and Entropy

- At what rate is information produced? How much "choice" is involved in the selection of the event or of how uncertain we are of the outcome? Measure of property $H(p_1, p_2, \cdots, p_n)$ needs to satisfy three properties:

  1. H should be continuous in p_i
  2. If all the p_i are equal ($p_i = \frac{1}{n}$), H is a monotonic increasing function of n (more choices = more uncertainty)
  3. If a choice is split into two successive choices, H = weighted sum of individual values of H (each choice should be factored in)

- **Theorem 2**: H = $-K\sum_{i=1}^{n} p_i log p_i$

- H(X) = entropy for random variable X, but X isn't argument to function it acts like a label

- INSERT FIGURE 7

- More properties:

  1. H = 0 iff all p_i but one = 0, meaning there's only one choice and no uncertainty
  2. H is max and = to log(n) if uniform distribution (all $p_i = \frac{1}{n}$)
  3. $H(x,y) \leq H(x) + H(y)$ The uncertainty of joint is less than/equal to sum of individual uncertainties
  4. Any change toward equalization of the probabilities increases H, especially any "averaging" operation
  5. $p_i(j) = \frac{p(i,j)}{\sum_j p(i,j) = \frac{p(i,j)}{p(i)}}$ This looks just like stuff done in CS70
     Conditional entropy $H_x(y) = -\sum_{i,j} p(i,j) log p_i(j)$
     $\implies H(x,y) = H(x) + H_x(y)$ Joint entropy = Uncertainty in X + what we don't know about Y given X (like innovation)
  6. $H(y) \geq H_x(y)$ They are equal if X and Y are independent; knowing X can only provide information so can only decrease uncertainty/entropy

12.1.7 The Entropy of an Information Source

- Entropy per second $H = \sum_i f_i H_i$ where F_i is average frequency of state i

- **Theorem 3**: When N is large, $H \approx \frac{log 1/p}{N}$ very close! MEANS entropy = normalization of -log p; the amount of information we gain is related to how low a probability of the sequence occurring

- **Theorem 4**: $\underset{N\to\infty}{Lim} \frac{log n(q)}{N} = H$ where n(q) = number of messages we must take from the set in order of decreasing probability to have total probability q of those taken; applying Theorem 3 to subsets of sequences

- **Theorem 5**: Let $p(B_i)$ = probability of sequence B_i, let $G_N = -\frac{1}{N}\sum_i p(B_i) log p(B_i)$ summing over all sequences B_i containing N symbols, then G_N is a monotonic decreasing function of N and $\underset{N\to\infty}{Lim} G_N = H$; This basically restates Theorem 2 but defines K for large N and applies it to longer sequences

- **Theorem 6**: Let $p(B_i, S_j)$ be probability of B_i followed by symbol S_j, let $F_N = -\sum_{i,j} p(B_i, S_j) log p_{B_i}(S_j)$, summing over total length N, F_N is monotonic decreasing in N and:
  $G_N = \frac{1}{N}\sum_{n=1}^{N} F_n \mid F_N \leq G_N \mid \underset{N\to\infty}{Lim} F_N = H$

- F_N is the entropy of the N^{th} order approximation of the information source; F_N is the conditonal entropy of the next symbol when (N-1) are known, G_N is the entropy per symbol of blocks of N symbols

- Relative entropy = ratio of entropy of source to its max value with same symbols = measure of max compression possible

- Redundancy = 1 - relative entropy

- Max entropy = max possible states/symbols/choices = max rate of transmission; redundancy = "safety check" of codes to help prevent corruption

12.1.8 Representations of the Encoding and Decoding Operations

Probably don't understand this part well enough

- Transducer = encoding/decoding info at transmitter/receiver
- **Theorem 7**: The output of a finite state transducer driven by a finite state statistical source is a finite state statistical source, with entropy (per unit time) $\leq$ to that of the input. If the transducer is non-singular they are equal
- Transducer (aka encoding/decoding) can only decrease entropy/uncertainty
- **Theorem 8:** Let the system of constraints considered a channel have capacity C = log W. If $p_{ij}^{(s)} = \frac{B_j}{B_i} W^{-l_{ij}^{(s)}}$, where $l_{ij}^{(s)}$ is duration of s^{th} symbol from i to j and $B_i = \sum_{s,j} B_j W^{l_{ij}^{(s)}}$ then H is maximized and = C.

12.1.9 The Fundamental Theorem for a Noiseless Channel

- **Theorem 9**: Let a source have entropy H (bits per symbol) and channel have capacity C (bits per second), then it's possible to go up to an average rate of information $\frac{C}{H}$ but not possible to exceed that
- QUESTION MARK REREAD THIS

12.1.10 Discussion and Examples

- "The source as seen from the channel through the transducer should have the same statistical structure as the source which maximizes the entropy of the channel"
- IMPORTANT/RELEVANT "In general, ideal or nearly ideal encoding requires a long delay in the transmitter and receiver. In the noiseless case which we have been considering, the main function of this delay is to allow reasonably good matching of probabilities to corresponding lengths of the sequences."
- Maximum entropy based on statistical conditions determines channel capacity

12.2 Discrete Noisy Systems

12.2.1 Representation of a Noisy Discrete Channel

$$H(x,y) = H(x) + H_x(y) = H(y) + H_y(x)$$

12.2.2 Equivocation and Channel Capacity

INSERT FIGURE 8

- The rate of actual transmission $R = H(x) - H_y(x)$
- **Theorem 10**: If the correction channel has a capacity equal to $H_y(x)$ it is possible to so encode correction data as to send it over this channel and correct all but an arbitrarily small fraction ϵ of the errors. This isn't possible if the channel capacity $\leq H_y(x)$. $\rightarrow H_y(x)$ is the amount of additional information that must be supplied per second to correct the received message.
- Rate of transmission $R =$
  - $H(x) - H_y(x)$ = the amount of information sent - the uncertainty of what was sent
  - $H(y) - H_x(y)$ = the amount received - noise
  - $H(x) + H(y) - H(x,y)$ = the number of bits per second common between sent message and received message
- C = max R = max $(H(x) - H_y(x))$

12.2.3 The Fundamental Theorem for a Discrete Channel with Noise

- **Theorem 11**: Let a discrete channel have the capacity C and a discrete source the entropy per second H. If $H \leq C$ there exists a coding system s.t. the output of the source can be transmitted over the channel with an arbitrarily small frequency of errors/equivocation. If $H > C$ the equivocation can be minimized to $H - C$.
- INSERT FIGURE 9 and 10

12.2.4 Discussion

- Redundancy combats noise; redundancy increases the distance between codes, so if noise "moves" a code it's still relatively closer to the original than to a different message (think Hamming distance!)
- "As in the noiseless case, a delay is generally required to approach the ideal encoding. It now has the additional function of allowing a large sample of noise to affect the signal before any judgment is made at the receiving point as to the original message." → More delay = more noise
- If N(T,q) = maximum number of signals s.t. the probability of incorrect interpretation $\leq q$, **Theorem 12**: $\underset{T\rightarrow\infty}{Lim} \frac{log N(T,q)}{T} = C$, provided $q \neq 0, 1$

12.2.5 Example of a Discrete Channel and its Capacity

12.2.6 The Channel Capacity in Certain Special Cases

- Suppose symbols used are split into several distinct groups (e.g. noise can't move A_1 to B_i), then:
- Total probability P_n of symbols in nth group $= \frac{2^{C_n}}{\sum 2^{C_n}}$, $C = log \sum 2^{C_n}$

12.2.7 An Example of Efficient Coding

13 Yury Polyanskiy: Channel Coding Rate in the Finite Blocklength Regime

- *Converse*: Upper bound on the size of any code with given arbitrary blocklength and error probability
- *Achievability*: Lower bound on the size of any code guaranteed to exist with given arbitrary blocklength and error probability
- *Asymptotics*: bounds on the log size of the code normalized by blocklength asymptotically coincide
- For ergodic channels, provided the blocklength is allowed to grow without bound, the channel capacity is the max rate of information
- → Maybe this is: most likely codes have shortest lengths, rarer words have longer lengths, as long as there's no limit you can reach channel capacity, but if there's a limit then you have to rearrange what the codes mean and you lose entropy
- Reliability function = asymptotic exponential decay of error probability when transmitting at any given fraction of capacity → error goes down?? If $a < 1$ for estimation and if Kalman filter with no state error?
- Nonasymptotic regime??
- Backoff from C can be characterized by channel dispersion V (backoff = when one user dominates a channel and other users can't access the channel?)
- Finite blocklength (n) coding rate $\frac{logM(n,\epsilon)}{n} \approx C - \sqrt{\frac{V}{n}}Q^{-1}(\epsilon)$
- Fundamental limit vs asymptotic limit
- SNR penalty as a function of blocklength to check suboptimality of code? (the shorter the block length the higher the SNR penalty because...less signal, same noise? but multiplicative noise?)
- Three bounds: RCU (random coding union), DT (dependency testing), and $\kappa\beta$ (Neyman-Pearson lemma)
- Three channels of importance: BEC (binary erasure channel), BSC (binary symmetric channel), AWGN channel (Additive White Gaussian Noise)

14 Cover and Thomas: Information Theory

14.1 Chapter 1: Introduction and Preview

When you find the time, insert notes here!!

14.2 Chapter 2: Entropy and Mutual Information

14.2.1 Entropy

- Entropy = uncertainty in a random variable
- If X has pmf p(x), $H(X) = -\sum p(x)logp(x) = \mathbb{E}_p[log\frac{1}{p(X)}]$
- This = $H(p)$ when p(x) is for a Bernoulli (because we're focused on bits/log 2)
- $H(X) \geq 0; H_b(X) = (log_b a)H_a(X)$
- INSERT FIGURE 2.1
- Minimum expected number of binary questions to determine X is between H(X) and H(X) + 1

14.2.2 Joint and Conditional Entropy

- $H(X,Y) = -\sum_{x\in X}\sum_{y\in Y} p(x,y)logp(x,y) = -\mathbb{E}[logp(X,Y)]$
- $H(Y|X) = -\sum_{x\in X}\sum_{y\in Y} p(x,y)logp(y|x) = -\mathbb{E}[logp(Y|X)]$
- Chain Rule: $H(X,Y) = H(X) + H(Y|X) \implies H(X,Y|Z) = H(X|Z) + H(Y|X,Z)$
- Note! $H(Y|X) \neq H(X|Y)$; $H(X) - H(X|Y) = H(Y) - H(Y|X)$

14.2.3 Relative Entropy and Mutual Information

- Relative entropy = measure of inefficiency of assuming distribution is q when the true distribution is p (If we know it's p, we use H(p), but if we thought it was q we would need H(p) + $D(p||q)$
- $D(p||q) = \sum_{x\in X} p(x)log\frac{p(x)}{q(x)} = \mathbb{E}_p[log\frac{p(X)}{q(X)}]$
- Not a true distance between distributions because not symmetric and doesn't satisfy triangle inequality, but useful to think of it that way $(D(p||q) \neq D(q||p)$
- Mutual Information

$$\begin{aligned} I(X;Y) &= \sum_{x\in X}\sum_{y\in Y} p(x,y)log\frac{p(x,y)}{p(x)p(y)} \\ &= D(p(x,y)||p(x)p(y)) \\ &= \mathbb{E}_{p(x,y)}[log\frac{p(X,Y)}{p(X)p(Y)}] \end{aligned}$$

-

14.2.4 Relationship between Entropy and Mutual Information

-

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) \\ &= H(Y) - H(Y|X) \\ &= H(X) + H(Y) - H(X,Y) \\ &= I(Y;X) \end{aligned}$$

- $I(X;X) = H(X)$
- INSERT FIGURE 2.2

14.2.5 Chain Rules

15 2014-07-14 2014-07-21 Kalman Filter Learning Crand

*****Abbreviated Version*****
Note! Control U(n) = 011111$\cdots$
Problem Setup:

(1) $X(n+1) = C_r(n+1) = C_r(n)$

(2) $Y(n) = C_r(n) + (-aV(n-1) + V(n))$

(3)

(4) $X(0) \sim N(\mu, \sigma^2)$

(5) $V(n) \sim N(0, \sigma_v^2)$

Goal: $\hat{X}(n) = \mathbb{E}[X(n)|Y^n]$

(6) $\mathbb{E}[X(n)|Y^{n-1}] = \hat{X}(n-1)$

(7) $\mathbb{E}[Y(n)|Y^{n-1}] = \mathbb{E}[X(n) - aV(n-1) + V(n)|Y^{n-1}]$

(8) $= \mathbb{E}[X(n)|Y^{n-1}]$

(9) $= \mathbb{E}[X(n-1)|Y^{n-1}] = \hat{X}(n-1)$

Kalman Filter $= \frac{cov(X(n), Y(n) - \hat{X}(n-1))}{cov(Y(n) - \hat{X}(n-1))}$
Numerator:

(10)
$cov(X(n), Y(n) - \hat{X}(n-1))$

(11)
$= cov(X(n) - \hat{X}(n-1), Y(n) - \hat{X}(n-1))$

(12)
$= cov(X(n) - \hat{X}(n-1), X(n) - aV(n-1) + V(n) - \hat{X}(n-1))$

(13)
$= cov(X(n) - \hat{X}(n-1), X(n) - \hat{X}(n-1))$

(14)
$= cov(X(n) - \hat{X}(n-1))$

(15)
$= cov(X(n-1) - \hat{X}(n-1)) = S_{n-1}$

Denominator:

(16)
$cov(Y(n) - \hat{X}(n-1))$

(17)
$= cov(X(n) - aV(n-1) + V(n) - \hat{X}(n-1))$

(18)
$= cov(X(n) - \hat{X}(n-1)) + cov(-aV(n-1)) + cov(V(n))$

(19)
$= cov(X(n-1) - \hat{X}(n-1)) + a^2\sigma_v^2 + \sigma_v^2$

(20)
$= S_{n-1} + (a^2+1)\sigma_v^2$

$$K_f = \boxed{\frac{S_{n-1}}{S_{n-1} + (a^2+1)\sigma_v^2}}$$

Estimate $= \boxed{\hat{X}(n) = \hat{X}(n-1) + K_f(Y(n) - \hat{X}(n-1))}$

(21)
$S_n = cov(X(n) - \hat{X}(n))$

(22)
$= cov(X(n) - \hat{X}(n-1) - K_f\tilde{Y}(n))$

(23)
$= \mathbb{E}[(X(n) - \hat{X}(n-1))^2 - 2(X(n) - \hat{X}(n-1))(K_f\tilde{Y}) + K_f^2\tilde{Y}^2(n$

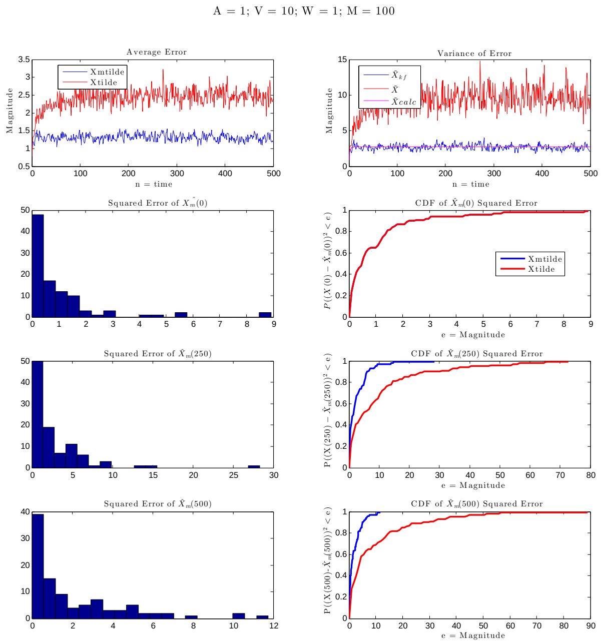
(24)
$= S_{n-1} - 2K_fS_{n-1} + K_fS_{n-1}$

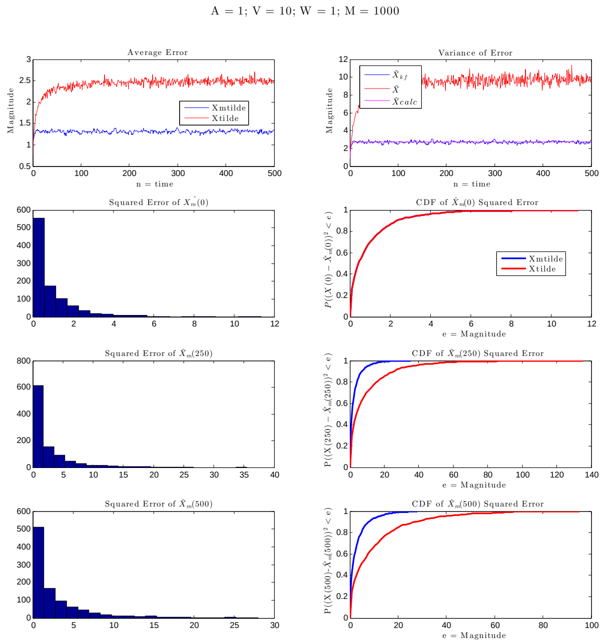
(25)
$= (1 - K_{f n-1})S_{n-1}$

When comparing U(n) = 0111... vs. U(n) = 0101..., they both can achieve the same asymptotic error. This is especially due to C_{rand} not increasing or decreasing, so time doesn't affect the state. The only difference is that the former can achieve asymptotic error (which converges to 0) at double the rate as the latter, since it can collect an observation at each timestep while the second control can only collect observations every two timesteps.

16 Results of Simulations

16.1 Kalman Filter Additive Noise





These two figures represent a Kalman Filter with Additive Noise estimating a system with only additive noise. Xtilde represents a memoryless estimate, while Xmtilde represents the Kalman Filter's estimate. Xcalc represents the theoretical variance, which matches the empirical variance and helps check the plots for correctness.

- The estimation error variance is bounded

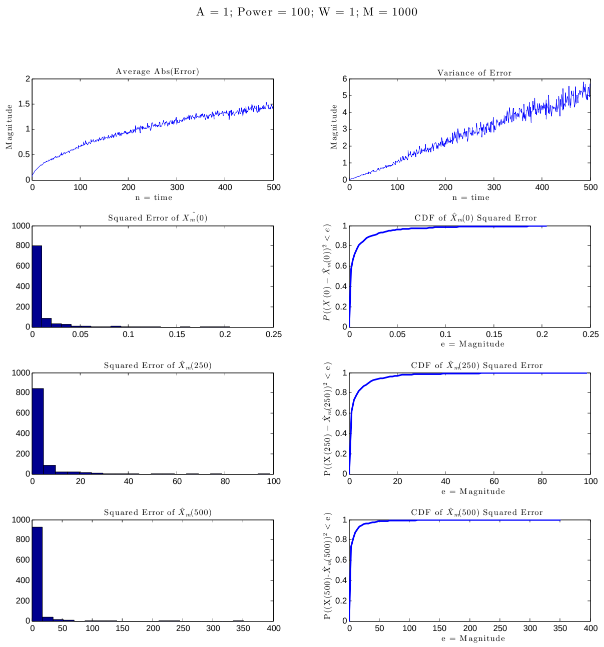
- The Kalman Filter performs better than the optimal memoryless estimator; over many trials it's clearer the error variance is lower

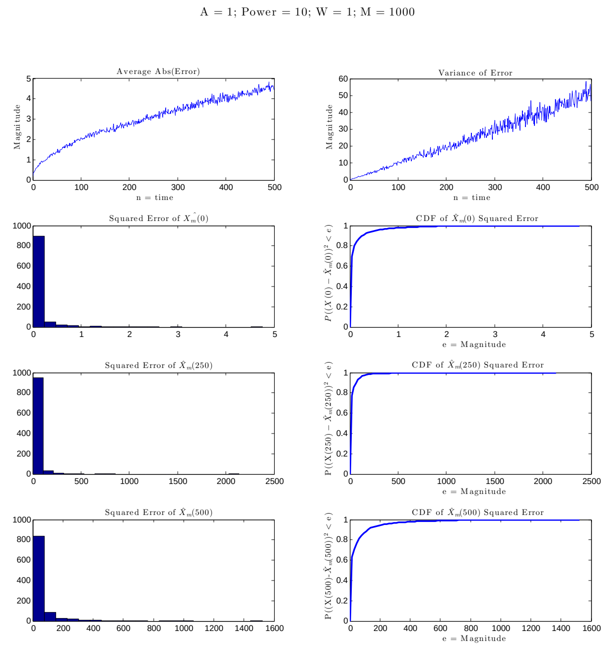
- If $0 < A < 1$, $Xmtilde \rightarrow Xtilde$. We showed earlier that the Kalman Filter is only intended for an estimation system, not a control system, and as the value of the state converges to 0 the estimation system behaves like the control problem. Since the value of the state is so close to 0 at each timestep, memory provides no additional benefit/utility to estimation.

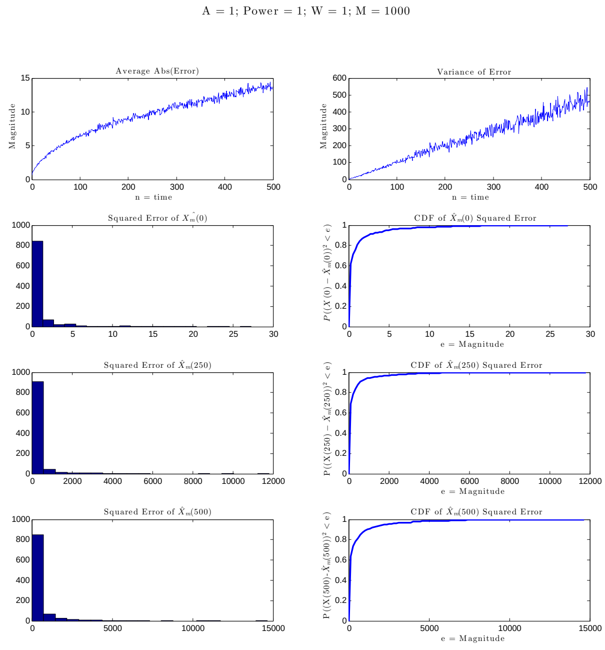
- The CDF of estimation error is affected by C, V, and W

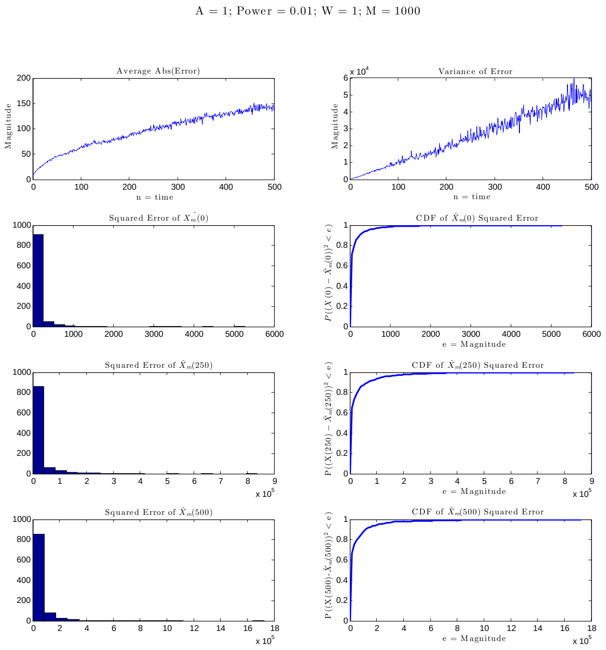
Note: CDF of $\hat{X}_m(n)$ Squared Error means the plots are of the estimation error.

16.2 Kalman Filter Ignoring Multiplicative Noise





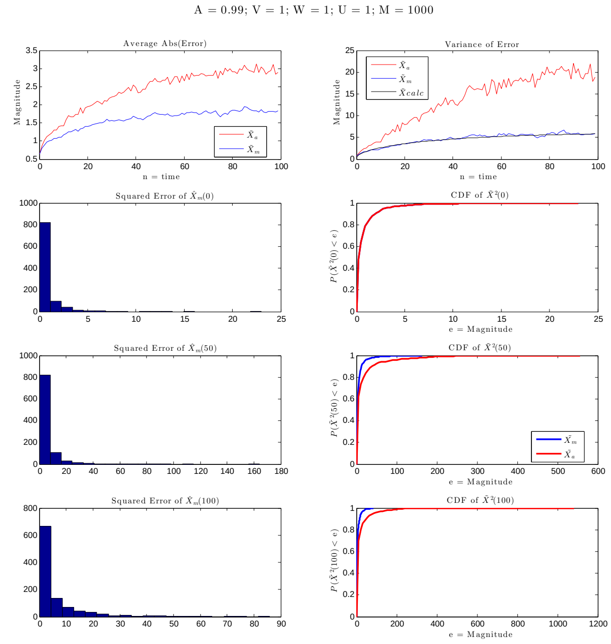


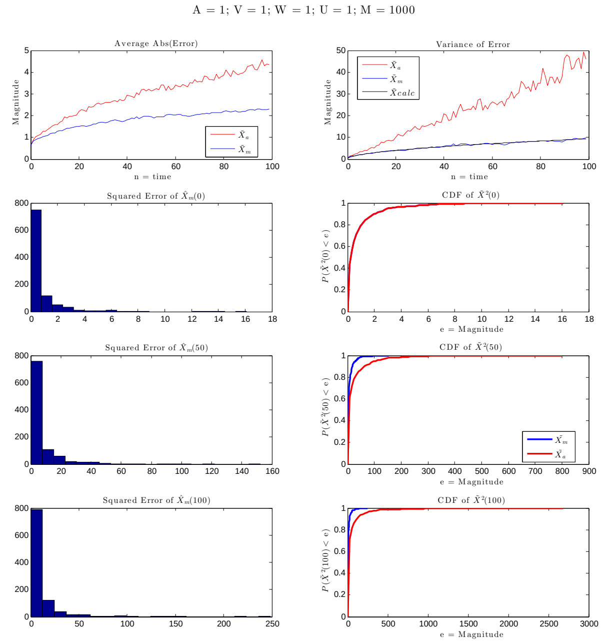


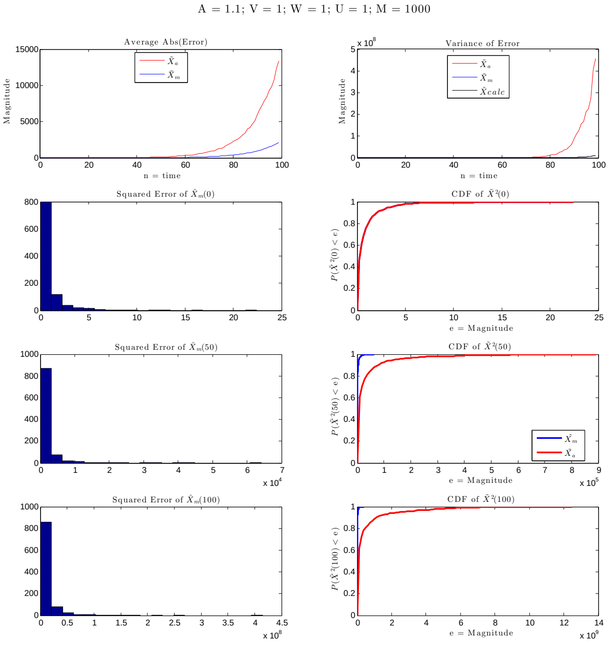
These figures represent a Kalman Filter for Additive Noise with a system that has state multiplicative noise and additive noise but no observation noise. The estimation error variance steadily increases because the multiplicative noise causes the signal to increase, thus increasing the error from the multiplicative noise. In this setup, the multiplicative error was $(1 + r(n)) = (1 + normrnd(0, varv))$. The four figures represent ascending levels of mul noise (or $\frac{1}{p}$), with the estimation error variance increasing as V increases.

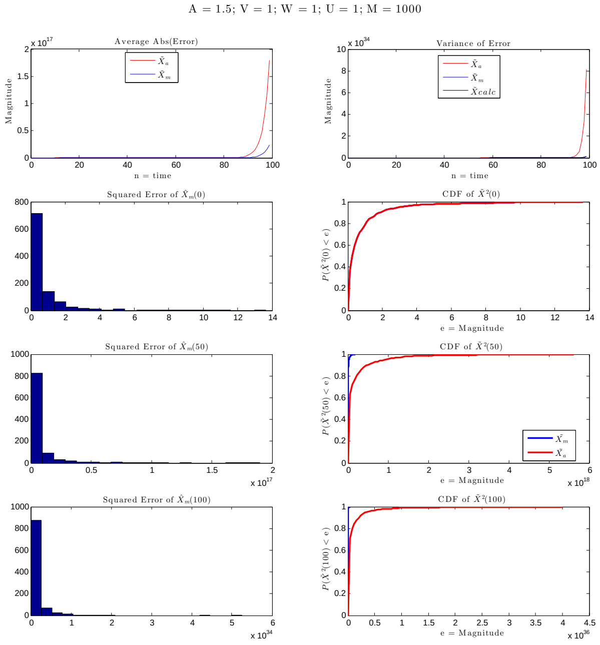
Note: CDF of $\hat{X}_m(n)$ Squared Error means the plots are of the estimation error.

16.3 Kalman Filter for Multiplicative Noise





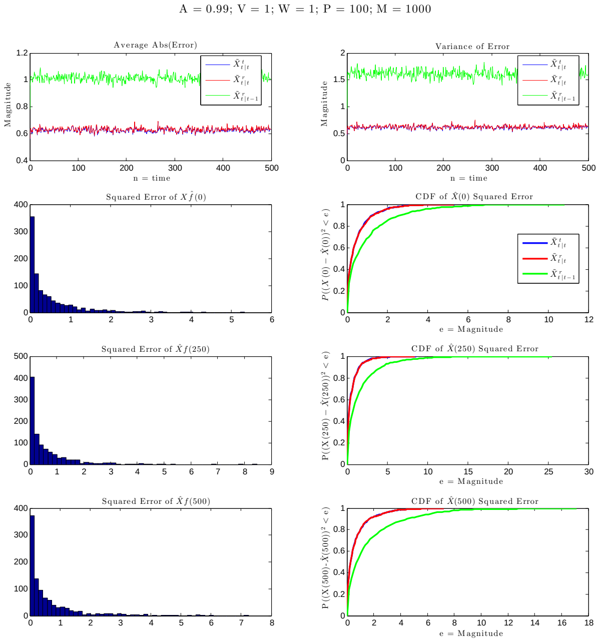


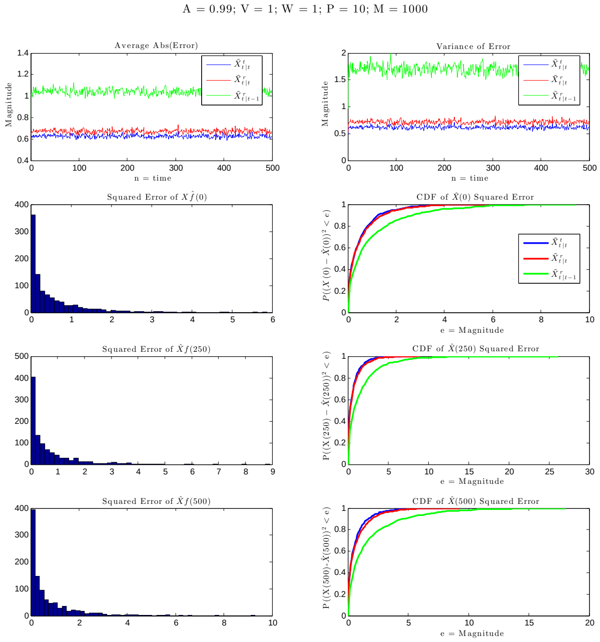


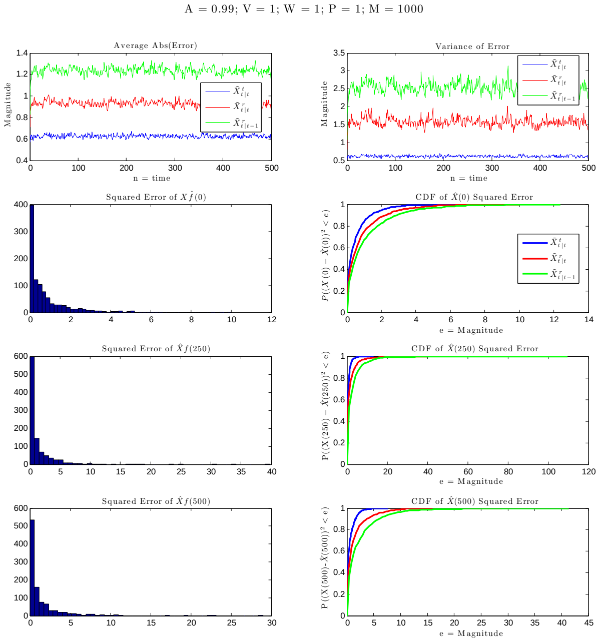
This is the implementation of a Kalman Filter for Multiplicative Noise (Rajasekaran) with a system that has both multiplicative and additive noise. $\tilde{X}_a$ represents the estimation error for a Kalman Filter for only additive noise, while $\tilde{X}_m$ represents the new Kalman Filter. $\tilde{X}calc$ represents the theoretical variance, as a check that it matches the empirical variance. It is apparent the new Kalman Filter is doing much better; both have bounded error variances, but $\tilde{X}_m$ is significantly lower.

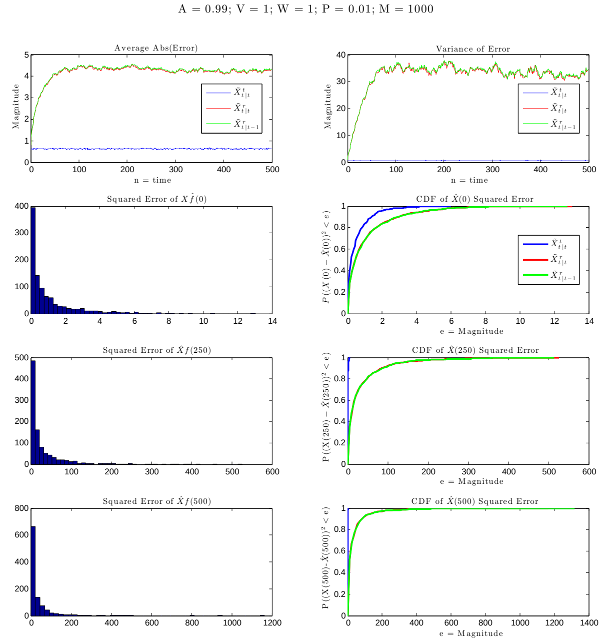
In the upper left corner, A = 0.99. Since A < 1, the state converges toward 0 and the estimation error variance for X_a is bounded. That being said, Var($\tilde{X}_m$) is still better (bounded at a lower value) than Var($\tilde{X}_a$). In the upper right, A = 1. Var($\tilde{X}_a$) matches the behavior we would expect, which is increasing linearly similar to the previous ”Ignoring Multiplicative Noise with A = 1” plots above. Var($\tilde{X}_m$) is bounded, again as we would expect since we're using a Kalman Filter intended for multiplicative noise. In the lower left, A = 1.1. In this case, you can see the estimation error variance explodes, with Var($\tilde{X}_a$) increasing up to $10^{8}2$. Var($\tilde{X}_m$), on the other hand, remains at its low bounded value. In the lower right, A = 1.5. The system is growing too fast over n = 100 timesteps, and Matlab can't handle the numbers because they're getting too large. Both estimation error variances appear large because of value truncation/roundoff error done in Matlab. This simply means values $A > 1.5$ aren't testable in Matlab, even if Rajasekaran's Kalman Filter applies.

16.4 Quantization Noise: Schenato



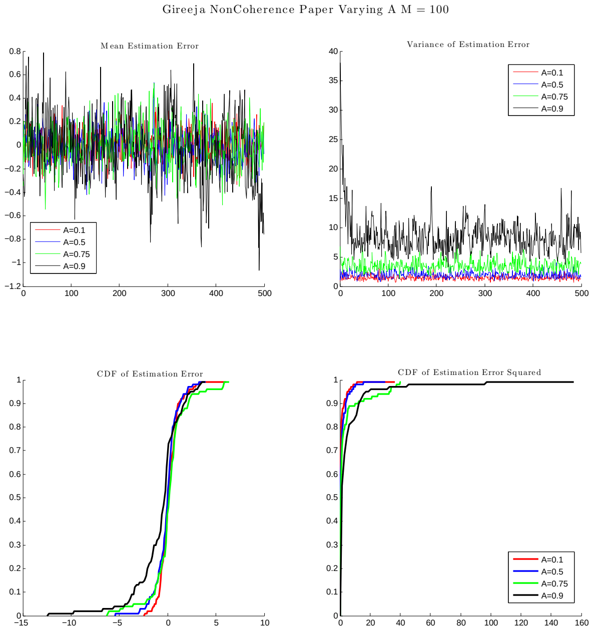


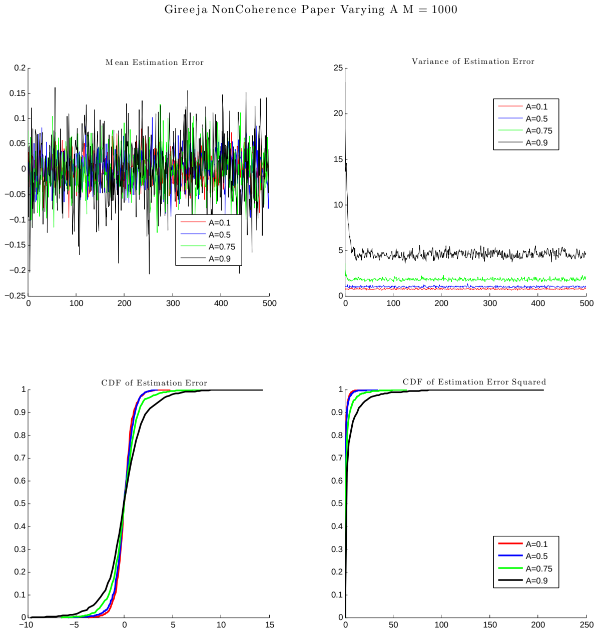


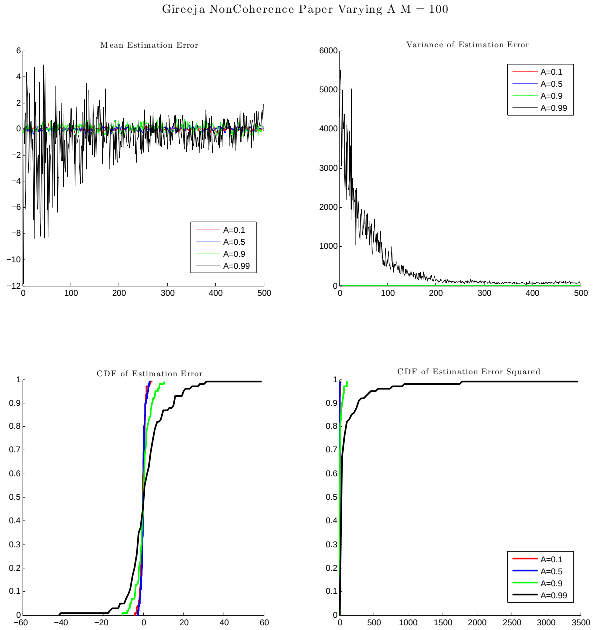


These figures are the implementation of Schenato's paper, with multiplicative noise but no packet drops. Multiplicative (quantization) noise happens over the channel between the transmitter and the receiver. $\tilde{X}^t_{t|t}$ represents the transmitter's estimate of the state, using a Kalman Filter for additive noise (because so far there has only been additive noise.) $\tilde{X}^r_{t|t-1}$ represents a prediction of the state post-quantization noise, and $\tilde{X}^r_{t|t}$ represents the estimation of the state post-quantization noise. When the power is extremely small (thus the multiplicative noise has large variance), $\tilde{X}^r_{t|t-1}$ does the same as $\tilde{X}^r_{t|t}$. This is because the multiplicative noise has such a huge effect, the Kalman Filter minimizes the observation to the point that it's not being used, and the estimation error is the same as the prediction error/no information error.

16.5 NonCoherence Estimation: Gireeja

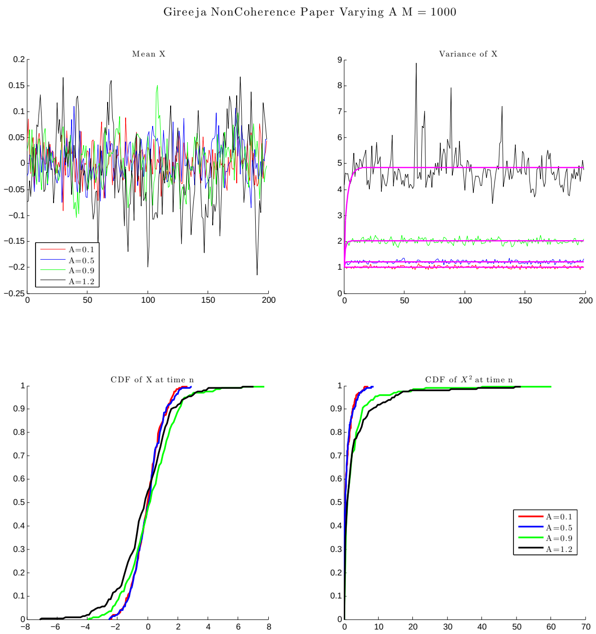


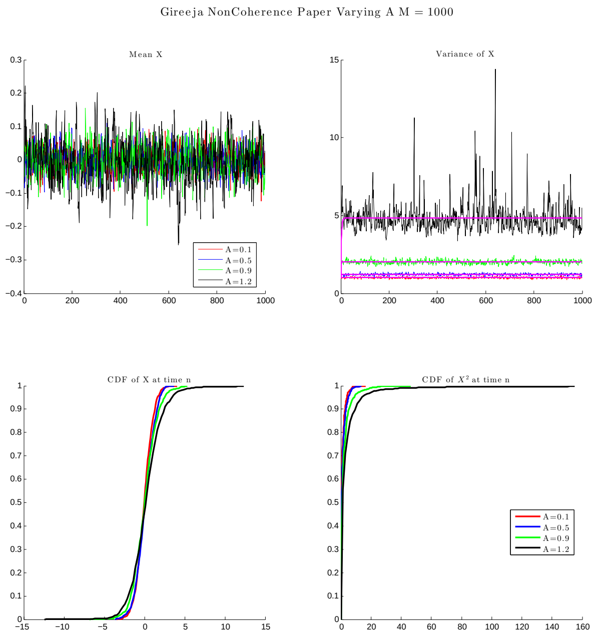




These figures are the implementation of Gireeja's Non-Coherence paper for estimation, not control. The figures above show that A affects the final bounded value of the estimation error variance. The figure to the left shows why A must be < 1; when A is close to 1 it takes longer to converge towards its asymptotic estimation error variance, and when A ≥ 1 the system will explode. After looking back at old code, I think the multiplicative noise was normrnd(1,1).

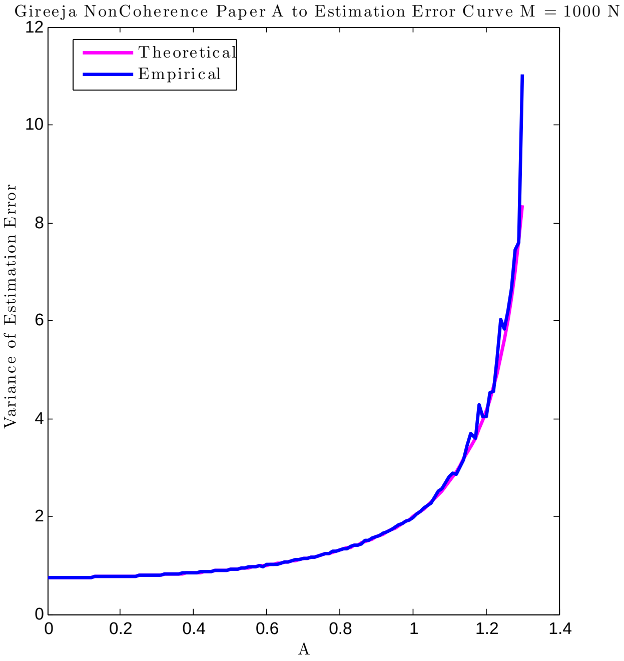
16.6 NonCoherence Control: Gireeja





This figure is Gireeja's NonCoherence Paper Control System, and it shows the system is stabilized for varying levels of $A < 1.4$. Magenta represents the theoretical calculation for Var(X), and this figure reflects that that calculation is accurate. Over time, Var(X) quickly converges to $Var(X)^{\infty}$. In addition, the greater the A the higher the state variance, and this can be calculated.

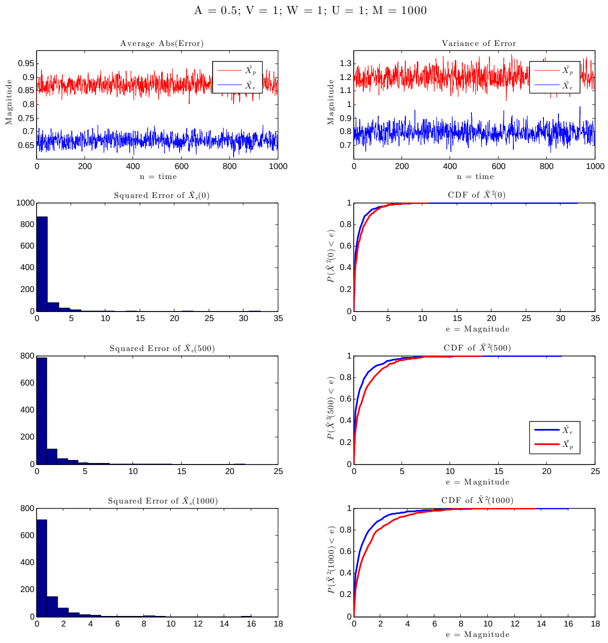
16.7 Varying A, NonCoherence Control, Gireeja

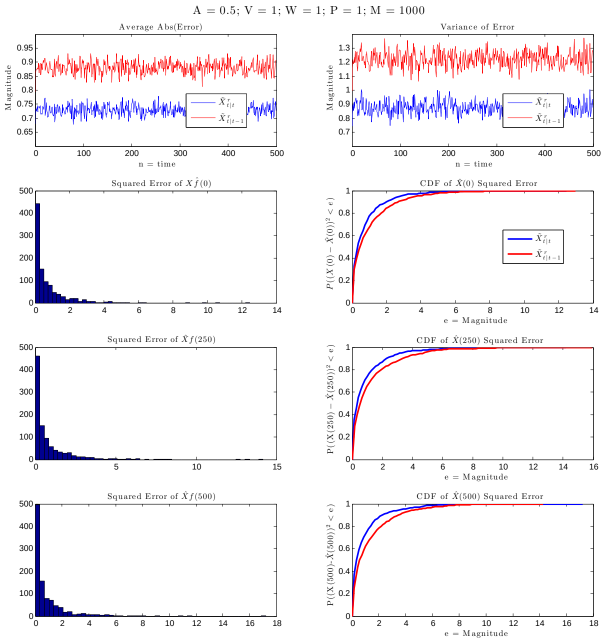


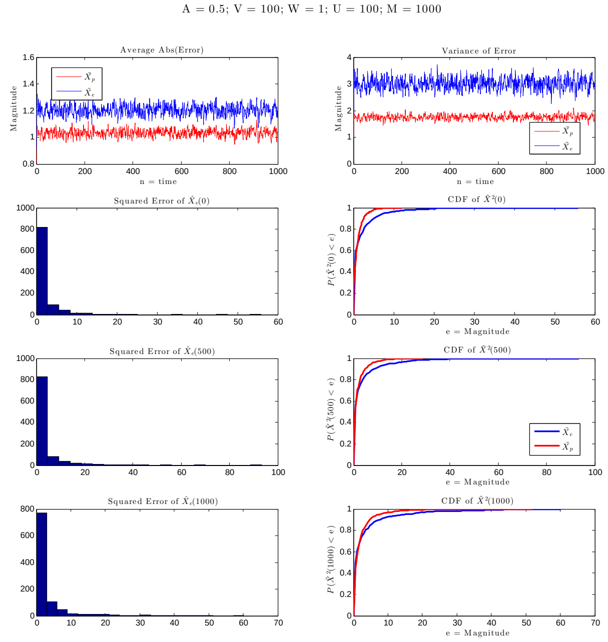
THIS NEEDS TO BE UPDATED!! In this figure, blue represents the asymptotic empirical variance for levels of A 0:0.01:1.3, while magenta represents the asymptotic theoretical variance. These values are mean(mean()), or averaged over both trials and timesteps. The curve appears roughly exponential, or at the very least when $A > 1$ the curve rises sharply. Based on Gireeja's paper, for these values the curve should asymptote at $\sqrt{2} = 1.414$, which is reflected in the curve. Since this curve was generated over n = 250 timesteps and M = 1000 trials, the spikes in the blue curve should *not* be attributed to randomness. Multiple runs could confirm whether those spikes are significant or not.

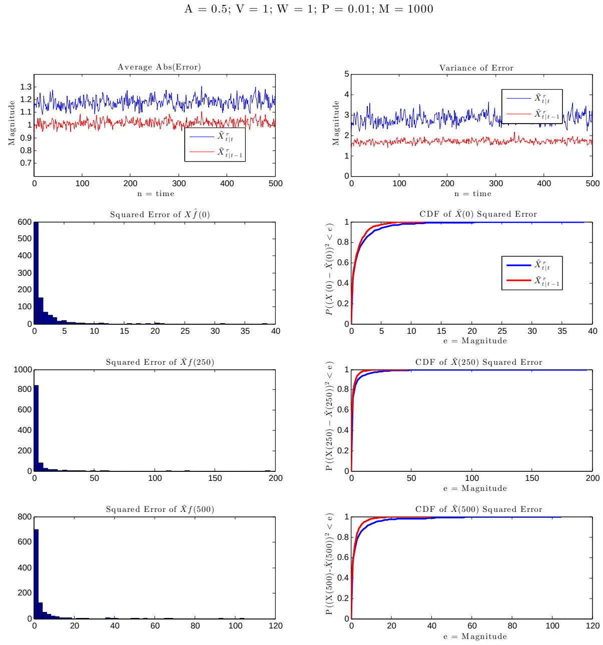
16.8 Rajasekaran vs Schenato: Prediction vs Estimation

*****NOTE: THIS SECTION IS NOW OUTDATED!! I'M KEEPING IT FOR RECORD PURPOSES*****





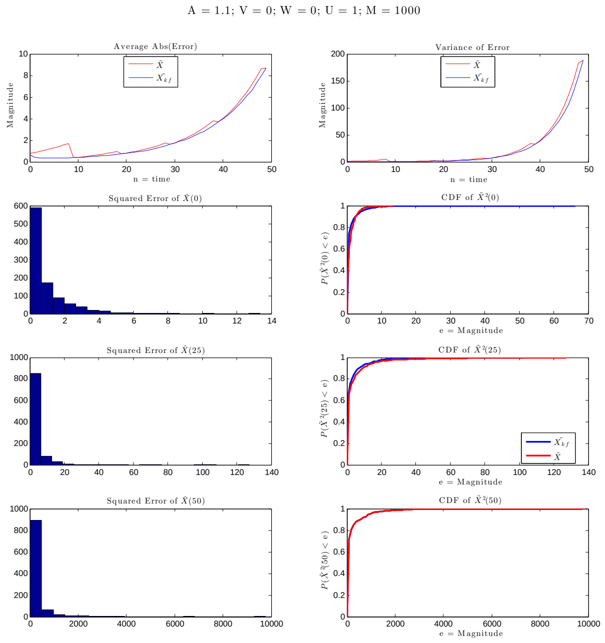


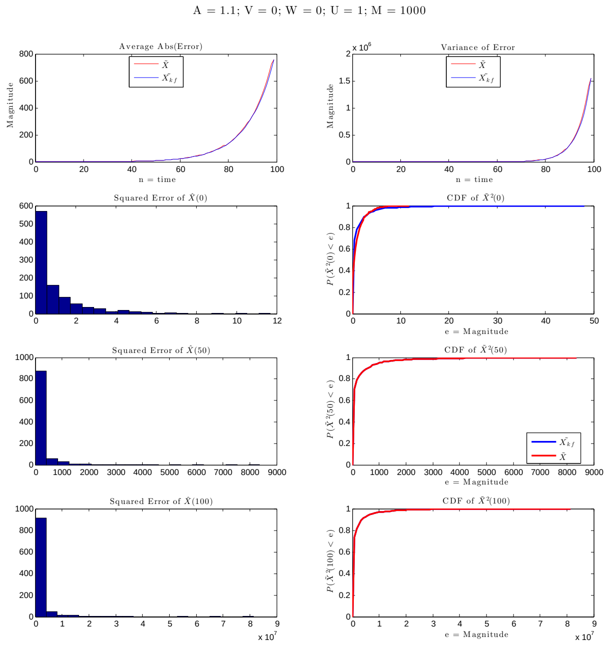


Left = Rajasekaran's Kalman Filter, Right = Schenato's Receiver Estimators.
In all four figures, red represents prediction and blue represents estimation.

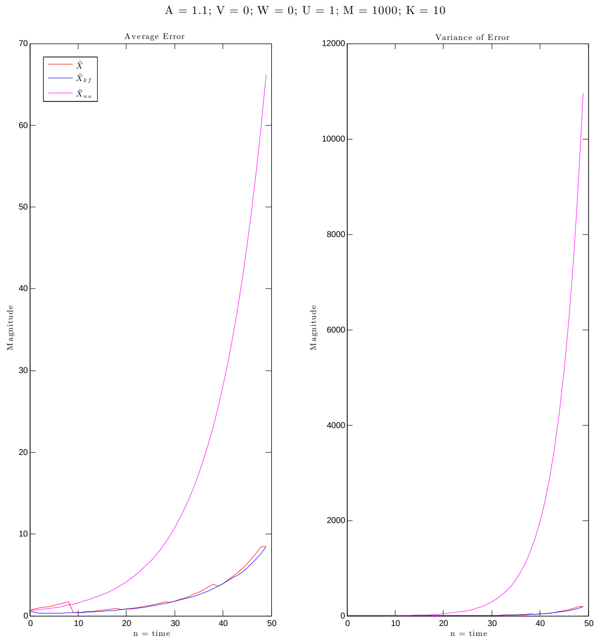
16.9 Delay Estimation: Quant Noise, No Drops

*****Note: k = 10 for all these plots!*****



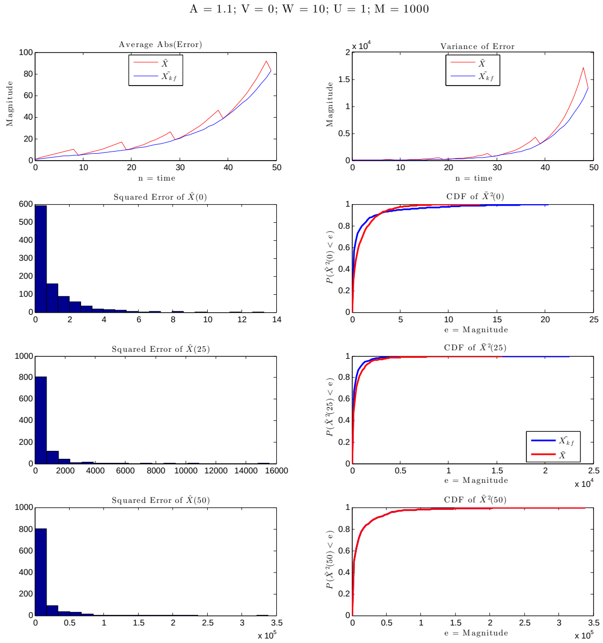


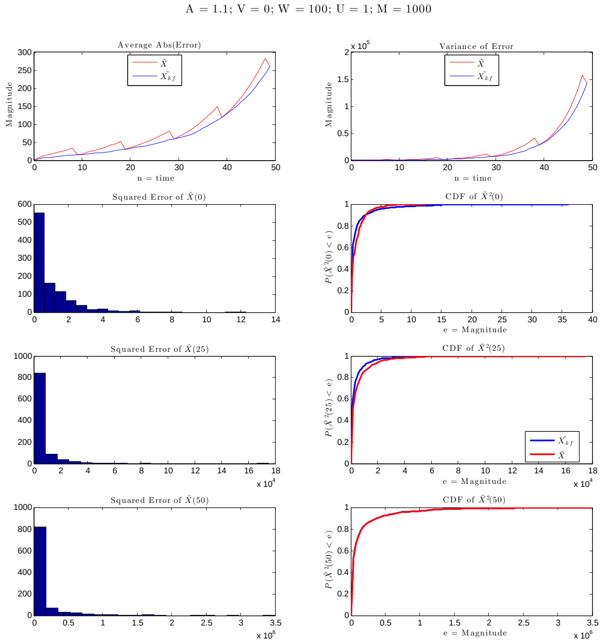
These plots are the basic delay problem with no state noise, channel additive noise or packet drops. There is, however, quantization noise. In this case, k = 10. The error behaves as we would expect, with a rough sawtooth shape hugging the kalman filter curve. Blue represents the error with no delay, and red represents with delay. The error is exponential because A = 1.1, so the error is unbounded. In the CDF of $\tilde{X}^2(25)$, "delay" does worse than "no delay" because 25 is between 20 and 30, so the error of the delay version has grown relative to "no delay". In the CDF of $\tilde{X}^2(50)$ there are two reasons the CDF is the same: 1) 50 mod 10 == 0 and 2) the error converges as $n \to \infty$ because it's exponential (the impact of pulling the delay error at time k matters less and less.)



In this figure, I added error from an estimate with absolutely no information. In this case, the first estimate of $\tilde{X}_{no}$ is the same as the first Kalman Filter estimate, but after that the estimate = a*previous_estimate. Although both are exponential and unbounded, the error with no information grows much faster than error with kalman filter.

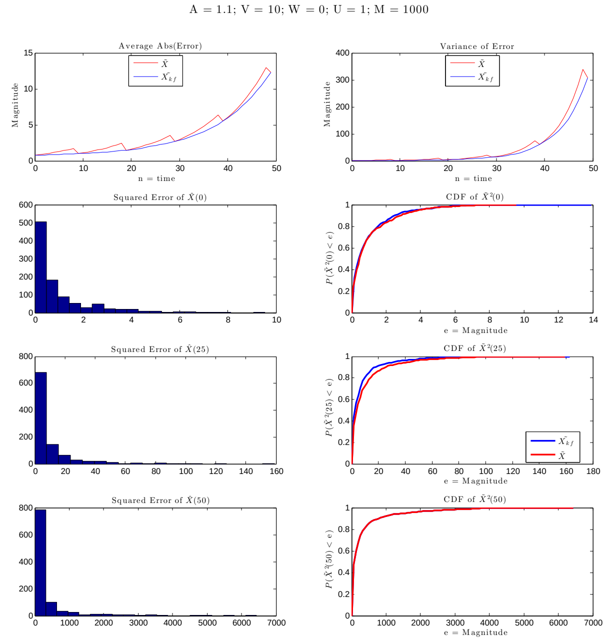
16.9.1 State Noise

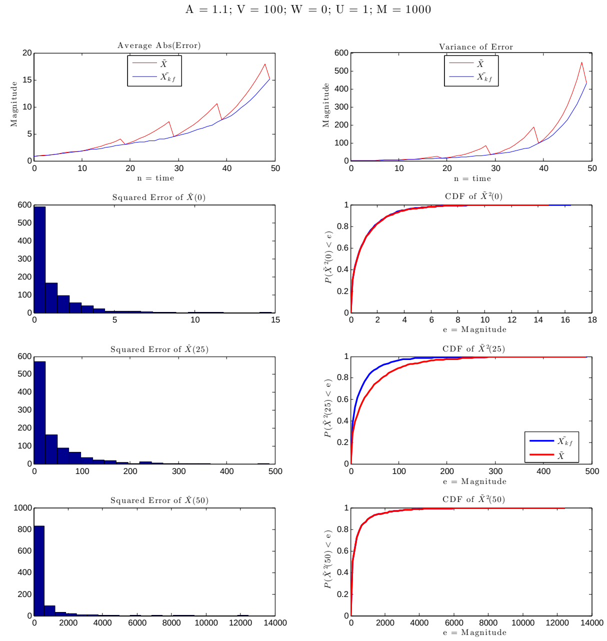




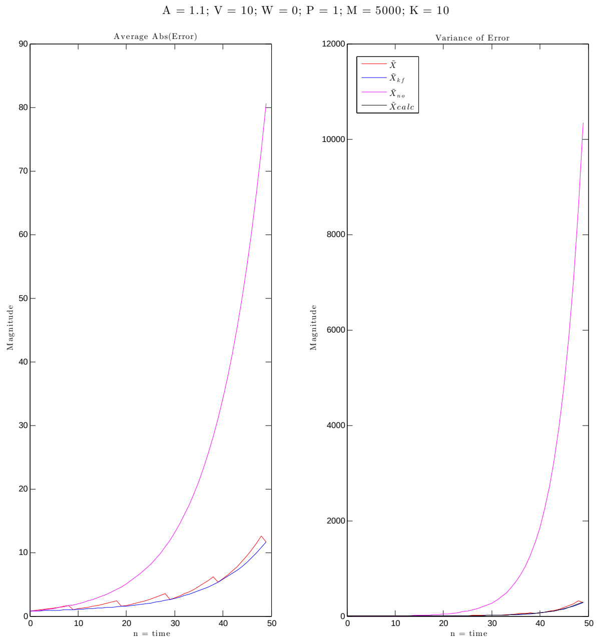
In this figure, I added state noise. I expect the "delay error" to do worse than before, because now the "feedback"/information from the observations matter even more.

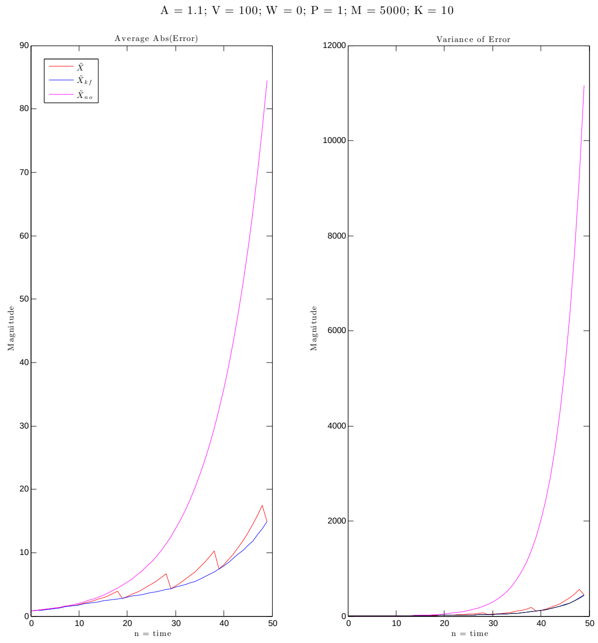
16.9.2 Channel Additive Noise



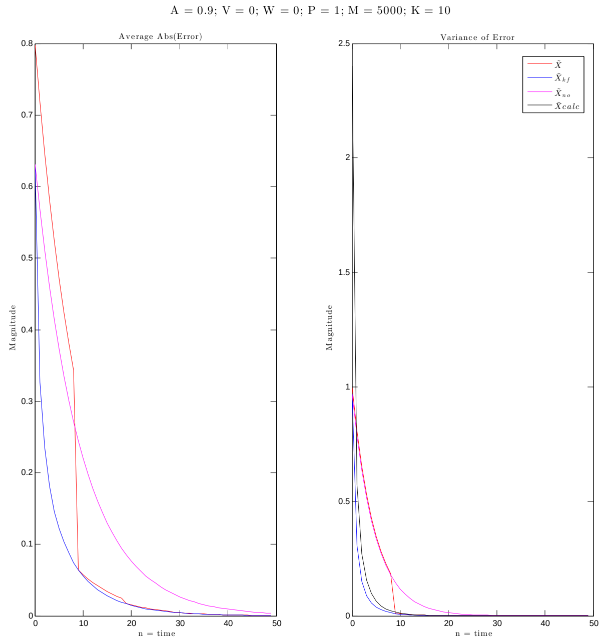


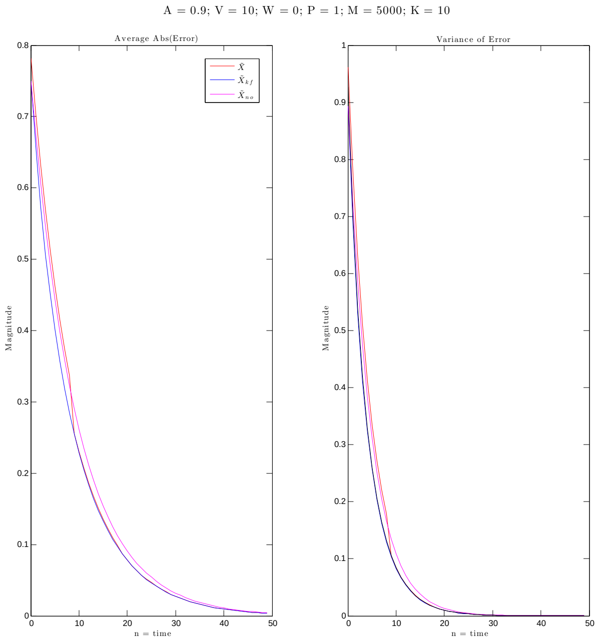
In this figure, there is no state noise but there is additive channel noise. **WRITEUP PLEASE.**

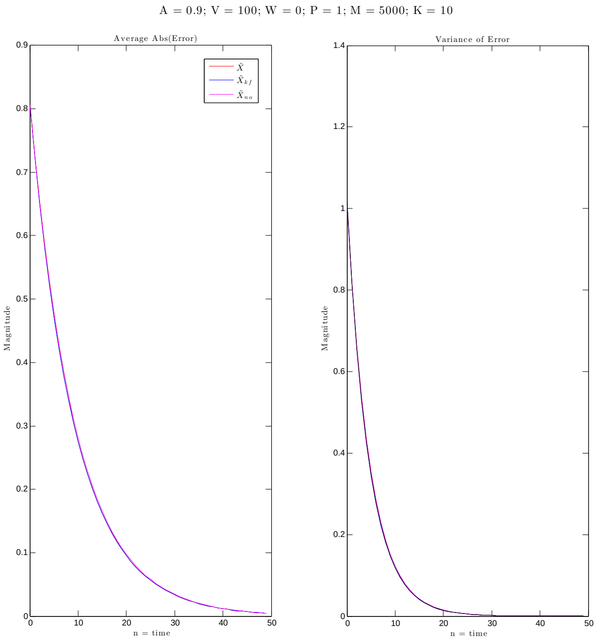




This figure compares "delay error", "no delay error", and "no information error".

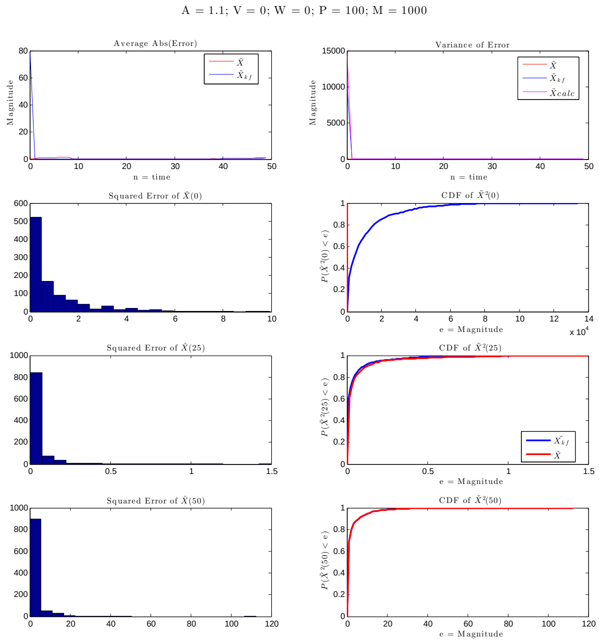


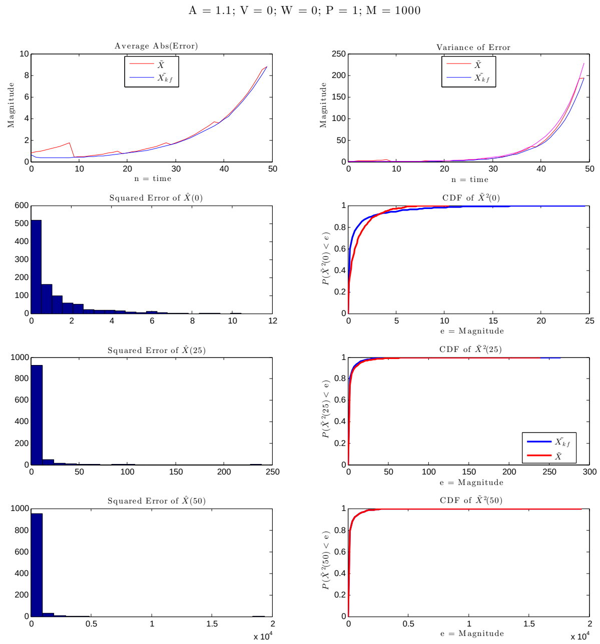


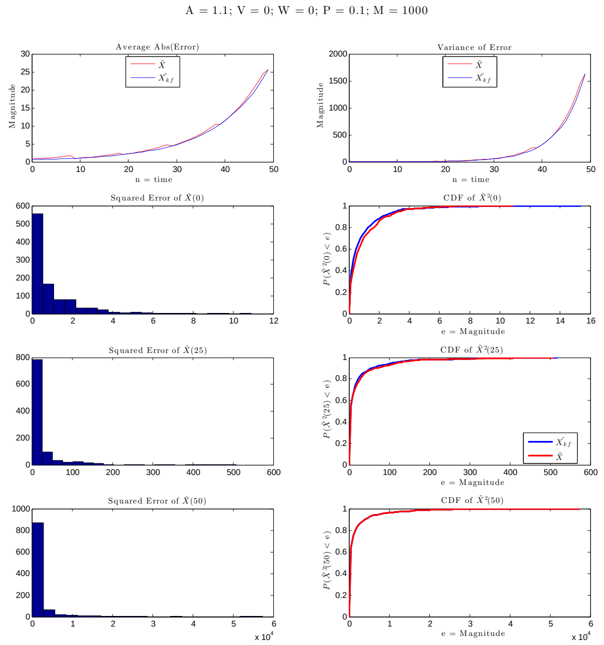


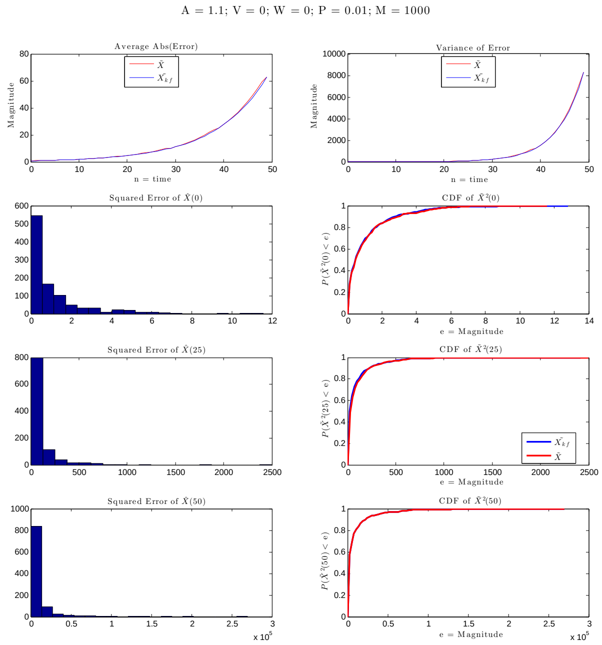
These figures are the same as the ones above except with A = 0.9 instead of A = 1.1. Note that since $A < 1$ both state and error converge to 0. With large amounts of channel noise, the observation is essentially useless so everything performs the same.

16.9.3 Multiplicative Noise

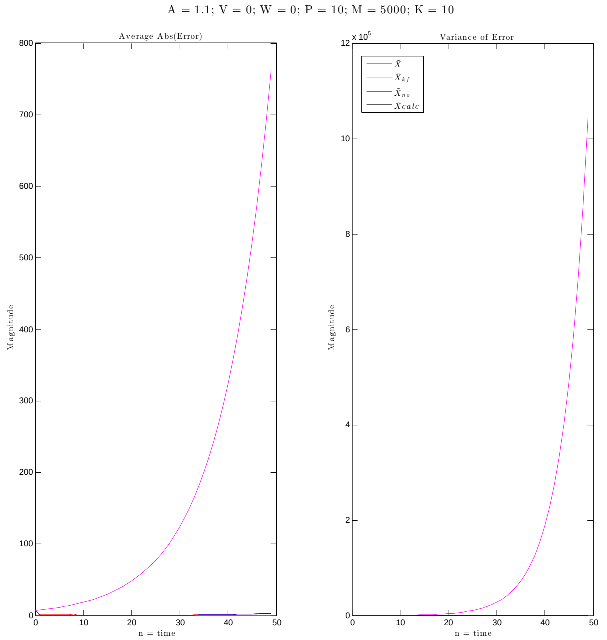


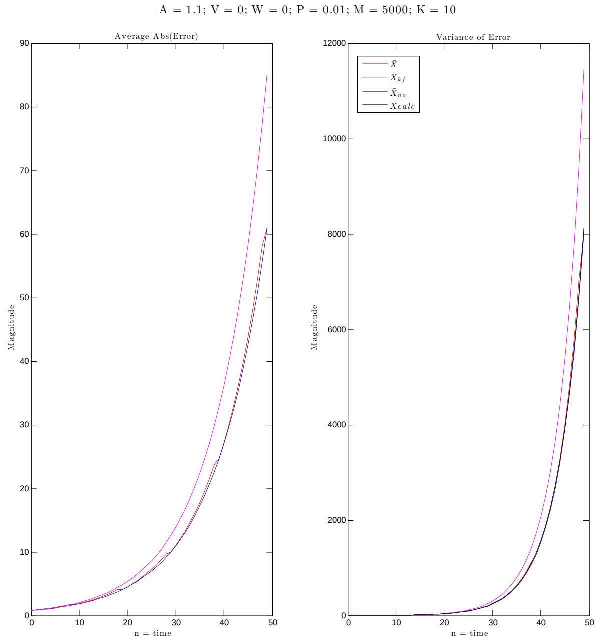






These figures show increasing amounts of multiplicative noise.

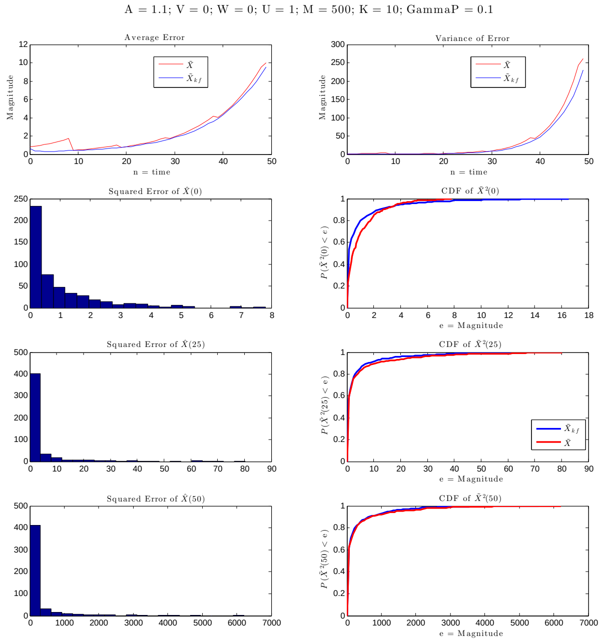


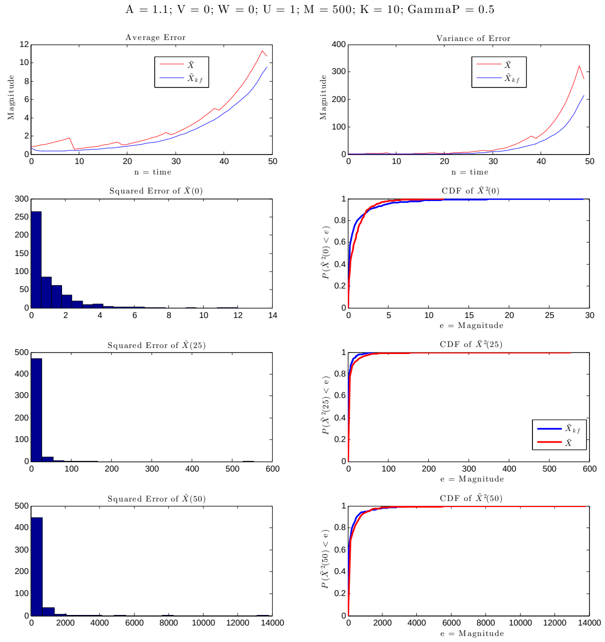


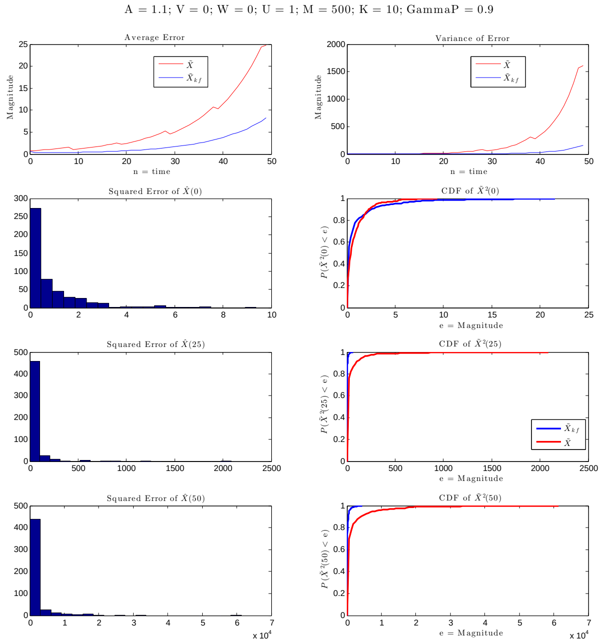
In the left diagram the power is higher, so we expect less multiplicative noise.

16.10 Delay Estimation: Quant Noise + Packet Drops

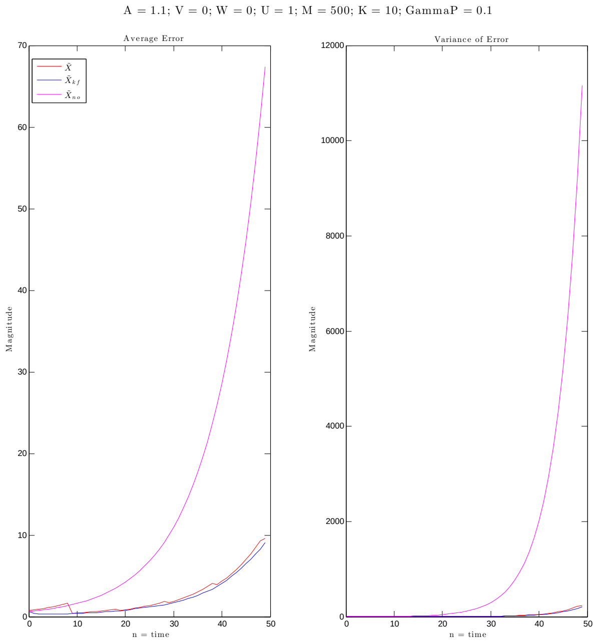
Note: K = 10 for these plots.

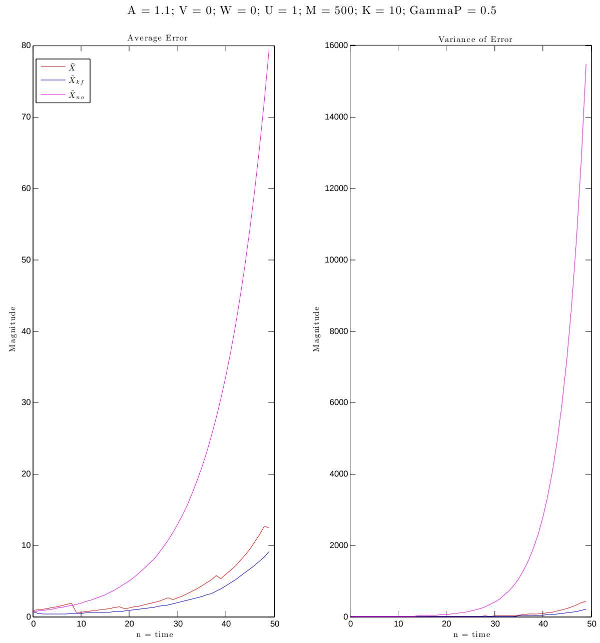


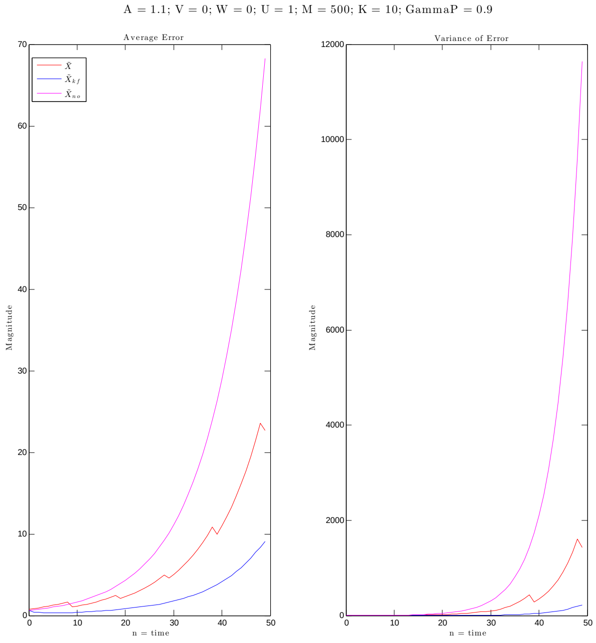


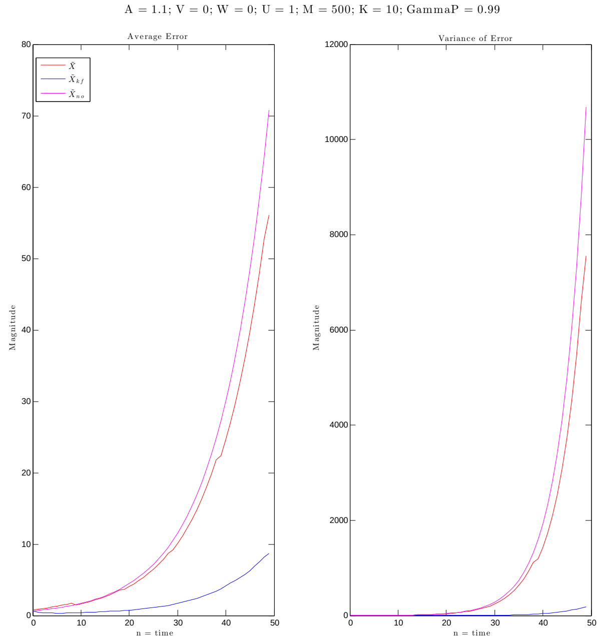


These figures show increasing probabilities of packet drop compared to an ideal kalman filter with no packet drops and no delay. When there are more drops, the ”delay+drop error” can’t be pulled down as much to the ”no delay error”, and since it’s increasing slightly faster as time passes the two errors diverge.



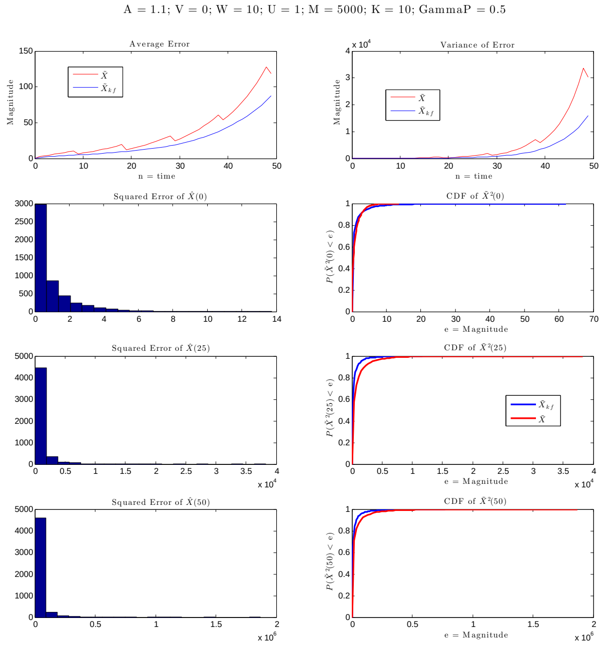


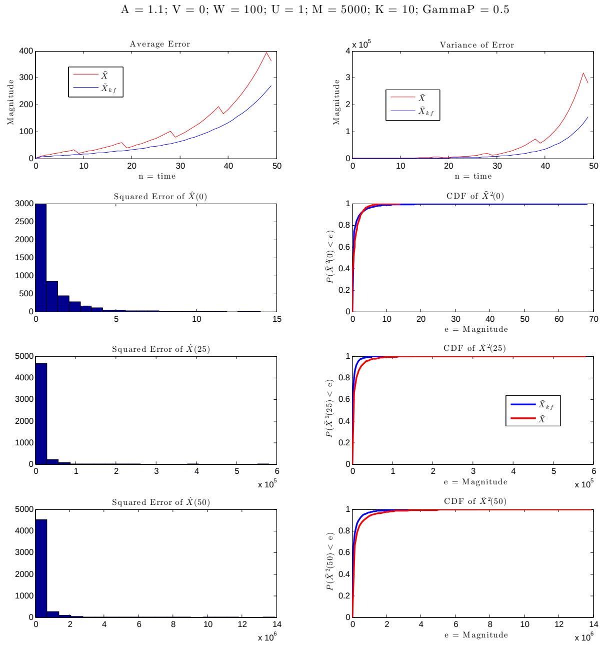




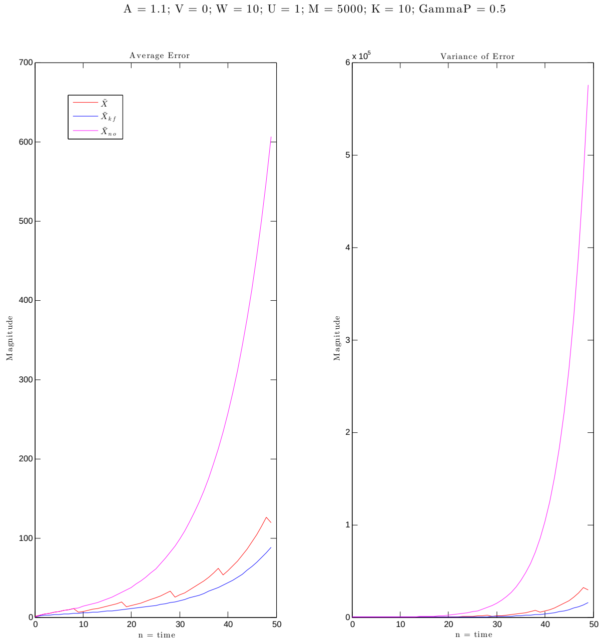
In these figures you can see that as the probability of packet drop increases, the "delay+drop error" approaches "no information error" since they are functionally behaving the same.

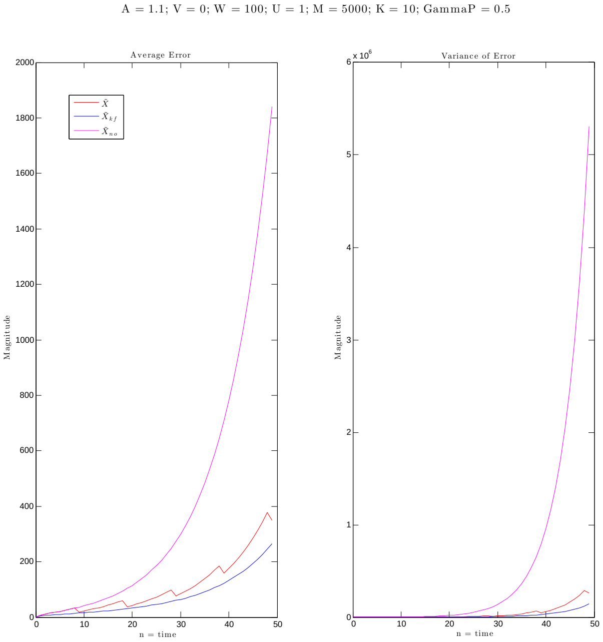
16.10.1 State Noise



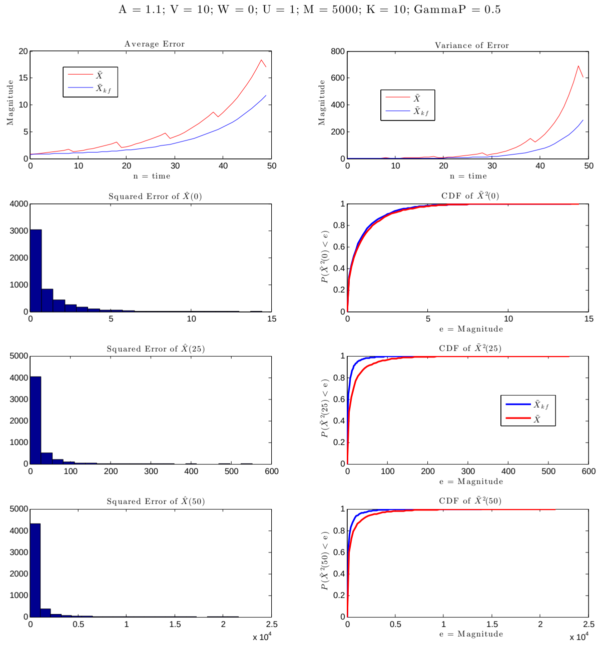


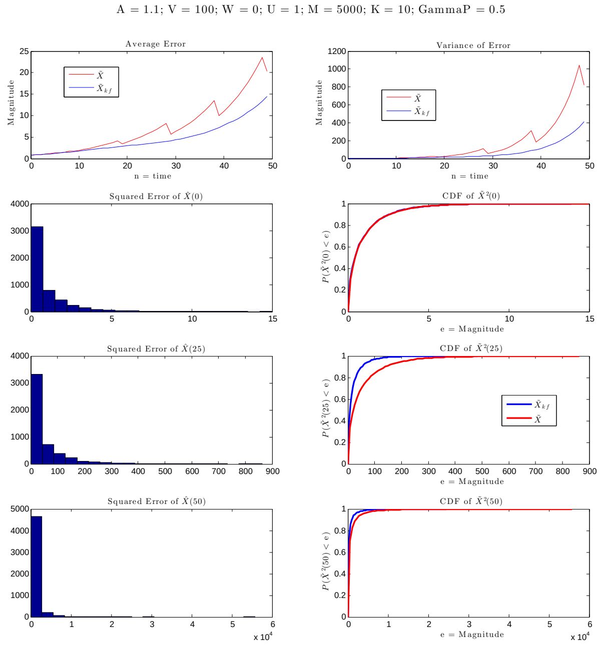
These figures compare a system with both delay and packet drops versus an ideal multiplicative noise Kalman Filter. In these figures I added state noise, but there is no channel noise and Var(mul noise) = 1. Although the magnitude of the error increases when Var(W) increases, there doesn't appear to be a difference in the relationship between the two curves.



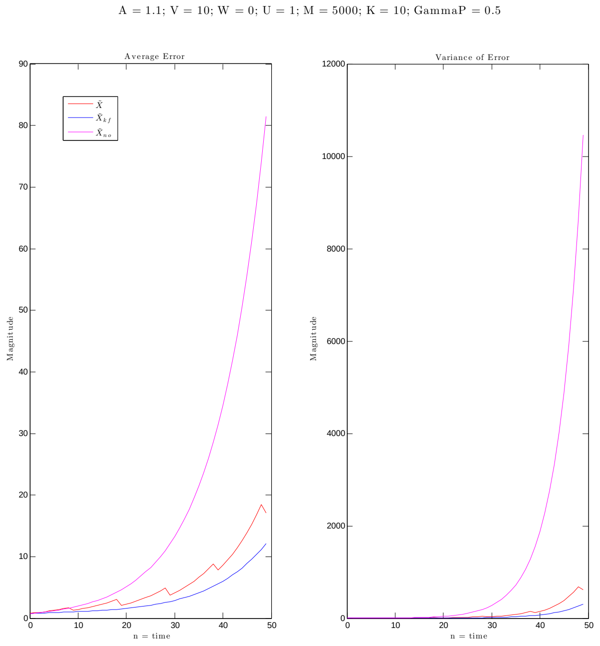


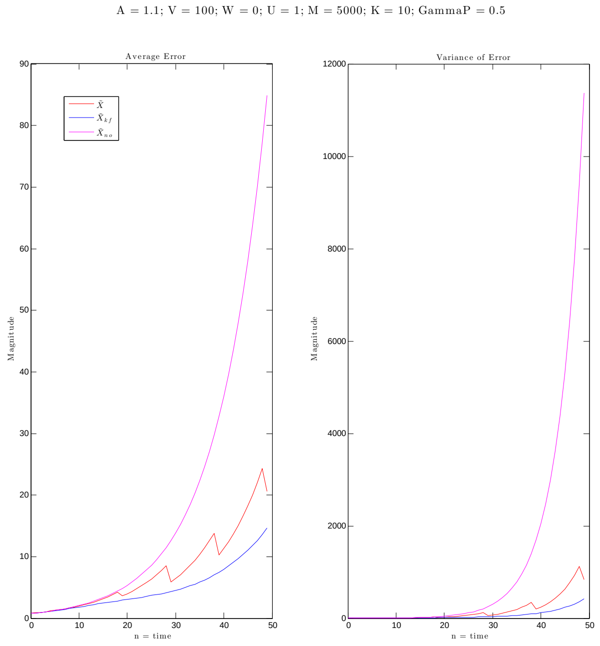
16.10.2 Channel Additive Noise



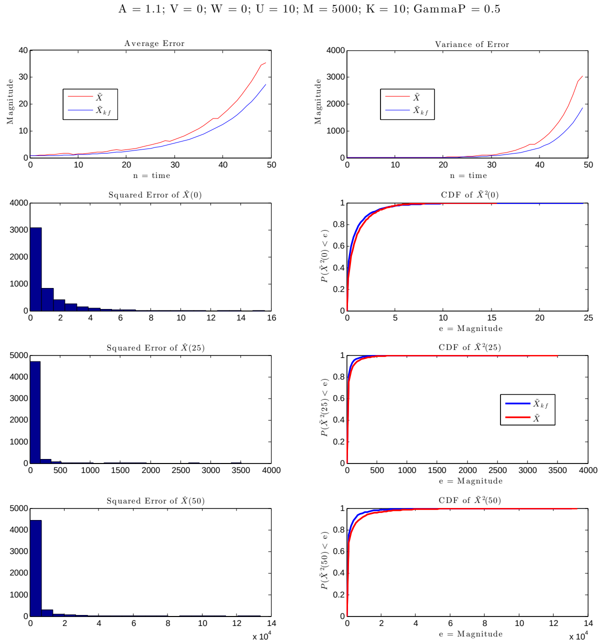


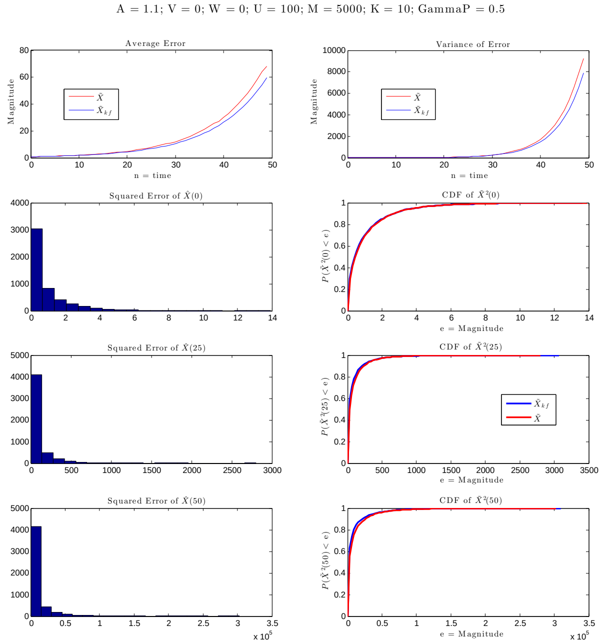
These figures have channel noise. The magnitude of the average error/variance of error doesn't increase significantly, but the red line draws further apart from the blue line, indicating delay/drops is creating a larger difference.



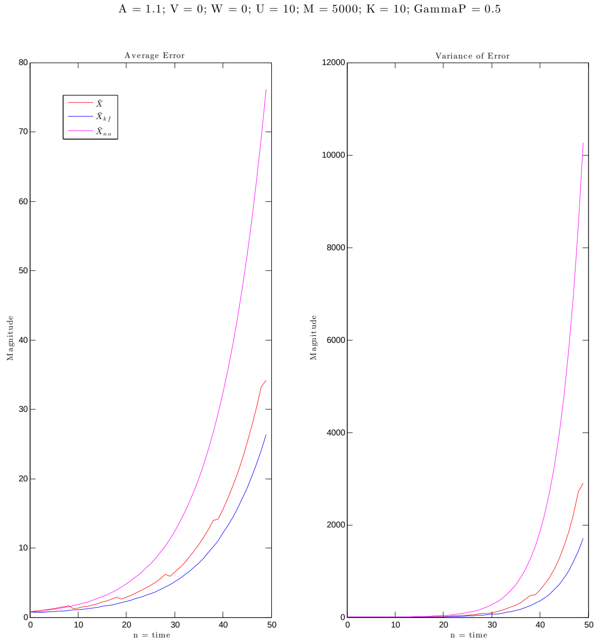


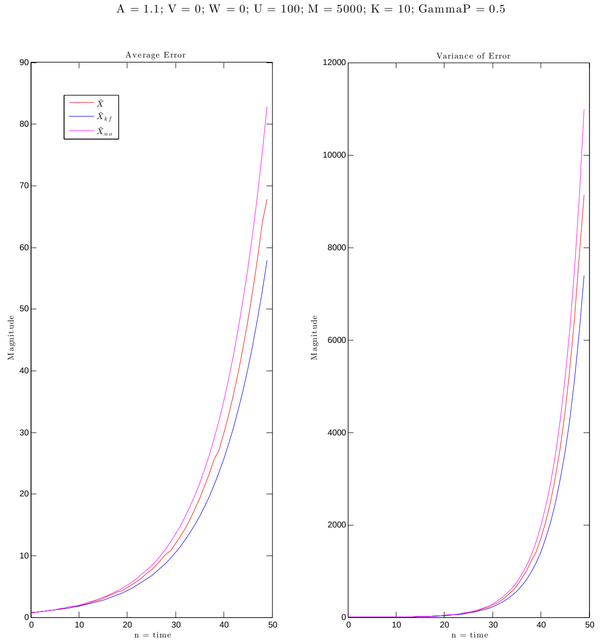
16.10.3 Quantization Noise





In these figures I have varied the power of the multiplicative noise. When the multiplicative noise is high, the observations are essentially useless and there is less impact from delay/drops.

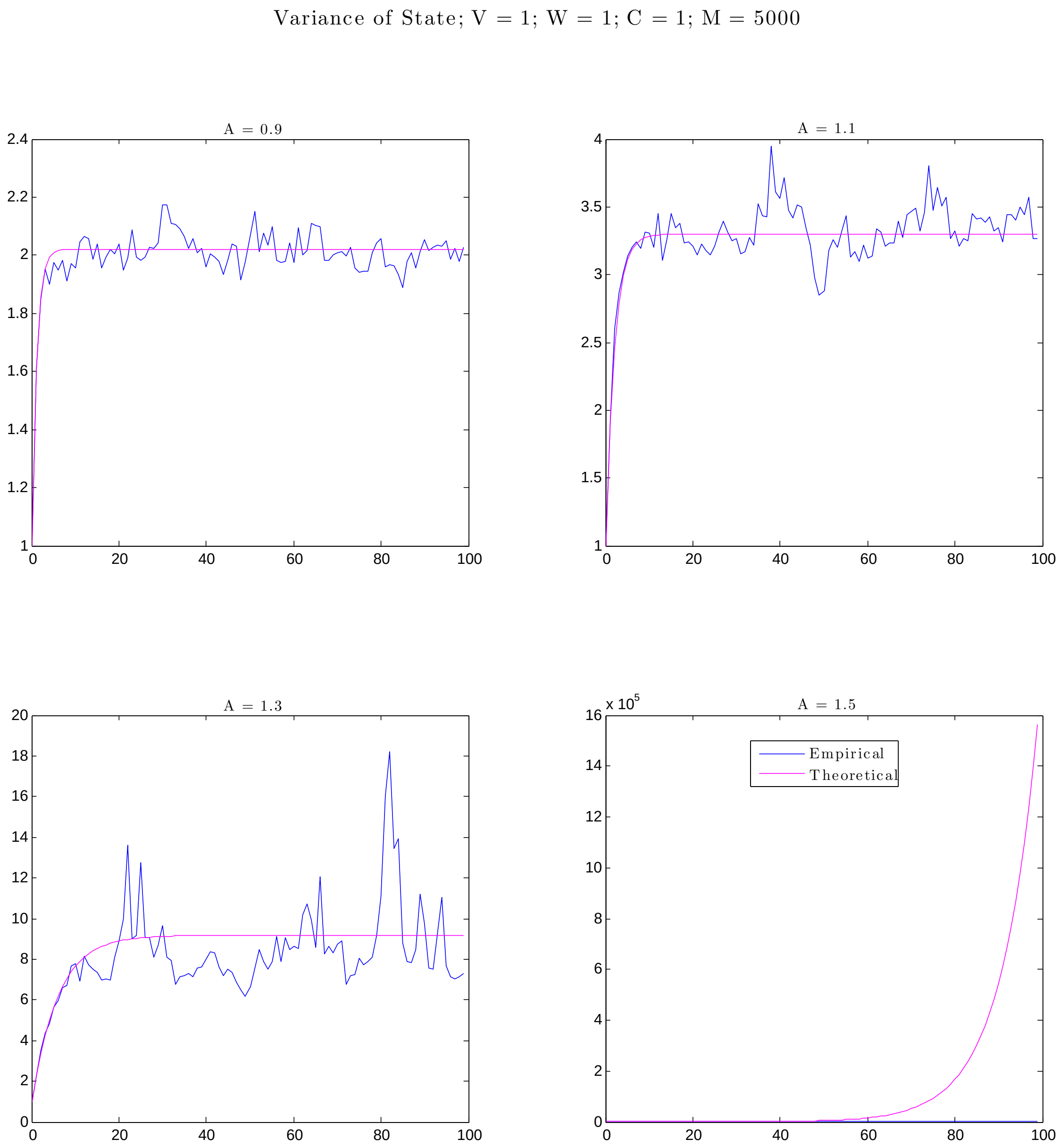




16.11 Gireeja's Control

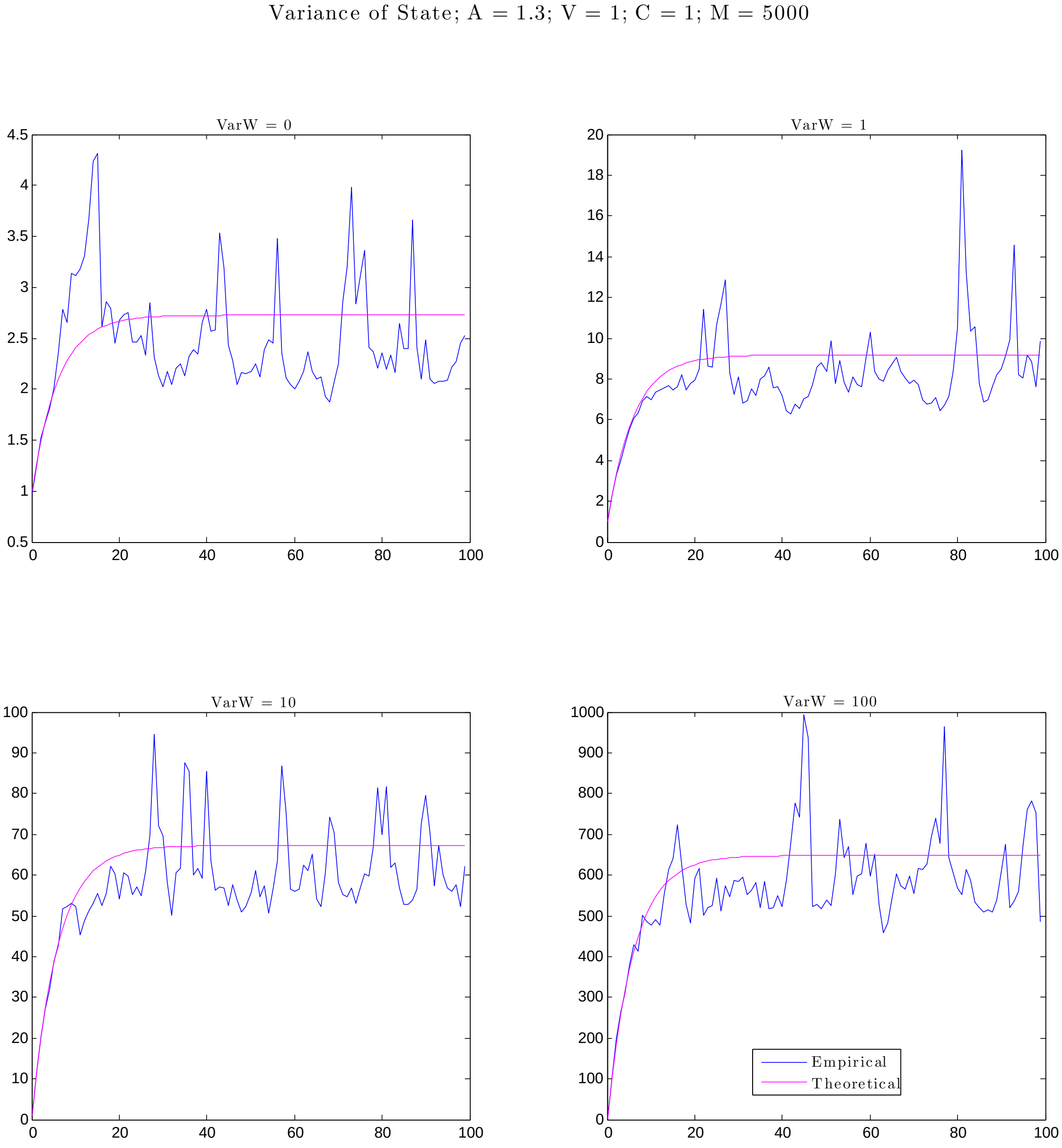
Although when A > Threshold, Theoretical Variance >> Empirical Variance, this is justified by theoretical variance serving as an upper bound of stability. In reality, the system might continue to be controlled for values above the threshold, but this control can't be guaranteed (there are peaks in the variance?)

The control is $d = \frac{a\mu_c}{\mu_c^2+\sigma_c^2}$ as per Gireeja's paper. This is calculated from minimizing the rate of growth of the variance of X, based on the previous variance of X.



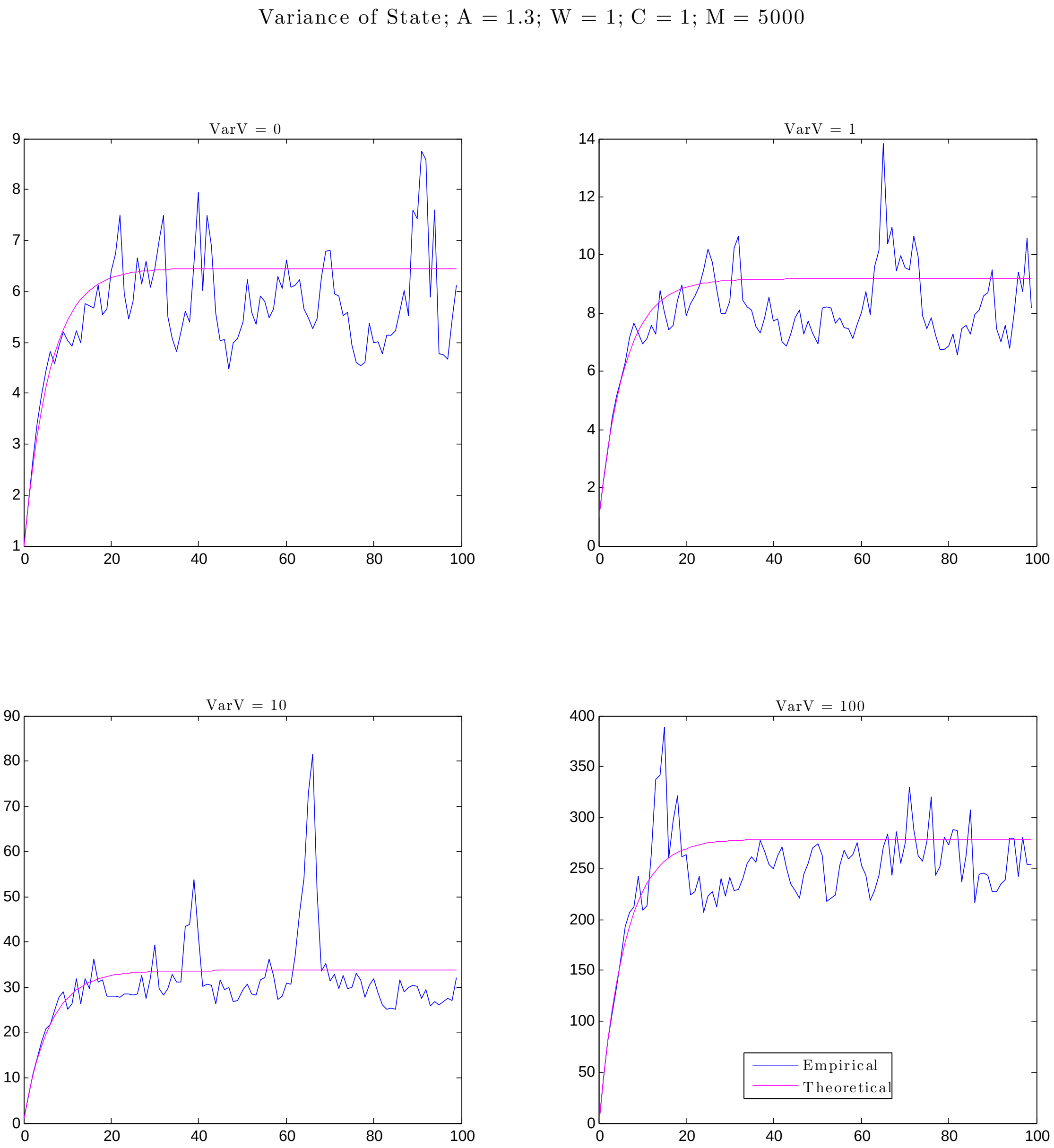
For the above graph, $\mu_c = 1$ and $\sigma_c^2 = 1$. Therefore the threshold for A is $\sqrt{1^2 + 1^2} = \sqrt{2} \approx 1.4$, as demonstrated by the plots.

16.11.1 State Noise



In these plots we can see that as state noise variance increases, the total error increases but the stability of the system doesn't change.

16.11.2 Additive Noise

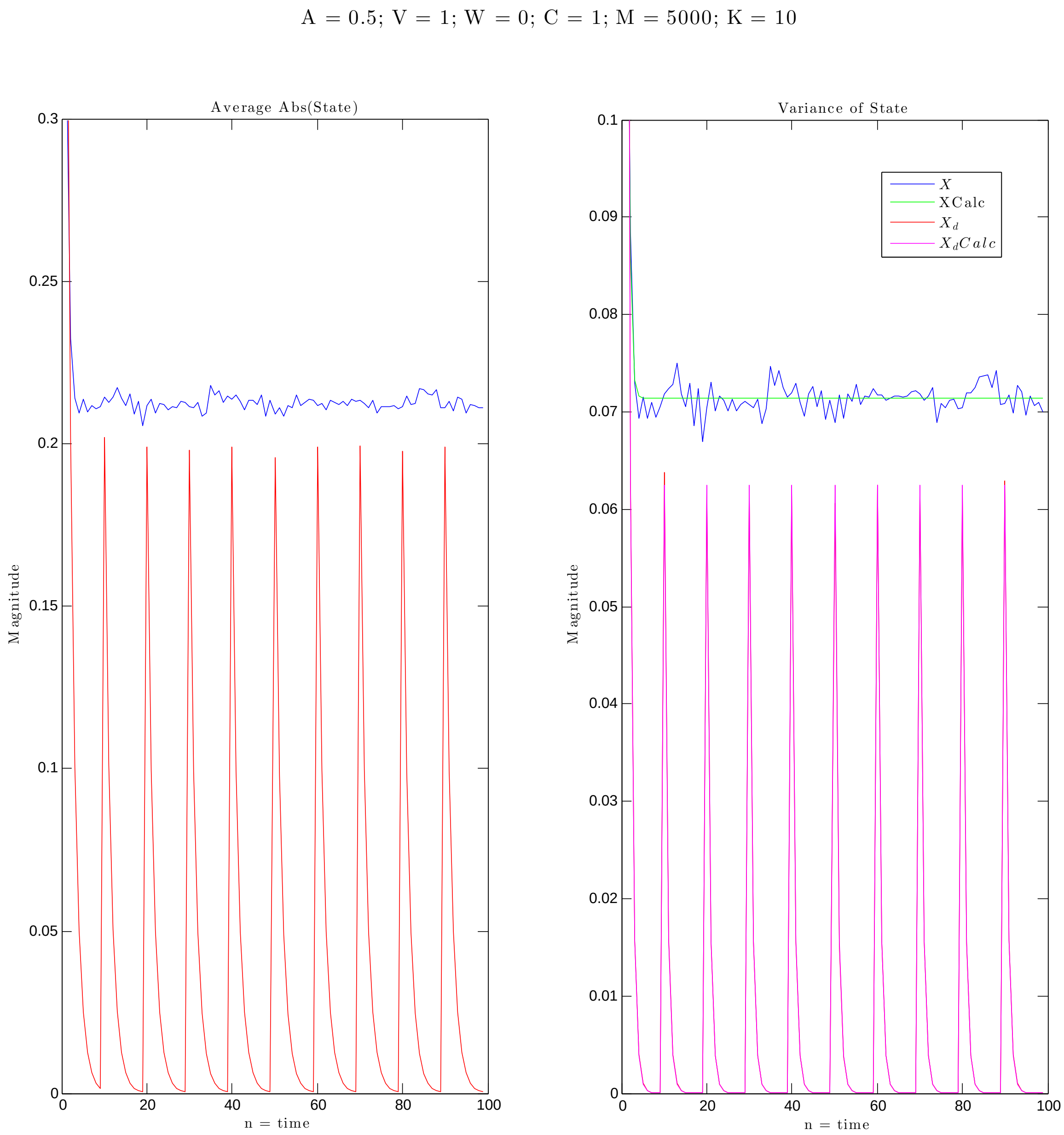


In these plots we see that additive noise behaviors similarly to state variance, except the effect isn't s pronounced. Additive noise variance increases total error, but not as much as state noise does, and it also doesn't affect the stability of the system.

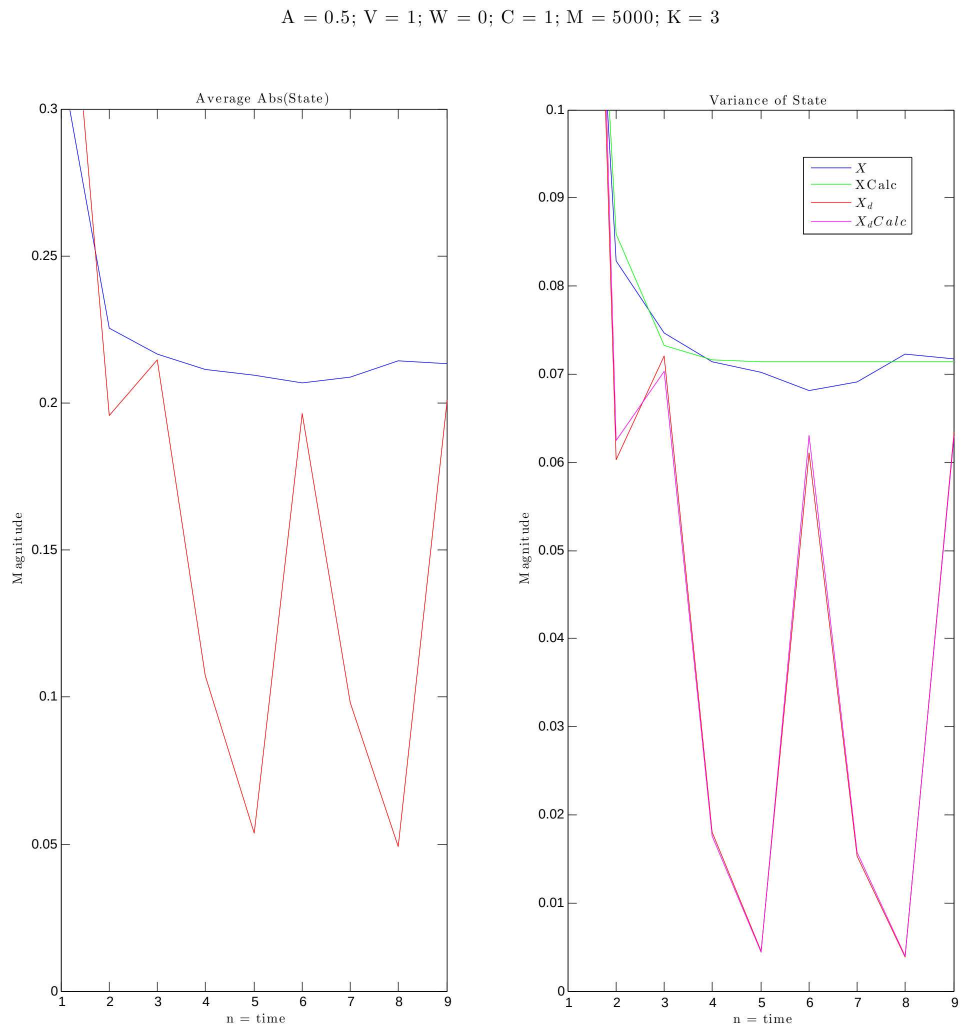
16.12 Delay Control: Quantization Noise, No Drops

16.12.1 A < 1

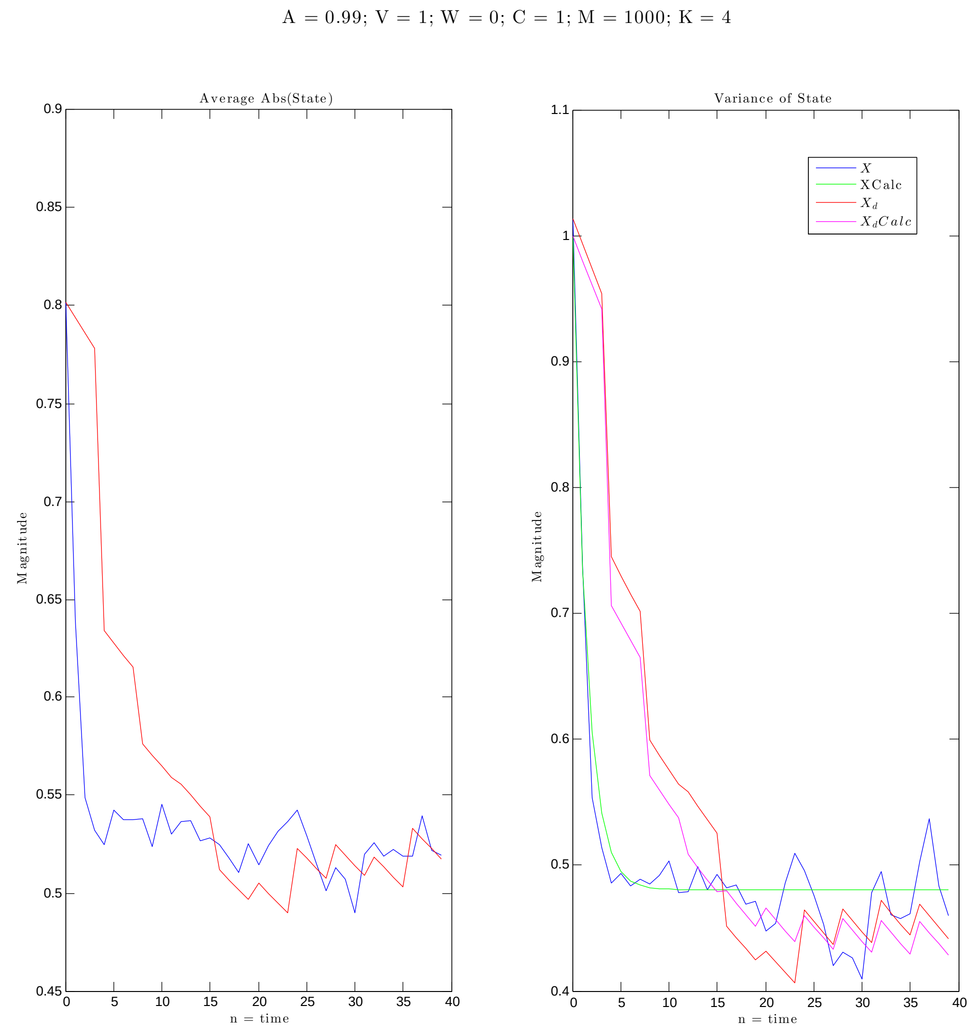
If $A < 1$, the system will be stable/variance will be bounded as per the equation: $\Sigma_\infty = \frac{b^2\Sigma_w}{1-a^2}$. When there is state noise, delay increases the state variance as per the equation $\sigma_n^2 = a^{2k}\sigma_{n-k}^2 + b^2\sigma_w^2 \sum_{i=0}^{k-1} a^{2i}$. If there is no state noise, the control *increases* the state variance. This is because leaving the system alone, it will naturally converge to 0. The controller has additive noise that can 'overshoot' and increase the error.



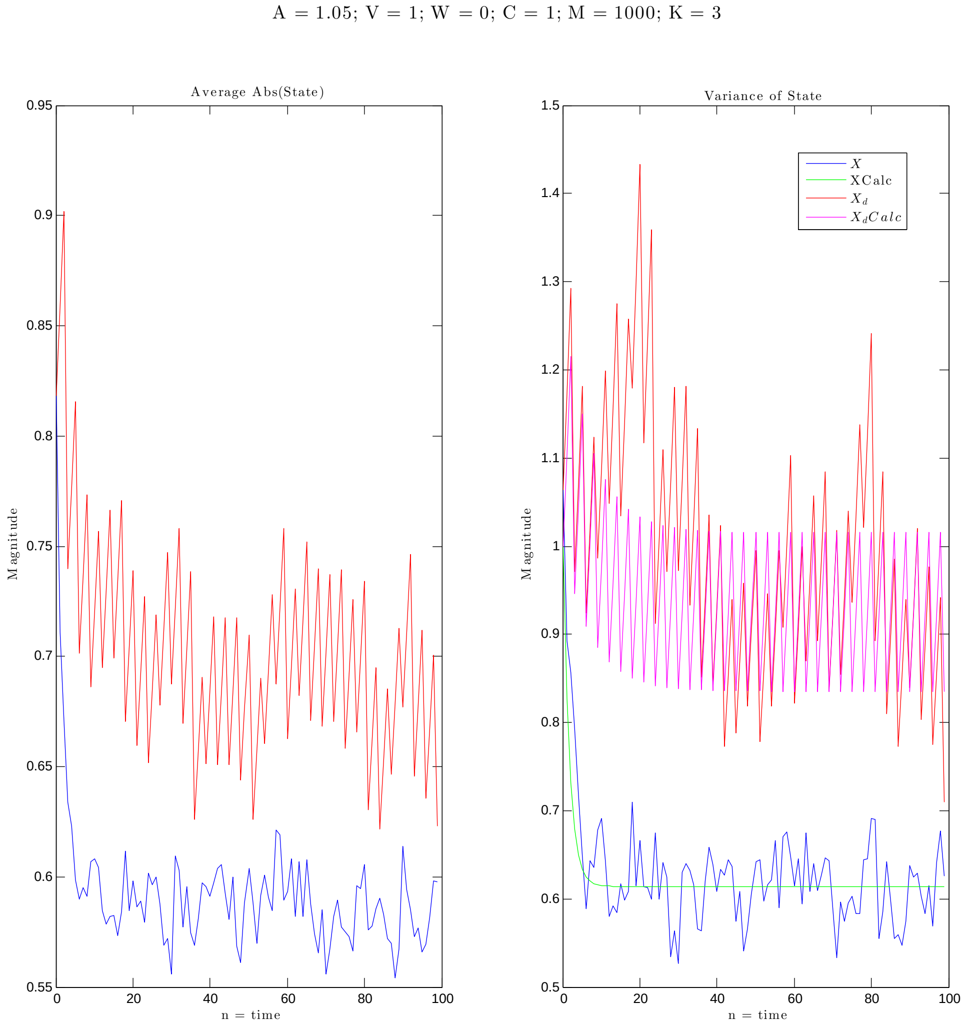
For example, in the above plot VarW = 0 and VarV = 1. The error of the system is $\Sigma_\infty = \frac{b^2\Sigma_w}{1-a^2} = 0$ while the error of the control is $\sigma_n^2 = \frac{a^2\sigma_c^2}{\mu_c^2+\sigma_c^2} * \frac{b^2\sigma_w^2}{1-a^2} + \left(\frac{a\mu_c^2}{\mu_c^2+\sigma_c^2}\right)^2 \sigma_v^2 + b^2\sigma_w^2 = \left(\frac{a^2\sigma_c^2}{\mu_c^2+\sigma_c^2} + 1\right) b^2\sigma_w^2 + \left(\frac{a\mu_c^2}{\mu_c^2+\sigma_c^2}\right)^2 \sigma_v^2 = \left(\frac{0.5*1}{1^2+1^2}\right)^2 * 1 = \frac{1}{16} \approx 0.0625$. The value for error after control in the plot after zooming in is 0.0625. Interestingly enough, the delay gives the system *time* for $\sigma_x^2 \to 0$. Without delay, the system converges earlier and asymptotes at ≈ 0.0714, which matches calculations $\left(\sigma_{x(n+1)}^2 = \frac{a^2\sigma_c^2}{\mu_c^2+\sigma_c^2} * \sigma_{x(n)}^2 + b^2\sigma_w^2 + \left(\frac{a\mu_c}{\mu_c^2+\sigma_c^2}\right)^2 \sigma_v^2\right)$.



In this plot, I changed $k = 10 \to k = 3$, and focused on the first 10 timesteps. You can see that the system jumps at the control, and uses the delay time to converge towards 0 error. Note that when $k = 10$ the system actually reached Variance = 0, whereas here there isn't enough time so the system is reaching Variance = 0.01 at best.



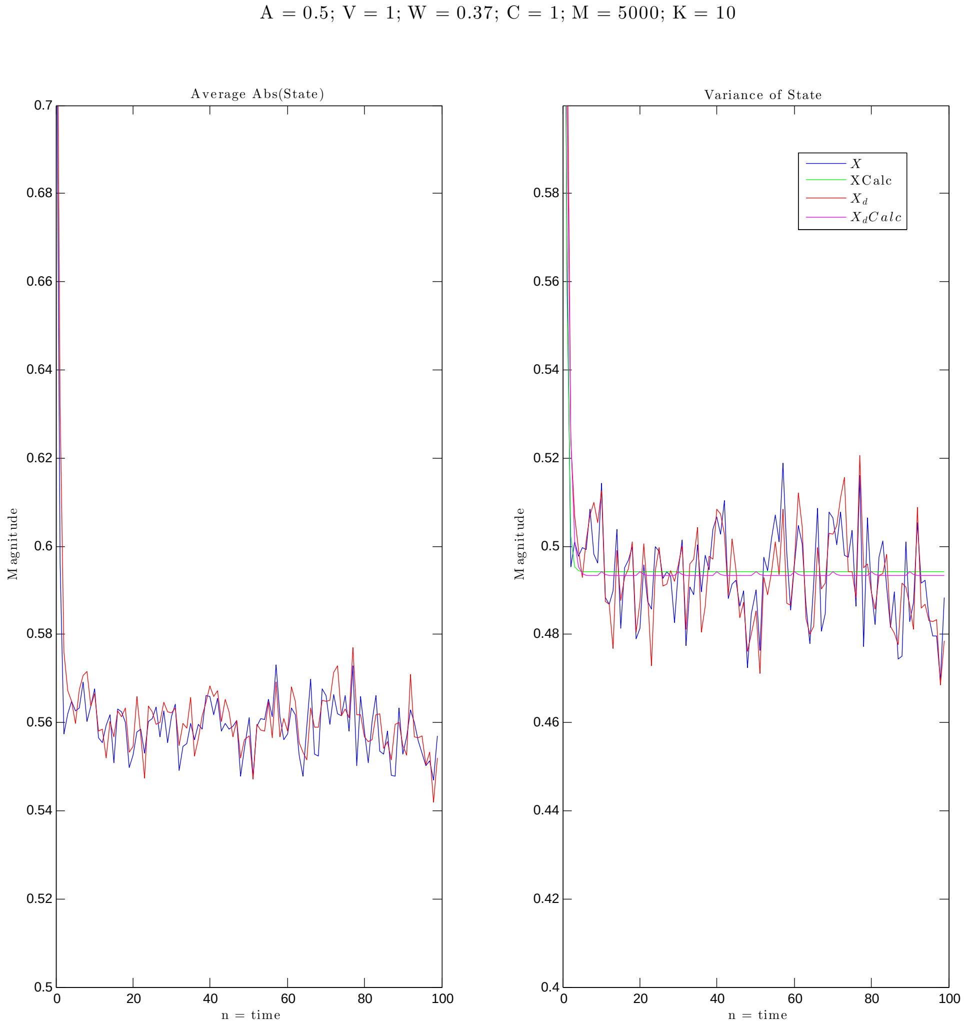
Another example of the effect, with A = 0.99. Note when the pink and green line coincide, which is when a control is added to both systems!! After zooming in, that pink/green line happens between the 15th and 16th timestep, the correct time for a control for the delay system.



In this plot we test $A > 1$. I had to change $k = 10 \rightarrow k = 3$ because it is almost impossible for a value $A > 1$ to sustain delay 10 (will be explained later.) Because $A > 1$, delay now causes error to increase. This means that *because of delay*, the system can't asymptote at a lower error variance. After being pulled down by the control, the system error variance immediately starts growing again. The less delay the lower the min/max error variances will be.

Theorem: The threshold for how much state noise can be tolerated for a certain value of A for the system to still experience a benefit from delay is given by $\sigma_w^2 < \frac{(1-a^{2(k-1)})\sigma_{x\infty}^2}{b^2 \sum\limits_{i=0}^{k-2} a^{2i}}$.

This calculation shows that as A increases, the tolerated σ_w^2 decreases. In addition, if $A > 1 \implies \sigma_w^2 < 0$, which is impossible.

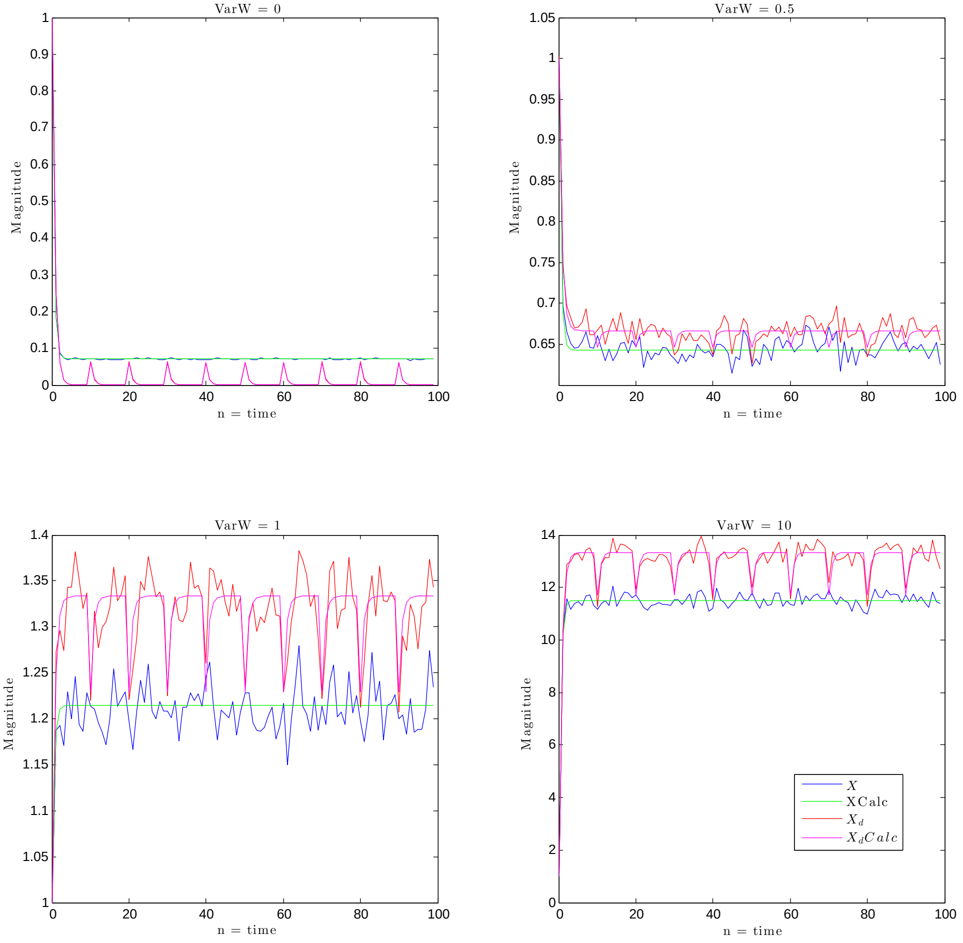


This plot shows calculations match the threshold of delay benefit. (Variance of W = 0.37, and the magenta line rises above the green line when W = 0.38.) Another example is A = 0.1, K = 10, $\sigma_w^2 = 0.5$. The magenta and green lines (theoretical error variances) will match up exactly, and $\sigma_{x\infty}^2 = 0.505$.

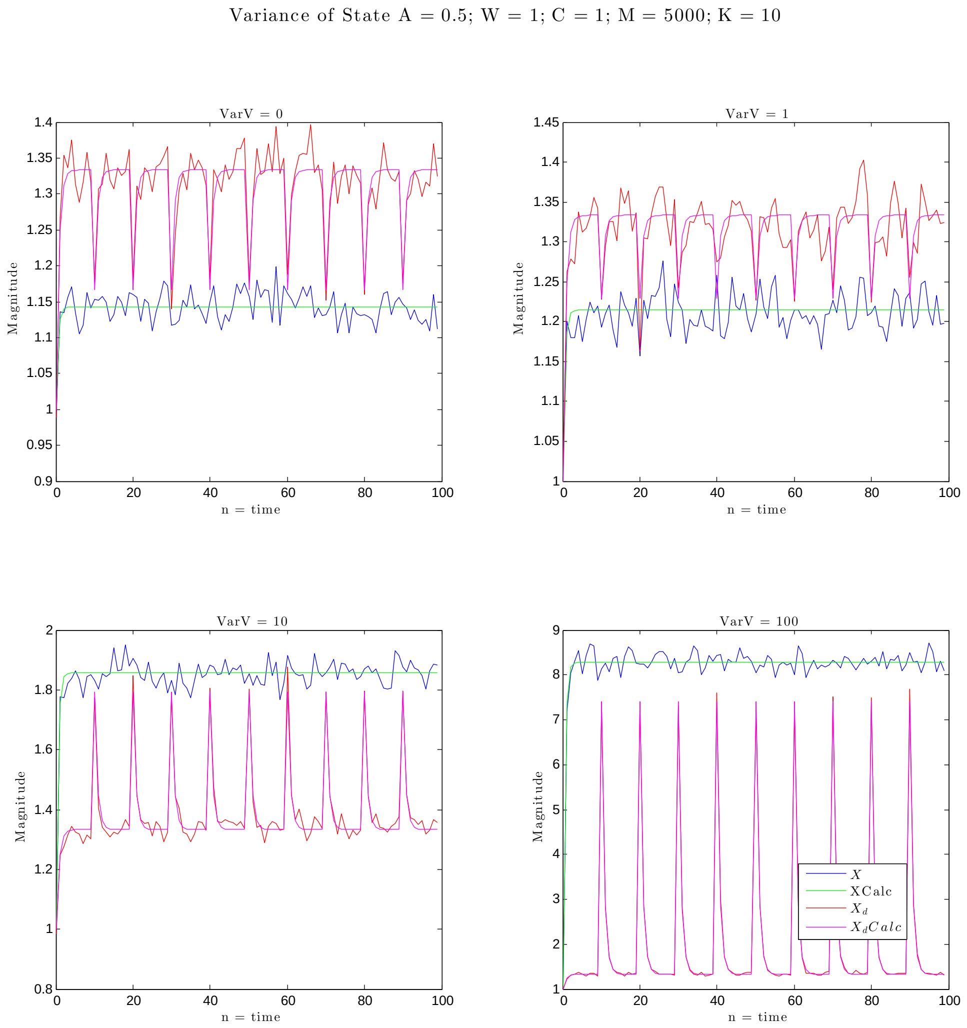
NOTE! MAJOR PROBLEM! $\sigma_{x\infty}^2$ depends on σ_w^2, and I can't just plug in $\sigma_{x\infty}^2 = \frac{b^2\sigma_w^2}{1-a^2}$. This means the theorem above can be used to check work, but is not good for calculating the threshold before constructing plots.

16.12.2 State Noise

Variance of State A = 0.5; V = 1; C = 1; M = 5000; K = 10



16.12.3 Additive Noise



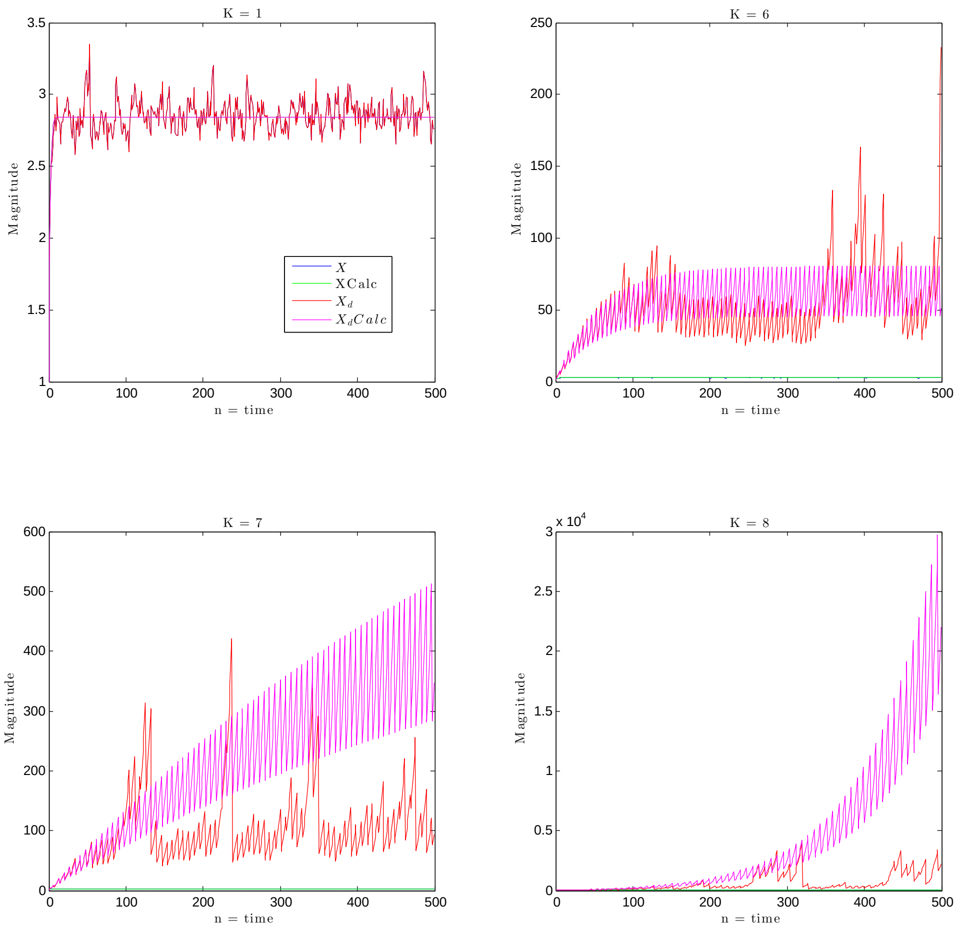
These plots indicate that increasing levels of additive noise might increase the benefits of delay, probably because the controller is noisier and tends to "overshoot" 0. **This needs to be explored.**

16.12.4 A > 1

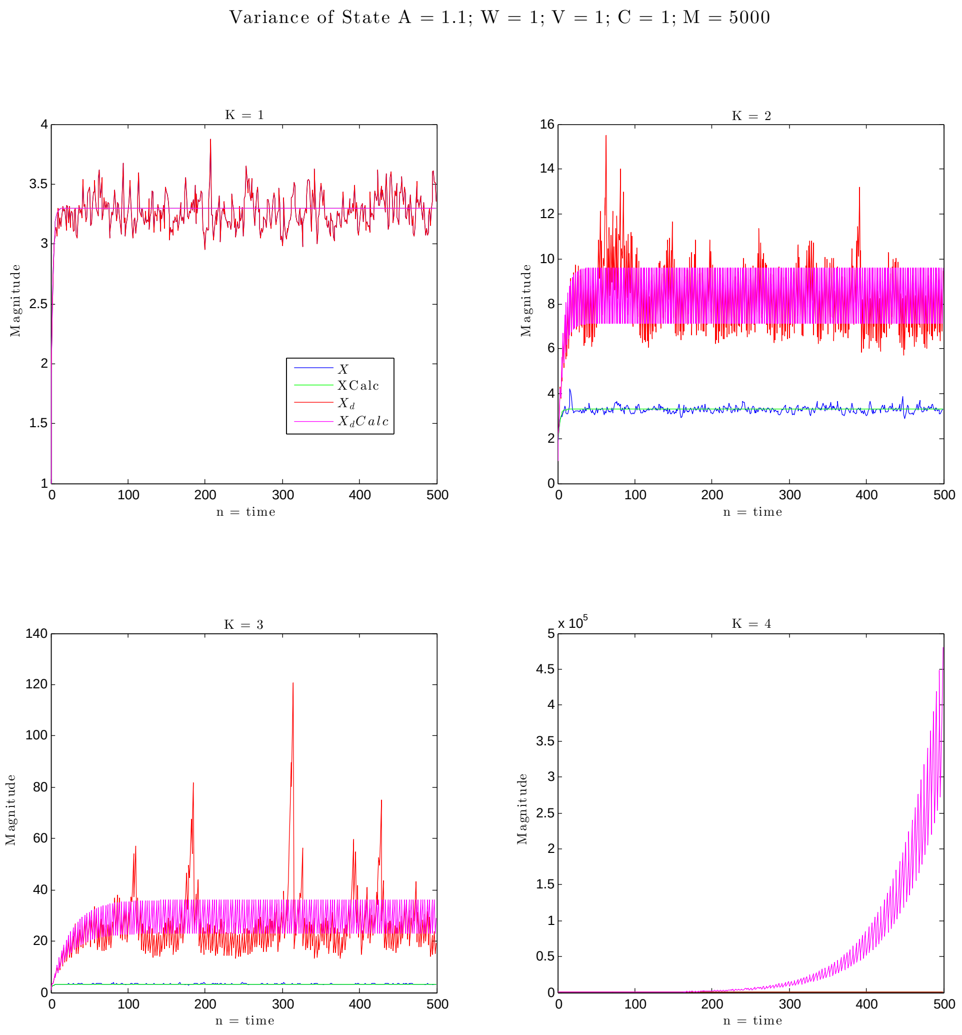
Theorem: A system with delay k will be stable if the following condition is satisfied: $1 > \frac{\sigma_c^2 * a^{2k}}{\mu_c^2+\sigma_c^2}$.

For example, we can calculate that with A = 1.05, $\mu_c = 1, \sigma_c^2 = 1$, a delay of k=7 is the threshold for stability. $1.05^{14}/2 = 0.99$

Variance of State A = 1.05; W = 1; V = 1; C = 1; M = 5000



For A = 1.1, $\mu_c = 1, \sigma_c^2 = 1$, the threshold is instead k = 4. $1.1^6/2 = 0.89, 1.1^8/2 = 1.07$



The above was discontinued because I introduced packet drops.

16.13 Delay Control: Quant Noise, Packet Drops

Variance after the control will be experienced via the following equation:

$$\sigma^2_{x(mk)} = (a^2 - 2\gamma^p a\mu_c d + \gamma^p d^2(\mu_c^2 + \sigma_c^2))\sigma^2_{x(mk-1)} + b^2\sigma_w^2 + \gamma^p d^2 \sigma_v^2$$

Where γ^p represents the probability the packet is received. With the choice for $d = \frac{a*\mu_c}{\mu_c^2+\sigma_c^2}$, this simplifies to:

$$\sigma^2_{x(mk)} = (a^2 - \gamma^p a\mu_c d)\sigma^2_{x(mk-1)} + b^2\sigma_w^2 + \gamma^p d^2 \sigma_v^2$$

$\sigma^2_{x(mk-1)}$ is the variance of the state after delay, or $\sigma^2_{x(mk-1)} = a^{2(k-1)}\sigma^2_{x(mk-k)} + b^2\sigma_w^2 \sum_{i=0}^{k-2} a^{2i}$. Thus:

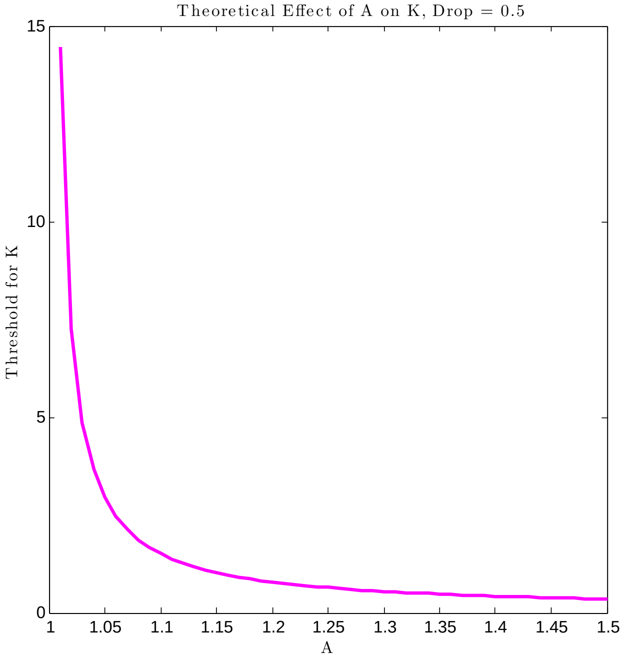
$$\sigma^2_{x(mk)} = (a^2 - \gamma^p a\mu_c d)\left(a^{2(k-1)}\sigma^2_{x(mk-k)} + b^2\sigma_w^2 \sum_{i=0}^{k-2} a^{2i}\right) + b^2\sigma_w^2 + \gamma^p d^2 \sigma_v^2$$

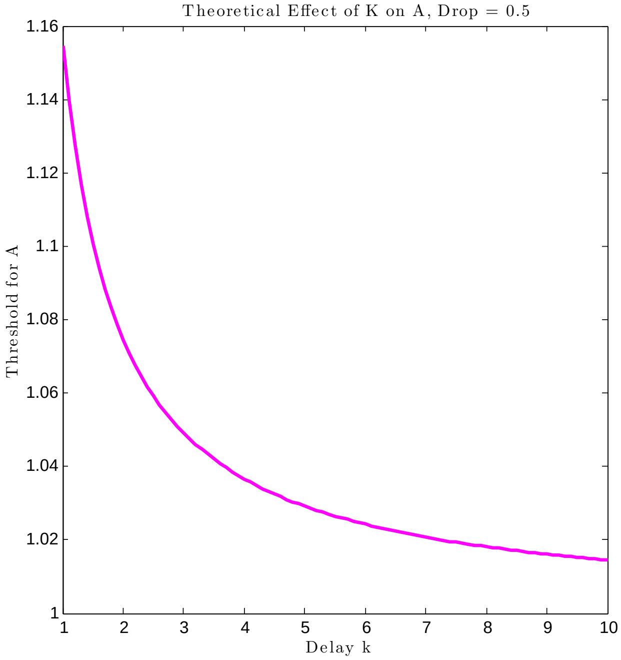
What matters most is the rate of growth of σ_x^2, which is:
$(a^2 - \gamma^p a\mu_c d) * a^{2(k-1)} = (a^2 - \gamma^p a\mu_c \frac{a*\mu_c}{\mu_c^2+\sigma_c^2}) * a^{2(k-1)} = a^2\left(1 - \gamma^p \frac{\mu_c^2}{\mu_c^2+\sigma_c^2}\right) * a^{2(k-1)} = a^{2k}\left(\frac{(1-\gamma^p)\mu_c^2+\sigma_c^2}{\mu_c^2+\sigma_c^2}\right) < 1.$

Since the expression in parentheses is by definition positive, $\boxed{a^{2k} < \frac{\mu_c^2+\sigma_c^2}{(1-\gamma^p)\mu_c^2+\sigma_c^2}}$.

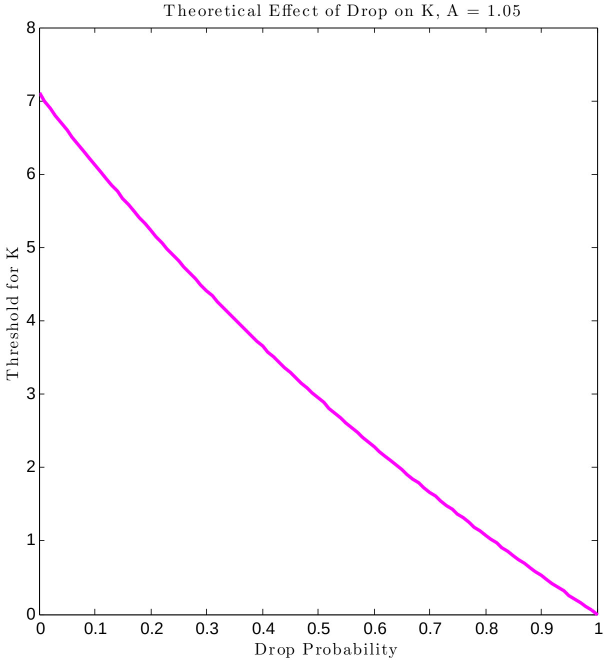
$$\text{This leads to } \boxed{k < \frac{\log_a\left(\frac{\mu_c^2+\sigma_c^2}{(1-\gamma^p)\mu_c^2+\sigma_c^2}\right)}{2} = \frac{log\left(\frac{\mu_c^2+\sigma_c^2}{(1-\gamma^p)\mu_c^2+\sigma_c^2}\right)}{2log(a)}}.$$

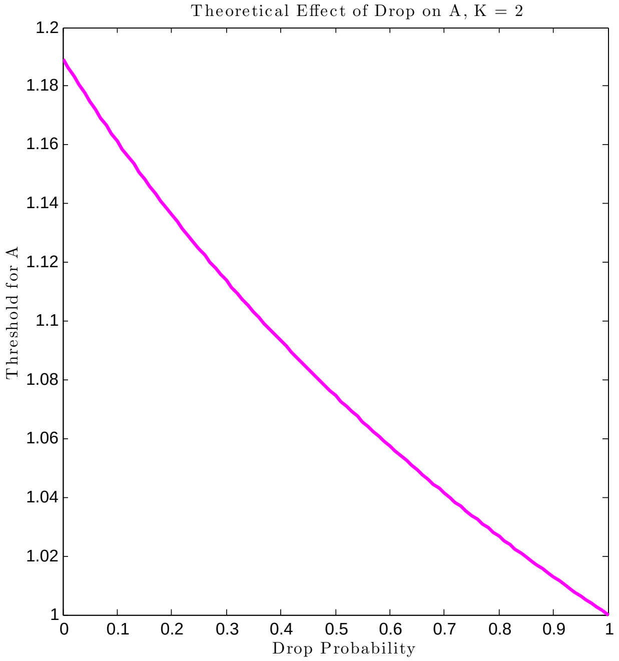
16.13.1 Theoretical Curves





Note that in the right plot the curve asymptotes towards 1. This is because the calculations assume $A > 1$, as that is the case we are concerned with. The right plot also begins with $k = 1$. This is because my definition of delay is that a packet is received every k^{th} timestep, so $k = 1$ represents no delay and it's impossible to have a $k < 1$.
The curve on the left clearly shows the sharpest decay of A with k, which makes sense since the equation looks roughly like $y = constant^{1/2x}$. The y limits of the curve on the right shows how limited values of A are.

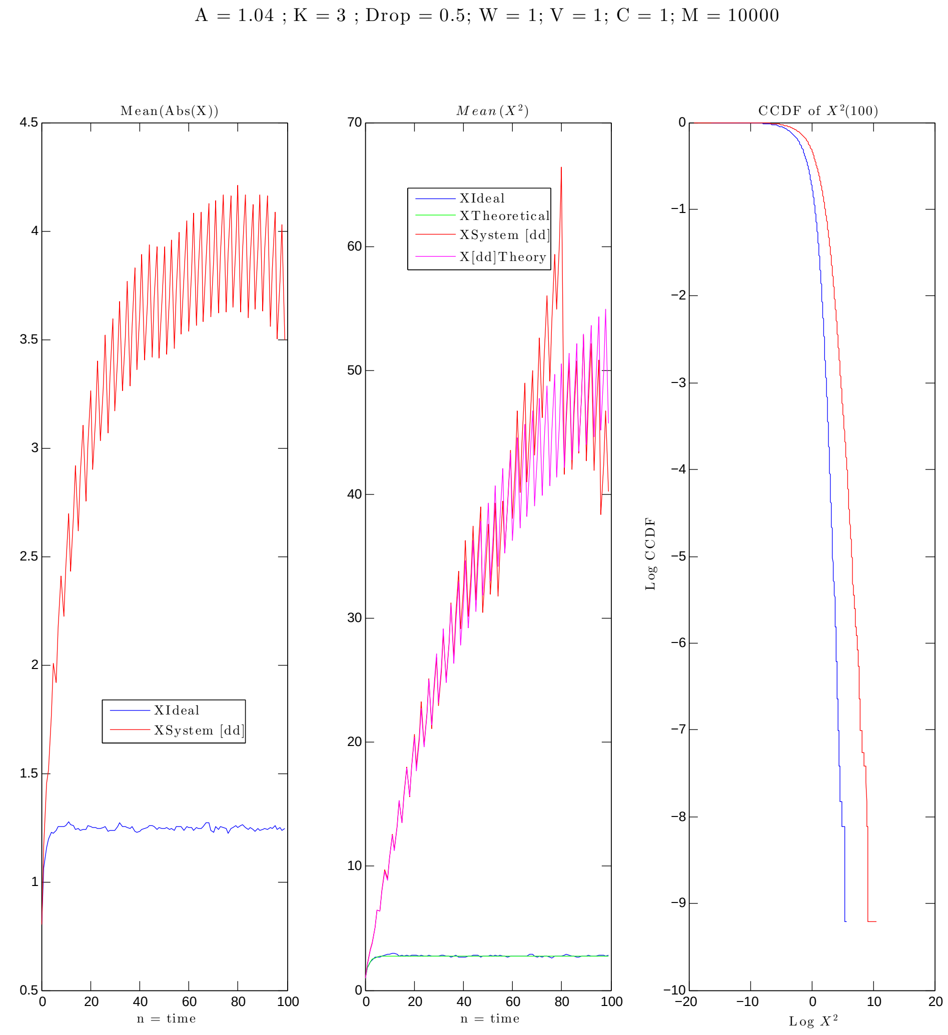




These plots show the effect of drop probability on delay k and growth A. The bottom left curve shows that for $A = 1.05$ the max K is 7. This was shown above with Delay Control: Quantization Noise + No Drops, confirming my calculation matches simpler cases. In addition, the curve is almost linear, but the slope is dependent on the value for A. For example, a quick run with $A = 1.1$ produced a slope ≈ -3.6 vs. this plot, which has slope ≈ -7.

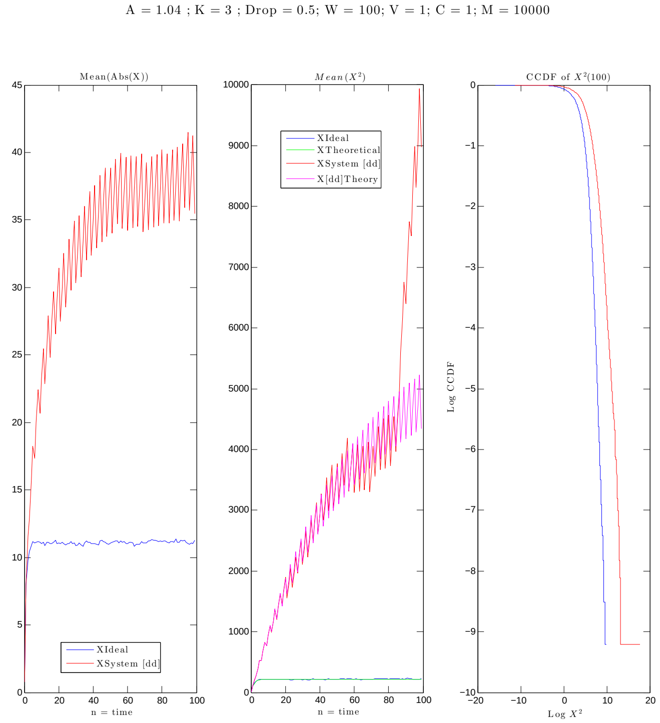
The bottom right curve is extremely shallow compared to the plots above, the result of the equation being roughly $y = \left(\frac{1}{x}\right)^{1/2k}$ with $0 < x < 1$. **MORE WRITEUP??**

This plot will serve as the baseline for the following three plots.



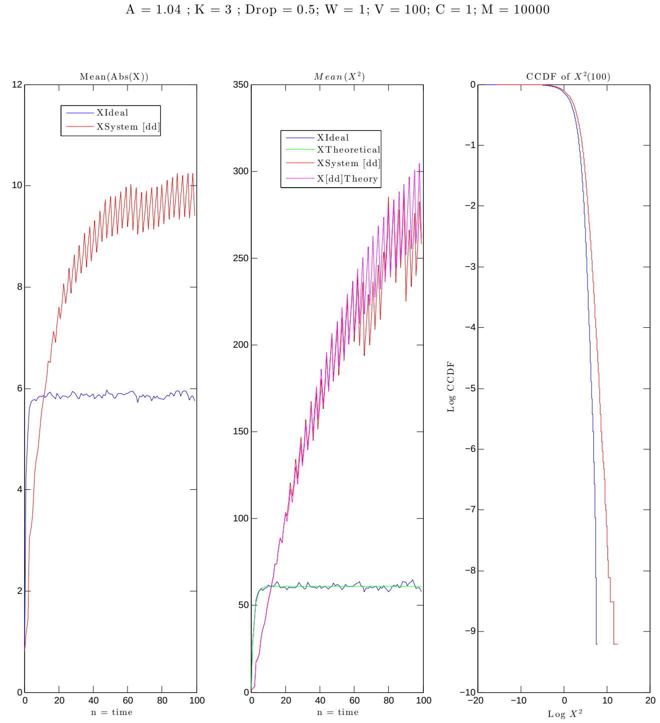
The polyfit for the log-log CCDF in this plot is -1.1816.

16.13.2 State Noise



In this plot, you can see that state noise doesn't affect stability. The total error and error variance increases significantly, but the shape of the theoretical curve remains the same.

16.13.3 Additive Noise

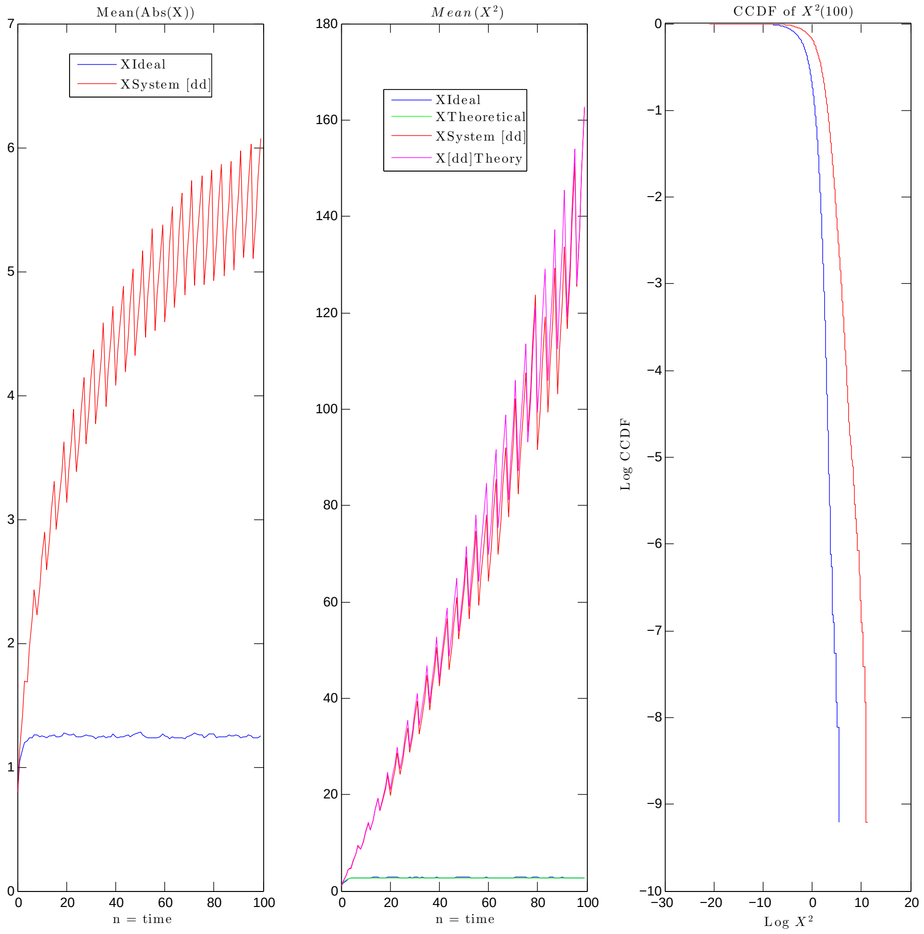


In this plot, you can see that additive noise doesn't have as large an effect as state noise, although it still increases the total error. In addition, relative to state noise, additive noise has a greater effect on both the delay-drop system and the ideal system compared to just the delay-drop system.

16.13.4 Threshold of Stability

The following plots show how the increase of delay by one will cross the threshold of stability, changing the shape of the variance (or second moment) curve.

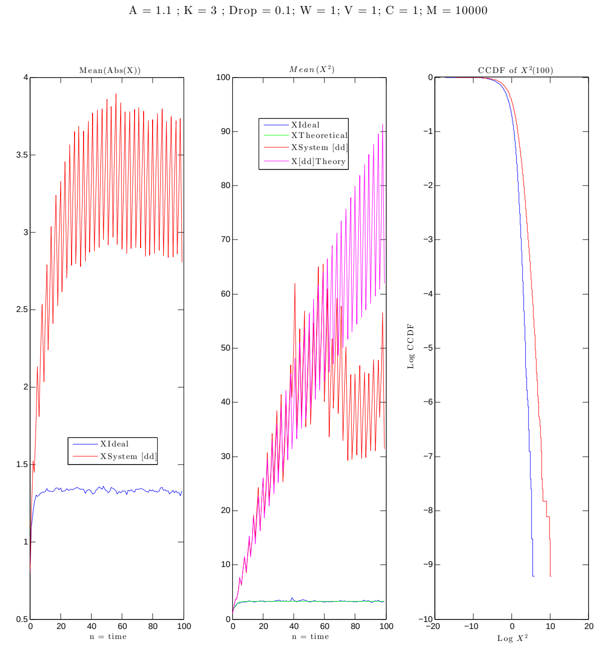
A = 1.04 ; K = 4 ; Drop = 0.5; W = 1; V = 1; C = 1; M = 10000

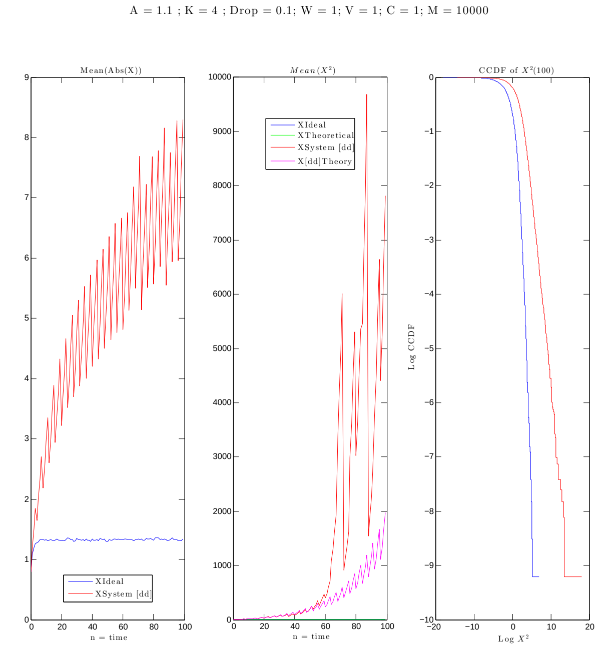


This plot should be compared to the baseline plot above. A polyfit of the tail of distribution (linear regression of log(vec2(8000:9000)),log(1-data2(8000:9000)) or the slope of the log-log CCDF near the bottom of the plot) gives α = -0.9331. Since this value is just slightly less than -1, it indicates the system is unstable and won't converge.

Note: vec2 is a vector length 1x10,000 because cdfld uses a logspace with partitions 10,000.

The following two plots show the threshold for a smaller drop probability and larger value of A.

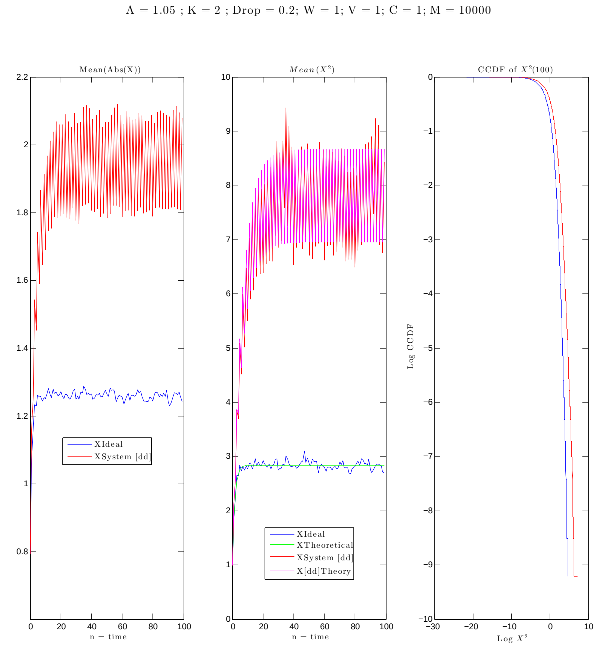


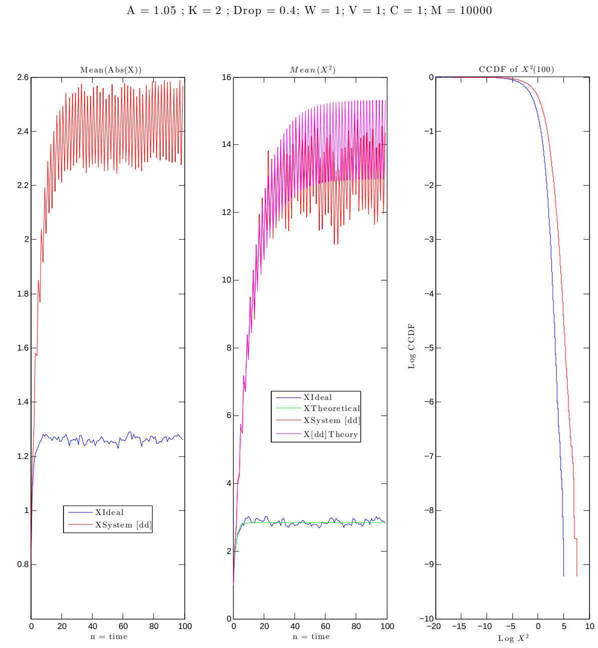


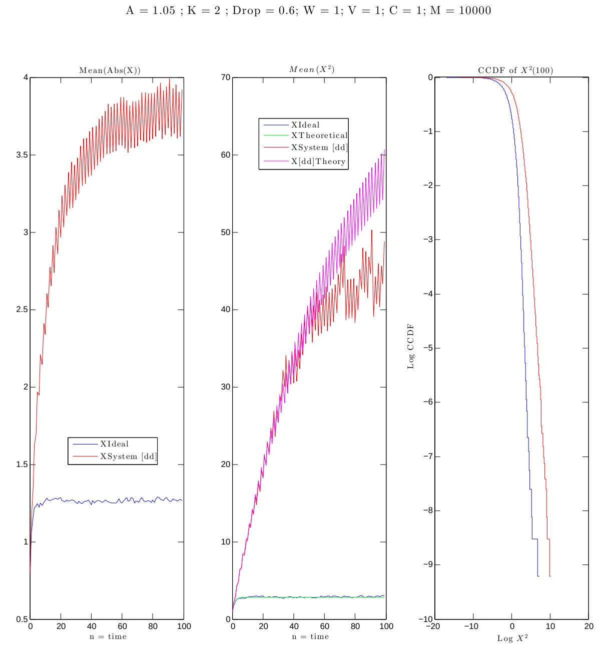
In the first graph, below the threshold, the polyfit of the log-log CCDF gave a slope $\alpha = -1.2444$. In the second graph, above the threshold, the polyfit gave a slope $\alpha = -0.8495$. This demonstrates that the increase in delay by 1 crossed a threshold of stability. Note that since the value for A is the same, the blue ”ideal” log-log CCDF is in the same position between the two graphs.

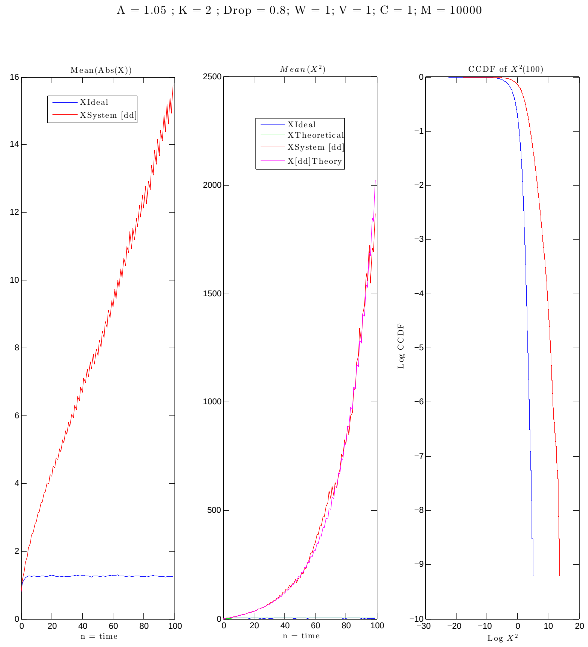
*****Note:***** For the following plots, make sure to pay attention to the values for A/K/P! They are not always the same, because the system blows up easily so I found values for A/K/P that would showcase effects of one parameter. In general, I chose $A = 1.05$ and $K = 2$ which is the minimum amount of delay.
Over the threshold, many of the plots don't show the ”heavy tail behavior” we saw previously. I think this is because the threshold is a lot more sensitive, so values ”just” over the threshold are actually high enough to cause visible explosion.

16.13.5 Varying Drop Probability



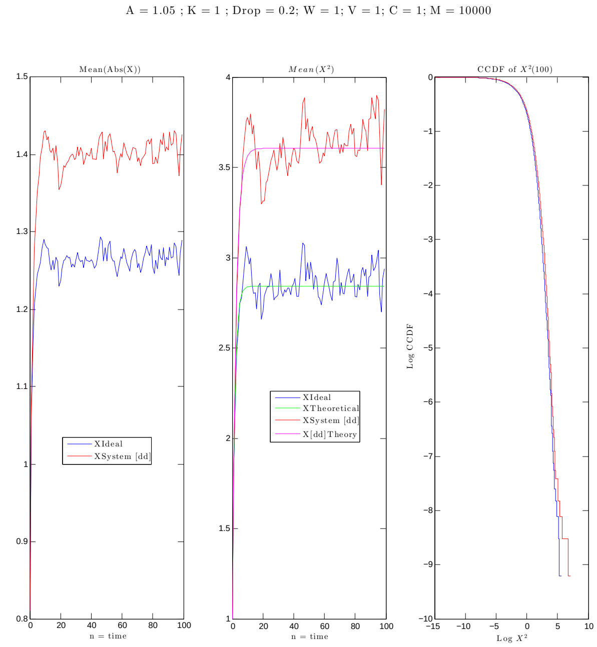


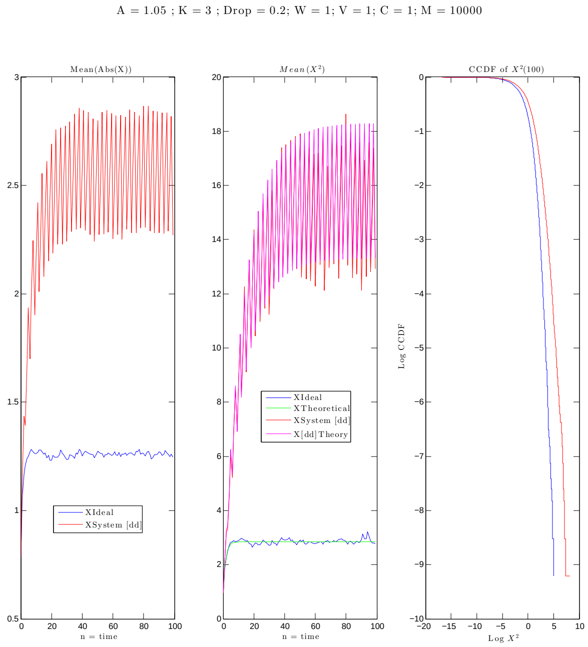


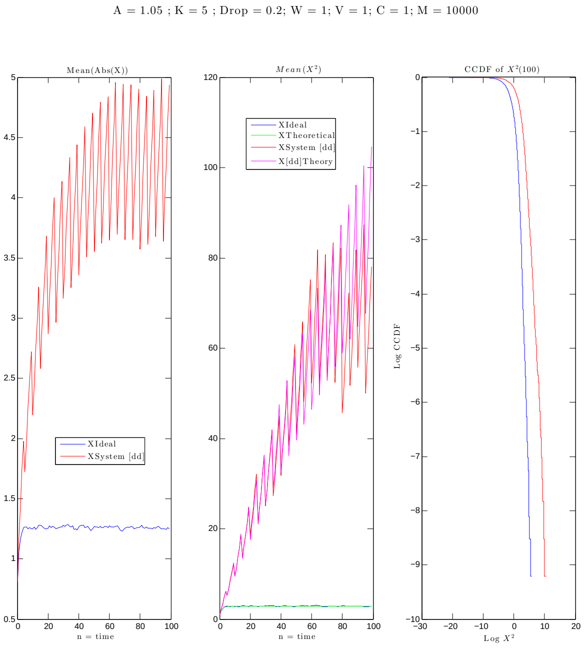


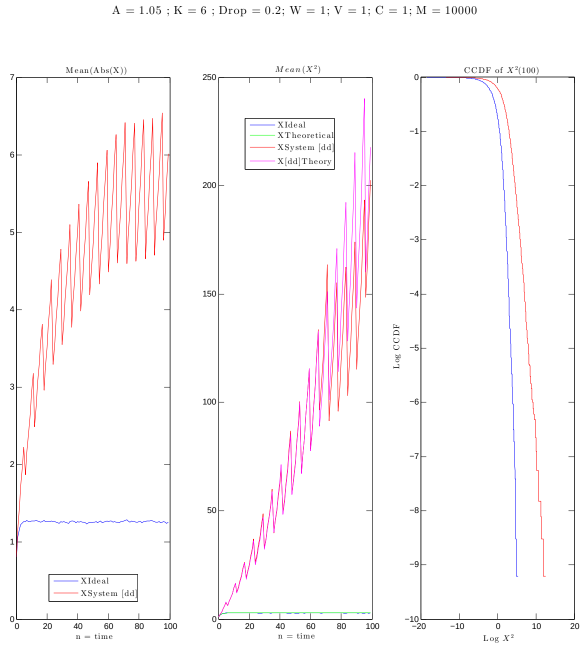
These results show the system with increasing values for drop probability. You can see just how markedly the Mean(Abs(X)) changes over the threshold, as well as how much longer it takes for the variance to asymptote/stabilize when drop probability increases. When packets are dropped, the system is effectively uncontrolled so X grows unboundedly. The polyfit for the bottom left curve's log-log CCDF is -1.1641, while the polyfit for the bottom right curve's log-log CCDF is -0.7981.

16.13.6 Varying Delay



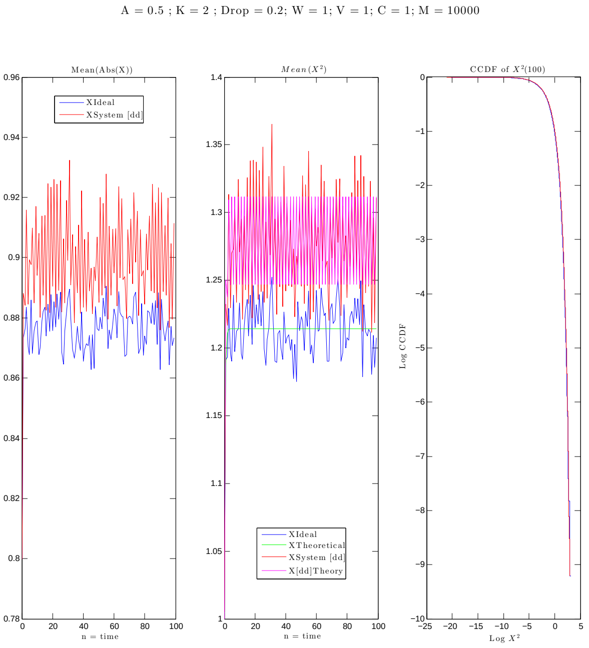


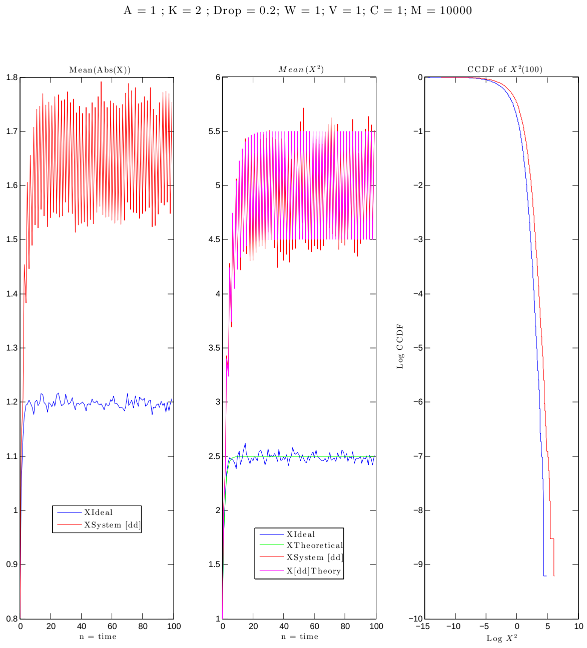


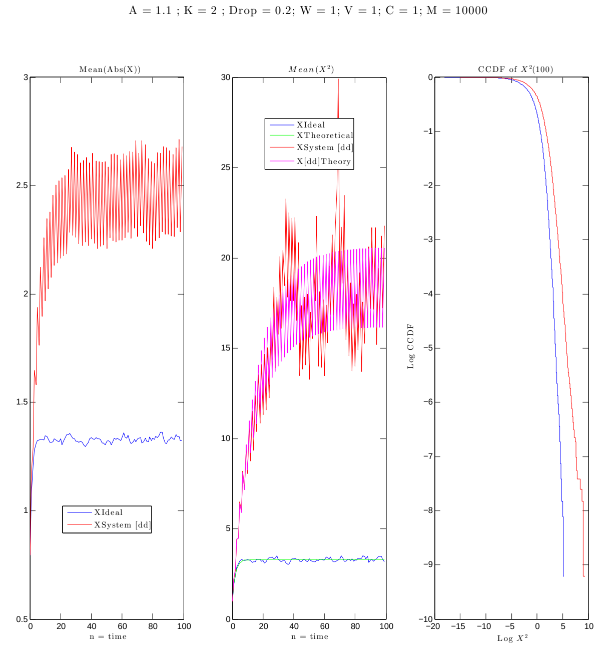


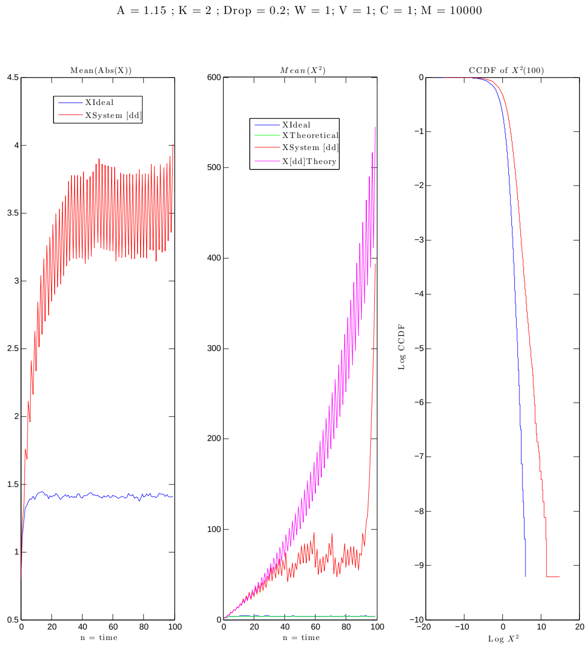
The plot in the upper left represents no delay (there's still a difference between the system and the ideal because of packet drops). You can see in the CCDF plot that that difference is *really* small compared to the effect of delay. For the bottom two plots, the polyfit of the left log-log CCDF is -1.0559, showing how close this system is to the threshold. The polyfit of the right plot is -0.8227. You can also eyeball a substantial difference in the CCDFs, since the blue "ideal" is a fixed position.

16.13.7 Varying A



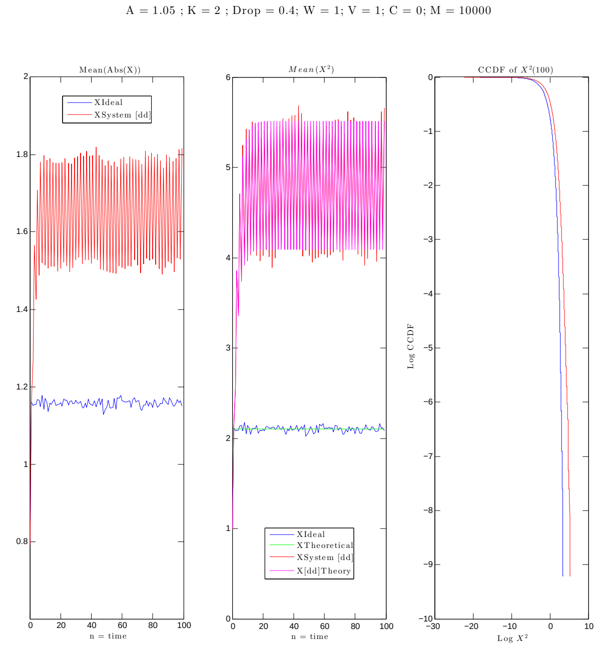


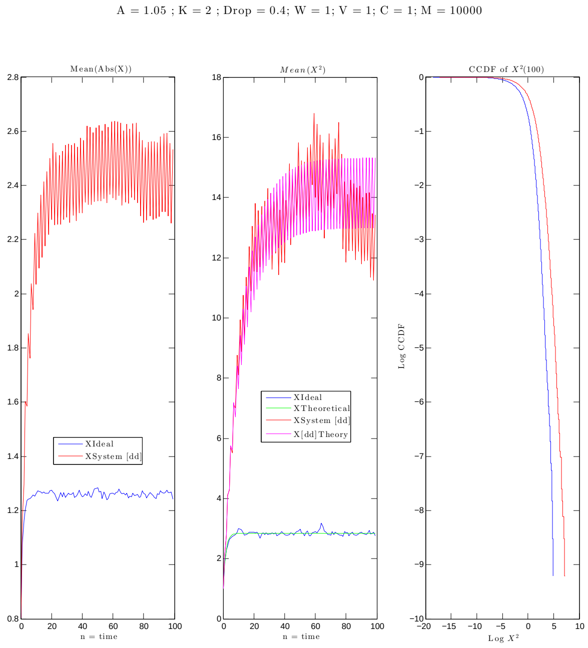


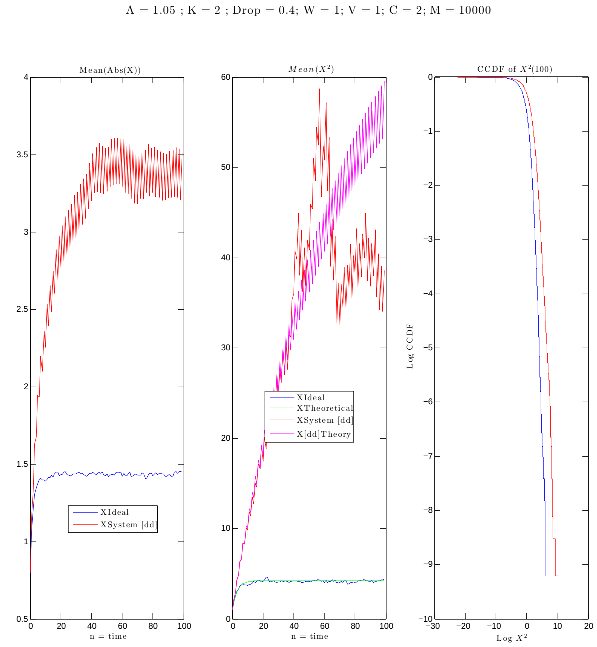


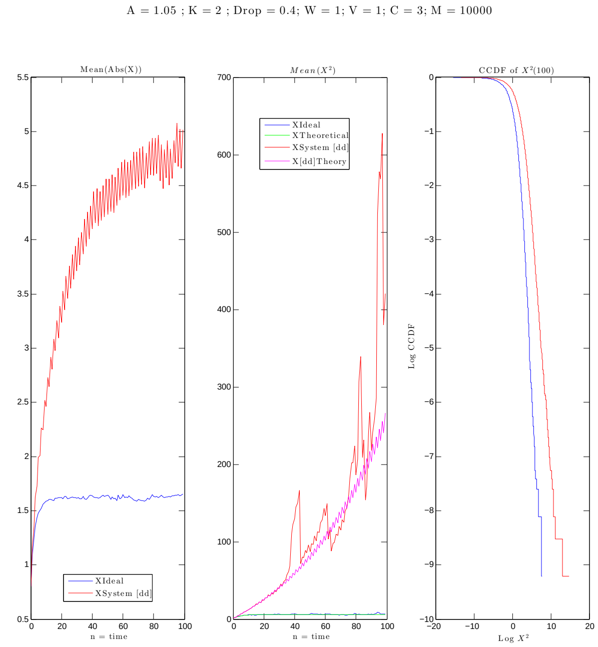
The upper left plot is the first time I tested this code with $A < 1$. You can see almost no difference in the CCDFs between ”ideal” and ”delay+drop system”. Although the system is still doing worse in terms of absolute value of state or variance, the difference is miniscule. This should be investigated further. When $A = 1$, the distance between the ideal curves and the system curves widens a lot more.
In the bottom two plots, the polyfit of the left side’s log-log CCDF is -1.1784, while the polyfit of the right side is -0.9040.

16.13.8 Varying Quant Noise









In my previous plots before Delay+Drop+Control, I would blow the variance of C up to 100. I can't do that anymore, because the threshold for σ_c^2 is now $\approx< 5$. Only very small values for quantization noise are allowed.
The top left curve has $\sigma_c^2 = 0$, or absolutely no quantization noise.
The polyfit for the bottom left curve is -1.0409, while the polyfit for the bottom right curve is -0.7981.