

# **Estimation using Quantized Innovations for Wireless Sensor Networks**

**Noele Norris**

Electrical Engineering, 2010

Minor in Control and Dynamical Systems

California Institute of Technology

*Mentor: Babak Hassibi*

*Co-mentor: Ravi Teja Sukhavasi*

## Acknowledgements

---

I would like to thank Babak Hassibi, my mentor who was always around to give valuable suggestions—for both my project and my academic endeavours, and Ravi Teja Sukhavasi, my brilliant grad-student mentor who kindly took me under his wing and who always took the time to explain and derive important concepts for me. I have learned so much under both of their guidance.

## Abstract

---

Recent advancements in integrated small scale micro-electromechanical system technology has created cheap, low power-consuming sensors that can be used in wireless sensor networks, an increasingly popular technology because of its potentially diverse applications. However, sensor networks have many constraints, such as limited bandwidth and power, which have inspired a considerable amount of research for the development of energy efficient detection and estimation algorithms using quantized observations. Though optimal estimation algorithms using quantized innovations have been recently developed to tackle this problem, bounds are not available on the error of the resulting optimal filter. Because tight bounds on the estimation error are essential in determining the stabilizability of the corresponding closed loop dynamical system and thus the applicability of a filter to a specific system, this project focuses on developing error bounds from a close study of the filtering algorithms. Initial attempts were unable to show that the estimation error of a system using quantized innovations followed a Ricatti recursion. Thus, a number of different algorithms and coder-estimator pairs were then analyzed to determine performance and to better understand means of proving stabilizability. Our primary goal is to have a better understanding of the evolution of the lower and upper bounds of estimation errors under measurement quantization, so that filters with verifiable performance specifications can be systematically designed for particular dynamical systems.

## Table of Contents

---

Acknowledgements .....	2
Abstract .....	3
Table of Figures.....	5
Introduction .....	6
Discussion and Results .....	8
Developing an Estimator for a Wireless Sensor Network.....	8
Quantizing the Measurement using the Sign-of-Innovation.....	8
The Kalman-Like Particle Filter.....	9
Simulating the KLPF with the SOI-KF .....	9
The Probabilistic Set-Membership Description .....	10
Literature Review .....	11
Minero's Data-Rate Theorem.....	12
Yuksel's Stochastic Drift Result .....	15
Markov Chains and Various Notions of Stochastic Stability .....	17
Conclusions .....	19
Methods.....	20
Problem Formulation and Sign of Innovations.....	20
The KLPF .....	20
KLPF Simulations .....	21
The Probabilistic Set-Membership Description .....	22
Minero and Yuksel Simulations .....	23
Appendices .....	27
Appendix A: Probabilistic Set-Membership Description Approach Lemma .....	27
Appendix B: Probabilistic Set-Membership Description Approach Recursion Attempt .....	27
Appendix C: Minero Simulation Code.....	30
MineroSim.m.....	30
Quantizer.m .....	32
Appendix D: Yuksel Simulation Code .....	35
YukselSim.m .....	35
Quantizer.m .....	37
References .....	38

## Table of Figures

---

Figure 1: KLPF Simulation Plot. ....	10
Figure 2: Minero's adaptive, exponential quantizer. ....	12
Figure 3: Minero Simulations.....	14
Figure 4: Yuksel Simulation .....	16
Figure 5: Probability distribution of state for KLPF .....	20
Figure 6: Simulation of KLPF in which error is higher than the SOI-KF Ricatti. ....	21
Figure 7: Simulation of KLPF in which error is lower than the SOI-KF Ricatti.....	22
Figure 8: Comparison of Minero and Yuksel with no Erasures.. ....	24
Figure 9: Comparison of Minero and Yuksel with no Erasures with no averaging.....	25
Figure 10: Yuksel simulation with 0.2 probability of erasure. ....	26

---

## Introduction

---

Wireless sensor networks (WSNs) are networks of distributed, independent devices that monitor various environments or objects and wirelessly transmit measurements to a central base that can evaluate the information. They have become an increasingly popular technology and active field of interest not only because of the many interdisciplinary advancements that are making efficient and inexpensive ones possible but also because of their potential applications in diverse areas. The advancements made in integrated small-scale micro-electromechanical systems that have low power consumption pave the way for smart environments, "physical world[s] that [are] richly and invisibly interwoven with sensors, actuators, displays, and computational elements, embedded seamlessly in the everyday objects of our lives, and connected through a continuous network."<sup>1</sup> In particular, WSNs can be used to monitor the ecosystem of a Redwood tree or monitor material responses to various vibrations and stresses,<sup>2</sup> to track a vehicle or survey traffic flow, and even to track chemicals or make medical diagnostics.

WSNs have many advantages. As the sensors composing a network are usually distributed in large quantities, there is great robustness to sensor failure because of redundancies. There are also possible improvements in signal-to-noise ratios because of the network's ability to reduce the distance between the sensors and their targets. Finally, the distribution of sensors across an environment creates the possibility of 'multi-hopping' transmissions over multiple nodes to a central base, which reduces required energy. However, the very benefits of WSNs create a number of challenges. Not only does the large distribution of sensors create design challenges because of the necessity of controlling a large number of independent devices, the small sensors have limited energy and communication capabilities as well as a limited ability to both process and store data.<sup>3</sup>

Thus, a great amount of research is currently being conducted to address both the various complexities and constraints of WSNs. One area of research tackles the particular challenge of energy and communication costs by considering the specific problem of tracking a noisy dynamical system (such as the position of a moving object) using discretized measurements from a number of sensors assuming that, at one measurement step, one sensor transmits only a constrained number of bits of information.

While the Kalman filter is the optimal estimator for a Gaussian system, constraining the measurements to a few bits creates nonlinearities that skew the probability distribution of the system so that the Gaussian assumption can no longer be assumed. Thus, the focus of this research was to design and analyze new methods of filtering for systems using quantized measurements. After analyzing an optimal particle filter designed by Sukhavasi ([3]) that uses the sign-of-innovation as the quantized measurement, various techniques were used to try to show that a linear dynamic system under sign-of-innovation measurements can be sufficiently tracked.

To do this, we first attempted to derive a recursive state estimation that used a *probabilistic set-membership description of uncertainty* that probabilistically bounds the noises

---

<sup>1</sup> Mark Weiser quoted in [1] D. Cook and S. Das, *Smart Environments : Technology, Protocols and Applications*. New York: Wiley-Interscience, 2004.

<sup>2</sup> These two applications are discussed in detail in [2] F. Zhao and L. J. Guibas, *Wireless Sensor Networks : An Information Processing Approach*. Greensboro: Morgan Kaufmann, 2004.

<sup>3</sup> Both advantages and challenges taken from [2] Ibid.

of the system in ellipsoids. We were hoping that this technique would show that the estimation error under quantized innovations evolved under a Ricatti-like recursion—similar to that of a Kalman filter. However, as this does not seem to be the case and we were unable to use this technique to prove estimation error bounds, we then looked to analyze and develop new techniques for creating coder-estimator pairs and proving specific performance measures. After studying and simulating algorithms and techniques developed by others, we have begun studying various notions of stochastic stability to see if we can use the theory of Markov chains to prove performance specifications for Sukhavasi's filter.

## Discussion and Results

---

### Developing an Estimator for a Wireless Sensor Network

---

#### Quantizing the Measurement using the Sign-of-Innovation

---

The Kalman filter was developed in the 1960s and is a ubiquitous method of state estimation for a linear dynamic system in which both the noise and the conditional probability distributions of the states given the measurements are Gaussian. It is a recursive, minimum mean-square-error estimator that has the form of a linear observer, in which the observer gains are computed from error covariance matrices that are in turn calculated from a Riccati recursion equation (a type of difference equation). Kalman filters are popular because, not only are they optimal for Gaussian systems, they are computationally efficient and easy to implement. However, the Kalman filter cannot be used to track systems using severely constrained wireless sensor networks for a number of reasons. For one, under measurement quantization, the required Gaussian assumption no longer holds. More importantly, because the sensors are distributed, there is no central observer that has access to all of the full measurements so not even an observer can calculate the Kalman filter estimate—as has been done in other work on communication constrained control systems.

However, we can still use some of the principles used to develop the Kalman filter for a WSN. The Kalman filter is based on what is called the innovations process, which is the process of the error—the difference between the most current measurement and the controller’s estimate of what that measurement should have been given all previous measurements. The innovations process is a white process and can be “regarded as the ‘new information’ or the ‘innovation’ in the observation after we remove all we can say ... about it from knowledge of past observations.”<sup>4</sup> In [5], Ribeiro *et al.* used this concept for WSNs by assuming that all the sensors can receive the controller’s estimate information (thus, assuming that it requires less energy to receive than to transmit), so that, in the case of using a communication channel with a one bit rate, the one bit of information that is transmitted from the sensor is the sign of the innovation (SOI) from its new measurement.

However, using the SOI, Ribeiro *et al.* incorrectly assumed that the conditional probability density function of the state given the quantized measurement *approximately* follows a normal distribution and, thus, proceeded to develop a modified Kalman filter (the SOI-KF) with a corresponding modified Riccati recursion which varies from the original Kalman filter Riccati only by a  $2/\pi$  factor in front of the gain term, which accounts for the degree of quantization of the measurement.

Though Ribeiro *et al.* presented simulations of a couple of systems in which the SOI-KF *seems* to work, later work done by Sukhavasi has shown that their filter fails and has diverging error performance under many different systems.<sup>5</sup>

---

<sup>4</sup> From [4] T. Kailath, *et al.*, *Linear Estimation*. Upper Saddle River, NJ: Prentice Hall, 2000.

<sup>5</sup> Sukhavasi did this using a general particle filter as explained in his paper [6] R. T. Sukhavasi and B. Hassibi, "Particle filtering for Quantized Innovations," presented at the Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, 2009.



## The Kalman-Like Particle Filter

---

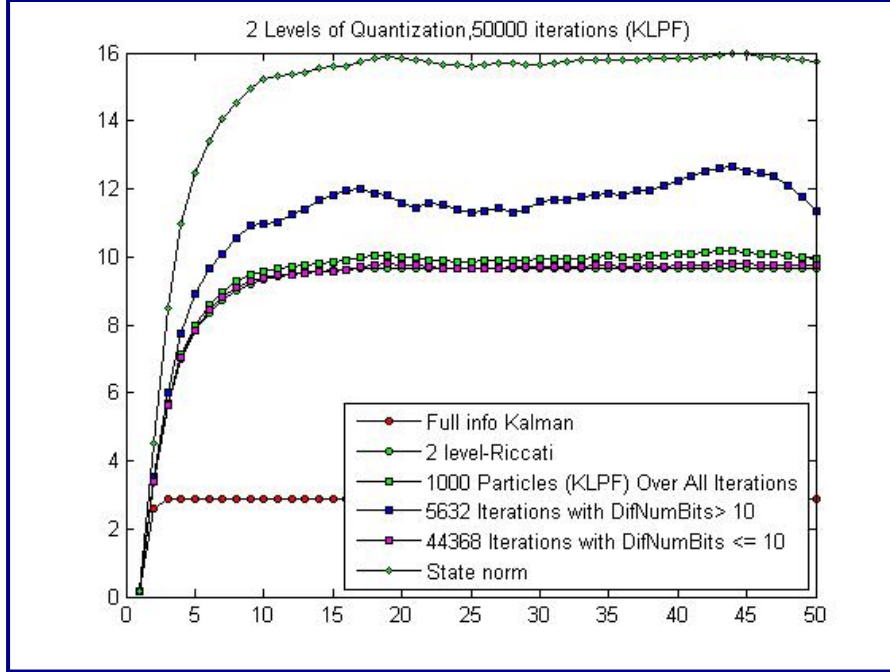
As a result, Sukhavasi developed another filter presented in [3], called the Kalman-Like Particle Filter (KLPF). The particle filter is used when the assumptions of the Kalman filter cannot be met. Instead of assuming a normal distribution, the particle filter represents a belief of a state in a nonparametric form by a set of randomly selected state samples. Each sample is called a particle and is essentially a single hypothesis about what the true state may be. An ‘importance factor’ gives a particular weight to each sample, representing using Bayes’ rule the probability of the state given the measurements. The weights are then used in a resampling process that has a greater probability of keeping particles with more weight.

Though optimal in the case of non-linear and non-Gaussian systems, particle filters can be very computationally expensive because they can require a huge number of particles. However, Sukhavasi developed a variant of the particle filter that greatly reduced the number of particles needed to optimally run the filter by breaking the probability model of the system into two parts—one that could be described as a Gaussian distribution and the other which had to be evolved using a particle filter. He noted that the original state-space model (without the measurement quantization) follows the Gaussian assumptions necessary to use a Kalman filter, while the one-bit quantization of the measurements creates nonlinearities that must be addressed by other estimation techniques. Thus, each particle addresses the skewed probability of the original measurement given its quantization, and the Kalman filter evolves the state estimate using the various guesses of the original (un-quantized) measurement. Therefore, this filter effectively runs a Kalman filter on each particle, where each particle represents a possible value of the original measurement based on the quantized measurement that is received.

## Simulating the KLPF with the SOI-KF

---

The KLPF has much better error performance than the SOI-KF developed in [5], but, oddly enough, in a number of simulations, the error performance of the KLPF seemed to track the Riccati recursion developed for the SOI-KF. Thus, we performed a large number of simulations on the KLPF, which varied the dynamics of the systems being analyzed, the number of particles, etc., to see how the error evolves in relation to the Riccati recursion used in [5]. It provided great insight on the effects of quantization on the estimation of the system. For example, one simulation showed that the error of the system is dependent on the bit sequence of measurements received; the plot below shows that the error is larger when the system continually overestimates or underestimates the state, thus returning more of one bit than the other. Finally, a number of other simulations (two are shown in KLPF Simulations, 21) were used to determine that the SOI-KF Riccati recursion did not in fact characterize the performance of the optimal filter in any way and could not serve as any kind of bound for the error of the KLPF.



**Figure 1: KLPF Simulation Plot.** Plot of the evolution of the error of a particular dynamical system averaged over 50,000 iterations, each using a particle filter with 1000 particles

### The Probabilistic Set-Membership Description

With the realization that there was no meaningful relationship between the SOI-KF Riccati and the evolution of the error of the KLPF, work shifted to trying to prove performance through a careful analysis of the original equations that described the system dynamics and quantized measurement updates. The performance of the KLPF could be verified if it could be shown that a marginally unstable system could be tracked with sufficiently bounded error when using only a sequence of bits, each representing the sign of the innovation.

To do this, an approach was tried that was inspired by work done by Schweppe and then Bertsekas in the late 60s ([7], [8]). Schweppe and Bertsekas derived recursive state estimations under a ‘set-membership description of uncertainty.’ The set-membership describes “input disturbances and observation errors [when they] are unknown except for the fact that they belong to given bounded sets.” [8] The sets they used were ellipsoids. They showed that intersecting ellipsoids, representing the bounds on the state both from the understanding of the state’s initial position and state evolution as well as from the measurement update, will result in an ellipsoid bounding the current state. This recursive process is not the best filter because the center of the resulting ellipsoid cannot be shown to be the optimal estimate of the state. However, if the ellipsoids remain bounded, it equivalently bounds the error in the estimate of the state of the system.

However, the difference with the system at hand and the one used by Schweppe and Bertsekas is that, here, the uncertainties are not bounded but are instead Gaussian. This created new difficulties because, unlike in the set-membership approach, the system can be ‘unlucky’ and get hit with large values of noise, though this, of course, happens with low probability.

Thus, in this system, the noise could only be bounded *probabilistically* within ellipsoids, which is equivalent to defining the variance of the noise using Gaussian confidence intervals.

Using Chernoff bounding techniques, a very similar recursive estimation process was developed for the probabilistic ellipsoids (shown in The Probabilistic Set-Membership Description, 22). With a recursion process at-hand, the goal became to show that, at each time step, an exponentially decaying probability bound held for the error measurement, which would then prove that the error covariance matrix remained bounded for all time. Using the state-dynamics equations, the probability bounds were evolved using the set-membership techniques to sum and intersect ellipsoids. Another main difference was that, instead of intersecting two ellipsoids to combine the bounds on the state based on its possible evolution from its initial position as well as from its measurement, in this case, the state-evolution ellipsoid could only be intersected with a half-plane that corresponded to the measurement bit received.

While this approach did give a method for evolving the probability bounds through time, the probability bound obtained at the end of a single recursion step was not tight enough to prove bounded error. Instead it had a very counterintuitive result, showing that the best estimate was obtained simply by evolving the mean estimate of the initial state and never using any of the quantized measurements. Thus, while simulations showed that the KLPF had bounded error, this technique could not verify it.

## Literature Review

---

Because the probabilistic set-membership approach cannot be used to develop a tight recursion for the system, it has neither shown that the KLPF can be used to track a system nor that it cannot be used. Therefore, still needing to develop techniques to prove performance for the KLPF, we analyzed previous research that had instead created various coder-estimator pairs to prove stabilizability for closed-loop control systems under communication constraints. One result in this area of work on networked control systems that is especially applicable to our system is called the data-rate theorem and is inspired by Shannon’s source coding theorem. It states that, to stabilize a scalar system, the rate of the communication channel must be greater than the *intrinsic entropy rate* of the system, which is the logarithm of the magnitude of its unstable pole:

$$R > \log_2 |\lambda|$$

Quite simply, this means that, to keep a bound on the estimation of the state, the rate used to communicate measurements for the estimation must be larger than the rate at which the system is growing.

We decided to analyze more closely two very different papers, each of which derives a specific data-rate theorem result for a particular system and communication channel. One is Minero’s “Data Rate Theorem for Stabilization over Time-Varying Feedback Channels.”[9] The other is Yuksel’s “A Random Time Stochastic Drift Result and Application to Stochastic Stabilization over Noisy Channels.”[10] The systems analyzed in these papers are very different from our system because their goal is not only to estimate the state but also to control it. The assumption these papers make that allows them to close the loop of the system with a controller is that the information pattern—the observer’s available inputs—of the system includes all full (un-quantized) measurements. While this is unlike the KLPF’s observer, which only has access

to the current measurement and the estimate from all passed quantized measurements, we hoped that the techniques presented would also be applicable to the KLPF.

### Minero's Data-Rate Theorem

Minero's formulation is for a system with Gaussian disturbances and uncertainties under a time-varying feedback channel. While Minero assumes that all bits sent over the channel are correctly received by the decoder without error, the rate at which the channel sends information changes randomly over time. However, because we might expect some correlation between the rates of consecutive time steps, Minero proposes a scenario in which the rate "remains constant in blocks of  $n$  consecutive channel uses and then varies according to an independent and identically distributed process across blocks." [9]

His result is that, for this particular system, the necessary and sufficient condition for stabilizability is that

$$E \left[ \left( \frac{\lambda^2}{2^{2R}} \right)^n \right] < 1$$

where  $n$  is again the length of the channel block in which the rate remains constant,  $R$  is the random variable rate, and  $\lambda$  is the eigenvalue (assumed to be unstable) of the scalar system.

The main differences with this result and the data-rate theorem for the scalar fixed-rate, non-erasure system are that now an expectation must be taken because  $R$  is a random variable. And, most importantly, that stabilizability depends on  $n$ . As the function  $f(x) = x^n$  is convex for  $n \geq 1$ , by Jensen's inequality ([11], Theorem 2.6.2)

$$[E \left( \frac{\lambda^2}{2^{2R}} \right)]^n \leq E \left[ \left( \frac{\lambda^2}{2^{2R}} \right)^n \right].$$

This means that, with  $n > 1$ , the necessary condition for stabilizability becomes harder to ensure. This makes intuitive sense if we consider the scenario in which  $n = 1000$  and there is a non-zero probability of having a rate of zero over a channel block.

Minero proves the necessity of this condition using an entropy-power inequality, which is a standard property used in information theory (See [11], Theorem 17.7.3). However, we were most interested in the paper's technique of proving sufficiency, which is an explicit construction of a quantizer and coder-decoder pair. To prove sufficiency, Minero relies on the property of an adaptive, exponential quantizer that has average quantization error that diminishes like the inverse square of the number of levels,  $2^{2v}$ , where  $v$  is the number of bits that can be used to partition the region.



Figure 2: Minero's quantizer is an adaptive, exponential quantizer. Picture taken from [12].

He also relies on the ability to use multiple time steps to transmit information about a single measurement so that the decoder's information about the measurement can be refined as much as necessary. This is done by using a fixed value  $\tau$  number of channel blocks (each with  $n$  channel uses in which the rate  $R$  remains constant) to send information about one measurement. Therefore, every  $n\tau$  steps (denoted by iterator  $j$ ), the decoder receives the measurement update and updates the estimator using  $v_j$  bits, where

$$v_j := nR_{j\tau} + nR_{j\tau+1} + \dots + nR_{(j+1)\tau-1}$$

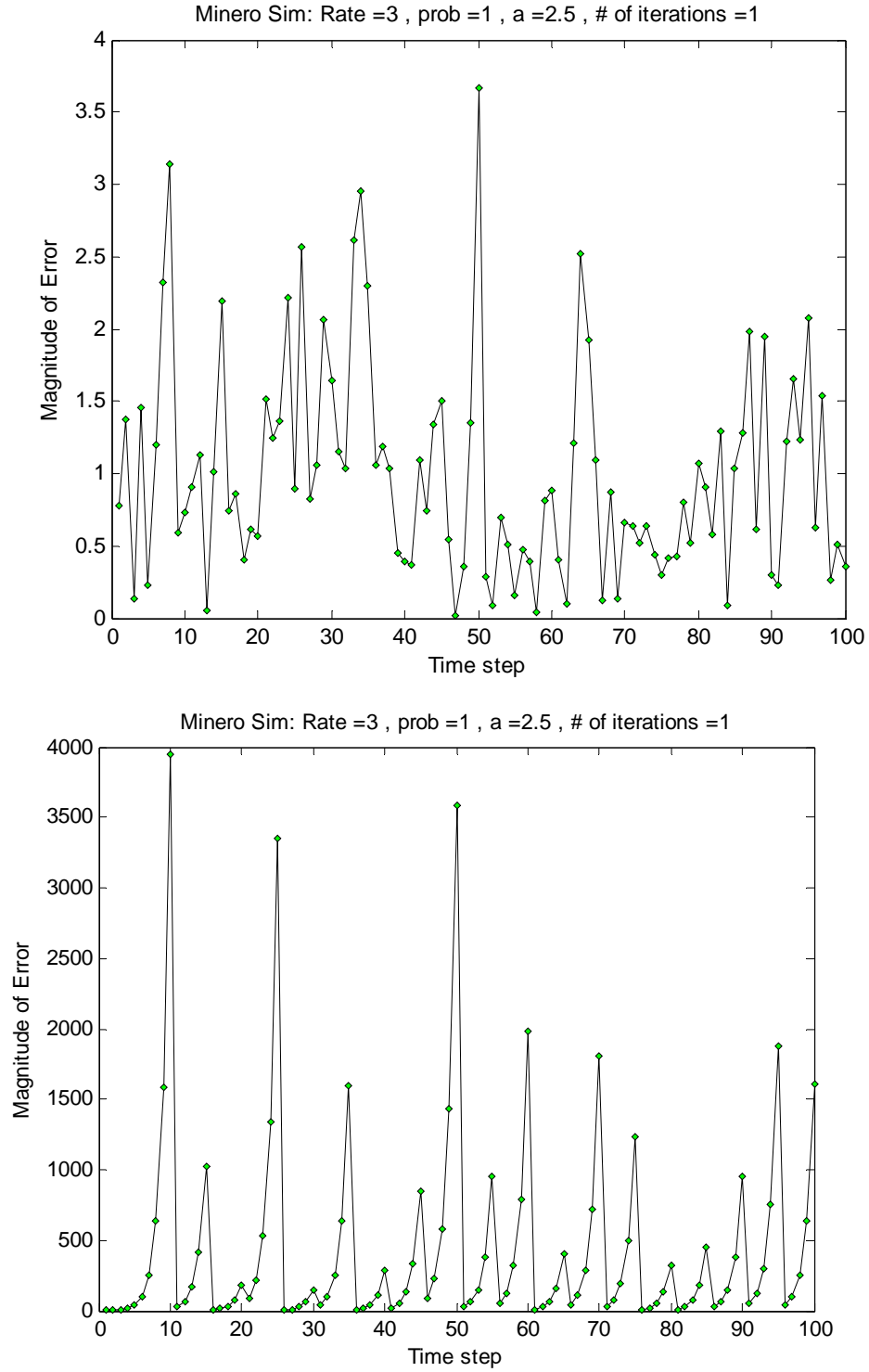
Hence, there is a tradeoff in selecting  $\tau$ : the smaller  $\tau$  is, the more measurements are transmitted to the decoder with less delay, and the greater  $\tau$  is, the more the decoder can refine the value of a single measurement.

### *Simulations of Minero's Algorithm*

---

While Minero's results did prove stabilizability for the system, we wanted to better understand the performance of the quantizer and encoder-decoder pair, so we simulated their algorithm on a number of various systems.

These simulations show the weakness of the algorithm's sufficiency proof in its reliance that  $\tau$  be "sufficiently large." As seen in the simulation below, because increasing  $\tau$  increases the delay, the errors increase in choosing  $\tau$  larger than absolutely necessary. Note, that in all simulations we have chosen  $\text{Prob}(R_i = 3) = 1$ , so that the rate is not actually time-varying. (More simulations of Minero's algorithms are found in Minero and Yuksel Simulations, 23.)



**Figure 3: Minero Simulations. In the top simulation,  $\tau = 1$ . In the bottom simulation,  $\tau = 5$ .**

Because Minero's approaches used rather standard control-theoretic and information-theoretic techniques and also did not account for any possible channel erasures, we also analyzed Yuksel's result in [10] for a stochastic communication channel that uses newer and what seem to be very powerful results from the theory of Markov chains on general spaces. Using notions of stochastic stability for Markov chains, Yuksel constructs a quantizer and encoder-decoder pair for estimating a state with measurements from a rate-constrained erasure channel and proves the filter's stabilizability.

Like Minero's work, Yuksel's formulation is for a scalar dynamical system with unstable mode  $a$ , Gaussian noise disturbance, and Gaussian initial condition. While constraining the data rate to  $R$ , Yuksel also allows for erasures of the form:

$$P(p_t = 1) = p$$

where  $0 < p \leq 1$  and  $p_t = 1$  is the event that there is no erasure and that the signal is transmitted with no error at time step  $t$ .

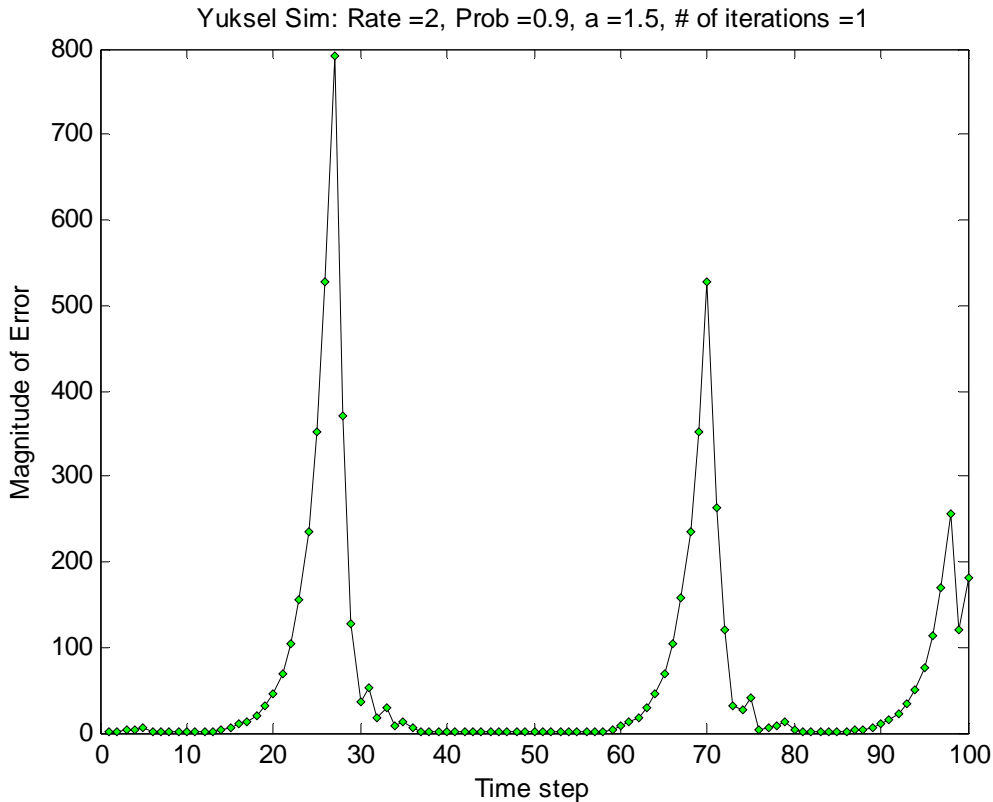
Yuksel's main result is that the necessary and sufficient condition for stabilizability of the plant is that

$$a^2 \left( 1 - p + \frac{p}{(2^R - 1)^2} \right) < 1$$

Notice that the result again follows the data-rate theorem: for  $p = 1$ , it reduces to the theorem and, for  $p = 0$ , the eigenvalue  $a$  must be stable.

To prove the sufficiency of this condition, Yuksel uses the language and techniques of Meyn and Tweedie [13]. (An overview of Markov chains and these techniques is presented below in Markov Chains and Various Notions of Stochastic Stability, 17.) The central feature of the algorithm that Yuksel develops is a random, adaptive quantizer that allows Yuksel to use Markov chain theory to show that the error will continually return to a bounded region (thus, that the bounded region is a recurrent set of the Markov chain describing the evolution of the estimation error). Yuksel's quantizer is actually incredibly simple: he uses a uniform quantizer with a bin size that is updated at each time step. Yet, it is this update that is the key to ensuring stabilizability; the update algorithm is chosen to ensure that the joint Markov process is recurrent with a unique invariant distribution.

While Yuksel does prove stabilizability of his system, the biggest problem with it is that it does no more than that. Thus, even systems with values that well satisfy the sufficiency condition can produce very large estimation errors at particular time steps. An example simulation is shown below. (For more simulations, look at Minero and Yuksel Simulations, 23.)



**Figure 4: Yuksel Simulation.** Although the unstable mode only has a value of 1.5 and there is only a 0.1 probability of erasure, because the rate has been constrained to two bits, there are particular time instances with high estimation error.

However, although the algorithm is not of practical use, Yuksel's result remains very important because it suggests new techniques for approaching the problem. Yuksel's work shows that the general theory of Markov chains can be used as a powerful tool for understanding and deriving stochastic control systems.



The last portion of our work has focused on better understanding the techniques used by Yuksel in [10] that are presented in Meyn and Tweedie [13]. Here is given a very brief introduction to Markov chains and some of the results and ideas presented in [13] that are most important to our current research.

First of all, a homogenous Markov chain is a particular type of random process that obeys the following very ‘nice,’ useful properties:

- (1) **Memoryless (Markov) property:** The future state of the chain depends only on the current state of the chain and not on all its past states. Or more formally, the “conditional distribution of  $X_{n+1}$  given  $(X_0, \dots, X_n)$  depends only on  $X_n$ .” ([14])
- (2) **Time homogeneity:** The conditional probability of a state given a particular value for the current state is the same for all time.

Examples of Markov chains include a random walk, a queuing system, and, of course, the state-space dynamical system that we have been analyzing. While the theory of Markov chains has existed since Markov’s pioneering work in the early 1900s, most of the work has focused on chains in countable spaces, that is, under the assumption that the state  $X$  can only take values from a countable set. Yet, this is clearly not the case for the Markov chain of the estimation error in our system. However, beginning in the 1950s with theories derived by Harris, it was shown that similar results could also be developed for chains on more general spaces. The theory of Markov chains on general spaces relies primarily on measure theory; instead of analyzing a collection of particular values of a countable set, we analyze sets of nonzero measure.

A Markov chain for a general space is defined by its transition probability kernel  $P(x, A)$ , where, as defined by Meyn & Tweedie,

- (i) For each  $A \in B(X)$ ,  $P(\cdot, A)$  is a non-negative measurable function on  $X$  and
- (ii) For each  $x \in X$ ,  $P(x, \cdot)$  is a probability measure on  $B(X)$

where  $B(X)$  is a “countably generated” sigma field on  $X$ . A time-homogenous Markov chain  $\Phi$  then satisfies the following equation for each  $n$ :

$$P_\mu(\Phi_0 \in A_0, \Phi_1 \in A_1, \dots, \Phi_n \in A_n) = \int_{y_0 \in A_0} \dots \int_{y_{n-1} \in A_{n-1}} \mu(dy_0) P(y_0, dy_1) \dots P(y_{n-1}, A_n)$$

where  $\mu$  is the initial distribution of the chain. (From [13], pages 59-61)

Meyn and Tweedie’s rigorous presentation of Markov chains is based on developing notions of stochastic stability. They write, “We will systematically develop a series of increasingly strong levels of communication and recurrence behavior within the state space of a Markov chain, which provide one unified framework within which we can discuss stability.” ([13], page 14) This communication that they refer to is the ability for the chain to reach, at some point in time, a set  $A$  of nonzero measure. This communication is described by requiring that the hitting time of set  $A$  is less than infinity:

$$\tau_A := \inf(n \geq 1 : \Phi_n \in A)$$

chain communicates with  $A$ :  $\tau_A < \infty$

In this framework, the goal of a controller is to ensure that the set  $A$ , in which the estimation error is bounded, is continually visited by the Markov chain that describes the

process. The weakest notion of stability in Meyn and Tweedie's approach is thus that of  $\psi$ -irreducibility, which is the property that sets of nonzero measure (as measured by the function  $\psi$ ) can be reached by the chain from every possible starting point:

$$\forall x \in X, \psi(A) > 0 \rightarrow P_x(\tau_A < \infty) > 0$$

But, we want to ensure that not only is the set  $A$  visited, but that it is visited infinitely often, which is the property of recurrence and, because of the property of irreducible Markov chains, is equivalent to requiring that the set  $A$  is reached in finite time:

$$\forall x \in X, \psi(A) > 0 \rightarrow E_x[\tau_A] < \infty$$

The strongest notion of stability that Meyn and Tweedie develop is ergodicity, which requires that the chain have "nice" limiting behavior in which its distribution converges to a limiting invariant distribution:

$$\pi(A) = \int \pi(dx)P(x, A)$$

Using these notions of stability, the goal of our work is then to show that the Markov chain described by the estimation error of the KLPF is recurrent with some invariant probability distribution. If we can show this, we will have equivalently shown that the set in which the estimation error remains small is continually visited and, therefore, that the KLPF can successfully track a given state under specified conditions.

## Conclusions

---

Although the Kalman-like Particle Filter (KLPF) exploits both the tractability of the Kalman filter and the optimality of particle filters and simulations have also shown good performance, we have still been unable to rigorously prove that it follows any performance specifications. However, in our attempts to prove stabilizability and other performance measures for the KLPF, we have determined that the optimal estimation error of a state under quantized measurements does not follow a Riccati-like recursion. We have also studied a number of results for control under communication constraints that have given us a toolbox of techniques and algorithms for approaching these problems.

While many of these techniques are founded on combining and modifying ideas and results from traditional information and control theories, Yuksel's approach using Markov chains in [10] relies primarily on analyzing the probabilistic nature of the control system. This very general technique that uses the theory of stochastic stability offers great potential, and further work needs to be done to use these types of techniques to prove the performance of the KLPF.

As theoretical work progresses and derives algorithms which provably satisfy stricter error bounds, our hope is that filters with verifiable performance specifications can be systematically designed for particular wireless sensors networks. This will allow us to develop energy-efficient networks that can monitor large multi-agent systems with possibly vast spatial distributions.

## Methods

---

### Problem Formulation and Sign of Innovations

---

The state-space system being measured can be summarized mathematically as

$$\begin{aligned} x(n+1) &= Ax(n) + w(n) \\ y(n) &= Hx(n) + v(n) \end{aligned}$$

where  $x(n)$  is the state evolving under a constant matrix  $A$  with additive Gaussian noise  $w(n) \sim N(0, Q)$  and  $y(n)$  is the measurement of the state also under additive Gaussian noise,  $v(n) \sim N(0, R)$ . While the original Kalman filter optimally estimates the filter by recursively using the measurements  $y_{1:n}$ , the goal of this research is to show that the system can be tracked under much harsher conditions—when the estimator only receives a bit sequence  $b_{1:n}$  from the sensors, where

$$b(n) = \text{sign}[y(n) - \hat{y}(n|n-1)]$$

the sign of the innovation of the measurement (the difference in the measurement and the estimate of the measurement given all previous measurements).

Our primary goal is to devise a filter under these conditions which ensures that the error of the system remains bounded for all time, or that

$$\sup_{k \in \mathbb{N}} E \left[ (x(k) - \hat{x}(k))^2 \right] < C,$$

where  $C$  is some constant.

### The KLPF

---

In [3], Sukhavasi shows that the probability distribution of the system is as follows:

Probability distribution of state conditioned on quantized measurement	
$x(n)   (B_{1:n} = b_{1:n}) \sim \underbrace{N(0, R_\Delta)}_{\text{Normal distribution}} + \underbrace{R_{xy}(n)(R_y(n))^{-1}(Y_{1:n}   B_{1:n} = b_{1:n})}_{\text{Truncated normal distribution}}$	
$R_y(n) \equiv   y(n)  ^2$	
$R_{xy}(n) \equiv \langle x(n), y(n) \rangle$	
$R_\Delta \equiv R_x(n) - R_{xy}(n)(R_y(n))^{-1}R_{yx}(n)$	

Figure 5: Probability distribution of state for KLPF

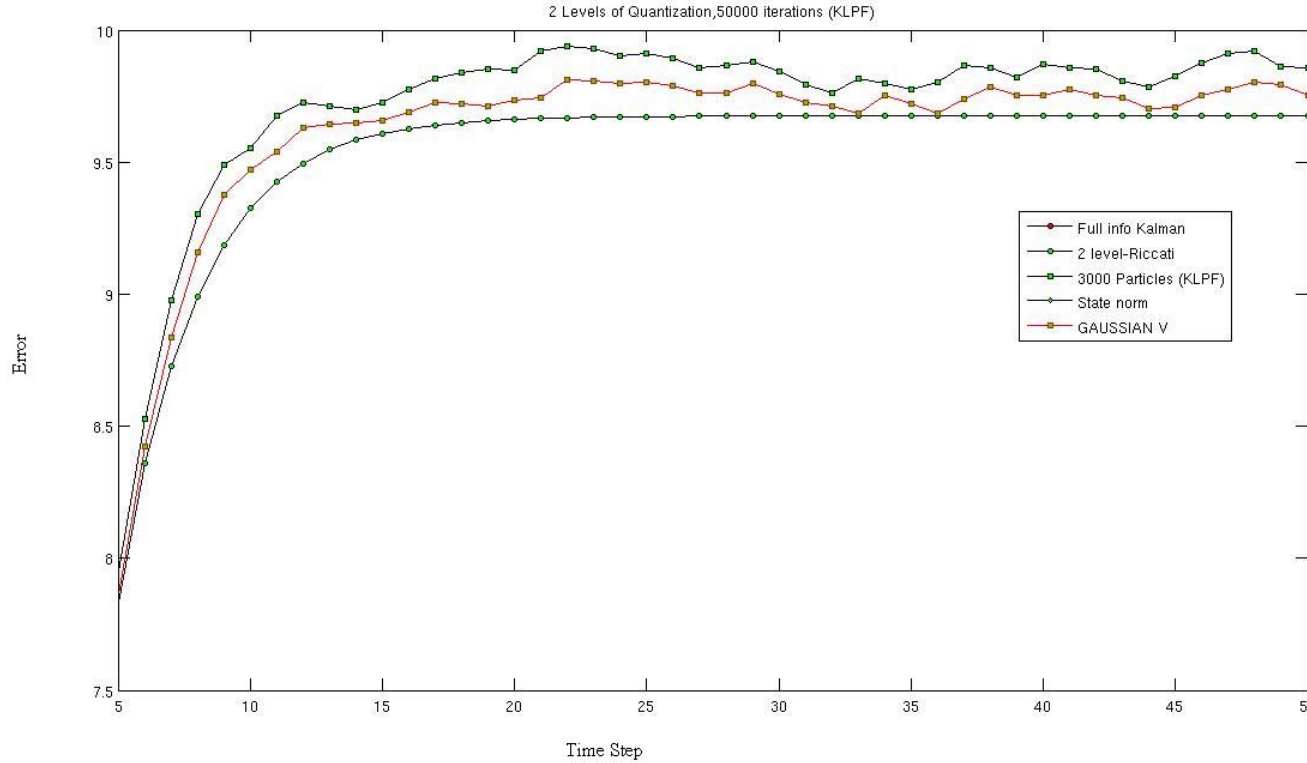
Thus, in the KLPF, the truncated normal distribution of  $y_{1:n} | b_{1:n}$  is estimated using a particle filter, while a Kalman filter evolves  $x(n) | y_{1:n}$  for each particle  $\{y_{1:n}^i\}_{i=1, \dots, N}$ .

## KLPF Simulations

Using Monte Carlo simulations of the KLPF with 50,000 iterations each with 3,000 particles, systems were found in which the KLPF Riccati was above and below the SOI-KF Riccati. Therefore, it was concluded that the SOI-KF Riccati has no significance and does not characterize or bound the error of the optimal filter. (For more detailed descriptions of the KLPF and the SOI-KF refer to [3] and [5], respectively.):

- System 5: In system 5, the KLPF error is higher than that assumed by the SOI-KF Riccati.

$$R = 10; \tau = 0.1; A = \begin{bmatrix} 1.1 & 45 \\ 0 & 1 \end{bmatrix}; Q = \begin{bmatrix} \frac{\tau^4}{4} & \frac{\tau^3}{2} \\ \frac{\tau^3}{2} & \tau^2 \end{bmatrix}; H = [1 \ 0]; P_0 = \begin{pmatrix} .01 & 0 \\ 0 & .01 \end{pmatrix}$$



**Figure 6: Simulation of KLPF in which error is higher than the SOI-KF Riccati.**

- System 12: However, in system 12, the KLPF has lower error than the SOI-KF.

$$R = 2.5; A = \begin{bmatrix} 0.95 & 1 & 1 & 0 & 0 \\ 0 & 0.9 & 7 & 1 & 0 \\ 0 & 0 & 0.6 & 2 & 0 \\ 0 & 0 & 0 & 0.7 & 0.8 \\ 0 & 0 & 0 & 0 & 0.5 \end{bmatrix}; Q = 2 * I; H = [1 \ 0 \ 1 \ 0 \ 2]; P_0 = .01I$$

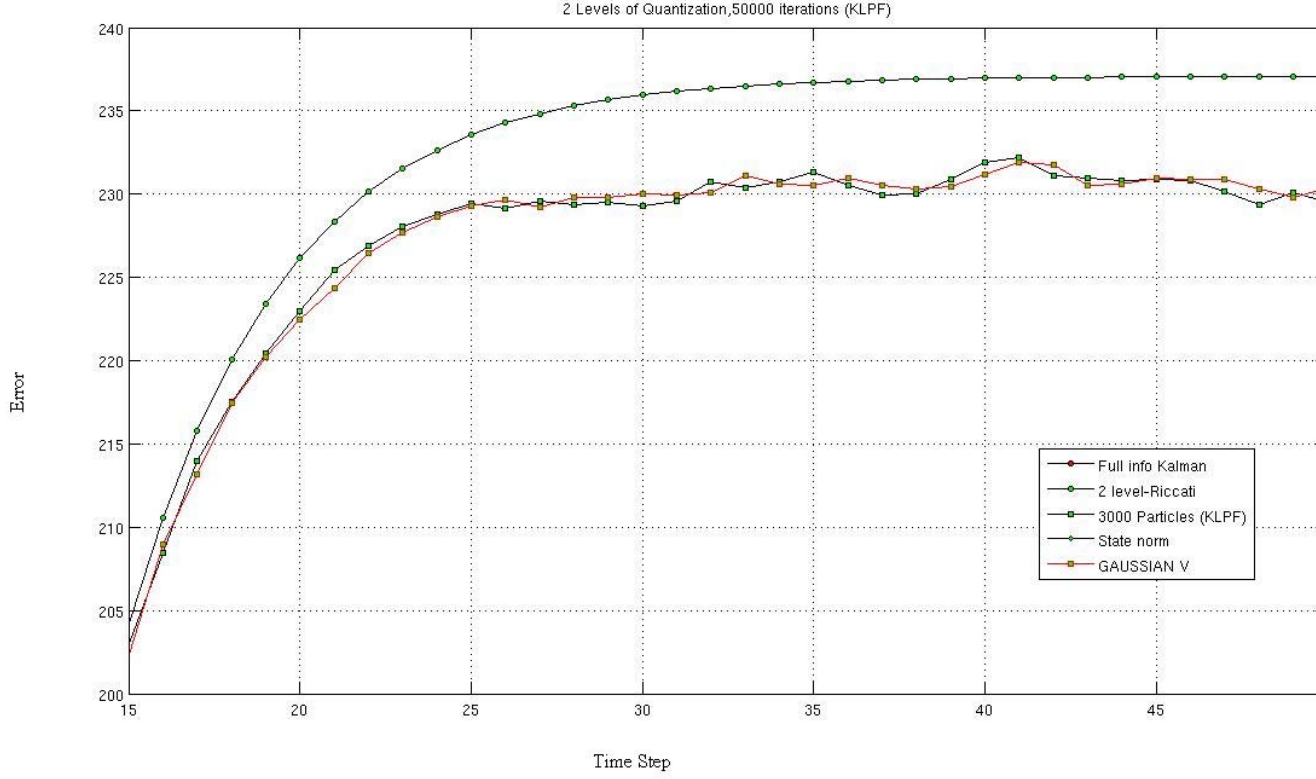


Figure 7: Simulation of KLPF in which error is lower than the SOI-KF Ricatti

## The Probabilistic Set-Membership Description

If, for a multidimensional random variable  $X$ ,

$$P(X^T Q^{-1} X \geq m) \leq \alpha e^{-\frac{m}{2}}$$

then its covariance matrix is finite (Appendix A.1). Thus, we want to recursively show that, at each time step, this exponentially decaying probability bound held for  $X = x(n) - \hat{x}(n|n)$ .

The recursion evolved with the following initial conditions, which were proven using Chernoff bounding techniques on the original Gaussian noise assumptions:

$$P[x^T(0)x(0) \geq m] \leq \left(\frac{me}{k}\right)^{\frac{k}{2}} e^{-\frac{m}{2}}$$

$$P[v^2 \geq n\sigma_v^2] \leq 2e^{-\frac{n}{2}}$$

$$P[w^T w \geq q] \leq \left(\frac{qe}{k}\right)^{\frac{k}{2}} e^{-\frac{q}{2}}$$

where now, without loss of generality,  $x(0) \sim N(0, I)$  is the initial position of the  $k$ -dimensional state,  $v \sim N(0, \sigma_v^2)$  is the measurement noise, and  $w \sim N(0, I)$  is the state evolution noise. Using the state-dynamics equations, the probability bounds on the state were evolved using the set-membership techniques to sum and intersect ellipsoids. The main difference is that instead of intersecting two ellipsoids to combine the bounds on the state based on its possible evolution from its initial position as well as from its measurement, the state-evolution ellipsoid was intersected with a half-plane that corresponded to the measurement bit,  $b(n)$ , received.

In this manner (Appendix A.2), starting with a probability bound

$$P[(x(t) - \hat{x}(t|t-1))^T P^{-1}(t)(x(t) - \hat{x}(t|t-1)) \geq m] \leq \alpha m^{\frac{k}{2}} e^{-\frac{m}{2}}$$

after one step of the measurement update recursion the probability bound was

$$P[(X - bz\sqrt{n})^T \tilde{P}^{-1}(t)(X - bz\sqrt{n}) \geq n] \leq 1 - (1 - \alpha m^{\frac{k}{2}} e^{-\frac{m}{2}})(1 - 2e^{-\frac{n}{2}})$$

where

$$\begin{aligned} X &= x(t) - \hat{x}(t|t-1) \\ z &= \rho\sqrt{R} \frac{(\theta_p - 1)}{2} P_p H^T \\ P_p &= P - \frac{P H^T H P}{\rho^{-1} + H P H^T} \\ \tilde{P} &= P_p \left( \theta + \rho R \theta_p + \rho^2 R \frac{(\theta_p - 1)^2}{4} H P_p H^T \right) \\ \theta_p &= \sqrt{\frac{H P H^T}{H Q H^T}}, m = n\theta \end{aligned}$$

But here there are some problems. The ellipsoidal bound on the state  $x(t|t)$  has a moving center; the center depends on  $n$ , which is simply the free variable that describes the probabilistically decaying bound. To prove a finite covariance, the center of the ellipsoid must be fixed. However, when trying to fix the center by any arbitrary shift corresponding to the bit received, the resulting covariance matrix is no smaller than if the estimate had not been updated based on the new measurement and was simply kept at the center at  $\hat{x}(t|t-1)$ . Thus, this recursion does not track the system. Thus, this current analysis cannot be used to prove bounded estimation error because the bounds it creates are not tight enough to show that the error covariance remains finite.

## Minero and Yuksel Simulations

---

Presented below are results from a few simulations of both Minero's and Yuksel's algorithms. The code used to run the simulations is presented in Appendix C: Minero Simulation Code, 30 and Appendix D: Yuksel Simulation Code, 35.

To compare the algorithms, the same parameter values have been used for both.

### Comparison of Minero's and Yuksel's Algorithm with no Erasures

Here results are averaged over 1000 iterations.

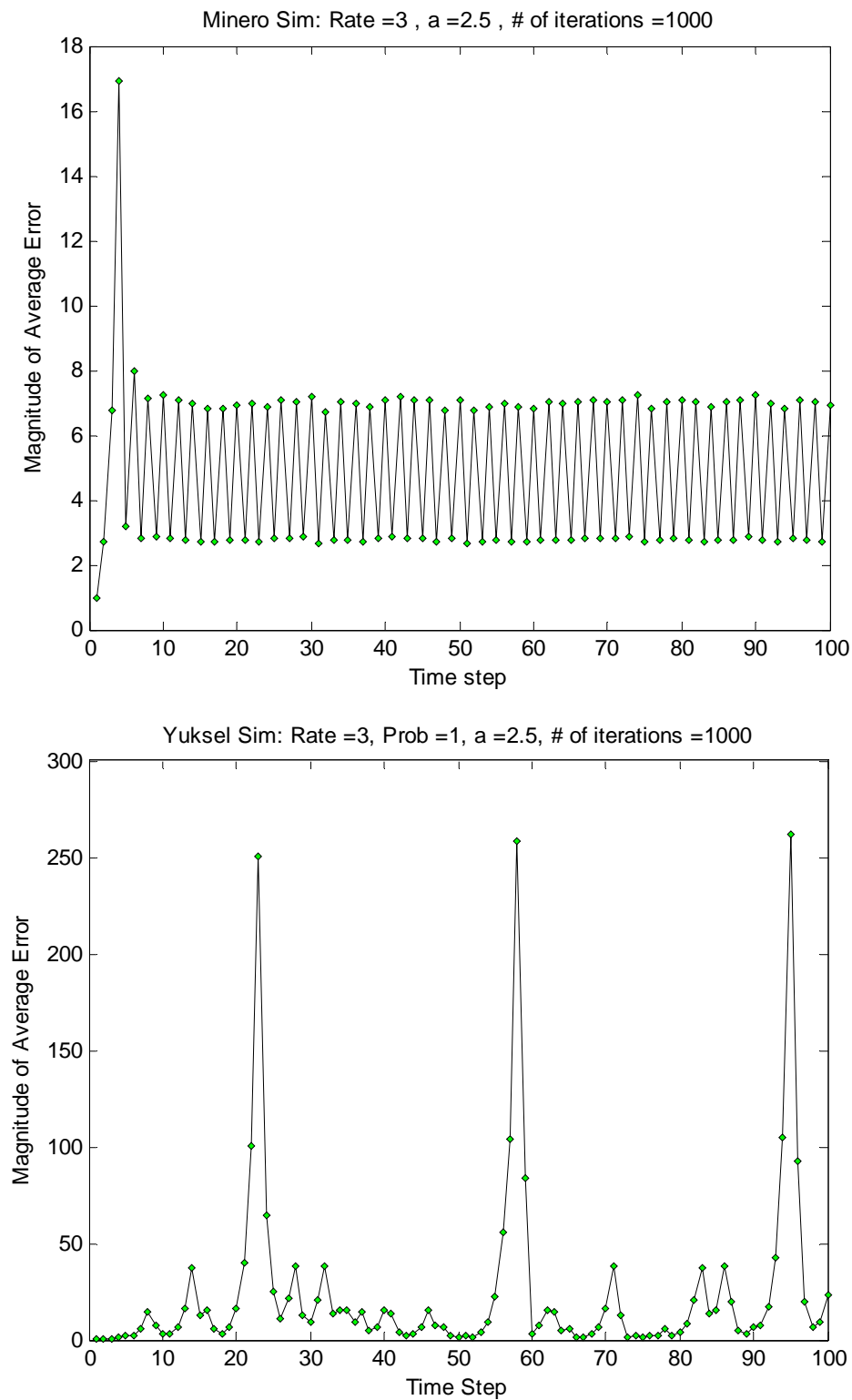
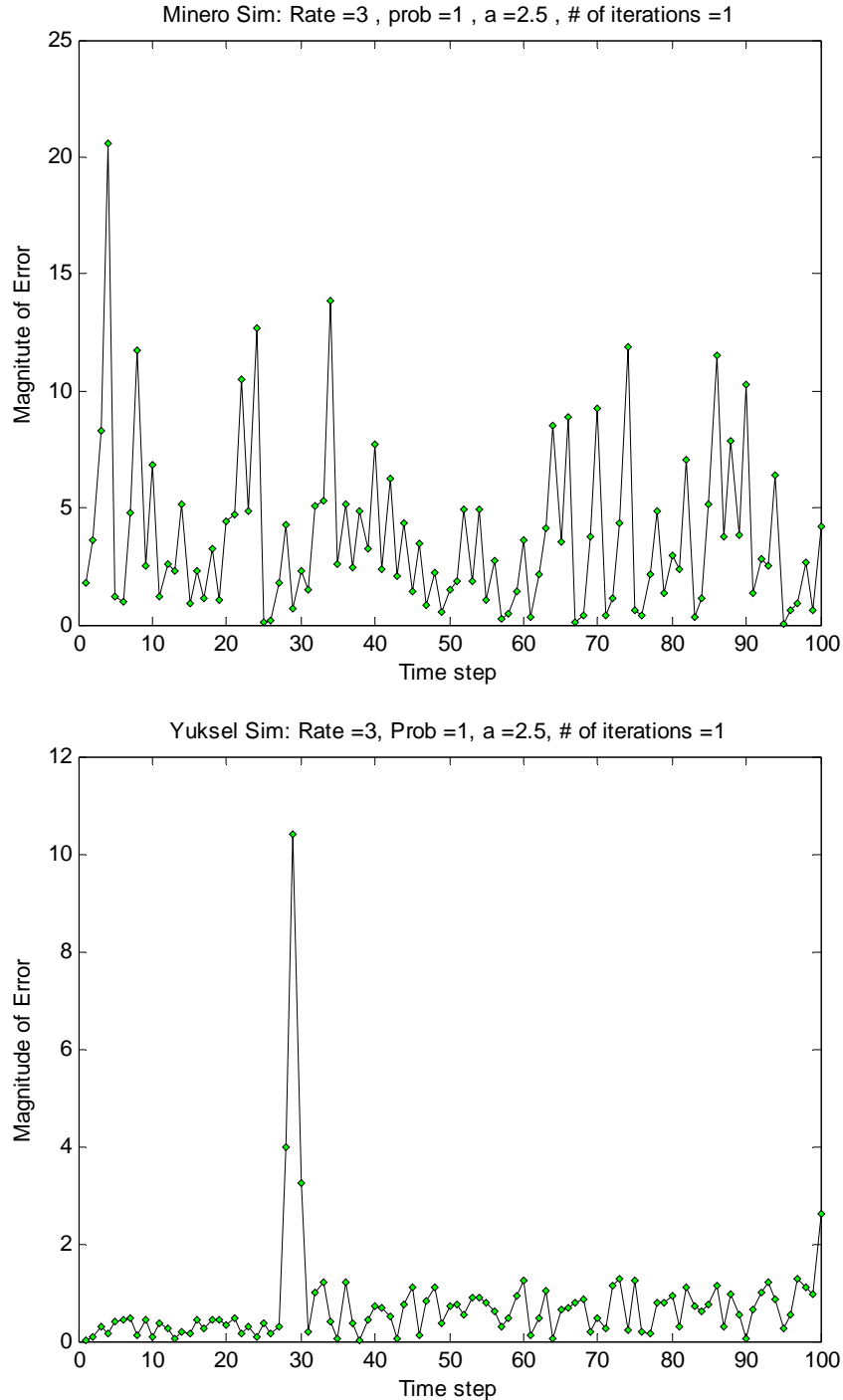


Figure 8: Comparison of Minero and Yuksel with no Erasures. In the top simulation of Minero's algorithm, the minimum sufficient  $\tau$  of value two is used.



### **Comparison of Yuksel's and Minero's Algorithm with no Erasures: No Averaging**

In the graphs below, only a single random iteration of the simulation was run. Thus, the averaging of Minero's errors much more clearly shows when each measurement update was made (at every other time step, because of the choice of  $\tau = 2$ ). For Yuksel's simulation, notice that times of high estimation error occur infrequently over a single iteration.



**Figure 9: Comparison of Minero and Yuksel with no Erasures with no averaging. In the top simulation of Minero's algorithm, the minimum sufficient  $\tau$  of value two is used.**

### Comparing the effects of erasures on Yuksel's simulation

Below are two graphs showing results from simulating Yuksel with a 0.2 probability of erasure. The first graph averages over 1000 iterations, while the second graph shows the outcome of a single iteration.

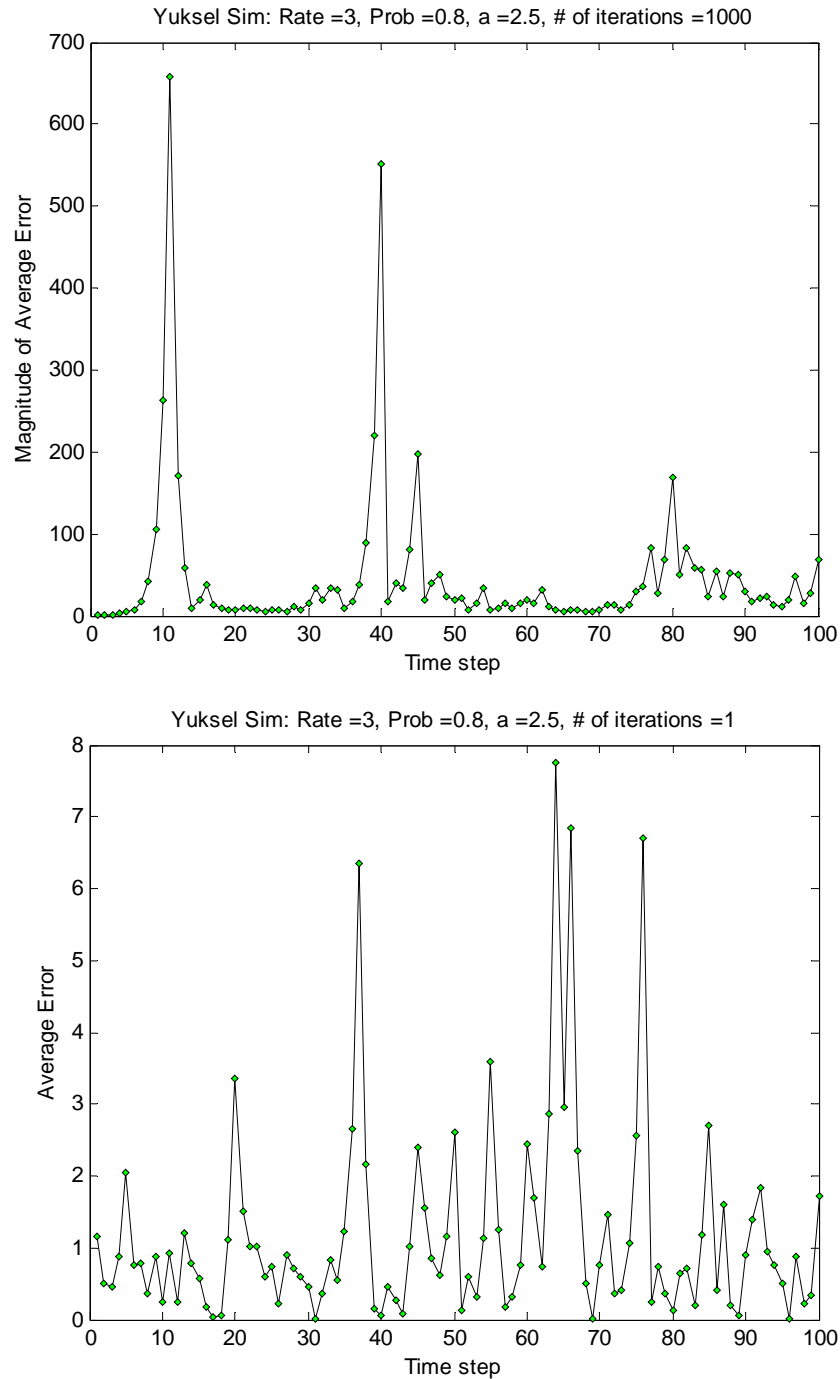


Figure 10: Yuksel simulation with 0.2 probability of erasure.

## Appendices

---

### Appendix A: Probabilistic Set-Membership Description Approach Lemma

---

If, for a multidimensional random variable  $X$ ,

$$P(x^T Q^{-1} X \geq a) \leq C e^{-a}$$

then its covariance matrix is finite.

This can be shown by writing  $Q^{-1}$  as a diagonal matrix: Since  $Q^{-1}$  is a real, symmetric matrix, it can be diagonalized by an orthogonal matrix  $A$  ( $A^T A = A A^T = I$ ).

$$A^T Q^{-1} A = D, \text{ where } D \text{ is a diagonal matrix}$$

$$Q^{-1} = I Q^{-1} I = (A A^T) Q (A A^T) = A D A^T$$

$$\rightarrow P(x^T A D A^T x \geq a) \leq C e^{-a}$$

$$y = A^T x$$

$$\rightarrow P(y^T D y \geq a) \leq C e^{-a}$$

Since  $D$  is the diagonalization of a positive definite matrix, its eigenvalues are positive. Thus, for  $i = 1, \dots, n$ ,

$$P(c_i y_i \geq a) \leq C e^{-a}$$

We have thus reduced the problem to proving this for the one-dimensional case---that the variance of each  $y_i$  is finite, as

$$E(x x^T) = Q E(y y^T) Q^T$$

For the one-dimensional case, we have:

$$\begin{aligned} P\left(\frac{(x-\mu)^2}{\sigma^2} \geq r^2\right) &\leq C e^{-r^2} \\ &= P(x - \mu \geq \sigma r) + P(x - \mu \leq -\sigma r) \\ &\rightarrow F_x(\mu - \sigma r) + 1 - F_x(\mu + \sigma r) \leq C e^{-r^2} \end{aligned}$$

Using this, we can easily show that  $\int (x - \mu)^2 p(x) dx$  is bounded. ■

### Appendix B: Probabilistic Set-Membership Description Approach Recursion Attempt

---

Optimizing measurement update ellipsoid when  $y$  is scalar.

Initial conditions:

$$P[(x(t) - \hat{x}(t))^T P^{-1}(t) (x(t) - \hat{x}(t)) \geq m] \leq a e^{-\frac{m}{2}}$$

$$y(t) = Hx(t) + v(t)$$

$$P[|v| \geq \sqrt{nR}] \leq n e^{-\frac{n}{2}}$$

Thus,

$$P[(Hx(t) - H\hat{x}(t))^T (HP(t)H^T)^{-1} (Hx(t) - H\hat{x}(t)) \geq m] \leq a e^{-\frac{m}{2}}$$

(1) scalar:

$$P\left[\frac{(Hx(t) - H\hat{x}(t))^2}{HP(t)H^T} \leq m\right] \geq 1 - a e^{-\frac{m}{2}}$$

Measurement:

▪  $b=1$ :

$$y(t) > \hat{y}(t) \rightarrow Hx(t) > \hat{y}(t) - v(t)$$

$$P[Hx(t) - \hat{y}(t) > -v(t)] = 1$$

$$P[|v| \geq \sqrt{nR}] \leq ne^{-\frac{n}{2}}$$

$$P[-\sqrt{nR} \leq v \leq \sqrt{nR}] \geq 1 - ne^{-\frac{n}{2}}$$

$$A_1 \cap A_2 \rightarrow A_3 \rightarrow P(A_3) \geq P(A_1 \cap A_2)$$

$z > -v(t) \& -\sqrt{nR} \leq v \leq \sqrt{nR}$  implies that:

$$z > -\sqrt{nR}$$

$$P[Hx(t) > \hat{y}(t) - \sqrt{nR}] \geq 1 - ne^{-\frac{n}{2}}$$

▪  $b=-1$ :

$$y(t) < \hat{y}(t) \rightarrow Hx(t) < \hat{y}(t) - v(t)$$

$$P[Hx(t) - \hat{y}(t) < -v(t)] = 1$$

Similarly, we can conclude for  $b = -1$ :

$$P[Hx(t) < \hat{y}(t) + \sqrt{nR}] \geq 1 - ne^{-\frac{n}{2}}$$

Thus, the general observation is:

$$(2) P[bHx(t) > b\hat{y}(t) - \sqrt{nR}] \geq 1 - ne^{-\frac{n}{2}}$$

Combining (1) and (2):

$$(1) -\sqrt{MHP(t)H^T} < bHx(t) - bH\hat{x}(t) < \sqrt{MHP(t)H^T}$$

$$(2) b\hat{y}(t) - bH\hat{x}(t) - \sqrt{nR} < bHx(t) - bH\hat{x}(t)$$

Measurement is useful if:

$$b\hat{y}(t) - bH\hat{x}(t) - \sqrt{nR} > -\sqrt{mHP(t)H^T}$$

Assume that we are filtering:  $\hat{y}(t) = H\hat{x}(t)$

Thus, the measurement is useful if:  $\sqrt{nR} < \sqrt{mHP(t)H^T}$

$$\rightarrow \frac{m}{n} > \frac{R}{HP(t)H^T}$$

Since  $P(t|t-1) = \alpha(AP(t-1|t-1)A^T + Q) > Q$ , choose

$$(3) \frac{m}{n} = \frac{R}{HQH^T} > \frac{R}{HP(t)H^T}$$

Finally, we have:

$$-\sqrt{nR} < bHx(t) - b\hat{y}(t) < \sqrt{mHP(t)H^T}$$

Thus, the new ellipse is:

Radius:

$$\frac{\sqrt{mHP(t)H^T} + \sqrt{nR}}{2}$$

Mean:

$$\frac{b(\sqrt{mHP(t)H^T} - \sqrt{nR})}{2}$$

Assume that  $m > n$  so that  $ne^{-\frac{n}{2}}$  is the dominant term.

Define:

$$m = n\theta; \theta = \frac{R}{HQH^T}$$

$$\theta_p = \sqrt{\frac{HPH^T}{HQH^T}}$$

This gives our new ellipsoidal bound:

$$(Hx - \hat{y} - \frac{b}{2}\sqrt{nR}(\theta_p - 1))^2 \leq \frac{1}{4}nR(\theta_p + 1)^2$$

We now need to intersect the original bound for  $x$  with the new bound from the measurement using a linear combination of the two bounds

$$1 \times (x - \hat{x})^T P^{-1} (x - \hat{x}) \leq m$$

$$((x - \hat{x})^T \quad 1) \begin{pmatrix} P^{-1} & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} x - \hat{x} \\ 1 \end{pmatrix} \leq m$$

$$\rho \times (Hx - \hat{y}(t) - \frac{b}{2}\sqrt{nR}(\theta_p - 1))^2 \leq \frac{1}{4}nR(\theta_p + 1)^2$$

$$\rho \times ((x - \hat{x})^T \quad 1) \begin{pmatrix} H^T H & -\frac{b}{2}\sqrt{nR}(\theta_p - 1)H^T \\ -\frac{b}{2}\sqrt{nR}(\theta_p - 1)H & \frac{nR}{4}(\theta_p - 1)^2 \end{pmatrix} \begin{pmatrix} x - \hat{x} \\ 1 \end{pmatrix} \leq \frac{nR(\theta_p + 1)^2}{4}$$

Adding together:

$$((x - \hat{x})^T \quad 1) \begin{pmatrix} P^{-1} + \rho H^T H & -\frac{\rho b}{2}\sqrt{nR}(\theta_p - 1)H^T \\ -\frac{\rho b}{2}\sqrt{nR}(\theta_p - 1)H & \frac{\rho nR}{4}(\theta_p - 1)^2 \end{pmatrix} \begin{pmatrix} x - \hat{x} \\ 1 \end{pmatrix} \leq m + \frac{\rho nR(\theta_p + 1)^2}{4}$$

Giving, after some algebraic manipulation:

$$\begin{aligned} & \left( x - \hat{x} - \frac{\rho b}{2}\sqrt{nR}(\theta_p - 1)(P^{-1} + \rho H^T H)^{-1}H^T \right)^T (P^{-1} + \rho H^T H)(...) \\ & \leq m + \rho nR\theta_p + \frac{\rho^2 nR}{4}(\theta_p - 1)^2 H(P^{-1} + \rho H^T H)^{-1}H^T \end{aligned}$$

The bound can be rewritten as :

$$n\left(\frac{m}{n} + \rho R\theta_p + \frac{\rho^2 R}{4}(\theta_p - 1)^2 H(P^{-1} + \rho H^T H)^{-1}H^T\right)$$

This gives us finally a bound of the form:

$$P \left[ (x - \hat{x}(t|t))^T P^{-1}(t|t)(...) \leq n \right] \geq \left(1 - me^{-\frac{m}{2}}\right) \left(1 - ne^{-\frac{n}{2}}\right)$$

with

$$\hat{x}(t|t) = \hat{x}(t|t-1) + \frac{\rho b(t)}{2}\sqrt{nR}(\theta_p - 1)(P^{-1} + \rho H^T H)^{-1}H^T$$

$$P(t|t) = (P^{-1} + \rho H^T H)^{-1}(\theta + \rho R\theta_p + \frac{\rho^2 R}{4}(\theta_p - 1)^2 H(P^{-1} + \rho H^T H)^{-1}H^T) \blacksquare$$

## Appendix C: Minero Simulation Code

---

### MineroSim.m

---

```
%Noele Norris
%Created: February 13, 2010
%Last Modified: April 29, 2010
%Simulate Coder-Decoder scheme in Minero's "Data Rate Theorem for
%Stabilization Over Time-Varying Feedback Channels"

%Currently: only implementing for fixed rate; can only use with scalar
%system though coding for vector system
%Also look at Nair's "Stabilizability of Stochastic Linear Systems with
%Finite Feedback Data Rates"

%inputs to function:
%   R : fixed rate
%   tau : number of channel uses for single measurement
%   p : rate of adaptive quantizer
%   T : number of time steps
%   montecarlo : # of iterations

R = 3;
tau = 5;
p = 2;
T = 100;
montecarlo = 1;

%System dynamics parameters
%Scalar system:  $x_{k+1} = \lambda x_k + u_k + v_k$ ;  $y_k = x_k + w_k$ 
%unstable system:  $|\lambda| \geq 1$ 
%   A : system dynamics matrix
%   Q : system disturbance covariance
%   H : measurement matrix
%   sigv2 : measurement noise covariance
%   P_orig : covariance of initial state  $x_0$ 
[A, Q, H, sigv2, P_orig] = System_params();
sigv2 = diag(sigv2);
sigv = sqrt(sigv2);

%Result: Mean square error
error_norm = zeros(1, T+1);
error_norm_kf = zeros(1, T+1);
state_norm = zeros(1, T+1);

%Kalman Riccati to be used in Kalman filter in coder
%Computes estimate covariance  $P(k|k)$  and Kalman gain

HP_kalH = zeros(1, T+1);
Kal_gain = zeros(1, T+1);

P_kal = P_orig;
```

```

for i=1:T+1
    %HP_kalH_quant(i) = diag(H*P_kal*H');

    HP_kalH(i) = H^2*P_kal;
    Kal_gain(i) = P_kal*H/(HP_kalH(i) + sigv2);
    %P_kal = P_kal*H(j,:)'*H(j,:)*P_kal/(HP_kalH(j, i) + sigv2(j));
    P_kal = (1 - Kal_gain(i)*H)*P_kal;

    P_kal = A^2*P_kal + Q;    %P(k+1|k) = AP(k|k)A^T + W(k)
end

%ITERATION
for iter = 1:montecarlo

    iter

    P = P_orig;
    %measurement noise
    V = normrnd(0, sigv, 1, T+1);

    %initialize realization of state
    X = zeros(1, T+1);
    mu_Xo = 0;
    X(1) = mvnrnd(mu_Xo, P)'; %Initializing state; X0 ~ N(0, P)

    %initialize realization of measurement
    Y = zeros(1, T+1);
    Z = zeros(1, T+1); %corresponding innovations
    Y(1) = H*X(1) + V(1);

    %initialize coder predictions (results of Kalman filter)

    Xkal_hat = zeros(1, T+1); %coder's prediction of state  $\hat{x}(k|k-1)$ 
    Xkal_updated = zeros(1, T+1); %coder's estimate of state  $\hat{x}(k|k)$ 

    X_hat = zeros(1, T+1); %decoder's prediction of state
    l = zeros(1, T+1); %scaling factor
    l(1) = A^tau; %initialization of scaling factor

    for t = 1:T+1

        %time update: realization of state and measurement with coder's
        %Kalman filtering update
        if t == 1
            Xkal_hat(t) = 0;
            X_hat(t) = 0;
        else
            u = -A*X_hat(t-1); %control input w/o quantization
            Xkal_hat(t) = A*Xkal_updated(t-1) + u;
            X(t) = A*X(t-1)+u + mvnrnd(0, Q)';
        end
    end
end

```

```

Y(t) = H*X(t) + V(t);

%decoder's estimate
if mod(t-1, tau) == 0
    %QUANTIZATION MEASUREMENT UPDATE
    z = Xkal_updated(t-tau)-X_hat(t-tau);
    [k,q] = Quantizer(R*tau, p, z/l(t+1-tau));
    X_hat(t) = pro(t)*A^(tau)*l(t+1-tau)*q;
    l(t+1) = max(l(1), k*l(t)*abs(A)^tau);
else
    l(t+1) = l(t);
    X_hat(t) = 0; %choose control input to bring state to zero
end
end

%measurement update
%calculate innovation
Z(t) = Y(t) -H*Xkal_hat(t);
Xkal_updated(t) = Xkal_hat(t) + Kal_gain(t)*Z(t);
end

error_norm = error_norm + sum((X - X_hat).^2, 1);
error_norm_kf = error_norm_kf + sum((X-Xkal_updated).^2,1);
state_norm = state_norm + sum(X.^2, 1);
end

error_norm = sqrt(error_norm/montecarlo);
error_norm_kf = sqrt(error_norm_kf/montecarlo);
state_norm = sqrt(state_norm/montecarlo);

time = 1:T;

figure;
plot(time, error_norm(1:T), '-kd','MarkerFaceColor','g','MarkerSize', 4);
title(strcat('Rate = ', num2str(R), ' , prob = ', num2str(prob), ' , a = ',
num2str(A) , ' , # of iterations = ' , num2str(montecarlo)));

```

---

## Quantizer.m

```

%Noele Norris
%Created: February 13, 2010
%Last Modified: April 29, 2010
%Simulate Coder-Decoder scheme in Minero's "Data Rate Theorem for
%Stabilization Over Time-Varying Feedback Channels"

%Currently:only implementing for fixed rate; can only use with scalar
%system though coding for vector system
%Also look at Nair's "Stabilizability of Stochastic Linear Systems with
%Finite Feedback Data Rates"

```



```

%inputs to function:
%   R : fixed rate
%   tau : number of channel uses for single measurement
%   p : rate of adaptive quantizer
%   T : number of time steps
%   montecarlo : # of iterations

R = 3;
tau = 5;
p = 2;
T = 100;
montecarlo = 1;

%System dynamics parameters
%Scalar system:  $x_{k+1} = \lambda x_k + u_k + v_k$ ;  $y_k = x_k + w_k$ 
%unstable system:  $|\lambda| \geq 1$ 
%   A : system dynamics matrix
%   Q : system disturbance covariance
%   H : measurement matrix
%   sigv2 : measurement noise covariance
%   P_orig : covariance of initial state  $x_0$ 
[A, Q, H, sigv2, P_orig] = System_params();
sigv2 = diag(sigv2);
sigv = sqrt(sigv2);

%Result: Mean square error
error_norm = zeros(1, T+1);
error_norm_kf = zeros(1, T+1);
state_norm = zeros(1, T+1);

%Kalman Riccati to be used in Kalman filter in coder
%Computes estimate covariance  $P(k|k)$  and Kalman gain

HP_kalH = zeros(1, T+1);
Kal_gain = zeros(1, T+1);

P_kal = P_orig;

for i=1:T+1
    %HP_kalH_quant(i) = diag(H*P_kal*H');

    HP_kalH(i) = H^2*P_kal;
    Kal_gain(i) = P_kal*H/(HP_kalH(i) + sigv2);
    %P_kal = P_kal*H(j,:)'*H(j,:)*P_kal/(HP_kalH(j, i) + sigv2(j));
    P_kal = (1 - Kal_gain(i)*H)*P_kal;

    P_kal = A^2*P_kal + Q;    % $P(k+1|k) = AP(k|k)A^T + W(k)$ 
end

%ITERATION
for iter = 1:montecarlo

```

```

iter

P = P_orig;
%measurement noise
V = normrnd(0, sigv, 1, T+1);

%initialize realization of state
X = zeros(1, T+1);
mu_Xo = 0;
X(1) = mvnrnd(mu_Xo, P)'; %Initializing state;  $X_0 \sim N(0, P)$ 

%initialize realization of measurement
Y = zeros(1, T+1);
Z = zeros(1, T+1); %corresponding innovations
Y(1) = H*X(1) + V(1);

%initialize coder predictions (results of Kalman filter)

Xkal_hat = zeros(1, T+1); %coder's prediction of state  $\hat{x}(k|k-1)$ 
Xkal_updated = zeros(1, T+1); %coder's estimate of state  $\hat{x}(k|k)$ 

X_hat = zeros(1, T+1); %decoder's prediction of state
l = zeros(1, T+1); %scaling factor
l(1) = A^tau; %initialization of scaling factor

for t = 1:T+1

    %time update: realization of state and measurement with coder's
    %Kalman filtering update
    if t == 1
        Xkal_hat(t) = 0;
        X_hat(t) = 0;
    else
        u = -A*X_hat(t-1); %control input w/o quantization
        Xkal_hat(t) = A*Xkal_updated(t-1) + u;
        X(t) = A*X(t-1)+u + mvnrnd(0, Q)';
        Y(t) = H*X(t) + V(t);

        %decoder's estimate
        if mod(t-1, tau) == 0
            %QUANTIZATION MEASUREMENT UPDATE
            z = Xkal_updated(t-tau)-X_hat(t-tau);
            [k,q] = Quantizer(R*tau, p, z/l(t+1-tau));
            X_hat(t) = pro(t)*A^(tau)*l(t+1-tau)*q;
            l(t+1) = max(l(1), k*l(t)*abs(A)^tau);
        else
            l(t+1) = l(t);
            X_hat(t) = 0; %choose control input to bring state to zero
        end
    end

    %measurement update

```

```

        %calculate innovation
        Z(t) = Y(t) -H*Xkal_hat(t);
        Xkal_updated(t) = Xkal_hat(t) + Kal_gain(t)*Z(t);
    end

    error_norm = error_norm + sum((X - X_hat).^2, 1);
    error_norm_kf = error_norm_kf + sum((X-Xkal_updated).^2,1);
    state_norm = state_norm + sum(X.^2, 1);
end

error_norm = sqrt(error_norm/montecarlo);
error_norm_kf = sqrt(error_norm_kf/montecarlo);
state_norm = sqrt(state_norm/montecarlo);

time = 1:T;

figure;
plot(time, error_norm(1:T), '-kd','MarkerFaceColor','g','MarkerSize',4);
title(strcat('Rate = ', num2str(R), ' , prob = ', num2str(prob), ' , a = ' ,
num2str(A) , ' , # of iterations = ' , num2str(montecarlo)));

```

## Appendix D: Yuksel Simulation Code

---

### YukselSim.m

---

```

%Noele Norris
%Created: February 23, 2010
%Modified: April 29, 2010
%Simulation of Yuksel's update rules in "A Random Time Stochastic Drift
%Result and Application to Stochastic Stabilization over Noisy Channels"

%inputs to function:
%   T : number of time steps
%   montecarlo : # of iterations
%   p : probability that signal is transmitted with no error over 1
%       channel use
%   R: rate
T = 100;
montecarlo = 1;
count = 0; %number of instances error exceeds choosen value

%System dynamics parameters
%Scalar system:  $x_{t+1} = a*x_t + u_t + d_t$ 
%unstable system:  $|a| \geq 1$ 
%Gaussian noise:
    a = 2.5;
    R = 3; %rate of transmission
    prob = 0.8; %1 - prob = probability of erasure
    Q = 1; %variance of  $d_t$ 

```

```

P_orig = 1;           %variance of x_0

K = 2^R;
R_p = log2(K-1);

%Values for determining quantizer bin size (look at Thm 3.2)
d = 0.05*abs(a);
eta = 0.05*(K-1);

%%%
L = (K - 1 - eta)/abs(a); %threshold = 'L' in the paper
L_p = 1;

%Result: Mean square error
error_norm = zeros(1, T);

for iter = 1:montecarlo

    iter

    %initialize realization of state
    X = zeros(1, T);

    %vector of erasures
    p = binornd(1, prob, [1 T]);

    %initalization
    del = zeros(1, T); %step size of adaptive quantizer
    del(1) = 1;
    X_hat = zeros(1, T); %decoder's prediction of state
    X(1) = normrnd(0, P_orig);

    for t = 2:T

        Q_output = Quantizer(K-1, del(t-1), X(t-1));
        if abs(abs(Q_output)-(K-1)*del(t-1)/2) < .0005
            overflow = 1;
            X_hat(t-1) = 0;
        else
            overflow = 0;
            X_hat(t-1) = p(t-1)*Q_output;
        end

        X(t) = a*(X(t-1)-X_hat(t-1)) + mvnrnd(0, Q); %state update,
with control input driving state to zero

        %determine change in step size
        if (p(t-1)== 0 || overflow == 1)
            delMult = abs(a) + d;
        elseif del(t-1) > L
            delMult = abs(a)/(2^R_p - eta);
        else

```

```

        delMult = 1;
    end
    del(t) = del(t-1)*delMult;
end

error_norm = error_norm + sum((X - X_hat).^2, 1);
if(max(X) >= 10^5)
    count = count + 1;
end

end

error_norm = sqrt(error_norm/montecarlo);

time = 1:T;

count
figure;
plot(time, error_norm(1:T), '-kd','MarkerFaceColor','g','MarkerSize',4);
title(strcat('Rate = ', num2str(R), ', Prob = ', num2str(prob), ', a = ',
num2str(a), ', # of iterations = ', num2str(montecarlo)));

```

## Quantizer.m

---

```

function y = Quantizer(K, delta, x)

%is written only for odd K

if mod(K,2) == 0
    error('K should be an odd positive integer');
end

y = zeros(size(x));

for i = 1:length(x)
    if abs(x(i)) >= K*delta/2
        y(i) = sign(x(i))*(K)*delta/2;
    else
        index = floor(2*abs(x(i))/delta);
        index_v2 = ceil(index/2);

        y(i) = sign(x(i))*index_v2*delta;
    end
end
end

```

## References

---

- [1] D. Cook and S. Das, *Smart Environments : Technology, Protocols and Applications*. New York: Wiley-Interscience, 2004.
- [2] F. Zhao and L. J. Guibas, *Wireless Sensor Networks : An Information Processing Approach*. Greensboro: Morgan Kaufmann, 2004.
- [3] R. T. Sukhavasi, "The Kalman Like Particle Filter : Optimal Estimation With Quantized Innovations/Measurements," presented at the 48th IEEE Conference on Decision and Control, Shanghai, China, 2009.
- [4] T. Kailath, *et al.*, *Linear Estimation*. Upper Saddle River, NJ: Prentice Hall, 2000.
- [5] A. Ribeiro, *et al.*, "SOI-KF: Distributed Kalman filtering with low-cost communications using the sign of innovations," *IEEE Transactions on Signal Processing*, vol. 54, pp. 4782-4795, Dec 2006.
- [6] R. T. Sukhavasi and B. Hassibi, "Particle filtering for Quantized Innovations," presented at the Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, 2009.
- [7] F. C. Schweppe, "Recursive State Estimation - Unknown but Bounded Errors and System Inputs," *IEEE Transactions on Automatic Control*, vol. AC13, pp. 22-&, 1968.
- [8] D. Bertsekas and I. B. Rhodes, "Recursive State Estimation for a Set-Membership Description of Uncertainty," *IEEE Transactions on Automatic Control*, vol. AC16, pp. 117-&, 1971.
- [9] P. Minero, *et al.*, "Data Rate Theorem for Stabilization Over Time-Varying Feedback Channels," *IEEE Transactions on Automatic Control*, vol. 54, pp. 243-255, 2009.
- [10] S. Yuksel, "A Random Time Stochastic Drift Result and Application to Stochastic Stabilization Over Noisy Channels," presented at the Allerton Conference, Illinois, USA, 2009.
- [11] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Second Edition ed. Hoboken, NJ: Joyn Wiley & Sons, Inc., 2006.
- [12] M. Franceschetti, "Data Rate Theorem for Stabilization over Time-Varying Feedback Channels: Workshop on Frontiers in Distributed Communication, Sensing and Control," ed, 2008.
- [13] S. Meyn and R. L. Tweedie, *Markov Chains and Stochastic Stability*, Second Edition ed. New York: Cambridge University Press, 2009.
- [14] H. Olle, *Finite Markov Chains and Algorithmic Applications*. New York: Cambridge University Press, 2002.