What can Coding do for Control?

Babak Hassibi

Joint work with Ravi Teja Sukhavasi

Department of Electrical Engineering

California Institute of Technology, Pasadena, CA 91125

ISL Colloquium

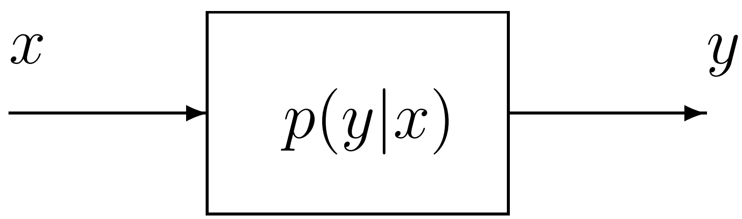
Stanford University, October 25, 2012

Outline

- **Introduction**
  - information theory and coding
  - control theory
- **Interplay between control and communication**
  - coding for interactive communications: tree codes
  - control over noisy channels: anytime capacity
- **Construction of linear tree codes**
  - existence with high probability
  - efficient decoding for erasure channels
  - examples: stabilizing plants over erasure links, consensus over lossy networks
- **Conclusion**
  - future work and open problems, *what can coding do for control*?

Single-User Information Theory

Single-user information (Shannon 1948) deals with the study of the fundamental limits of reliable information transmission between a sender and a receiver over a noisy channel



$$C = \max_{p_X(\cdot)} \{H(x) + H(y) - H(x,y)\}$$

Key idea: Block coding

- the behavior of the channel over a single use is unpredictable
- but the behavior over many channel uses is:
  - if the channel introduces errors with probability p, say, over $n \gg 1$ channel uses it will introduce $\approx np$ errors

Information Theory Lives in Asymptopia

- start with $b = \{b_i\}_{i=1}^m$ bits (the message), and map them to $c = \{c_i\}_{i=1}^n \in \mathcal{C}$ bits (encoding, rate $= \frac{m}{n}$)
- if $y = \{y_i\}_{i=1}^n$ is received, then perform ML decoding: $\hat{c} = \arg\max_{c \in \mathcal{C}} p(y|c)$

If $\frac{m}{n} < C$, *there exists a sequence of codes*, such that (Shannon)

$$\lim_{n \to \infty} P(\hat{c} \neq c) = 0.$$

This is nice theory and an elegant result. However,

- it may require unlimited computational resources at the transmitter and receiver (encoding and decoding may need exponential time)
- it assumes asymptotically long delays ($n \to \infty$)
  - encoding can be done only *after* all the bits $\{b_i\}_{i=1}^n$ are available
  - decoding can be done only *after* all the outputs $\{y_i\}_{i=1}^n$ are observed

In Practice...

- one cares about the probability-of-error as a function of the rate and the length of the code (error exponents)
- one cares about codes that can be efficiently encoded and decoded
  - algebraic codes
    - Reed-Solomon, Reed-Muller, algebraic geometry
    - Berlekamp-Massey, list-decoding (Guruswamy-Sudan), etc.
  - graph-based codes
    - turbo codes, LDPC codes, expander codes
    - message-passing, bit-flipping, LP decoding, etc.
  - polar codes

In summary, after 60 years of work, we have practical codes that come close to the Shannon limits in many cases.

A Critique...

An early criticism (albeit philosphical) of information was that it *did not involve time*

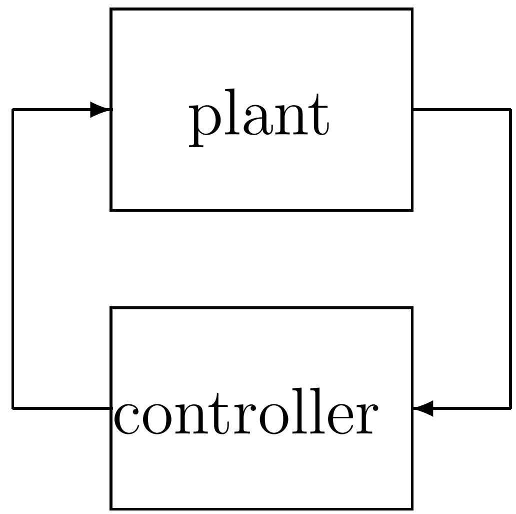
- the "information" obtained about knowledge of an event, depends only on the probability of that event

$$\log \frac{1}{p},$$

  not on *when* this knowledge is revealed or when we want to take action on this knowledge

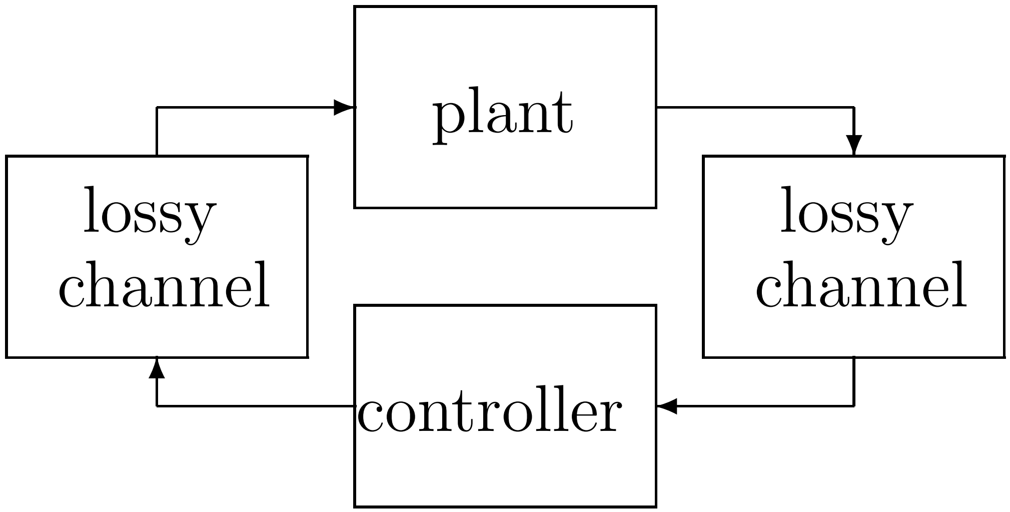
- issue never quite resolved (things like directed mutual information, or the entropy rate of a random process don't quite cut it)
- problem is that information-theoretic quantities often require some form of ergodicity to have operational significance

Control Theory



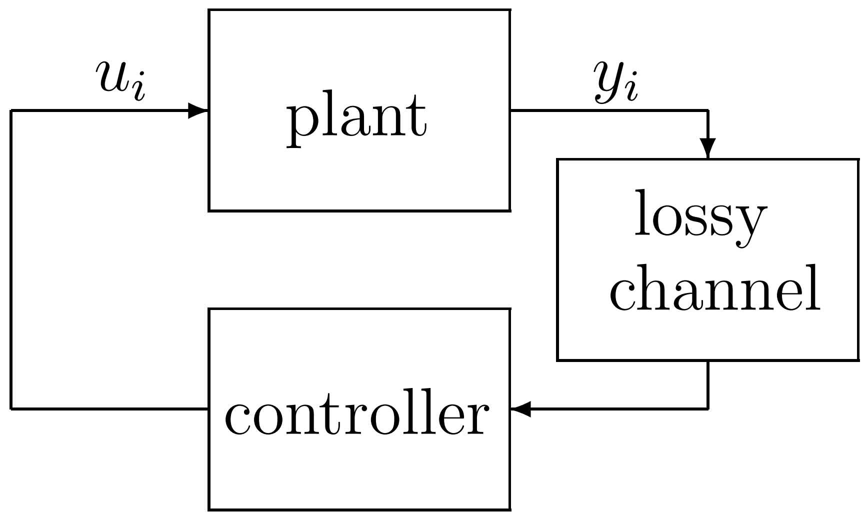
- in control theory we observe the output of a dynamical system (plant) and design a controller to regulate its behavior
- controller needs to react and generate control signals *in real-time*
- delay can result in loss of performance and/or instability
- very rich theory has been developed, especially in the LTI case, (LQG control, H^∞ control, Kalman filtering, separation principle)

But what if...?



- increasingly we have applications where systems (autonomous agents, sensor/actuator networks, smart grid, etc.) are remotely controlled and where measurement and control signals are transmitted across noisy channels
- conventional channel coding does not work - the delay is intolerable
- no coding does not work - even optimal control can lead to instability if there is no coding (Sinopoli et al, 2005)

A Simple Example



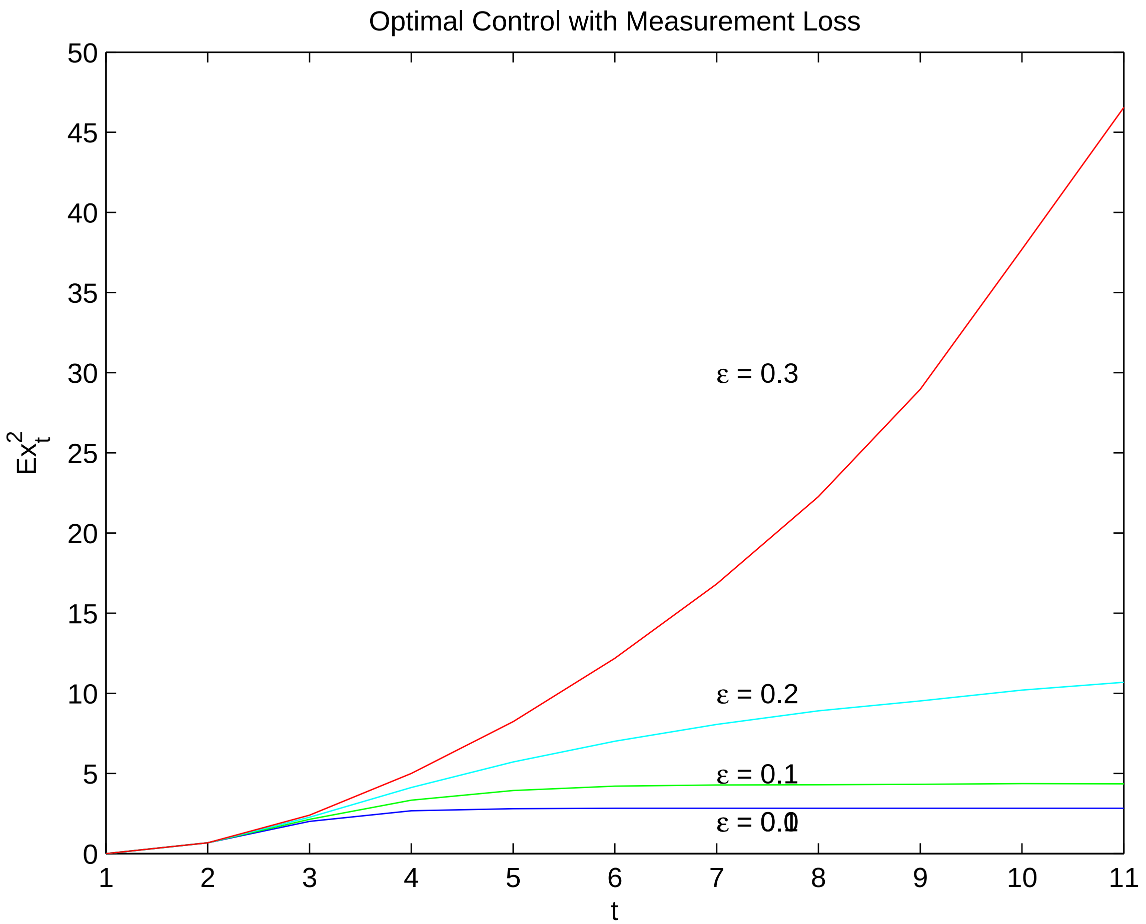
- Take the scalar LTI plant

$$\begin{cases} x_{i+1} & = & 2x_i + w_i + u_i \\ y_i & = & x_i + v_i \end{cases}$$

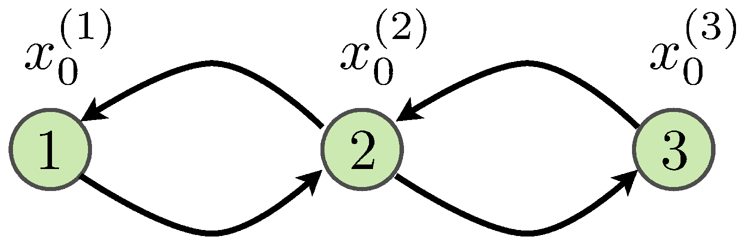
  where w_i and v_i are uniform over $[-1, 1]$.

- Let the channel from the plant to controller be an erasure channel with erasure probability ϵ

The Performance of the *Optimal* Controller



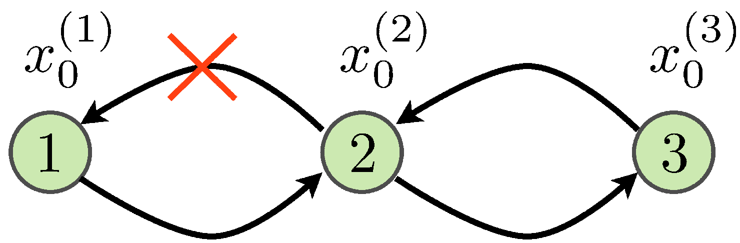
Another Example: Distributed Consensus



$$
\begin{cases}
x_{t+1}^{(1)} & = & x_t^{(1)} + \alpha(x_t^{(2)} - x_t^{(1)}) \\
x_{t+1}^{(2)} & = & x_t^{(2)} + \alpha(x_t^{(1)} - x_t^{(2)}) + \alpha(x_t^{(3)} - x_t^{(2)}) \\
x_{t+1}^{(3)} & = & x_t^{(3)} + \alpha(x_t^{(2)} - x_t^{(3)})
\end{cases}
$$

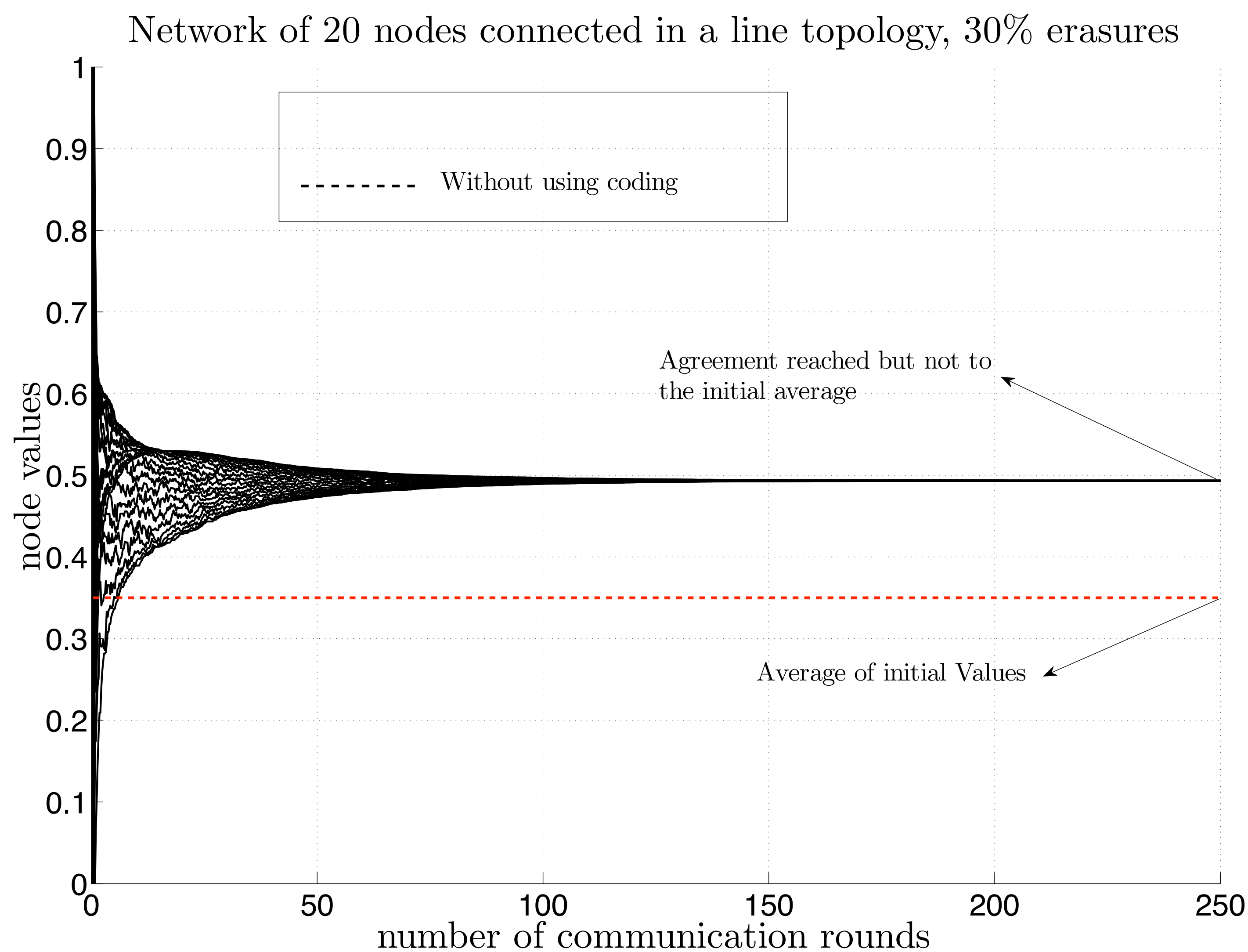
$$
W = \begin{bmatrix} 1-\alpha & \alpha & 0 \\ \alpha & 1-2\alpha & \alpha \\ 0 & \alpha & 1-\alpha \end{bmatrix} \quad , \quad \lim_{t\to\infty} W^t = \frac{1}{3}\mathbf{1}\mathbf{1}^T.
$$

Distributed Consensus with Erasures?

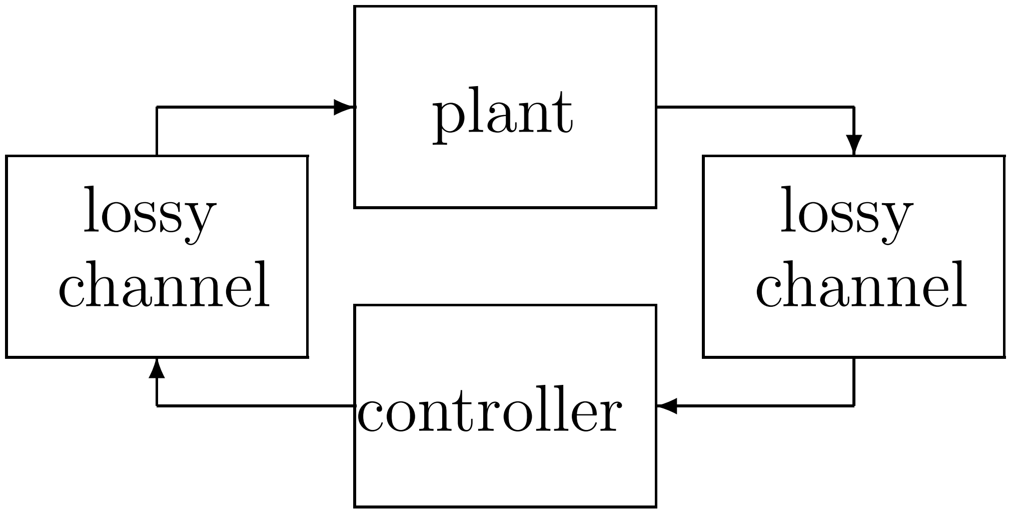


$$\begin{cases} x_{t+1}^{(1)} &= x_t^{(1)} \\ x_{t+1}^{(2)} &= x_t^{(2)} + \alpha(x_t^{(1)} - x_t^{(2)}) + \alpha(x_t^{(3)} - x_t^{(2)}) \\ x_{t+1}^{(3)} &= x_t^{(3)} + \alpha(x_t^{(2)} - x_t^{(3)}) \end{cases}$$

$$W_t = \begin{bmatrix} 1 & 0 & 0 \\ \alpha & 1-2\alpha & \alpha \\ 0 & \alpha & 1-\alpha \end{bmatrix} \quad , \quad \prod_t W_t \text{ is not doubly stochastic.}$$


Network of 20 nodes connected in a line topology, 30% erasures
Without using coding
Agreement reached but not to the initial average
Average of initial Values
node values
number of communication rounds
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
50
100
150
200
250

What to Do?



- the problem is that if we cannot tolerate large delays, we cannot make the noisy channels reliable
- but do we need to do that?
- what do we need to guarantee the stability of the closed loop system?
- what do we need to do in the consensus problem?

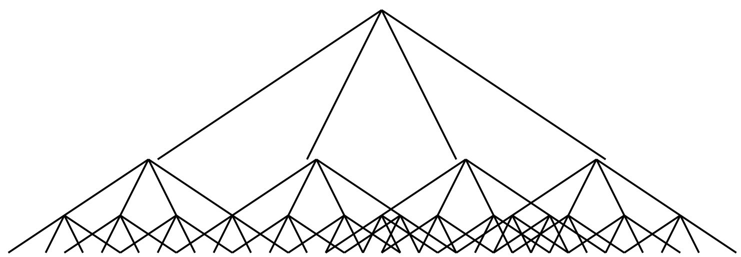
Coding for Interactive Communications

Consider a two-party communication system

$$
\begin{array}{ccc}
\text{Alice, } x & & \text{Bob, } y \\
 & s_1 = f_1(x) & \\
 & - - - - - -- \longrightarrow & \\
 & s_2 = f_2(y, s_1) & \\
 & \longleftarrow - - - - - -- & \\
 & s_3 = f_3(x, s_1, s_2) & \\
 & - - - - - -- \longrightarrow & \\
 & \vdots &
\end{array}
$$

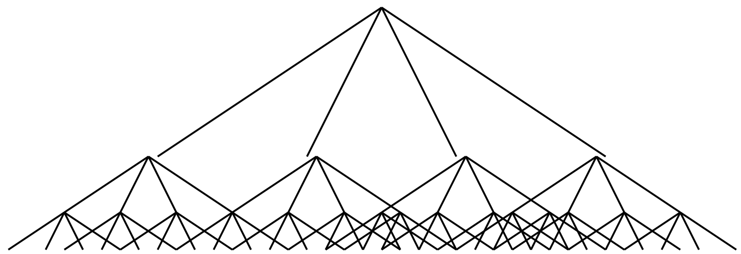
Can one do this reliably over noisy links?

Tree Codes (Schulman, 1993)



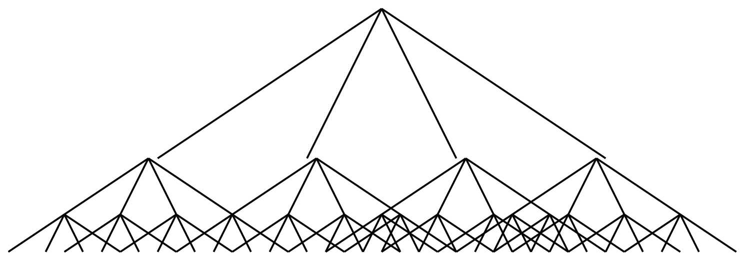
- semi-infinite d-ary tree
- each edge labeled by a symbol in an alphabet of size $d' > d$
- maps a sequence $\{s_i\}_{i=1}^{\infty}$ to a sequence $\{c_i\}_{i=1}^{\infty}$, where $s_i \in \{0, 1, \ldots, d-1\}$ and $c_i \in \{0, 1, \ldots, d'-1\}$
- represents a *causal* code; each path is a codeword; $R = \frac{\log d}{\log d'}$

Tree Codes (Schulman, 1993)



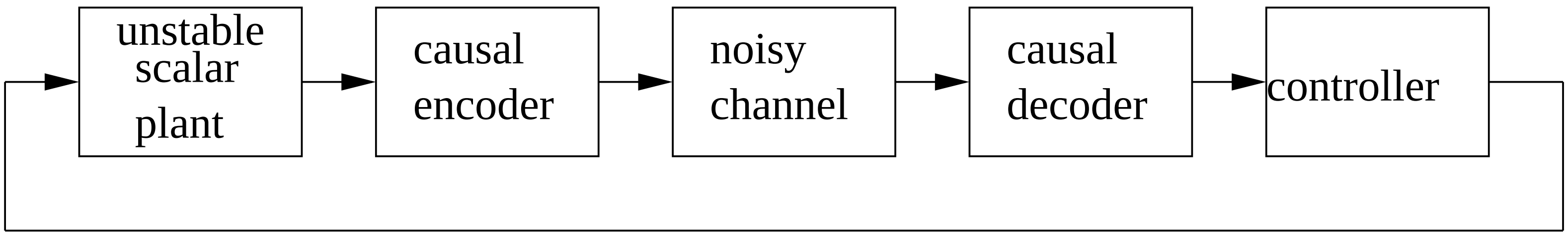
- for *every* pair of paths with a common ancestor and length n, say, we require that the "Hamming distance" between the paths to be *at least a fixed proportion of* n
- Schulman proved the existence of tree codes
- along with ML decoding, allows reliable interactive communication over a noisy link
- **Problem:** No explicit constructions; no tractable decoding; existence result is not with high probability

Tree Codes (Schulman, 1993)



- for the above reasons there has been scant progress in interactive communication over noisy links since Schulman's work
- **Some very recent results:**
  - *potent tree codes:* relax the requirements of tree codes and hence show existence with high probability; good enough for some problems; not good enough for control (Gelles and Sahai, 2011)
  - improvements to Schulman's protocol (Braverman and Rao, 2011)

Anytime Capacity (Sahai, 2001)



- scalar unstable LTI system
- noisy channel from plant output to controller
- each measured output quantized to k bits; *causally* encoded and transmitted across channel
- controller attempts to *causally* decode transmitted bits, estimate state of the system, and generate a control signal
- given that we cannot reliably recover the transmitted bits, can we even stabilize the system?

Toy Example: Tracking an Unstable Plant

$$x_{i+1} = ax_i + w_i, \quad |a| > 1, \quad w_i \in \{-1, 1\} \ \ \text{(unknown)}$$

Assume the initial state $x_0 = 0$ is known to the encoder and controller.

- clearly, at each time instant, the encoder should try to convey 1 bit of information to the controller indicating whether $w_i = 1$ or $w_i = -1$
- the encoder will causally encode this sequence of bits $\{b_i\}_{i=0}^{\infty}$ and send them across the channel
- the decoder, at each time instant i, will attempt to decode the *entire* bit sequence $\{b_j\}_{j=0}^{i}$ and obtain $\{\hat{b}_{j|i}\}_{j=0}^{i}$

Toy Example: Tracking an Unstable Plant

The probability that the first error happens d time steps in the past is:

$$P_e(i,d) = \mathrm{Prob}(\hat{b}_{j|i} = b_j, \forall j < i-d, \hat{b}_{i-d|i} \neq b_{i-d})$$

- Then the mean-square error is bounded by

$$E(x_{i+1} - \hat{x}_{i+1|i})^2 \leq \sum_{d=1}^{i} \left(\frac{a^d - 1}{a-1}\right)^2 P_e(i,d) < \frac{1}{(a-1)^2} \sum_{d=1}^{\infty} |a|^{2d} P_e(i,d).$$

- Clearly, if there exists K, ϵ and Δ, such that for all i and $d > \Delta$:

$$P_e(i,d) < K|a|^{-2d-\epsilon},$$

  we will have *mean-square stability*, i.e., $E(x_{i+1} - \hat{x}_{i+1|i})^2 < \infty$.

- **Remark:** For mean absolute stability, we need $P_e(i,d) < K|a|^{-d-\epsilon}$

Conclusion: We do not need arbitrary reliability. Only a reliability that decays appropriately exponentially fast with the delay.

Anytime Capacity

- *Definition:* A channel will be said to have "anytime capacity" $C_{any}(\lambda)$, for some parameter $\lambda > 1$, if for all rates $R < C_{any}(\lambda)$, there exists causal encoding and decoding schemes such that

$$P_e(i,d) < K\lambda^{-d-\epsilon}, \quad \forall i,\ \forall d > \Delta$$

- *Theorem:* Consider a scalar LTI system

$$\begin{cases} x_{i+1} = \lambda x_i + w_i + u_i \\ y_i = x_i + v_i \end{cases} \qquad |\lambda| > 1$$

where w_i and v_i are *bounded* disturbances. Then to stabilize this system over a noisy channel it is necessary and sufficient that

1. $k > \log|\lambda|$
2. $R = \frac{k}{n} < C_{any}(|\lambda|)$

- The theorem is based on the use of tree codes

Some Problems

This is an elegant result, but...

- there are no explicit constructions of tree codes with efficient decoding
- there is very little hope of actually computing $C_{any}(\lambda)$
  - this requires computing optimal error exponents for tree codes
  - even for block codes optimal error exponents have not been computed
- there are no "necessary and sufficient" conditions for systems with vector states

What to do...?

Linear Tree Codes?

Linear Tree Codes

- linear codes can be represented by *generator* or *parity check* matrices
- for linear tree codes, these matrices will be (block) lower triangular

$$\begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \end{bmatrix} = \begin{bmatrix} G_{11} & & & \\ G_{21} & G_{22} & & \\ G_{31} & G_{32} & G_{33} & \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \end{bmatrix} , \quad \begin{bmatrix} P_{11} & & & \\ P_{21} & P_{22} & & \\ P_{31} & P_{32} & P_{33} & \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \end{bmatrix} = 0$$

where $b_i \in GF_2^k$, $c_i \in GF_2^n$, $G_{ij} \in GF_2^{n \times k}$ and $P_{ij} \in GF_2^{(n-k) \times n}$

- *Do linear tree codes exist?* Requires $P_e(i, d) < K\lambda^{-d-\epsilon}$, for all i and $d > \Delta$. This requires two union bounds (which kills things)—hence Schulman's approach

Toeplitz Linear Tree Codes

- trick is to make the code Toeplitz

$$\begin{bmatrix} G_0 & & & \\ G_1 & G_0 & & \\ G_2 & G_1 & G_0 & \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

and

$$\begin{bmatrix} P_0 & & & \\ P_1 & P_0 & & \\ P_2 & P_1 & P_0 & \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

- this makes the code "look the same" at all times i, and so we avoid the union bound over i

Existence of Tree Codes with High Probability

Theorem: Choose the entries of the entries of the matrices $\{G_i\}_{i=1}^{\infty}$ (or $\{P_i\}_{i=1}^{\infty}$) independently from Bernouli($\frac{1}{2}$) and consider a binary-input channel with Bhattacharya parameter

$$\zeta = \int_{-\infty}^{\infty} \sqrt{p(y|b=1)p(y|b=0)}dy.$$

Then with probability $1 - 2^{-\Omega(n\Delta)}$, for all rates satisfing

$$R < 1 - \log(1+\zeta),$$

there exists a K such that the probability of ML decoding satisfies

$$P_e(i,d) < K2^{-\beta d}, \quad \forall i,\ \forall d > \Delta$$

where

$$\beta < H^{-1}(1-R)\left(\log\frac{1}{\zeta} + \log(2^{1-R}-1)\right).$$

But What to do About ML Decoding?

- to get the error performance we need, we must do ML decoding *at each time instant*
- for block codes doing this even once is too hard....

However, there is one exception....

But not for Erasure Channels

- Consider a random block linear code with $(N-K)\times N$ parity check matrix P ($R = \frac{K}{N}$)
- Suppose the codeword c is transmitted and partition it onto the observed entries c_o and the erased entries c_e, i.e., $c = \begin{bmatrix} c_o \\ c_e \end{bmatrix}$.
- Now due to the parity check condition $Pc = \begin{bmatrix} P_o & P_e \end{bmatrix} \begin{bmatrix} c_o \\ c_e \end{bmatrix} = 0$, we have

$$P_e c_e = P_o c_o$$

- $$P_e c_e = P_o c_o$$

  If the erasure probability of the channel is ϵ, then c_e will have size $\approx N\epsilon$

- If $R = \frac{K}{N} < C = 1 - \epsilon$, then $N - K > N\epsilon$ and the system of linear equations $P_e c_e = P_o c_o$, will, with high probability, have a unique solution.

Conclusion: For erasure channels ML decoding is simply matrix inversion

But What About the Lower Triangular Case?

- for tree codes the matrices P_e and P_o are lower triangular
- therefore even though the matrix P_e has more rows/equations $(N-K)$ than columns/unknowns ($\approx N\epsilon$), the system of equations

$$P_e c_e = P_o c_o,$$

  will most likely *not* have a unique solution (otherwise tree codes would have the same performance of block codes!)
- however, if the tree code corrects all errors above a delay of d, this means that if we partition $c_e = \begin{bmatrix} c_{e1} \\ c_{e2} \end{bmatrix}$, where c_e are all the erased entries with delay more than d, we must have

$$P_e c_e' = P_e c_e'' \quad \text{implies} \quad c_{e1}' = c_{e2}''$$

- partitioning P_e, P_o and c_o similarly, we have

$$\begin{bmatrix} P_{e11} & \\ P_{e21} & P_{e22} \end{bmatrix} \begin{bmatrix} c_{e1} \\ c_{e2} \end{bmatrix} = \begin{bmatrix} P_{o11} & \\ P_{o21} & P_{o22} \end{bmatrix} \begin{bmatrix} c_{o1} \\ c_{o2} \end{bmatrix}$$

  or

$$\begin{bmatrix} P_{e11}c_{e1} \\ P_{e21}c_{e1} + P_{e22}c_{e2} \end{bmatrix} = \begin{bmatrix} P_{o11}c_{o1} \\ P_{o21}c_{o1} + P_{o22}c_{o2} \end{bmatrix}$$

  pre-multiplying the second set of equations by $P_{e22}^{\perp}$, the orthogonal complement of P_{e22} yields

$$\begin{bmatrix} P_{e1} \\ P_{e22}^{\perp}P_{e21} \end{bmatrix} c_{e1} = \begin{bmatrix} P_{o11}c_{o1} \\ P_{e22}^{\perp}P_{o21}c_{o1} + P_{e22}^{\perp}P_{o22}c_{o2} \end{bmatrix}$$

  But this latter system of equations must have a *unique solution* for c_{e1}.

An Efficient Algorithm

1. suppose at time i the bits up to delay d have not yet been decoded (this happens with probability $P_e(i,d) < K\lambda^{-d}$)

2. for these bits, partition $c = \begin{bmatrix} c_e \\ c_o \end{bmatrix}$ and $P = \begin{bmatrix} P_e & P_o \end{bmatrix}$

3. starting with delays $d' = 1, 2, \ldots, d$ check whether the matrix

$$\begin{bmatrix} P_{e1} \\ P_{e22}^{\perp} P_{e21} \end{bmatrix}$$

   has full column rank

4. if so, solve for c_{e1} in the system of equations

$$\begin{bmatrix} P_{e1} \\ P_{e22}^{\perp} P_{e21} \end{bmatrix} c_{e1} = \begin{bmatrix} P_{o11} c_{o1} \\ P_{e22}^{\perp} P_{o21} c_{o1} + P_{e22}^{\perp} P_{o22} c_{o2} \end{bmatrix}$$

5. if this does not happen for any $d' = 1, 2, \ldots d$, go to next time instant

The expected complexity per time instant is constant:

$$\sum_{d=1}^{\infty} K' d^3 \lambda^{-d}$$

Furthermore, the probability that the complexity at any given time instant exceeds $O(d^3)$ decays as $O(\lambda^{-d})$.

Remark: With feedback, encoding can also be done with constant expected complexity.

A Scalar Example

- Take the scalar LTI system

$$\begin{cases} x_{i+1} &= 2x_i + w_i + u_i \\ y_i &= x_i + v_i \end{cases}$$

  where w_i is uniform over $[-30, 30]$ and v_i is uniform over $[-1, 1]$.

- Suppose we want to stabilize this over an erasure channel with erasure probability $\epsilon = 0.3$ and that we have $n = 15$ bits per measurement at our disposal.

- We need an error exponent $2^{-\beta} < \frac{1}{2}$. Using the theorem we can see that we need a rate less than $R < 0.40$, i.e., we should quantize the measurements to at most $k = 5$ bits.

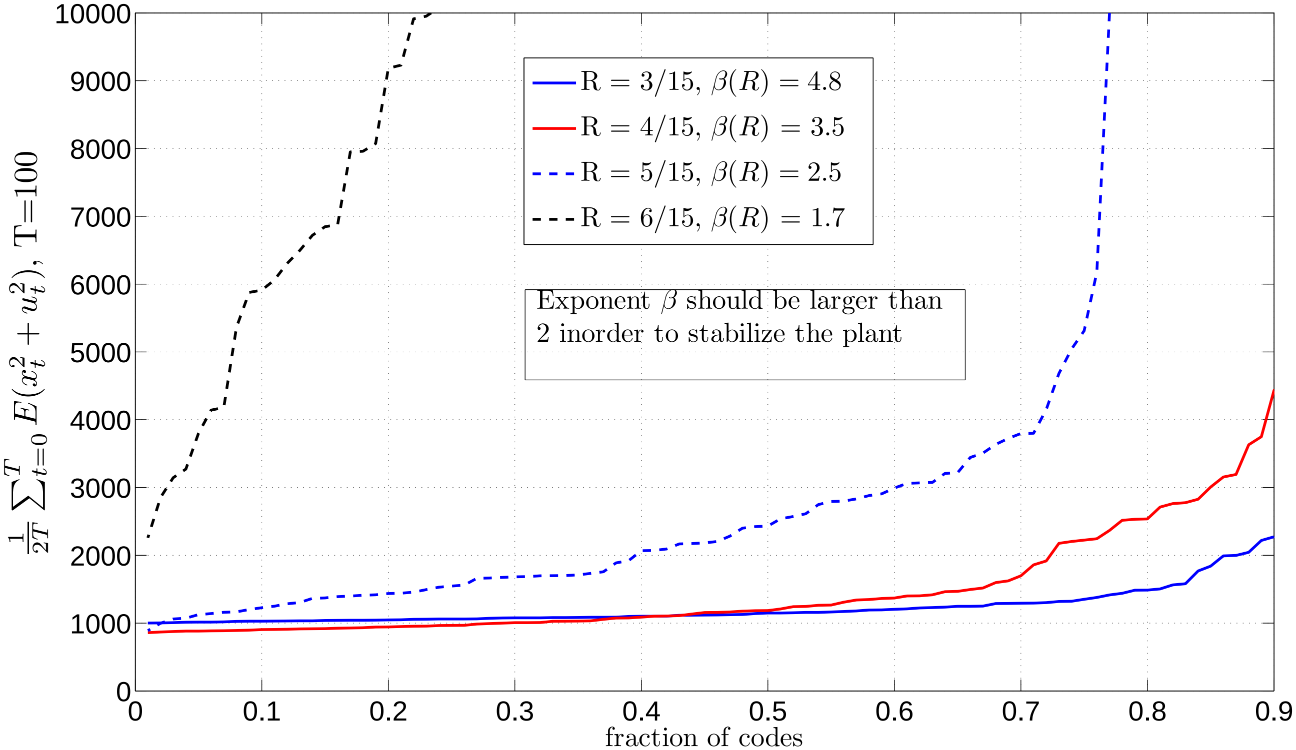


Figure 1: CDF of LQR costs for different realizations of the codes.

The Vector Case

- Consider an observable and controllable LTI system with characteristic polynomial $a(z) = z^n + a_1 z^{n-1} + \ldots a_n$.
- Let $\lambda > 1$ be the smallest positive number such that the matrix

$$\begin{bmatrix} \frac{|a_1|}{\lambda} & 1 & 0 & \ldots & 0 \\ \frac{|a_2|}{\lambda} & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \frac{|a_n|}{\lambda} & 0 & 0 & \ldots & 1 \end{bmatrix} \quad \text{is stable.}$$

- Then we can (mean-square) stabilize the system by appropriately quantizing each output to k bits and using a tree code, such that
  1. $k > \log \lambda$
  2. $P_e(i,d) < K|\lambda_{max}|^{-2d-\epsilon}$ for all i and $d > \Delta$, where $|\lambda_{max}|$ is the largest root of $a(z)$ (in absolute value).

A Vector Example

$$
\begin{aligned}
x_{i+1} &= \begin{bmatrix} 2 & 1 & 0 \\ 0.25 & 0 & 1 \\ -0.5 & 0 & 0 \end{bmatrix} x_i + w_i + u_i \\
y_i &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} x_i + v_i
\end{aligned}
$$

where w_i and v_i are truncated $N(0,1)$ normals to lie in $[-2.5, 2.5]$.
$\lambda_{max} = 2$.

- We would like to stabilize the plant over an erasure channel with $\epsilon = 0.3$.
- We have $n = 15$ bits per measurement available.
- We need an exponent $< \frac{1}{2}$: an application of the theorem shows that we need a rate $R < 0.43$, hence $k < 7$.

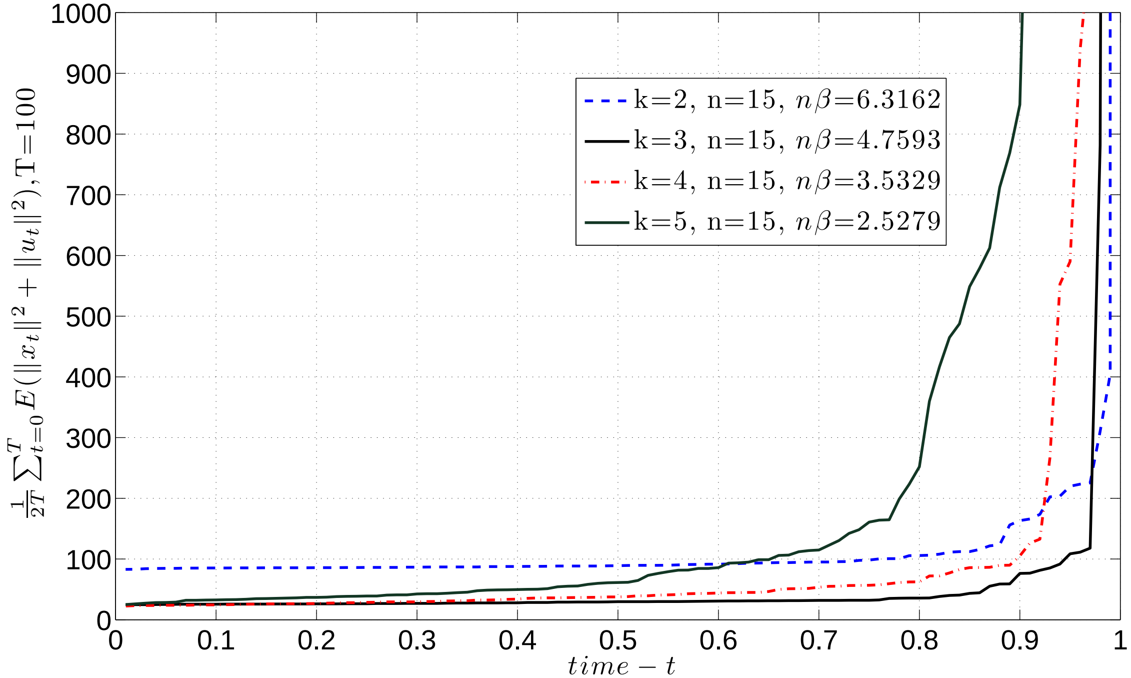
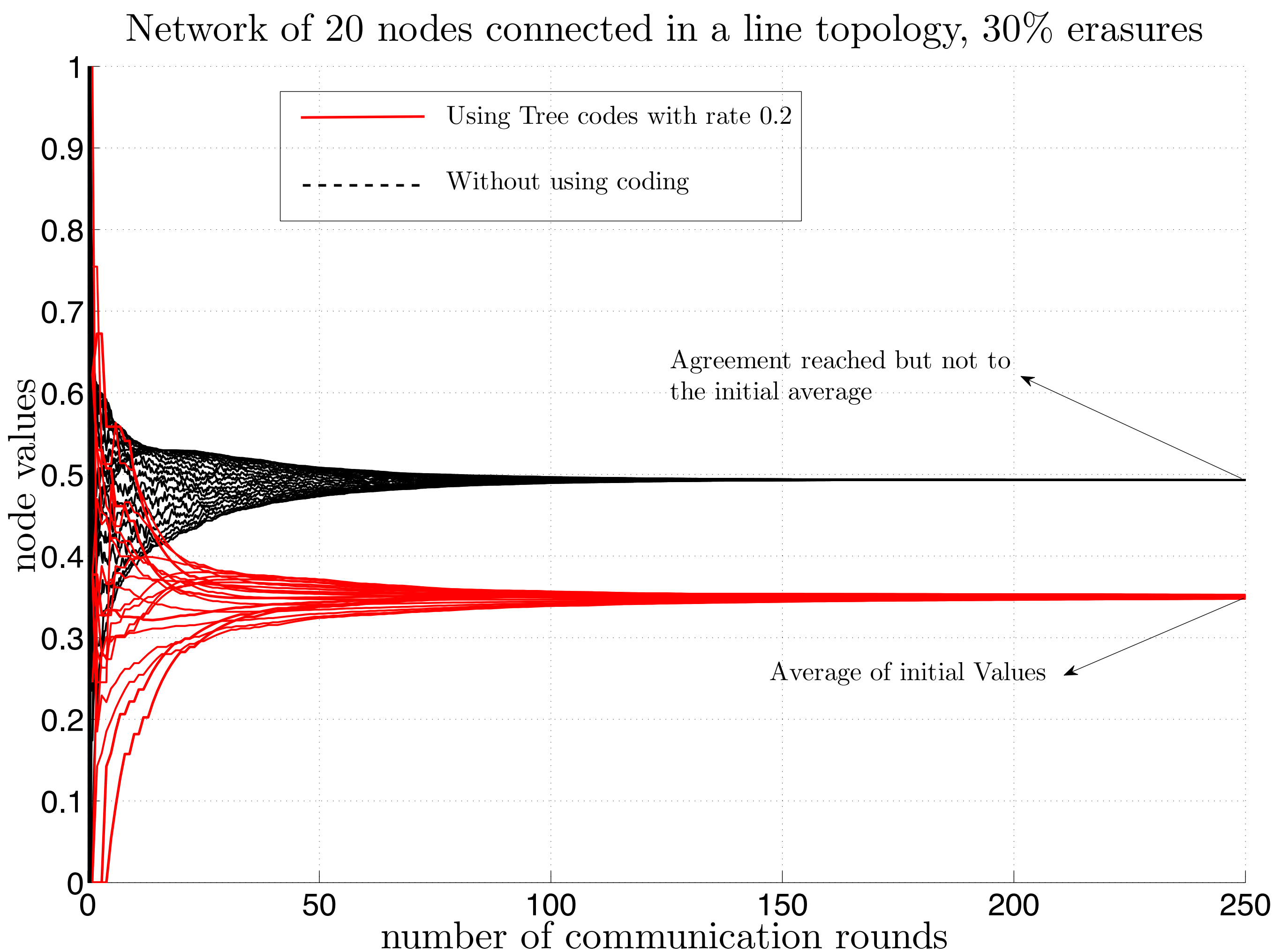


Figure 2: CDF of LQR costs for different realizations of the codes.

Tree Codes for Interactive Protocols: Distributed Consensus

- As mentioned, tree codes can be used to implement *any* interactive protocol over noisy links
- Consider the problem of distributed consensus over a given graph
  - it is well known that if nodes average their and their neighbors' values, then one reaches consensus to the *true* average with a rate given by the second largest eigenvalue of a certain doubly-stochastic matrix
  - if links are erased symmetrically, consensus to the true average still occurs (Jadbabaie and Morse)
- However, symmetric erasures is not a practically reasonable assumption. When erasures are asymmetric, consensus is still reached, *but not to the true average*



Network of 20 nodes connected in a line topology, 30% erasures
Using Tree codes with rate 0.2
Without using coding
Agreement reached but not to
the initial average
Average of initial Values
node values
number of communication rounds
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
50
100
150
200
250

$$R_s \geq 0.175$$

Protocols over Erasure Networks

Consider a graph with maximum degree Δ with asymmetric erasure links

Theorem (Sukhavasi and Hassibi, 2012) A tree code of rate r and reliability exponent β, such that $n\beta > 2\log(1+\Delta)$, guarantees:

- exponentially small error probability in protocol length
- a simulation rate of $R_s \geq r\rho(r)$, where

$$\rho(r) = \max_{s>0} \left\{ s \;\middle|\; (1-s)\frac{n\beta}{2} \geq H(s) + \log(1+\Delta) \right\}$$

Some Remarks

- We have developed a universal and efficient method for stabilizing plants, and implementing interactive protocols, over erasure channels
- Erausre channels are perhaps the most practically interesting case: most systems quantize their measurements to some number of bits, put them in packets and send them across a lossy network (where packets will either be received or dropped)
- Not clear how to deal with unbounded noise (say, Gaussian). Seems to require perfect feedback. Not clear if this is important in practice.
- Stabilizing a plant is the first step. Optimizing performance is next.
  - this will require studying the trade-offs between control and communication resources
  - should we quantize coarsely, but heavily protect the bits, or quantize finely and moderately protect the bits?

Other Channels

- It would be interesting to develop efficiently-decodable tree codes for other types of channels, especially, the BSC and the AWGNC.
- These appear to be much more challenging, since ML decoding is out of the question.
  - even in the block coding case, the codes that achieve capacity over these channels (such as LDPCs or polar codes) do not do so with an error exponent
  - for example, polar codes approach capacity with a probability-of-error $e^{-\alpha\sqrt{N}}$
- We care about having an error exponent much more than the rate
  - in the block coding case, decoders that achieve an error exponent are those that can correct a fixed fraction of errors
  - Reed-Solomon codes; LDPC and expander codes with bit-flipping and/or LP decoding

What does Coding do for Control?

it replaces a lossy link with a lossless link (of lower rate), but with *random delay*.

Conclusion

- Traditional information theory lives in Asymptopia—not appropriate for real-time constraints
- Control has long ignored information theory...
- Controlling an unstable plant over a noisy channel is one place where the two must meet
  - the key object is a "tree code" (essentially a causal code), rather than a block code
  - the key criterion is the interplay between the rate and the decay of the probability of error as a function of the delay (anytime capacity)