

(Almost) Practical Tree Codes

Anatoly Khina, Wael Halbawi and Babak Hassibi

Abstract—We consider the problem of stabilizing an unstable plant driven by bounded noise over a digital noisy communication link, a scenario at the heart of networked control. To stabilize such a plant, one needs real-time encoding and decoding with an error probability profile that decays exponentially with the decoding delay. The works of Schulman and Sahai over the past two decades have developed the notions of *tree codes* and *anytime capacity*, and provided the theoretical framework for studying such problems. Nonetheless, there has been little practical progress in this area due to the absence of explicit constructions of tree codes with efficient encoding and decoding algorithms. Recently, linear time-invariant tree codes were proposed to achieve the desired result under maximum-likelihood decoding. In this work, we take one more step towards practicality, by showing that these codes can be efficiently decoded using sequential decoding algorithms, up to some loss in performance (and with some practical complexity caveats). We supplement our theoretical results with numerical simulations that demonstrate the effectiveness of the decoder in a control system setting.

I. INTRODUCTION

Control theory deals with stabilizing and regulating the behavior of a dynamical system (“plant”) via real-time causal feedback. Traditional control theory was mainly concerned and used in well-crafted closed engineering systems, which are characterized by the measurement and control modules being co-located. The theory and practice for this setup are now well established; see, e.g., [1].

Nevertheless, in the current technological era of ubiquitous wireless connectivity, the demand for control over wireless media is ever growing. This *networked control* setup presents more challenges due to its distributed nature: The plant output and the controller are no longer co-located and are separated by an unreliable link (see Fig. 1).

To stabilize an unstable plant using the unreliable feedback link, an error-correcting code needs to be employed over the latter. In one-way communications — the cornerstone of information theory — all the source data are assumed to be known in advance (*non-causally*) and are recovered only when the reception ends. In contrast, in coding for control, the source data are known only causally, as the new data at each time instant are dependent upon the dynamical random process. Moreover, the controller cannot wait until a large block is received; it needs to constantly produce estimates of the system’s state, such that the fidelity of earlier data improves as time advances. Both of these goals are achieved via causal coding, which receives the data sequentially in a causal fashion and encodes it in a way such that the error probability of

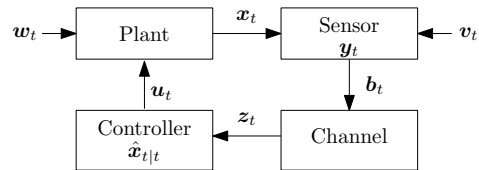


Fig. 1. Basic networked control system.

recovering the source data at a fixed time instant improves constantly with the reception of more code symbols.

Sahai and Mitter [2] provided necessary and sufficient conditions on the required communication reliability over the unreliable feedback link to the controller. To that end, they defined the notion of *anytime capacity* as the appropriate figure of merit for this setting, which is essentially the maximal possible rate of a causal code that at *any* time t recovers a source bit at time $(t - d)$ with error probability that decays exponentially with d , for *all* d . They further recognized that such codes have a natural *tree code* structure, which is similar to the codes developed by Schulman for the related problem of interactive computation [3].

Unfortunately, the result by Schulman (and consequently also the ones by Sahai and Mitter) only proves the existence of a tree code with the desired properties and does not guarantee that a random tree code would be good with high probability. The main difficulty comes from the fact that proving that the *random ensemble* achieves the desired exponential decay does not guarantee that the *same code* achieves this for *every time instant* and *every delay*.

Sukhavasi and Hassibi [4] circumvented this problem by introducing linear time-invariant (LTI) tree codes. The time-invariance property means that the behavior of the code at every time instant is the same, which suggests, in turn, that the performance guarantees for a *random (time-invariant) ensemble* are easily translated to similar guarantees for a *specific code* chosen at random, *with high probability*.

However, this result assumes maximum likelihood (ML) decoding, which is impractical except for binary erasure channels (in which case it amounts to solving linear equations which has polynomial computational complexity).

Sequential decoding was proposed by Wozencraft [5] and subsequently improved by others as a means to recover random tree codes with reduced complexity with some compromise in performance; specifically, for the expected complexity to be finite, the maximal communication rate should be lower than the cutoff rate. For a thorough account of sequential decoding, see [6, Ch. 10], [7, Sec. 6.9], [8, Ch. 6]. This technique was subsequently adopted by Sahai and Palaiyanur [9] for the purpose of decoding (time-varying) tree codes for networked control. Unfortunately, this result relies on an exponential bound on the error probability by Jelinek [10, Th. 2] that is valid for the binary symmetric channel (BSC) (and other

The authors are with the Dept. of Electrical Engineering, Caltech, Pasadena, CA 91125, USA (e-mail: {khina, whalbawi, hassibi}@caltech.edu).

This work was supported in part by the National Science Foundation under grants CNS-0932428, CCF-1018927, CCF-1423663 and CCF-1409204, by a grant from Qualcomm Inc., by NASA’s Jet Propulsion Laboratory through the President and Directors Fund, and by King Abdullah University of Science and Technology.

cases of interest) only when the expected complexity of the sequential decoder goes to infinity [11].

In this work we propose the usage of sequential decoding for the recovery of LTI tree codes. To that end, similarly to Sahai and Palaiyanur [9], we extend a (different) result developed by Jelinek [6, Th. 10.2] for general (non-linear and time-variant) random codes to LTI tree codes.

II. PROBLEM SETUP AND MOTIVATION

We are interested in stabilizing an unstable plant driven by bounded noise over a noisy communication link. In particular, an observer of the plant measures at every time instant t a noisy version $y_t \in \mathbb{R}$ (with bounded noise) of the state of the plant $\mathbf{x}_t \in \mathbb{R}^m$. The observer then quantizes y_t to $\mathbf{b}_t \in \mathbb{Z}_2^k$, and encodes — using a causal code — all quantized measurements $\{\mathbf{b}_i\}_{i=1}^t$ to produce $\mathbf{c}_t \in \mathbb{Z}_2^n$. This packet \mathbf{c}_t is transmitted over a noisy communication link to the controller, which receives $\mathbf{z}_t \in \mathcal{Z}^n$, where \mathcal{Z} is the channel output alphabet. The controller then decodes $\{\mathbf{z}_i\}_{i=1}^t$ to produce the estimates $\{\hat{\mathbf{b}}_{i|t}\}_{i=1}^t$, where $\hat{\mathbf{b}}_{i|t}$ denotes the estimate of \mathbf{b}_i when decoded at time t . These estimates are mapped back to measurement estimates $\{\hat{y}_{i|t}\}_{i=1}^t$ which are then used to give an estimate $\hat{\mathbf{x}}_{t|t}$ of the current plant state. Finally, the controller computes a control signal \mathbf{u}_t from $\hat{\mathbf{x}}_{t|t}$ and applies it to the plant.

The need for causally sending measurements of the state in real time motivates the use of causal codes in this problem. Generally speaking, a causal code maps, at each time instant t , the current and all previous quantized measurements to a packet of n bits $\mathbf{c}_t = f_t(\{\mathbf{b}_i\}_{i=1}^t)$.

For a linear code, one has $\mathbf{c}_t = \sum_{i=1}^t \mathbf{G}_{t,i} \mathbf{b}_i$. The decoder computes a function $g_t(\{\mathbf{z}_i\}_{i=1}^t)$ to produce $\{\mathbf{b}_{i|t}\}_{i=1}^t$. One is then assigned the task of choosing a sequence of matrices $\{\mathbf{G}_{t,i} | i = 1, \dots, t; t = 1, \dots, \infty\}$ that provides anytime reliability. We recall this definition as stated in [4].

Definition 1. Define the probability of the first error event as $P_e(t, d) \triangleq P(b_{t-d} \neq \hat{\mathbf{b}}_{t-d|t}, \forall \delta > d, b_{t-\delta} = \hat{\mathbf{b}}_{t-\delta|t})$, where the probability is over the randomness of the plant and the channel noise. Suppose we are assigned a budget of n channel uses per time step of the evolution of the plant. Then, an encoder–decoder pair is called (R, β) anytime reliable if there exists $d_0 \in \mathbb{N}$, such that

$$P_e(t, d) \leq 2^{-\beta nd}, \quad \forall t, d \geq d_0,$$

where β is called the anytime exponent.

According to the definition, anytime reliability has to hold for every decoding instant t and every delay d . Sukhvasi and Hassibi proposed in [4] a code construction based on Toeplitz block-lower triangular parity-check matrices, that provides an error exponent for all t and d . The Toeplitz property avoids the need to compute a double union bound; for details see [12].

In this work, the link between the observer and the controller is assumed to be a memoryless binary-input output-symmetric (MBIOS) channel: $w(z_i | c_i = 0) = w(-z_i | c_i = 1)$, where w is the channel transition distribution, $z_i \in \mathcal{Z}$ and $c_i \in \mathbb{Z}_2$.

III. PRELIMINARIES: CONVOLUTIONAL CODES

We next review known result for several random ensembles of convolutional codes, in Section III-A. The codes within

each ensemble can be either linear or not; linear ensembles can be further either time variant or time invariant. We further discuss sequential decoding algorithms and their performance in Section III-B which will be used in the sequel for tree codes.

A. Bounds on the Error Probability under ML Decoding

We now recall exponential bounds for convolutional codes under certain decoding regimes.

A compact representation (and implementation) of a convolutional code is via a shift register: The delay-line (shift register) length is denoted by d , whereas its width k is the number of information bits entering the shift register at each stage. Thus, the total memory size is equal to dk bits. At each stage, n code bits are generated by evaluating n functionals over the dk memory bits and the new k information bits. We refer to these n bits as a single branch. Therefore, the rate of the code is equal to $R = k/n$ bits per channel use. In general, these functionals may be either linear or not, resulting in linear or non-linear convolutional codes, respectively, and stay fixed or vary across time, resulting in time-invariant or time-variant convolutional codes. We further denote the total length of the convolutional code frame upon truncation by N .

Typically, the total length of the convolutional code frame is chosen to be much larger than d , i.e., $N \gg d$. We shall see in Section IV, that in the context of tree codes, a decoding delay of d time steps of the evolution of the plant into the past corresponds to a convolutional code with delay-length d . Since each time step corresponds to n uses of the communication link, the relevant regime for the current work is $N = nd$.

Theorem 1 ([8, Sec. 5.6], [13, Eq. (14)]). *The probability of the first error event of random time-variant and LTI convolutional codes, under optimal (maximum likelihood) decoding, is bounded from the above by*

$$\bar{P}_e(d) \leq 2^{-E_G(R)nd} \quad (1)$$

where $E_G(R)$ is Gallager's error exponent function [7, Sec. 5.6], defined as (see also Fig. 2):

$$E_G(R) \triangleq \max_{0 \leq \rho \leq 1} [E_0(\rho) - \rho R], \quad (2a)$$

$$E_0(\rho) \triangleq 1 + \rho - \log \left\{ \sum_{z \in \mathcal{Z}} \left[w^{\frac{1}{1+\rho}}(z|0) + w^{\frac{1}{1+\rho}}(z|1) \right]^{1+\rho} \right\}. \quad (2b)$$

We note that in the common work regime of $N \gg d$, the optimal achievable error exponent was proved by Yudkin and by Viterbi to be much better than $E_G(R)$ [8, Ch. 5].

Unfortunately, the computational complexity of maximum-likelihood decoding grows exponentially with the delay-line length d , prohibiting its use in practice for large values of d .

We therefore review next a suboptimal decoding procedure, the complexity of which does not grow rapidly with d but still achieves exponential decay in d of the BER.

B. Sequential Decoding

The Viterbi algorithm [8, Sec. 4.2] offers an efficient implementation of (frame-wise) ML decoding for fixed d and growing N . Unfortunately, the complexity of this algorithm grows exponentially with N when the two are coupled, i.e.,

$N = nd$. Prior to the adaptation of the Viterbi algorithm as the preferred decoding algorithm of convolutional codes, sequential decoding was served as the *de facto* standard. A wide class of algorithms fall under the umbrella of “sequential decoding”. Common to all is the fact that they explore only a subset of the (likely) codeword paths, such that their complexity does not grow (much) with d , and are therefore applicable for the decoding of tree codes.¹

In this work we shall concentrate on the two popular variants of this algorithm — the Stack and the Fano (which is characterized by a quantization parameter Δ) algorithms.

We next summarize the relevant properties of these decoding algorithms when using the generalized Fano metric (see, e.g., [6, Ch. 10]) to compare possible codeword paths:

$$M(c_1, \dots, c_N) = \sum_{t=1}^N M(c_t), \quad (3a)$$

$$M(c_t) \triangleq \log \frac{w(z_t|c_t)}{p(z_t)} - B, \quad (3b)$$

where B is referred to as the metric bias and penalizes longer paths when the metrics of different-length paths are compared. In contrast to ML decoding, where all possible paths (of length N) are explored to determine the path with the total maximal metric,² using the stack sequential decoding algorithm, a list of partially explored paths is stored, where at each step the path with the highest metric is further explored and replaced with its immediate descendants and their metrics. The Fano algorithm achieves the same without storing all these potential paths, at the price of a constant increase in the error probability and computational complexity; for a detailed descriptions of both algorithms see [6, Ch. 10], [7, Sec. 6.9], [8, Ch. 6].

The choice $B = R$ is known to minimize the expected complexity, and is therefore the most popular choice in practice. Moreover, for rates below the cutoff rate $R < R_0 \triangleq E_0(\rho = 1)$, the expected number of metric evaluations (3b) is finite and does not depend on d , for any $B \leq R_0$ [7, Sec. 6.9], [6, Ch. 10]. Thus, the only increase in expected complexity of this algorithm with d comes from an increase in the complexity of evaluating the metric of a single symbol (3b). Since the latter increases (at most) linearly with d , the total complexity of the algorithm grows polynomially in d . Furthermore, for rates above the cutoff rate, $R > R_0$, the expected complexity is known to grow rapidly with N for *any metric* [11], implying that the algorithm is applicable only for $R < R_0$.

Most results concerning the error probability under sequential decoding consider an infinite code stream (over a trellis graph) and evaluate the probability of an erroneous path to diverge from the correct path and re-merge with the correct path, which can only happen for $N > nd$. Such analyses are not adequate for our case of interest, in which $N = nd$. The following theorem provides a bound for our case.

Theorem 2 ([6, Ch. 10]). *The probability of the first error event of general random convolutional codes, using the Fano or stack sequential decoders and the Fano metric with bias*

¹Interestingly, the idea of tree codes was conceived and used already in the early works on sequential decoding [5].

²Note that optimizing (3a) in this case is equivalent to ML decoding.

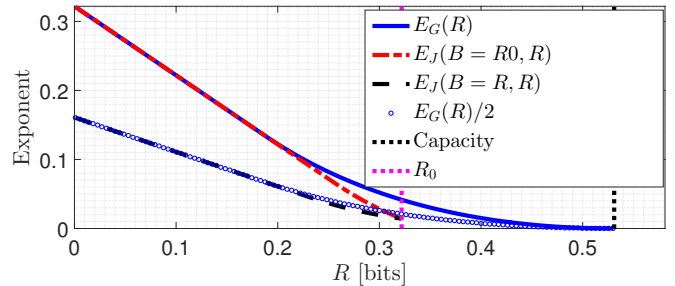


Fig. 2. Error exponents $E_G(R)$, $E_J(B = R_0, R)$ and $E_J(B = R, R)$ for a BSC with crossover probability $p = 0.1$.

B , is bounded from the above by:

$$\bar{P}_e(t, d) \leq A 2^{-E_J(B, R)nd}, \quad (4a)$$

$$E_J(B, R) \triangleq \max_{0 \leq \rho \leq 1} \frac{\rho}{1 + \rho} \left\{ E_0(\rho) + B - (1 + \rho)R \right\}, \quad (4b)$$

where A is finite for $B < R_0$ and is upper bounded by³

$$A \leq e^{\frac{\rho}{1+\rho}\Delta} \frac{1 - e^{-t[E_0(\rho) - \rho B]}}{1 - e^{-[E_0(\rho) - \rho B]}} \leq \frac{e^{\frac{\rho}{1+\rho}\Delta}}{1 - e^{-[E_0(\rho) - \rho B]}} < \infty, \quad (5)$$

for a quantization parameter Δ in the Fano algorithm; for the stack algorithm (5) holds true with $\Delta = 0$.

Since, $E_J(B, R)$ is a monotonically increasing function of B , choosing $B = R_0$ maximizes the exponential decay of $\bar{P}_e(d)$ in d .⁴ Interestingly, for this choice of bias, $E_J(B = R_0, R) = E_G(R)$ whenever $E_G(R)$ is achieved by $\rho = 1$ in (2a), i.e., for rates below the critical rate $R < R_{\text{crit}} \triangleq E'_0(\rho)$. For other values of ρ , $E_J(B = R_0, R)$ is strictly smaller than $E_G(R)$ (see Fig. 2).

For the choice of bias that optimizes complexity, $B = R$, on the other hand, an error exponent which equals to at least half the exponent under ML decoding (1) is achieved whenever $E_G(R)$ is achieved by $\rho = 1$: $E_J(B = R, R) \geq E_G(R)/2$ (see Fig. 2).

We now turn to bounding the number of branch computations per node of the code tree under sequential decoding.

Definition 2. Denote by W_t the number of branch computations of node t performed by a sequential decoding algorithm.

We note that W_t is a random variable (which depends on the received vector and the underlying tree code used). Since W_t is equal to just one more than the number of branch computations in the incorrect sub-tree of that node, it has the same distribution for any t , for general random and LTI random codes.

We use the following result which can be found in [8, Sec. 6.2] for general random convolutional codes, and in [7, Sec. 6.9] for LTI convolutional codes; for both, delay-line length d is assumed to be infinite.⁵

Theorem 3. *The probability that W_t of a general random convolutional code or an LTI convolutional code with infinite*

³Note that $E_0(\rho)/\rho$ is a monotonically decreasing function of ρ , therefore $B < R_0 = E_0(1)$ guarantees that $E_0(\rho) - \rho B > 0$.

⁴For finite values of d a lower choice of B might be better, since the constant A might be smaller in this case.

⁵For finite delay-length convolutional codes, the computational complexity can only be smaller than that of infinite delay-length codes.

delay-length is larger than $m \in \mathbb{N}$ is upper bounded by

$$\Pr(W_t \geq m) \leq Am^{-\rho},$$

where A is finite for $B, R < R_0$, $R < \frac{B+R_0}{2\rho}$, and $\rho \in (0, 1]$.

An immediate consequence of this result is that the expected complexity per branch $E[W_t]$ is bounded if $R < R_0$. Moreover, a converse result by Arkan [11] states that the expected complexity is unbounded for $R > R_0$.

The next result provides a lower bound on the error probability.

Theorem 4 (See [8, Sec. 6.4]). *The probability that W_t of any convolutional code, where no decoding error occurs, is greater than $m \in \mathbb{N}$, is lower bounded by*

$$\Pr(W_t \geq m) \geq (1 - o(m))m^{-\rho},$$

where $o(m) \rightarrow 0$ for $m \rightarrow \infty$, and $R = \frac{E_0(\rho)}{\rho}$ and any $\rho > 0$.⁶

This result is known to be tight [8] for *general* random convolutional code, and is widely believed to be true for random LTI convolutional codes, although no formal proof exists for the latter.

IV. LINEAR TIME INVARIANT ANYTIME RELIABLE CODES

In this section, we recall the construction of LTI anytime reliable codes as presented in [4]. A causal linear time-invariant code has a parity-check matrix with the following lower triangular Toeplitz structure. The generator matrix of an (n, R) LTI code of rate R and blocklength n per time step is given by $\mathbf{c} = \mathbf{G}_{n,R}\mathbf{b}$, where

$$\mathbf{G}_{n,R} = \begin{bmatrix} \mathbf{G}_1 & \mathbf{0} & \cdots & \cdots & \cdots \\ \mathbf{G}_2 & \mathbf{G}_1 & \mathbf{0} & \cdots & \cdots \\ \vdots & \vdots & \ddots & \ddots & \cdots \\ \mathbf{G}_t & \mathbf{G}_{t-1} & \cdots & \mathbf{G}_1 & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \mathbf{b} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_t \\ \vdots \end{bmatrix}, \mathbf{c} = \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \\ \vdots \\ \mathbf{c}_t \\ \vdots \end{bmatrix} \quad (6)$$

and $\mathbf{G}_t \in \mathbb{Z}_2^{n \times k}$. The following definition, given in [4], dictates the way to choose the \mathbf{G}_i 's.

Definition 3 (LTI Ensemble). Fix \mathbf{G}_1 to be a full rank matrix, and generate the entries of \mathbf{G}_t independently and uniformly at random, for $t \geq 2$.

For the purpose of this paper, we shall view an (n, R) LTI code as a convolutional code with infinite delay-line length, k information bits and n code bits. As a result of this interpretation, the results of Section III apply directly to this Toeplitz ensemble.

In the extended version of this paper [12], we recap why the LTI ensemble is indeed anytime reliable.

V. SEQUENTIAL DECODING OF LINEAR TIME-INVARIANT ANYTIME RELIABLE CODES

In this section we show that the upper bound on the probability of the first error event under sequential decoding for convolutional codes (4), holds true also for LTI convolutional

⁶Recall that $E_0(\rho)/\rho$ is a decreasing function of ρ and therefore $\rho > 1$ implies that $R < R_0$.

codes, which for $N = nd$, identifies with the LTI tree codes of (6).

To prove this, we adopt the proof technique of [7, Sec. 6.2], where the exponential bounds on the error probability of random block codes are shown to hold also for linear random blocks codes.

Theorem 5. *The probability of the first error event of the LTI random tree ensemble of Definition 3, using the Fano or stack sequential decoders and the Fano metric with bias B , is bounded from the above by (4).*

Proof sketch: A thorough inspection of the proof of Theorem 2, as it appears in [6, Ch. 10], reveals that the following two requirements for this bound to be valid are needed:

- 1) **Pairwise independence.** Every two paths are independent starting from the first branch that corresponds to source branches that disagree.
- 2) **Individual codeword distribution.** The entries of each codeword are i.i.d. and uniform.

We next show how these two requirements are met for the affine linear ensemble. The codes in this ensemble are as in (6) up to an additive translation $\mathbf{v} = [v_1 \ v_2 \ \cdots \ v_t \ \cdots]^T$, where $v_t \in \mathbb{Z}_2^k$, with $\mathbf{c} = \mathbf{G}\mathbf{b} + \mathbf{v}$. The entries of \mathbf{G} and \mathbf{v} are sampled independently and uniformly at random.

Now, assume that two source words \mathbf{b} and $\tilde{\mathbf{b}}$ are identical for $i < t$ and differ in at least one bit in branch t , i.e., $\mathbf{b}_i = \tilde{\mathbf{b}}_i$ for $i < t$ and $\mathbf{b}_t \neq \tilde{\mathbf{b}}_t$. Then, the causal structure of \mathbf{G} guarantees that also $\mathbf{c}_i = \tilde{\mathbf{c}}_i$ for $i < t$. Moreover, $\mathbf{b}_t \neq \tilde{\mathbf{b}}_t$ along with the random construction of \mathbf{G} suggest that the two code paths starting from branch t , $[\mathbf{c}_t^T \ \mathbf{c}_{t+1}^T \ \cdots]^T$ and $[\tilde{\mathbf{c}}_t^T \ \tilde{\mathbf{c}}_{t+1}^T \ \cdots]^T$ are independent. This establishes the first requirement.

To establish the second requirement we note that the addition of a random uniform translation vector \mathbf{v} guarantees that the entries of each codeword are i.i.d. and uniform. This establishes the second requirement and hence also the validity of the proof of [6, Ch. 10].

Finally, note that since the channel is MBIOS, the same error probability is achieved for any translation vector \mathbf{v} . ■

Since Theorem 5 holds for LTI codes, a specific code chosen from the LTI ensemble is anytime reliable with (4) (see [12]).

VI. SIMULATION OF A CONTROL SYSTEM

To demonstrate the effectiveness of the sequential decoder in stabilizing an unstable plant driven by bounded noise, we simulate a cart–stick balancer controlled by an actuator that obtains noisy measurements of the state of the cart through a BSC. The example is the same one from [4]. The plant dynamics evolve as,

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t + \mathbf{w}_t, \quad y_t = \mathbf{C}\mathbf{x}_t + v_t,$$

where \mathbf{u}_t is the control input signal that depends only on the estimate of the current state, i.e. $\mathbf{u}_t = \mathbf{K}\hat{\mathbf{x}}_{t|t}$. The system noise \mathbf{w}_t is a vector of i.i.d. Gaussian random variables with mean $\mu = 0$ and variance $\sigma^2 = 0.01$, truncated to $[-0.025, 0.025]$. The measurement noise v_t is also a truncated Gaussian random

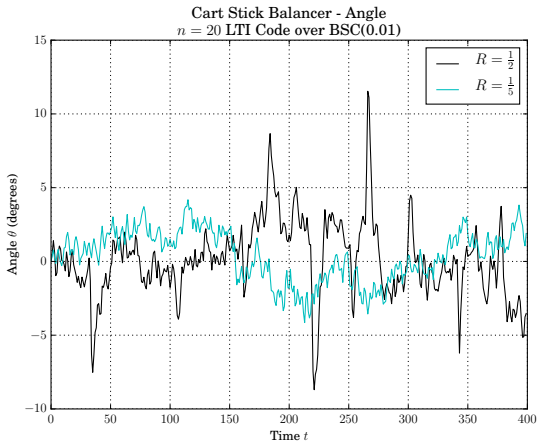


Fig. 3. Stability of the Cart–Stick Balancer is demonstrated using a sequentially decoded LTI code.

variable of the same parameters. We assume that the system is in observer canonical form:

$$\mathbf{A} = \begin{bmatrix} 3.3010 & 1 & 0 \\ -3.2750 & 0 & 1 \\ 0.9801 & 0 & 0 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} -0.0300 \\ -0.0072 \\ 0.0376 \end{bmatrix}, \mathbf{C} = [1 \ 0 \ 0],$$

$$\mathbf{K} = [-55.6920 \ -32.3333 \ -19.0476].$$

The state \mathbf{x}_t , before the transformation to observer canonical form, is composed of the stick’s angle, the stick’s angular velocity and the cart’s velocity. The system is unstable with the largest eigenvalue of \mathbf{A} being 1.75. The channel between the observer and the controller is a BSC with bit-flip probability $p = 0.01$, which has a cutoff rate $R_0 = 0.7382$. We fix a budget of $n = 20$ channel uses. Using Theorem 8.1 in [4], the minimum required number of quantization bits is $k_{\min} = 3$ and the minimum required exponent is $\beta_{\min} = 0.2052$.

For the first experiment, we use a code of rate $R = 1/2$, with $k = 10$ bits for a lattice quantizer with bin width $\delta = 0.1$. From (4b), a sequential decoder with bias R_0 will guarantee an error exponent of $\beta = 0.2382$. As is evident from the dark curve in Fig. 3, the stick on the cart does not deviate by more than 12 degrees.

For the second experiment, we use a lower code rate $R = 1/5$, which provides $\beta = 0.5382$. The light curve in Fig. 3 shows that the deviation is reduced to 4 degrees.

Although these simulations might suggest that a lower rate code always results in better stability of the system, it is not *a-priori* clear that this is truly the case. A lower code rate uses a coarser quantizer than a higher rate code. As a result, there could be some loss due to this coarseness. A common metric used to quantify the performance of the closed-loop stability of a dynamical system is the linear quadratic regulator (LQR) cost for a finite time horizon T given by

$$J = \mathbb{E} \left[\frac{1}{2T} \sum_{t=1}^T (\|\mathbf{x}_t\|^2 + \|\mathbf{u}_t\|^2) \right],$$

where the expectation is w.r.t. the randomness of the plant and the channel. For our example, we simulated the system using three different quantization levels, with 100 codes per quantization level and 40 experiments per code. The LQR costs are 206.0, 86.4 and 873.0 for $k=4, 5$ and 10, respectively.

VII. DISCUSSION

We showed that sequential decoding algorithms have several desired features: Error exponential decay, memory that grows linearly (in contrast to the exponential growth under ML decoding) and expected complexity per branch that grows linearly (similarly to the encoding process of LTI tree codes). However, the complexity distribution is heavy tailed (recall Theorems 3 and 4). This means that there is a substantial probability that the computational complexity is going to be very large, which will cause, in turn, a failure in stabilizing the system. Specifically, by allowing only a finite backtracking length to the past, the computational complexity can be bounded at the expense of introducing an error due to failure.

From a practical point of view, the control specifications of the problem determine a probability of error threshold under which a branch \mathbf{c}_t is considered to be reliable. This can be used to set a limit on the delay-line length of the tree code, which in turn converts it to a convolutional code with a finite delay-line length.

Finally, note that tighter bounds can be derived below the cutoff rate via expurgation. Compared to random block codes, linear codes achieve better exponents than Gallager’s exponent, below the cutoff rate [14]. Indeed, using this technique better bounds under ML decoding were derived in [4] and a similar improvement seems plausible for sequential decoding.

VIII. ACKNOWLEDGMENTS

We thank R. T. Sukhvasi for many helpful discussions.

REFERENCES

- [1] B. Hassibi, A. H. Sayed, and T. Kailath, *Indefinite-Quadratic Estimation and Control: A Unified Approach to H2 and H-infinity Theories*. New York: SIAM Studies in Applied Mathematics, 1998.
- [2] A. Sahai and S. K. Mitter, “The necessity and sufficiency of anytime capacity for stabilization of a linear system over a noisy communication link—part I: Scalar systems,” *IEEE Trans. Inf. Theory*, vol. 52, no. 8, pp. 3369–3395, Aug. 2006.
- [3] L. J. Schulman, “Coding for interactive communication,” *IEEE Trans. Inf. Theory*, vol. 42, pp. 1745–1756, 1996.
- [4] R. T. Sukhvasi and B. Hassibi, “Error correcting codes for distributed control,” *IEEE Trans. Auto. Control*, accepted, Jan. 2016.
- [5] J. M. Wozencraft, “Sequential decoding for reliable communications,” Res. Lab. of Electron., Massachusetts Institute of Technology, Cambridge, MA, USA, Tech. Rep., 1957.
- [6] F. Jelinek, *Probabilistic Information Theory: Discrete and Memoryless Models*. New York: McGraw-Hill, 1968.
- [7] R. G. Gallager, *Information Theory and Reliable Communication*. New York: John Wiley & Sons, 1968.
- [8] A. J. Viterbi and J. K. Omura, *Principles of Digital Communication and Coding*. New York: McGraw-Hill, 1979.
- [9] A. Sahai and H. Palaiyanur, “A simple encoding and decoding strategy for stabilization discrete memoryless channels,” in *Proc. Annual Allerton Conf. on Comm., Control, and Comput.*, Monticello, IL, USA, Sep. 2005.
- [10] F. Jelinek, “Upper bounds on sequential decoding performance parameters,” *IEEE Trans. Inf. Theory*, vol. 20, no. 2, pp. 227–239, Mar. 1974.
- [11] E. Arkan, “An upper bound on the cutoff rate of sequential decoding,” *IEEE Trans. Inf. Theory*, vol. 34, no. 1, pp. 55–63, Jan. 1988.
- [12] A. Khina, W. Halbawi, and B. Hassibi, “(Almost) practical tree codes,” Tech. Rep., Jan. 2016. [Online]. Available: www.its.caltech.edu/~khina/papers/conferences/tree_codes_isit2016.pdf
- [13] N. Shulman and M. Feder, “Improved error exponent for time-invariant and periodically time-variant convolutional codes,” *IEEE Trans. Inf. Theory*, vol. 46, pp. 97–103, 2000.
- [14] A. Barg and G. D. Forney Jr., “Random codes: Minimum distances and error exponents,” *IEEE Trans. Inf. Theory*, vol. 48, no. 9, pp. 2568–2573, Sep., 2002.