(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 an unreliable digital 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 further design these codes to be universal over the class of all binary-input memoryless output-symmetric channels with a given capacity. We supplement our theoretical results with numerical simulations that demonstrate the effectiveness of the decoder in a control system setting.

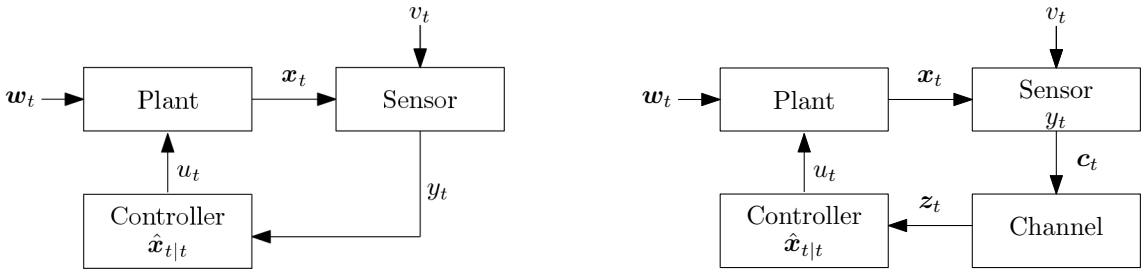
Index Terms

Tree codes, anytime-reliable codes, linear codes, convolutional codes, sequential decoding, networked control, universal codes, compound channel.

The work of A. Khina was supported in part by a Fulbright fellowship, Rothschild fellowship and has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 708932. The work of W. Halbawi was supported by the Qatar Foundation — Research Division. The work of B. Hassibi 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 Director's Fund, by King Abdulaziz University, and by King Abdullah University of Science and Technology. The material in this paper was presented in part at the 2016 IEEE International Symposium of Information Theory.

A. Khina and B. Hassibi are with the Department of Electrical Engineering, California Institute of Technology, Pasadena, CA 91125, USA (e-mail: {khina, hassibi}@caltech.edu).

W. Halbawi was with the Department of Electrical Engineering, California Institute of Technology, Pasadena, CA 91125, USA. He is now with Oracle Corporation, Santa Clara, CA 95054, USA (e-mail: whalbawi@alumni.caltech.edu).



(a) Traditional control system

(b) Networked control system

Fig. 1. Basic traditional and networked control systems.

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 (see Fig. 1a). The theory and practice for this setup are now well established; see, e.g., [1]–[3].

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. 1b).

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

Sahai and Mitter [4] provided necessary and sufficient conditions for 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 reminiscent of the codes developed by Schulman [5] for the related problem of interactive communication.

Unfortunately, the result by Schulman (and consequently also the ones by Sahai and Mitter) only proves the existence of tree codes 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 [6] 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 over binary erasure channels (in which case it amounts to solving linear equations which requires only polynomial computational complexity). Consequently efforts in developing practical anytime reliable codes over other channels have been made, building primarily on the spatially-coupled and convolutional low-density parity-check (LDPC) code constructions [7]–[9].

Sequential decoding was proposed by Wozencraft [10] 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 [11]. For a thorough account of sequential decoding, see [12, Ch. 10], [13, Sec. 6.9], [14, Ch. 6], [15, Ch. 6], [16, Ch. 6.4]. This technique was subsequently adopted by Sahai and Palaiyanur [17] 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 [18, Th. 2] that is valid for the binary symmetric channel (BSC) (and other cases of interest) only when the expected complexity of the sequential decoder goes to infinity.

In this work we revisit the construction of anytime reliable codes through the lens of convolutional codes. This allows us to leverage the existing exponential error bounds for truncated convolutional codes under ML decoding — for both general (non-linear and time-variant) random [14, Sec. 5.6], [15, Sec. 4.8] and random LTI ensembles [19], [20, Chs. 4 and 5] — and rederive the existing results for anytime reliable codes under ML decoding.

We further propose the use of sequential decoding for the recovery of LTI tree codes. To that end, similarly to Sahai and Palaiyanur [17], we extend a (different) result developed by Jelinek [12, Th. 10.2]

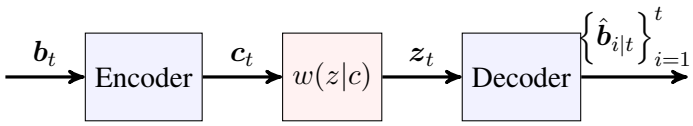


Fig. 2. MBIOS channel with reconstructions of all past information bits.

for general (non-linear and time-variant) random codes to LTI tree codes.

By appealing to the extremal properties of error exponents [21], [22], and the universality properties of the uniform prior [23], [24], we design an anytime reliable code that is good for the whole class of memoryless binary-input output-symmetric (MBIOS) channels with a given capacity — a scenario previously considered by Draper and Sahai [25] for the more stringent variant where, in addition to the encoder, the decoder is also oblivious of the true channel realization.

The rest of the paper is organized as follows. We set and motivate the problem in Section II. We then provide the necessary information theoretic notions and relevant results for capacity and error exponents in Section III. Convolutional codes and their performance under ML decoding and sequential decoding are discussed in Section IV, and are subsequently used in Sections V and VI to construct (universal) anytime reliable tree codes under these decoding procedures. We provide numerical evidence for the effectiveness of the proposed method in Section VII and conclude the paper in Section VIII.

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 $\boldsymbol{x}_t \in \mathbb{R}^m$ (depicted in Fig. 1b; see also Section VII and Fig. 8). The observer then quantizes y_t into k bits: $\boldsymbol{b}_t \in \mathbb{Z}_2^k$, and encodes — using a causal code ((or *tree code*) — all quantized measurements $\{\boldsymbol{b}_i\}_{i=1}^t$ to produce n bits: $\boldsymbol{c}_t \in \mathbb{Z}_2^n$; that is, the encoder uses a code rate of $R = k/n$. This packet $\boldsymbol{c}_t$ is transmitted over a noisy communication link to the controller, which receives $\boldsymbol{z}_t \in \mathcal{Z}^n$, where $\mathcal{Z}$ is the channel output alphabet. The controller then decodes $\{\boldsymbol{z}_i\}_{i=1}^t$ to produce the estimates $\{\hat{\boldsymbol{b}}_{i|t}\}_{i=1}^t$, where $\hat{\boldsymbol{b}}_{i|t}$ denotes the estimate of $\boldsymbol{b}_i$ when decoded at time t. These estimates are mapped back to measurement estimates $\{\hat{y}_{i|t}\}_{i=1}^t$ which, in turn, are used to construct an estimate $\hat{\boldsymbol{x}}_{t|t}$ of the current plant state of the plant. Finally, the controller computes a control signal $\boldsymbol{u}_t$ based on $\hat{\boldsymbol{x}}_{t|t}$ 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 $\boldsymbol{c}_t$,

$$\boldsymbol{c}_t = f_t\left(\{\boldsymbol{b}_i\}_{i=1}^t\right), \tag{1}$$

where $f_t : \{0,1\}^{kt} \to \{0,1\}^n$ is a known encoding function agreed upon by the encoder and the decoder prior to transmission. Since the coded packets depend on the entire history, they can be viewed conveniently as a tree, as illustrated in Fig. 3.

When restricted to linear codes, each function f_t can be characterized by a set of matrices $\{\mathbf{G}_{t,1}, \ldots, \mathbf{G}_{t,t}\}$, where $\mathbf{G}_{t,i} \in \mathbb{Z}_2^{n\times k}$. The sequence of quantized measurements at time t, $\{\boldsymbol{b}_i\}_{i=1}^t$, is encoded as,

$$\boldsymbol{c}_t = \mathbf{G}_{t,1}\boldsymbol{b}_1 + \mathbf{G}_{t,2}\boldsymbol{b}_2 + \cdots + \mathbf{G}_{t,t}\boldsymbol{b}_t\,, \tag{2}$$

or equivalently in matrix form:

$$\boldsymbol{c} = \boldsymbol{\mathcal{G}}_{n;R}\boldsymbol{b},$$

with

$$\boldsymbol{\mathcal{G}}_{n,R} = \begin{bmatrix} \mathbf{G}_{1,1} & \mathbf{0} & \mathbf{0} & \cdots & \cdots & \cdots \\ \mathbf{G}_{2,1} & \mathbf{G}_{2,2} & \mathbf{0} & \cdots & \cdots & \cdots \\ \vdots & \vdots & \ddots & \ddots & \cdots & \cdots \\ \mathbf{G}_{t,1} & \mathbf{G}_{t,2} & \cdots & \mathbf{G}_{t,t} & \mathbf{0} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \ddots \end{bmatrix}, \tag{3a}$$

$$\boldsymbol{b} = \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \vdots \\ \boldsymbol{b}_t \\ \vdots \end{bmatrix}, \quad \boldsymbol{c} = \begin{bmatrix} \boldsymbol{c}_1 \\ \boldsymbol{c}_2 \\ \vdots \\ \boldsymbol{c}_t \\ \vdots \end{bmatrix}. \tag{3b}$$

The decoder computes a function $g_t\left(\{\boldsymbol{z}_i\}_{i=1}^t\right)$ to produce $\{\boldsymbol{b}_{i|t}\}_{i=1}^t$. One is then assigned the task of choosing a sequence of functions $\{g_t | t = 1, \ldots, \infty\}$ (or of matrices $\{\mathbf{G}_{t,i} \,|\, i = 1, \ldots, t\,;\, t = 1, \ldots, \infty\}$ in the linear case) that provides anytime reliability. We recall this definition as stated in [6].

Definition II.1 (Anytime reliability)**.** Define the probability of the first error event at time t and delay d as

$$P_e(t,d) \triangleq P\left(\boldsymbol{b}_{t-d} \neq \hat{\boldsymbol{b}}_{t-d|t}, \forall \delta > d, \boldsymbol{b}_{t-\delta} = \hat{\boldsymbol{b}}_{t-\delta|t}\right), \tag{4}$$

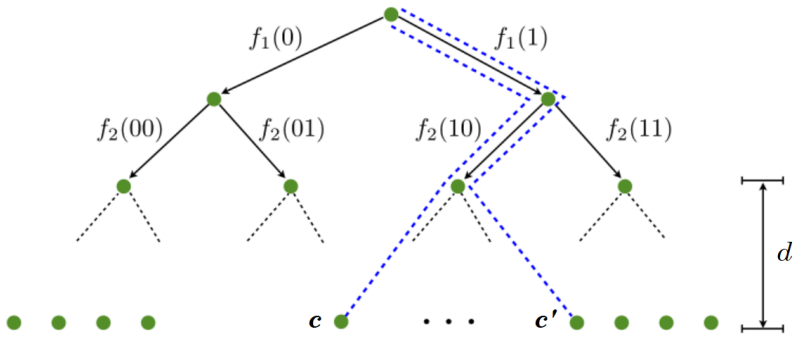


Fig. 3. Binary tree-code illustration with $k = 1$. $\boldsymbol{c}$ and $\boldsymbol{c}'$ are codewords that correspond to information words that are identical during the first two steps and differ in the first step.

where the probability is over the randomness of the information bits $\{b_t\}$ 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 exist $A \in \mathbb{R}$ and $d_0 \in \mathbb{N}$, such that

$$P_e(t, d) \leq A 2^{-\beta n d}, \qquad \forall t, d \geq d_0, \tag{5}$$

where β is called the anytime exponent.

Remark II.1. The requirement of $d \geq d_0$ in (5) can always be dropped, by replacing A by a larger constant. Conversely, A can be replaced with 1 by reducing β by $\epsilon > 0$, however small, and taking a large enough d_0. Nonetheless, we use both A and d_0 in the definition for convenience.

According to the definition, anytime reliability has to hold for every decoding instant t and every delay d. Sukhavasi and Hassibi proposed in [6] 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, in turn, avoids the need to compute a double union bound. We shall explicitly show this later in Section V, where we introduce the LTI ensemble.

We shall concentrate on MBIOS channels, defined formally as follows.

Definition II.2 (MBIOS channel)**.** a *binary-input channel* is a system with binary input alphabet $\{0, 1\}$, output alphabet $\mathcal{Z}$ and two probability transition functions: $w(z|0)$ for input $c = 0$ and $w(z|1)$ for input $c = 1$. It is said to be *memoryless* if the probability distribution of the output depends only on the input at that time and is conditionally independent of previous and future channel inputs and outputs. It is further said to be *output-symmetric* if there exists an involution $\pi : \mathcal{Z} \to \mathcal{Z}$, i.e., a permutation that satisfies

$\pi^{-1} = \pi$, such that

$$w(\pi(z)|0) = w(z|1)$$

for all $z \in \mathcal{Z}$.

III. Preliminaries: Information Theoretic Notions

Throughout this work, we shall concentrate on the class of MBIOS channels. We shall assume that the channel output alphabet $\mathcal{Z}$ has finite size, although the results hold for well-behaved channels with infinite (possibly uncountable) alphabet size, like the binary additive white Gaussian noise (BAWGN) channel [21].

A. Capacities

Definition III.1 (Achievable rate)**.** A rate R is said to be achievable over a channel w if there exists a sequence of mappings ("codes") in n from $\lceil nR \rceil$ information bits ("information word") to n coded bits ("codeword") that are sent over the channel, such that the average bit error probability (BER) of recovering the information bits from the resulting n channel output can be made arbitrarily small for a large enough n.

Definition III.2 (Capacity)**.** The capacity C of a channel w is defined as the supremum over all achievable rates over this channel.

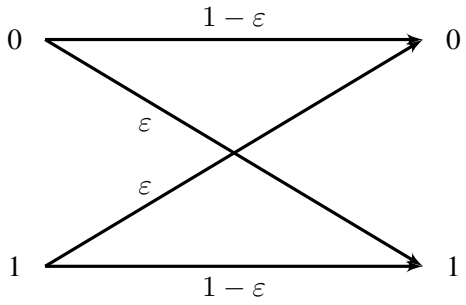
Theorem III.1 (MBIOS channel capacity; see [13], [26])**.** *The capacity* C *of an MBIOS channel* w *is given by*

$$\mathrm{C} = \sum_{z \in \mathcal{Z}} \sum_{c \in \{0,1\}} \frac{1}{2} w(z|c) \log \frac{w(z|c)}{\frac{1}{2}w(z|0) + \frac{1}{2}w(z|1)} \ .$$

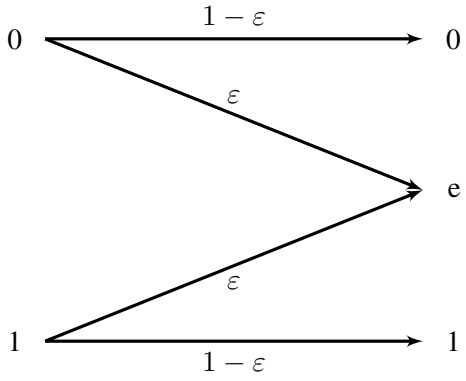
Remark III.1. The common Shannon-theoretic definitions of the achievable rate and capacity with respect to a vanishing *block error probability* coincide with the BER-based definitions used in this work; this easily follows from the source–channel separation theorem [27, Sec. 10.4] with a Hamming distortion measure.

Example III.1 (BSC). The BSC with crossover probability ε, depicted in Fig. 4a, is defined as

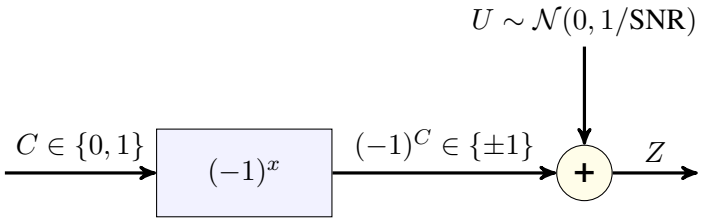
$$w(z|c) = \begin{cases} 1-\varepsilon & c = z \in \{0,1\} \\ \varepsilon & c \neq z; \ c, z \in \{0,1\} \end{cases}$$



(a) BSC



(b) BEC



(c) BAWGN channel

Fig. 4. Three important MBIOS channels: (a) BSC. (b) BEC. (c) BAWGN channel.

Its capacity is equal to

$$\mathrm{C} = 1 - \mathrm{H}_b(\varepsilon),$$

where $\mathrm{H}_b(\varepsilon) \triangleq -\varepsilon \log(\varepsilon) - (1-\varepsilon)\log(1-\varepsilon)$ is the binary entropy function.

Example III.2 (BEC). The binary-erasure channel (BEC) with erasure probability ε, depicted in Fig. 4b, is defined as

$$w(z|c) = \begin{cases} 1-\varepsilon & c = z \in \{0,1\} \\ \varepsilon & z = e \end{cases}$$

Its capacity is equal to

$$\mathrm{C} = 1 - \varepsilon.$$

Example III.3 (BAWGN channel). The BAWGN channel with signal-to-noise ratio (SNR) SNR, depicted in Fig. 4c, is defined as

$$\begin{aligned} w(z|c) &= \mathcal{N}\big((-1)^c, 1/\mathrm{SNR}\big) \\ &= \frac{1}{\sqrt{\tau/\mathrm{SNR}}} \exp\left\{-\frac{(z-(-1)^c)^2}{2/\mathrm{SNR}}\right\}, \end{aligned}$$

for $c \in \{0,1\}$, where $\exp\{x\} = \mathrm{e}^x$ denotes the natural exponential function and $\tau = 2\pi$ is the circle constant. That is,

$$Z = (-1)^C + U, \tag{6}$$

where U is a zero mean Gaussian noise of variance $1/\mathrm{SNR}$ and $C \in \{0,1\}$.

By Theorem III.1, the capacity of this channel is equal to

$$\mathrm{C} = \mathrm{h}\left(Z\right) - \frac{1}{2}\log\left(\tau \mathrm{e}/\mathrm{SNR}\right),$$

where Z is the resulting output in (6) for $q(0) = q(1) = 1/2$ and $\mathrm{h}\left(Z\right)$ denotes its differential entropy:

$$\begin{aligned} \mathrm{h}\left(Z\right) &\triangleq -\int_{-\infty}^{\infty} p(z)\log p(z) dz, \\ p(z) &= \frac{1}{2}\left[w(z|0) + w(z|1)\right]. \end{aligned}$$

We next consider the problem of constructing codes that are simultaneously good for all MBIOS channels with a given capacity. We formally define such classes of channels using the notion of compound channels [28]–[31] (see also [32]).

Definition III.3 (Compound channel)**.** A compound channel comprises a class $\mathcal{W}$ of possible channels. The exact channel transition distribution $w \in \mathcal{W}$ is known to the receiver but not to the the transmitter who only knows the class $\mathcal{W}$.[1]

The class of interest in this paper is that of all MBIOS channels with a given capacity C, denoted by $\mathrm{MBIOS}\left(\mathrm{C}\right)$. The following theorem, whose proof can be found in [28]–[32], states that there is no

[1]Sometimes a channel is said to be compound if the exact channel realization $w \in \mathcal{W}$ is known to neither sides. The capacity in both of these scenarios remains the same. For a further discussion of the difference between these two scenarios see [31, Ch. 4], [32].

tension between the different channels in $\mathrm{MBIOS}\,(\mathrm{C})$, i.e., there exist codes that achieve capacity for all MBIOS channels with a given capacity, simultaneously.

Theorem III.2 (MBIOS channel worst-cast capacity)**.** *The worst-case capacity of the compound channel* $\mathrm{MBIOS}\,(\mathrm{C})$ *is equal to* C*.*

B. Error Exponents

Definition III.4 (Error exponent)**.** A channel w is said to have an error exponent E if there exists a series of codes $\{\mathcal{C}_n\}$ in the blocklength n of rate R, such that their error probability sequence P_e satisfies

$$\mathrm{E} \leq \liminf_{n\to\infty} -\frac{\log P_e}{n}\,.$$

Remark III.2. We use here a more restrictive definition with a $\liminf$ instead of the more common definition that uses a $\limsup$. This restriction is useful for the derivation of anytime reliable codes. Furthermore, the next theorem holds for both of these definitions.

Theorem III.3 (Random coding error exponents [33, Ch. 9], [13, Sec. 5.6], [12, Ch. 7])**.** *Let w be an MBIOS channel, and consider an ensemble of $2^{\lceil nR\rceil}$ codewords of length n each, such that the letters of all the codewords are i.i.d. and uniform. Then, for every information word $1 \leq m \leq 2^{\lceil nR\rceil}$ the average (over the ensemble) decoding error probability under optimal (ML) decoding is bounded from above by*

$$\mathbb{E}\left[P_{e,m}\right] \leq 2^{n\mathrm{E}_G(R)}, \tag{7}$$

where

$$\mathrm{E}_G(R) \triangleq \max_{0\leq\rho\leq 1} \left[\mathrm{E}_0(\rho) - \rho R\right], \tag{8a}$$

with

$$\mathrm{E}_0(\rho) \triangleq -\log\left\{\sum_{z\in\mathcal{Z}}\left[\frac{1}{2}w^{\frac{1}{1+\rho}}(z|0) + \frac{1}{2}w^{\frac{1}{1+\rho}}(z|1)\right]^{1+\rho}\right\} \tag{8b}$$

The error exponent can be attained by linear codes.

Theorem III.4 (Linear random coding error exponents [13, Sec. 6.2])**.** *Let w be an MBIOS channel, and consider an ensemble of $2^{\lceil nR\rceil}$ codewords of length n each, that is generated as follows. For a specific codebook in the ensemble, the information word $\boldsymbol{b}$ is mapped to a codeword $\boldsymbol{c}$ via a linear map:*

$$\boldsymbol{c} = \mathbf{G}\boldsymbol{b},$$

where $\mathbf{G} \in \mathbb{Z}_2^{n\times\lceil nR\rceil}$ is a fixed (for a specific codebook) code generating matrix whose entries (across different codebooks in the ensemble) are i.i.d. and uniform. Then, for information word $m \in \left[1, 2^{\lceil nR\rceil}\right]$,

the average (over the ensemble) decoding error probability under ML decoding is bounded from above as in (7) *with an error exponent of* (8)*.*

Corollary III.1 (Good code generation)**.** *The decoding error probability P_e of a specific code from the random code ensemble of length n and rate R, generated w.r.t. probability distribution q, satisfies*

$$\Pr\left(P_e \geq 2^{n[\mathrm{E}_G(\rho,q)-\epsilon]}\right) \leq 2^{-\epsilon n}.$$

Proof. The proof is immediate by the Markov inequality and the upper bound on the expected value of the error probability (7). □

Remark III.3. Theorems III.3 and III.4 prove that any rate below the capacity is achievable as a consequence, since $\mathrm{E}_G(R)$ is strictly positive for all $R < \mathrm{C}$.

Two useful notions in the context of error exponents are those of the critical rate and the cutoff rate, the significance of which will become apparent in the analysis of sequential decoding in Section IV-B.

Definition III.5 (Critical rate)**.** Let w be an MBIOS channel with $\mathrm{E}_0(\rho)$ as defined in (8b). Then, the critical rate $\mathrm{R}_{\mathrm{crit}}$ is defined as

$$\mathrm{R}_{\mathrm{crit}} \triangleq \mathrm{E}_0'(1) = \left.\frac{d\mathrm{E}_0(\rho)}{d\rho}\right|_{\rho=1}.$$

Definition III.6 (Cutoff rate)**.** Let w be an MBIOS channel with $\mathrm{E}_0(\rho)$ as defined in (8b). Then, the cutoff rate R_0 is defined as

$$\mathrm{R}_0 \triangleq \mathrm{E}_0(1).$$

The critical rate is the rate above which the optimal value of ρ that maximizes $\mathrm{E}_G(R)$ in (8) is strictly lower than 1. The cutoff rate, on the other hand, is the intersection with the R axis ("x axis") of the continuation of the error-exponent straight line (which is attained below the critical rate).

Example III.4 (BSC). The error exponent of the BSC with crossover probability ε (recall Example III.1), depicted in Fig. 5, is equal to

$$\mathrm{E}_G(R) = \max_{0\leq\rho\leq 1}\left[\mathrm{E}_0(\rho) - \rho R\right],$$
$$\mathrm{E}_0(\rho) = \rho - (1+\rho)\log\left\{\varepsilon^{\frac{1}{1+\rho}} + (1-\varepsilon)^{\frac{1}{1+\rho}}\right\};$$

see [34], [13, Sec. 5.6] for further details.

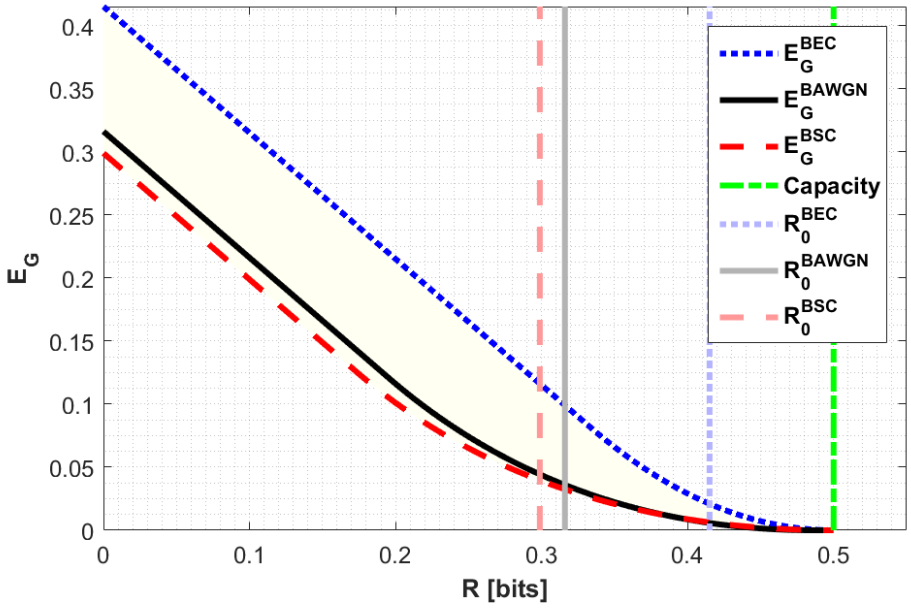


Fig. 5. Error exponents and cutoff rates for $\mathrm{C} = 0.5$ of the BSC, the BAWGN channel and the BEC.

Example III.5 (BEC). The error exponent of the BEC with erasure probability ε (recall Example III.1), depicted in Fig. 5, is equal to

$$\mathrm{E}_G(R) = \max_{0 \le \rho \le 1} \left[\mathrm{E}_0(\rho) - \rho R\right],$$
$$\mathrm{E}_0(\rho) = -\log\left\{\varepsilon + 2^{-\rho}(1-\varepsilon)\right\};$$

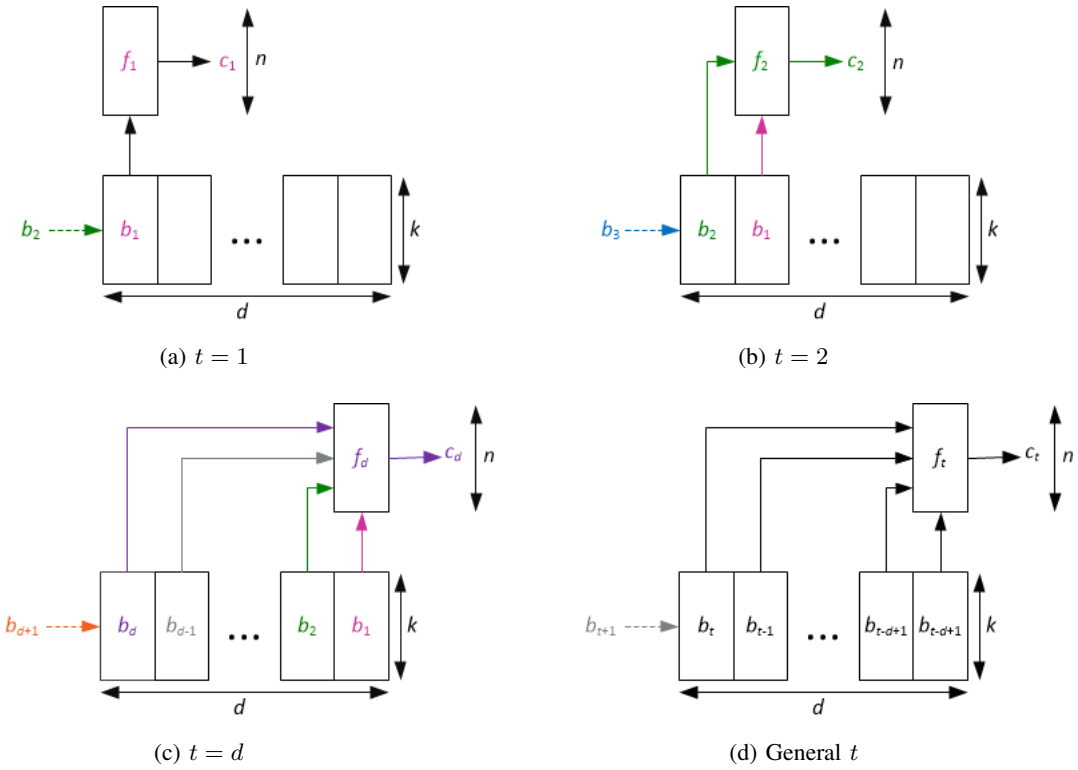
see [34] for further details.

Example III.6 (BAWGN channel). The error exponent of the BAWGN channel with a given SNR (recall Example III.3), depicted in Fig. 5, is given by

$$\mathrm{E}_G(R) = \max_{0 \le \rho \le 1} \left[\mathrm{E}_0(\rho) - \rho R\right],$$
$$\mathrm{E}_0(\rho) = -\log\left(\int_{-\infty}^{\infty} \left[\frac{1}{2} w^{\frac{1}{1+\rho}}(z|0) + \frac{1}{2} w^{\frac{1}{1+\rho}}(z|1)\right]^{1+\rho} dz\right)$$

The following theorem, illustrated also in Fig. 5, states that the error exponents of the BSC and the BEC with a given capacity C, are the worst and the best, respectively, out of the whole class of MBIOS channels with the same capacity C; we denote these channels by $\mathrm{BSC}(\mathrm{C})$ and $\mathrm{BEC}(\mathrm{C})$, respectively.

Theorem III.5 (Extremality of error exponents [21], [22])**.** *Let w be any MBIOS channel with capacity* C *and* $\mathrm{E}_0^w(\rho)$ *defined as in* (8b)*. Denote by* $\mathrm{E}_0^{\mathrm{BSC}}(\rho)$ *the* E_0 *of* (8b)*, of* $\mathrm{BSC}(\mathrm{C})$*, and by* $\mathrm{E}_0^{\mathrm{BEC}}(\rho)$ *the* E_0 *of* (8b)*, of* $\mathrm{BEC}(\mathrm{C})$*. Then, the following relations hold*



(a) $t = 1$ (b) $t = 2$

(c) $t = d$ (d) General t

Fig. 6. Tree code generation via a shift register.

$$\mathrm{E}_0^{\mathrm{BSC}}(\rho) \le \mathrm{E}_0^{w}(\rho) \le \mathrm{E}_0^{\mathrm{BEC}}(\rho) \tag{9}$$

for all $\rho \in [0, 1]$, *and consequently also*

$$\mathrm{R}_0^{\mathrm{BSC}} \le \mathrm{R}_0^{w} \le \mathrm{R}_0^{\mathrm{BEC}},$$

$$\mathrm{E}_G^{\mathrm{BSC}}(R) \le \mathrm{E}_G^{w}(R) \le \mathrm{E}_G^{\mathrm{BEC}}(R), \qquad \forall R \in [0, \mathrm{C}].$$

for all $R \in [0, \mathrm{C}]$.

IV. Review of Convolutional Codes

In this section we review known results for several random ensembles of convolutional codes and connect such codes to tree codes, in Section IV-A. The codes within each ensemble can be either linear or not; linear ensembles can further be either time-varying or time invariant. We further discuss sequential decoding algorithms and their performance in Section IV-B which will be applied 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, as depicted in Fig. 6: 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 (we consider the newest k information bits as being part of the the delay-line, for convenience). 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 (including 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-varying convolutional codes.

We observe that this representation is in fact equivalent to that of (1) with the encoding function $f_t : \{0,1\}^{dk} \to \{0,1\}^n$ at time t being a function of only the last d packets, i.e.,

$$\boldsymbol{c}_t = f_t\left(\{\boldsymbol{b}_i\}_{i=t-d+1}^{t}\right), \tag{10}$$

with $f_t : \{0,1\}^{dk} \to \{0,1\}^n$.

In this work we shall concentrate on linear functionals, resulting in linear convolutional codes. Linear convolutional codes are given as in (2)–(3), with

$$\mathbf{G}_{i,j} = 0, \qquad\qquad i \geq j + d, \tag{11}$$

namely,

$$\boldsymbol{\mathcal{G}}_{n;R} = \begin{bmatrix} \mathbf{G}_{1,1} & \mathbf{0} & \mathbf{0} & \cdots & \cdots & \cdots \\ \mathbf{G}_{2,1} & \mathbf{G}_{2,2} & \mathbf{0} & \cdots & \cdots & \cdots \\ \vdots & \vdots & \ddots & \ddots & \vdots & \cdots \\ \mathbf{G}_{d,1} & \mathbf{G}_{d,2} & \cdots & \mathbf{G}_{d,d} & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{G}_{d+1,2} & \cdots & \mathbf{G}_{d+1,d} & \ddots & \ddots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{G}_{d+2,d} & \ddots & \ddots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}. \tag{12}$$

For LTI convolutional codes, we further have

$$\mathbf{G}_{i,j} = \mathbf{G}_{i+1,j+1} = \cdots = \mathbf{G}_{j+d-1,j+d-1} \triangleq \mathbf{G}_{i-j+1} \ , \tag{13}$$

that is,

$$\boldsymbol{\mathcal{G}}_{n;R} = \begin{bmatrix} \mathbf{G}_1 & \mathbf{0} & \mathbf{0} & \cdots & \cdots & \cdots \\ \mathbf{G}_2 & \mathbf{G}_1 & \mathbf{0} & \cdots & \cdots & \cdots \\ \vdots & \vdots & \ddots & \ddots & \vdots & \cdots \\ \mathbf{G}_d & \mathbf{G}_{d-1} & \cdots & \mathbf{G}_1 & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{G}_d & \cdots & \mathbf{G}_2 & \ddots & \ddots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{G}_3 & \ddots & \ddots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}. \tag{14}$$

As is suggested in Section III, using linear codes incurs no loss in performance provided that the optimal prior is uniform — which is the case for $\mathrm{MBIOS}\,(\mathrm{C})$, and is more attractive in practice.

We shall 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 V, 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-line length d. Since each time step corresponds to n uses of the communication link, the relevant regime for the current work is $N = nd$.

We next define the three convolutional code ensembles that will be used in this work.

Definition IV.1 (General convolutional code ensemble)**.** A general (non-linear) convolutional code ensemble of rate $R = k/n$, that maps k information bits into n bits at every step, has a delay-line length of d, where the functions f_t in (10) are generated uniformly at random and independently of each other.

Definition IV.2 (LTV convolutional code ensemble)**.** An LTV convolutional code ensemble of rate $R = k/n$, and delay-line length d, that maps kd information bits into n bits at every step, where the entries in all $\{\mathbf{G}_{i,j}\}$ of $\boldsymbol{\mathcal{G}}_{n,R}$ of (12) are i.i.d. and uniform.

Definition IV.3 (LTI convolutional code ensemble)**.** An LTI convolutional code ensemble of rate $R = k/n$ and delay-line length d, that maps kd information bits into n bits at every step, where the entries in all $\{\mathbf{G}_i\}$ of $\boldsymbol{\mathcal{G}}_{n;R}$ of (14) are i.i.d. and uniform.

Theorem IV.1 ([14, Sec. 5.6], [15, Sec. 4.8])**.** *Let w be an MBIOS channel, and consider the LTV convolutional code ensemble of Definition IV.2 with rate R and delay-line length d. Then, the probability of the first error event* (5) *at any time t and delay that is equal to the delay-line length d of the code,*

under optimal (ML) decoding, is bounded from above by

$$\bar{P}_e(t,d) \leq 2^{-\mathrm{E}_G(R)nd}, \tag{15}$$

with $\mathrm{E}_G(R)$ *of* (8)*.*

Remark IV.1. 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 $\mathrm{E}_G(R)$ [14, Ch. 5]. Unfortunately, this result does not hold for the case of $N = nd$ which is the relevant regime for this work.

Interestingly, whereas LTV codes are known to achieve better error exponents than LTI ones when $N \gg d$, this gain vanishes when $N = nd$, as is suggested by the following theorem.

Theorem IV.2 ([19, Eq. (14)], [20, Chs. 4 and 5])**.** *Let* w *be an MBIOS channel, and consider the LTI convolutional code ensemble of Definition IV.2 with rate* R *and delay-line length* d*. Then, the probability of the first error event* (5) *at any time* t *and delay that is equal to the delay-line length* d *of the code, under optimal (ML) decoding, is bounded as in* (15)*.*

Thus, (15) remains valid for LTI codes.

Remark IV.2. Somewhat surprisingly, allowing larger delays for decoding does not improve the bound on the probability of the first error event (15) in the case of LTI codes, using the existing techniques; see [19], [20, Chs. 4 and 5] for details.

Unfortunately, the computational complexity of ML decoding grows exponentially with the delay-line length d, prohibiting its use in practice for large values of d. We therefore next review a suboptimal decoding procedure, the complexity of which does not grow rapidly with d but still achieves exponential decay in d of the probability of the first error event.

B. Sequential Decoding

The Viterbi algorithm [14, Sec. 4.2] offers an efficient implementation of (frame-wise) ML decoding[2] for fixed d and growing N. Unfortunately, the complexity of this algorithm grows exponentially with N when the two are coupled, i.e., $N = nd$.[3] Prior to the adaptation of the Viterbi algorithm as the preferred decoding algorithm of convolutional codes, sequential decoding had served as the *de facto* standard. A

[2]For bitwise ML decoding, the BCJR algorithm [35] needs to be used.

[3]This is true with the exception of the binary erasure channel, for which ML decoding amounts to solving a system of equations, the complexity of which is polynomial.

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 the delay-line length d, and are therefore applicable for the decoding of tree codes.[4]

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., [12, Ch. 10]) to compare possible codeword paths:

$$M(\boldsymbol{c}_1, \ldots, \boldsymbol{c}_N) = \sum_{t=1}^{N} M(\boldsymbol{c}_t), \tag{16a}$$

$$M(\boldsymbol{c}_t) \triangleq \log \frac{w(\boldsymbol{z}_t|\boldsymbol{c}_t)}{p(\boldsymbol{z}_t)} - nB, \tag{16b}$$

where B is referred to as the metric bias and penalizes longer paths when the metrics of different-length paths are compared.

Example IV.1. Consider the BSC of Example III.1. Let $\boldsymbol{c}_{1:j}$ and $\boldsymbol{z}_{1:j}$ denote the codeword and received word up to the j^{th} time-step, respectively. Let δ be the Hamming distance between $\boldsymbol{c}_{1:j}$ and $\boldsymbol{z}_{1:j}$. Then, the metric associated with the pair $\boldsymbol{c}_{1:j}$ and $\boldsymbol{z}_{1:j}$ is given by

$$M(\boldsymbol{c}_{1:j}) = \delta \left[\log(\varepsilon) + 1 - B\right] + (nj - \delta) \left[\log(1-\varepsilon) + 1 - B\right].$$

In contrast to ML decoding, where all possible paths (of length N) are explored to determine the path with the total maximal metric,[5] using the stack sequential decoding algorithm, a list of partially explored paths is stored in a priority queue, where at each step the path with the highest metric is further explored and replaced with its immediate descendants and their metrics. The Stack Algorithm is outlined in Algorithm 1 and implemented in [36], [37]. 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 description of both algorithms and variants thereof, see [12, Ch. 10], [13, Sec. 6.9], [14, Ch. 6], [15, Ch. 6].

The choice $B = R$ is known to minimize the expected computational complexity, and is therefore the most popular choice in practice. Moreover, for rates below the cutoff rate $R < \mathrm{R}_0$, the expected number of metric evaluations (16b) is finite and does not depend on d, for any $B \leq \mathrm{R}_0$ [13, Sec. 6.9], [12, Ch. 10]. Thus, the only increase in expected complexity of this algorithm with d comes from an

[4]Interestingly, the idea of tree codes was conceived and used already in the early works on sequential decoding [10]. These codes were used primarily for the classical communication problem, and not for interactive communication or control.

[5]Note that optimizing (16a) in this case is equivalent to ML decoding.

increase in the complexity of evaluating the metric of a single symbol (16b). 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, $R > \mathrm{R}_0$, the expected complexity is known to grow rapidly with N for *any metric* [11], implying that the algorithm is applicable only for rates below the cutoff rate.[6]

Algorithm 1 Stack Algorithm

$Q \leftarrow$ `MaxPriorityQueue()` ▷ Maintain a list explored nodes by decreasing metric

`metric` $\leftarrow 0$

Q.`push(root, metric)`

while Q.`top().level` $< d - 1$ **do** ▷ Continue until a leaf is reached

`current_node, current_metric` $\leftarrow Q$.`pop()` ▷ Explore path with largest metric

for `child` $\in$ `current_node.children` **do**

$t \leftarrow$ `child.level`

`child_metric` $\leftarrow$ `current_metric` $+$ COMPUTEMETRIC(`child`, $\boldsymbol{z}_t$) ▷ Compute child's metric using (16b)

Q.`push(child, child_metric)`

return Q.`top().codeword` ▷ Reconstruct the input sequence by backtracking

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 IV.3 ([12, Ch. 10])**.** *Let w be an MBIOS channel, and consider the random general (non-linear) convolutional code ensemble of Definition IV.1 with rate R and delay-line length d. Then, the probability of the first error event* (II.1) *at any time t and delay that is equal to the delay-line length d of the code, using the Fano or stack sequential decoders and the Fano metric with bias B, is bounded from above by:*

$$\bar{P}_e(t,d) \leq A\, 2^{-\mathrm{E}_J(B,R)nd}, \tag{17a}$$

$$\mathrm{E}_J(B,R) \triangleq \max_{0\leq\rho\leq 1} \frac{\rho}{1+\rho}\Big\{\mathrm{E}_0(\rho) + B - (1+\rho)R\Big\}, \tag{17b}$$

[6]This holds for *any* convolutional code for rates above the cutoff rate, $R > \mathrm{R}_0$.

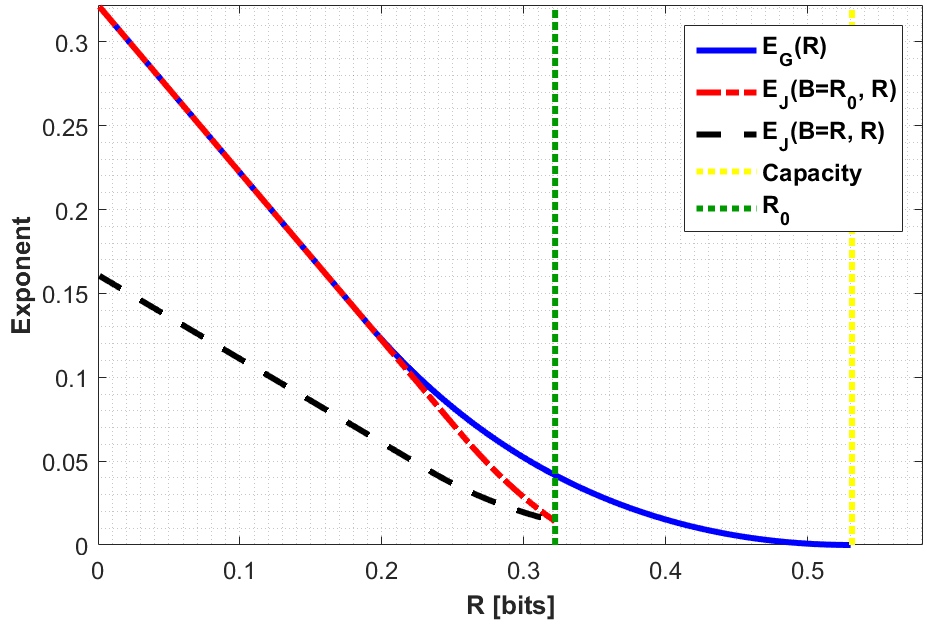


Fig. 7. Error exponents $\mathrm{E}_G(R)$, $\mathrm{E}_J(B=\mathrm{R}_0, R)$ and $\mathrm{E}_J(B=R, R)$ for a BSC with crossover probability $p = 0.1$.

where A is finite for $B < \mathrm{R}_0$ and is bounded from above by[7]

$$A \leq e^{\frac{\rho}{1+\rho}\Delta} \frac{1 - e^{-t[\mathrm{E}_0(\rho)-\rho B]}}{1 - e^{-[\mathrm{E}_0(\rho)-\rho B]}} \leq \frac{e^{\frac{\rho}{1+\rho}\Delta}}{1 - e^{-[\mathrm{E}_0(\rho)-\rho B]}} < \infty, \tag{18}$$

for a quantization parameter Δ in the Fano algorithm; for the stack algorithm, (18) *holds with $\Delta = 0$.*

Since $\mathrm{E}_J(B, R)$ is a monotonically increasing function of B, choosing $B = \mathrm{R}_0$ maximizes the exponential decay of $\bar{P}_e(d)$ in d.[8] Interestingly, for this choice of bias, we have $\mathrm{E}_J(B = \mathrm{R}_0, R) = \mathrm{E}_G(R)$ whenever $\mathrm{E}_G(R)$ is achieved by $\rho = 1$ in (8), i.e., for rates below the critical rate $\mathrm{R}_{\mathrm{crit}}$ (recall Definition III.5). For other values of ρ, $\mathrm{E}_J(B = \mathrm{R}_0, R)$ is strictly smaller than $\mathrm{E}_G(R)$ (see Fig. 7).

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 (15) is achieved whenever $\mathrm{E}_G(R)$ is achieved by $\rho = 1$: $\mathrm{E}_J(B = R, R) \geq \mathrm{E}_G(R)/2$ (see Fig. 7).

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

Definition IV.4. Denote by W_t the number of branch computations of node t performed by a sequential decoding algorithm.

[7]Note that $\mathrm{E}_0(\rho)/\rho$ is a monotonically decreasing function of ρ, therefore $B < \mathrm{R}_0 = \mathrm{E}_0(1)$ guarantees that $\mathrm{E}_0(\rho) - \rho B > 0$.

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

We note that W_t is a random variable (which depends on the received vector and the underlying 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 random general convolutional codes, random LTV convolutional codes and LTI random convolutional codes.

The next result provides a lower bound on the complexity.

Theorem IV.4 ([38]; see also [14, Sec. 6.4])**.** *Let w be an MBIOS channel. Then, the probability that W_t is greater than $m \in \mathbb{N}$ for any convolutional code, where no decoding error occurs under sequential decoding, is bounded from below by*

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

where $o(m) \to 0$ for $m \to \infty$, and $R = \mathrm{E}_0(\rho)/\rho$ for any $\rho > 0$.[9]

This bound was proved to be tight by Savage [39] for random general convolutional codes.

Theorem IV.5. *Let w be an MBIOS channel. Then, the probability that W_t is greater than $m \in \mathbb{N}$ for a random general convolutional code with infinite delay-line length, under sequential decoding, is bounded from above by*

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

where A is finite for $B, R < \mathrm{R}_0$, $R < \frac{B+\mathrm{R}_0}{2\rho}$, $\rho \in (0, 1]$, and R_0 is the cutoff rate (recall Definition III.6).

This result is widely believed to be true for random LTV and LTI convolutional codes, although no formal proof exists to date. Nonetheless, the following weaker result was established for LTV convolutional codes [14, Sec. 6.2] and for LTI convolutional codes [13, Sec. 6.9].

Theorem IV.6. *Let w be an MBIOS channel. Then, the probability that W_t is greater than $m \in \mathbb{N}$ for a random LTV convolutional code and a random LTI convolutional code with infinite delay-line lengths, under sequential decoding, is bounded from above by*

$$\Pr\left(W_t \geq m\right) \leq \frac{A}{m},$$

where A is finite for $B, R < \mathrm{R}_0$, $R < \frac{B+\mathrm{R}_0}{2\rho}$, $\rho \in (0, 1]$, and R_0 is the cutoff rate.

An immediate consequence of Theorems IV.5 and IV.6 is that the expected complexity per branch $E[W_t]$ is bounded for $R < \mathrm{R}_0$. Moreover, a converse result by Arıkan [11] states that the expected complexity is unbounded for rates exceeding the cutoff rate.

[9]Recall that $\mathrm{E}_0(\rho)/\rho$ is a decreasing function of ρ and therefore $\rho > 1$ implies that $R < \mathrm{R}_0$.

V. Linear Time Invariant Anytime Reliable Codes

As we alluded to in Section IV, tree codes are no more than convolutional codes with an infinite delay-line length. This simple observation, allows us to utilize all the results of Section IV to tree codes.

We start by recalling the construction of LTI anytime reliable (tree) codes, proposed first in [6]. The generating matrix of an LTI tree code satisfies (14) with $d \to \infty$; that is, it is of the form

$$\boldsymbol{\mathcal{G}}_{n,R} = \begin{bmatrix} \mathbf{G}_1 & \mathbf{0} & \mathbf{0} & \cdots & \cdots \\ \mathbf{G}_2 & \mathbf{G}_1 & \mathbf{0} & \cdots & \cdots \\ \vdots & \vdots & \ddots & \ddots & \cdots \\ \mathbf{G}_t & \mathbf{G}_{t-1} & \cdots & \mathbf{G}_1 & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}. \tag{19}$$

Similarly to the LTI convolutional code ensemble of Definition IV.3, we define the following ensemble, which can again be viewed as its infinite delay-line length limit.

Definition V.1 (LTI tree code ensemble)**.** An LTI tree ensemble of rate $R = k/n$, that maps kt information bits into n bits at time step t, where the entries in all $\{\mathbf{G}_i\}$ of $\boldsymbol{\mathcal{G}}_{n;R}$ of (19) are i.i.d. and uniform.

Remark V.1. In contrast to the ensemble considered by Sukhavasi and Hassibi [6], we do not require the first generating matrix $\mathbf{G}_1$ to be a full-rank matrix in our analysis.

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

Sukhavasi and Hassibi [6] proved that the LTI ensemble is anytime reliable, as defined in (5), with high probability. We next prove this result using Theorem IV.2.

Theorem V.1 (Anytime-reliable LTI tree codes)**.** *Let w be an MBIOS channel with error exponent $\mathrm{E}_G(R)$ (8). Let $\epsilon > 0$ and $d_0 \in \mathbb{N}$. Then, the probability that a particular code from the random LTI tree code ensemble of Definition V.1 has an anytime exponent* (5) *of $\mathrm{E}_G(R) - \epsilon$, for all $t \in \mathbb{N}$ and $d > d_0$, under optimal (ML) decoding, is bounded from below by*

$$\Pr\left(\bigcap_{t=1}^{\infty}\bigcap_{d=d_0}^{t}\left\{P_e(t,d) \le 2^{-[\mathrm{E}_G(R)-\epsilon]nd}\right\}\right) \ge 1 - \frac{2^{-\epsilon n d_0}}{1-2^{-\epsilon n}}$$

Thus, for any $\epsilon > 0$, however small, this probability can be made arbitrarily close to 1 by taking d_0 to be large enough.

Proof. By using the Markov inequality along with the result of Theorem IV.2, the probability that a particular code from the ensemble has an exponent that is strictly smaller than $\mathrm{E}_G(R)-\epsilon$ at time t and delay d is bounded from above by

$$\Pr\left(P_e(t,d) \geq 2^{-(\mathrm{E}_G(R)-\epsilon)nd}\right) \leq 2^{-\epsilon nd}. \tag{20}$$

For a code to be anytime reliable it needs to satisfy (5) for *every* t and $d_0 \leq d \leq t$.

To that end, we note that the time-invariance property suggests that for a fixed d, the event

$$\left\{P_e(t,d) \geq 2^{-(\mathrm{E}_G(R)-\epsilon)nd}\right\}$$

is identical for all t. Thus, by applying the union bound and (20) to

$$\bigcup_{t=1}^{\infty}\bigcup_{d=d_0}^{t}\left\{P_e(t,d) \geq 2^{-(\mathrm{E}_G(R)-\epsilon)nd}\right\} \tag{21}$$

gives rise to

$$\Pr\left(\bigcup_{t=1}^{\infty}\bigcup_{d=d_0}^{t}\left\{P_e(t,d) \geq 2^{-(\mathrm{E}_G(R)-\epsilon)nd}\right\}\right) = \Pr\left(\bigcup_{d=d_0}^{\infty}\left\{P_e(t,d) \geq 2^{-(\mathrm{E}_G(R)-\epsilon)nd}\right\}\right) \tag{22a}$$

$$\leq \sum_{d=d_0}^{\infty} 2^{-\epsilon nd} \tag{22b}$$

$$= \frac{2^{-\epsilon nd_0}}{1-2^{-\epsilon n}}\,. \tag{22c}$$

As a result, a large enough d_0 guarantees that a specific code selected at random from the LTI tree code ensemble achieves (5) with exponent $\beta = \mathrm{E}_G(R)-\epsilon$, for all t and $d_0 \leq d \leq t$, with high probability. □

Remark V.2 (LTV codes). This proof technique fails for LTV tree codes: For a code to be anytime reliable it needs to satisfy (5) for *every* t and $d_0 \leq d \leq t$. Unfortunately, applying the union bound and (20) to (21) gives a trivial upper bound, for LTV codes. The advantage of using LTI codes is that for a fixed d, the event $\left\{P_e(t,d) \geq 2^{-(\mathrm{E}_G(R)-\epsilon)nd}\right\}$ is identical for all t which reduces the union bound to a summation with respect to only d (22a) instead of a double summation over d and t.

The following extends this result to $\mathrm{MBIOS}\,(\mathrm{C})$, and is similar to the technique proposed in [40] for the construction of LDPC code ensembles that universally achieve capacity under belief propagation decoding.

Corollary V.1 (Universal anytime-reliable LTI tree codes)**.** *Let $\mathcal{W}$ be any (finite) subset of* $\mathrm{MBIOS}\,(\mathrm{C})$ *and denote by* $\mathrm{E}_G^{\mathrm{BSC}}(R)$ *the random error exponent* (8) *of* $\mathrm{BSC}(C)$*. Let $\epsilon > 0$ and $d_0 \in \mathbb{N}$. Then, the probability that a particular code from the random LTI tree code ensemble of Definition V.1 has an*

anytime exponent (5) *of* $\mathrm{E}_G^{\mathrm{BSC}}(R) - \epsilon$*, for all* $t \in \mathbb{N}$ *and* $d > d_0$*, and for all channels* $w \in \mathcal{W}$*, under optimal (ML) decoding, is bounded from above by*

$$\Pr\left(\bigcap_{w\in\mathcal{W}}\bigcap_{t=1}^{\infty}\bigcap_{d=d_0}^{t}\left\{P_e^w(t,d) \le 2^{-[\mathrm{E}_G^{\mathrm{BSC}}(R)-\epsilon]nd}\right\}\right) \ge 1 - \frac{2^{-\epsilon n d_0}|\mathcal{W}|}{1-2^{-\epsilon n}} \ ,$$

where $P_e^w(t,d)$ *denotes* $P_e(t,d)$ *of* (4) *with respect to the channel* w*.*

Thus, for any $\epsilon > 0$*, however small, this probability can be made arbitrarily close to* 1 *by taking* d_0 *to be large enough.*

This corollary can also be extended to include (countably and uncountably) infinite sets of channels in a similar fashion to the analysis carried by Kuderkar et al. [41].

VI. 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 (17) under sequential decoding for the random general (non-linear) convolutional code ensemble of Theorem IV.3, holds true also for LTI convolutional codes which, for $N = nd$, identifies with the LTI tree codes of (19).

To prove this, we adopt the proof technique of Theorem III.4 in [13, Sec. 6.2].

Theorem VI.1. *Let* w *be an MBIOS channel, and consider the random LTI convolutional code ensemble of Definition IV.1 with rate* R *and delay-line length* d*. Then, the probability of the first error event* (5) *at any time* t *and delay that is equal to the delay-line length* d *of the code, using the Fano or stack sequential decoders and the Fano metric with bias* B*, is bounded from above by* (17)*.*

Proof. A thorough inspection of the proof of Theorem IV.3, as it appears in [12, 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 prove that these two requirements are met for the affine linear ensemble, which can be regarded, in turn, as a special case of [12, Th. 10.7]. The codes in this ensemble are as in Definition V.1 up to an additive translation:

$$\boldsymbol{c} = \boldsymbol{\mathcal{G}}_{n;R}\boldsymbol{b} + \boldsymbol{v},$$

$$\boldsymbol{v} = \begin{bmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \\ \vdots \\ \boldsymbol{v}_t \\ \vdots \end{bmatrix},$$

where $\boldsymbol{v}_t \in \mathbb{Z}_2^n$; the entries in all $\{\mathbf{G}_i\}$ of $\boldsymbol{\mathcal{G}}_{n;R}$ (19) and $\boldsymbol{v}$ are i.i.d. and uniform.

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

To establish the second requirement we note that the addition of a random uniform translation vector $\boldsymbol{v}$ 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 [12, Ch. 10].

Finally note that, since the channel is MBIOS, the same error probability is achieved for any translation vector $\boldsymbol{v}$, including the all-zero vector; this concludes the proof. □

This theorem implies also the following results which parallel their ML decoding counterparts in Theorem V.1 and Corollary V.1, with essentially the same proofs.

Theorem VI.2 (Anytime-reliable LTI tree codes)**.** *Let w be an MBIOS channel with error exponent $\mathrm{E}_J(R)$* (17b) *under sequential decoding. Let $\epsilon > 0$ and $d_0 \in \mathbb{N}$. Then, the probability that a particular code from the random LTI tree code ensemble of Definition V.1 has an anytime exponent* (5) *of $\mathrm{E}_J(R) - \epsilon$, for all $t \in \mathbb{N}$ and $d > d_0$, under sequential decoding, is bounded from below by*

$$\Pr\left(\bigcap_{t=1}^{\infty}\bigcap_{d=d_0}^{t}\left\{P_e(t,d) \leq A 2^{-[\mathrm{E}_J(R)-\epsilon]nd}\right\}\right) \geq 1 - \frac{2^{-\epsilon n d_0}}{1-2^{-\epsilon n}}$$

where A is bounded from above as in (18)*.*

Thus, for any $\epsilon > 0$, however small, this probability can be made arbitrarily close to 1 by taking d_0 to be large enough.

The following is an immediate consequence of the definition of E_J (17b) and the extremality property (9) of E_0.

Corollary VI.1 (sequential-decoding exponent extremalities)**.** *Let w be any MBIOS channel with capacity* C *and* $\mathrm{E}_J^w(\rho)$ *defined as in* (17b)*. Denote by* $\mathrm{E}_J^{\mathrm{BSC}}(\rho)$ *the* E_J *of* (17b)*, of* $\mathrm{BSC}(\mathrm{C})$*, and by* $\mathrm{E}_J^{\mathrm{BEC}}(\rho)$ *the* E_J *of* (17b)*, of* $\mathrm{BEC}(\mathrm{C})$*. Then, the following relations hold*

$$\mathrm{E}_J^{\mathrm{BSC}}(R) \le \mathrm{E}_J^w(R) \le \mathrm{E}_J^{\mathrm{BEC}}(R),$$

for all $R \in [0, \mathrm{C}]$.

Applying Corollary VI.1 to Theorem VI.2 gives rise to the following universality result.

Corollary VI.2 (Universal anytime-reliable LTI tree codes)**.** *Let $\mathcal{W}$ be any (finite) subset of* $\mathrm{MBIOS}\,(\mathrm{C})$ *and denote by* $\mathrm{E}_J^{\mathrm{BSC}}(R)$ *the random error exponent* (17b) *of* $\mathrm{BSC}(C)$*. Let* $\epsilon > 0$ *and* $d_0 \in \mathbb{N}$*. Then, the probability that a particular code from the random LTI tree code ensemble of Definition V.1 has an anytime exponent* (5) *of* $\mathrm{E}_J^{\mathrm{BSC}}(R) - \epsilon$*, for all* $t \in \mathbb{N}$ *and* $d > d_0$*, and for all channels* $w \in \mathcal{W}$*, under sequential decoding, is bounded from above by*

$$\Pr\left(\bigcap_{w\in\mathcal{W}}\bigcap_{t=1}^{\infty}\bigcap_{d=d_0}^{t}\left\{P_e^w(t,d) \le A^{\mathrm{BSC}}2^{-[\mathrm{E}_G^{\mathrm{BSC}}(R)-\epsilon]nd}\right\}\right) \ge 1 - \frac{2^{-\epsilon n d_0}|\mathcal{W}|}{1-2^{-\epsilon n}}\ ,$$

where $P_e^w(t,d)$ *denotes* $P_e(t,d)$ *of* (4) *with respect to the channel w, and* A^{BSC} *is bounded as A in* (18) *with* E_0 *of* $\mathrm{BSC}(\mathrm{C})$.

Thus, for any $\epsilon > 0$*, however small, this probability can be made arbitrarily close to* 1 *by taking* d_0 *to be large enough.*

VII. Numerical Results

A. Simulation of a Control System

To demonstrate the effectiveness of the sequential decoder in stabilizing an unstable plant driven by bounded noise, depicted in Fig. 8, we simulate a cart–stick balancer controlled by an actuator that obtains noisy measurements of the angle (partial state) of the cart. We consider three channel scenarios: a BSC, an AWGN channel and a BEC, and use the same anytime-reliable LTI tree code for all of them. All channels are of capacity $C = 0.85$. The example is the same one from [6] which is originally from [42]. The plant dynamics evolve as,

$$\boldsymbol{x}_{t+1} = \mathbf{A}\boldsymbol{x}_t + \mathbf{B}\boldsymbol{u}_t + \boldsymbol{w}_t$$

$$y_t = \mathbf{C}\boldsymbol{x}_t + v_t,$$

where $\boldsymbol{u}_t$ is the control input signal that depends only on the estimate of the current state, i.e., $\boldsymbol{u}_t = -\mathbf{K}\hat{\boldsymbol{x}}_{t|t}$. The system noise $\boldsymbol{w}_t$ is a vector of i.i.d. Gaussian random variables with mean $\mu = 0$ and

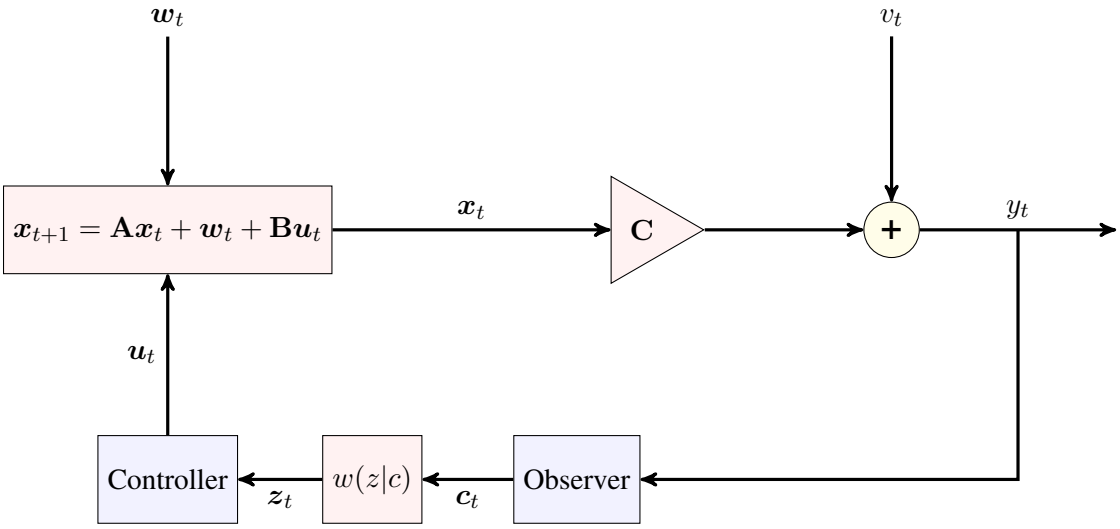


Fig. 8. Linear control system with a noisy-channel feedback.

variance $\sigma^2 = 0.01$, truncated to $[-0.025, 0.025]$. The measurement noise v_t is also a truncated Gaussian random variable of the same parameters. We assume that the system is in observer canonical form:

$$\mathbf{A} = \begin{bmatrix} 3.3010 & 1 & 0 \\ -3.2750 & 0 & 1 \\ 0.9801 & 0 & 0 \end{bmatrix}, \tag{23a}$$

$$\mathbf{B} = \begin{bmatrix} -0.0300 \\ -0.0072 \\ 0.0376 \end{bmatrix}, \tag{23b}$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, \tag{23c}$$

$$\mathbf{K} = \begin{bmatrix} 55.6920 & 32.3333 & 19.0476 \end{bmatrix}. \tag{23d}$$

The state $\boldsymbol{x}_t$, 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 LTI tree code used is designed for the BSC with capacity $C = 0.85$, which has a cutoff rate $\mathrm{R}_0 = 0.6325$. We fix a budget of $n = 25$ channel uses. Using [6, Theorem 8.1], the minimum required number of quantization bits is $k_{\min} = 3$ and the minimum required exponent is $\beta_{\min} = 0.1641$.

For the three scenarios, we use a code of rate $R = 1/5$, with $k = 5$ bits for a lattice quantizer with bin width $\delta = 0.07$. From (17a), a sequential decoder with bias $B = R$ will guarantee an error exponent of $\beta = 0.2163 > \beta_{\min}$. As is evident from the curves in Fig. 9, the stick on the cart does not deviate by

more than a few degrees in all three scenarios.

B. Stability versus Coding Rate Trade-off

One might wonder whether a lower code rate results in improved stability (lower cost due to channel coding) of the dynamical system in question. We note that for a fixed budget of n channel uses, a lower code rate translates to a coarser quantizer and consequently a larger cost due to quantization. As a result, this might suggest a tension between the quantization level being used and the performance of the tree code. To exhibit this behavior, we evolve the same dynamical system presented in Section VII-A using three different quantization levels. The dynamical system is evolved for a total of $T = 100$ time steps, each corresponding to $n = 20$ channel uses, where the channel is a BSC with crossover probability 0.01. 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

$$J_T \triangleq \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\left(Q_t \left\|\boldsymbol{x}_t\right\|^2 + R_t \left\|\boldsymbol{u}_t\right\|^2\right)\right], \tag{24}$$

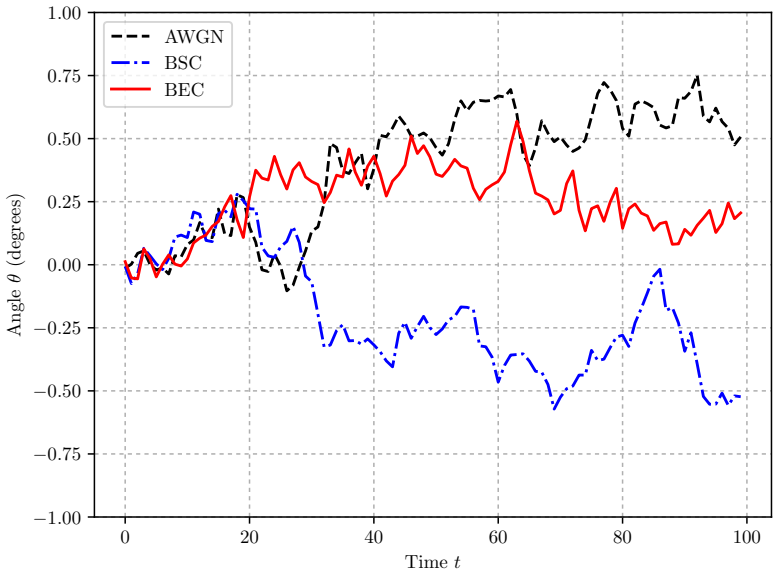
where the expectation is with respect to the randomness of the plant and the channel. The LQR weights $\{Q_t\}$ and $\{R_t\}$ penalize the state deviation and actuation effort, respectively. For our example, we used $Q_t \equiv R_t \equiv 1/2$. We average the results over 100 codes per quantization level and 40 experiments per code.

The data is tabulated as follows.

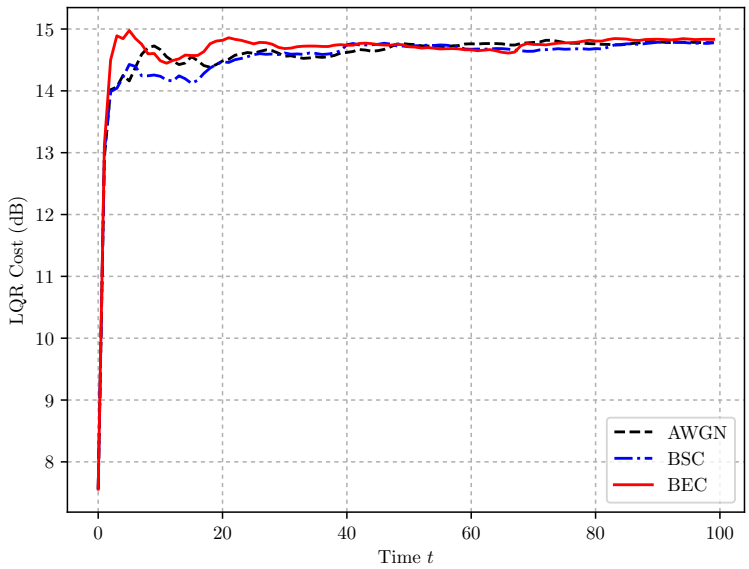
k	LQR Cost
4	206.0
5	86.4
10	873.0

For the setup considered, the minimum number of quantization bits is $k_{\mathsf{min}} = 3$. Nonetheless, we can see how one extra quantization level (going from 4 bits to 5) results in a significant decrease in the stabilization effort.

Remark VII.1. In principle, one would randomly sample an LTI generator matrix where each subblock $\mathbf{G}_i \in \mathbb{Z}_2^{n\times k}$. Nonetheless, from an implementation efficiency point of view, there is no loss in the anytime exponent if we pick $\mathbf{G}'_i \in \mathbb{Z}_2^{n'\times k'}$, where $n = n'\mathsf{gcd}(n,k)$ and $k = k'\mathsf{gcd}(n,k)$, and then encode the information sub-blocks of size k' using $\{\mathbf{G}'_i\}$.



(a) The empirical average angle deviation versus time. The plot shows that this deviation is minimal.



(b) The empirical LQR cost J_t (24) versus time t for weights $Q_t \equiv R_t \equiv \frac{1}{2}$.

Fig. 9. Stability of the Cart–Stick Balancer is demonstrated using a sequentially decoded LTI tree code with $k = 5$ and $n = 20$ that was generated at random and designed for the plant described by (23) when the measurements are transmitted across a BSC with capacity $C = 0.85$. This code is also used over the AWGN and the BEC with the same capacity. The plots are a result of averaging 50 experiments of random $\{\boldsymbol{w}_t, \boldsymbol{v}_t\}$ for the same code.

VIII. Discussion

A. Random Complexity

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 IV.4, IV.5 and IV.6). This means that there is a substantial probability that the computational complexity will 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 $\boldsymbol{c}_t$ 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.

B. Expurgated Error Exponents

Tighter bounds can be derived below the cutoff rate via expurgation. Moreover, random linear block codes are known to achieve the expurgated bound for block codes (with no need in expurgation) [43]. Indeed, using this technique, better bounds under ML decoding were derived in [14, Sec. 5.3] for LTV convolutional codes and in [6] — for LTI tree codes (see also [15, Ch. 4]); similar improvement is plausible for sequential decoding.

We further note that the extremality properties of Theorem III.5 carry on to the expurgated error exponent, i.e., the BEC and the BSC have the best and worst expurgated error exponents, respectively, out of the whole class of $\mathrm{MBIOS}\,(\mathrm{C})$ [21], [22].

C. Asymmetric Channels

Interestingly, the loss due to using a uniform prior instead of the capacity achieving one for memoryless binary-input asymmetric channels is no more than 0.011 bits and no more than 5.8% [23], [24], and is the largest for the Z-channel, for any fixed capacity [23].

Furthermore, for a given *symmetric capacity* [26] — the highest achievable rate with the additional constraint that the channel-input letters are used with equal probability — the extremality properties of Theorem III.5 extend to the wider class of all memoryless binary-input channels with a given symmetric capacity [22].

These results may further allow to extend the results of this paper beyond the realm of symmetric channels and are left for future research.

D. Separated and Joint Source–Channel Coding Schemes

The separation between the source-compression and the channel-coding tasks is very attractive, both conceptually and in practice, and is known to be optimal for point-to-point (one-way) communication when large encoding and decoding delays are allowed [44, Ch. 3.9].

Tree codes allow to extend the source–channel coding separation paradigm to control in conjunction with suitable compression (see [45], [46] and references therein).

Unfortunately, the optimality guarantees for source–channel coding separation do not hold for the case of short (not to mention zero) delays, which are essential due to the real-time nature of control.

Furthermore, in order to guarantee bounded larger moments, higher anytime exponents (5) are required [4]. Consequently, only a finite number of moments can be made bounded using tree codes.

Both of these issues can be circumvented by the use of joint source–channel mappings, which may offer a boost in both stability and computational complexity, as was recently demonstrated in [47] for the simple case of a scalar control system and a Gaussian communication channel. Exploring this direction further remains an interesting direction for future research.

ACKNOWLEDGMENT

The authors thank Dr. R. T. Sukhavasi for many helpful discussions.

REFERENCES

[1] K. J. Åström, *Introduction to Stochastic Control Theory*. New York, NY, USA: Academic Press, 1970, vol. 70.

[2] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, 3rd ed. Belmont, MA, USA: Athena Scientific, 2007, vol. I and II.

[3] B. Hassibi, A. H. Sayed, and T. Kailath, *Indefinite-Quadraric Estimation and Control: A Unified Approach to H2 and H-infinity Theories*. New York: SIAM Studies in Applied Mathematics, 1998.

[4] 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.

[5] L. J. Schulman, "Coding for interactive communication," *IEEE Trans. Inf. Theory*, vol. 42, pp. 1745–1756, 1996.

[6] R. T. Sukhavasi and B. Hassibi, "Error correcting codes for distributed control," *IEEE Trans. Autom. Control*, accepted, Jan. 2016.

[7] L. Grosjean, L. K. Rasmussen, R. Thobaben, and M. Skoglund, "Systematic LDPC convolutional codes: Asymptotic and finite-length anytime properties," *IEEE Trans. Inf. Theory*, pp. 4165–4183, Dec. 2014.

[8] M. Noor-A-Rahim, K. D. Nguyen, and G. Lechner, "Anytime reliability of spatially coupled codes," *IEEE Trans. Comm.*, pp. 1069–1080, Apr. 2015.

[9] N. Zhang, M. Noor-A-Rahim, B. N. Vellambi, and K. D. Nguyen, "Anytime characteristics of protograph-based LDPC convolutional codes," *IEEE Trans. Inf. Theory*, pp. 4047–4069, Oct. 2016.

[10] J. M. Wozencraft, "Sequential decoding for reliable communications," MIT Research Lab. of Elect., Cambridge, MA, USA, Tech. Rep., 1957.

[11] E. Arıkan, "An upper bound on the cutoff rate of sequential decoding," *IEEE Trans. Inf. Theory*, vol. 34, no. 1, pp. 55–63, Jan. 1988.

[12] F. Jelinek, *Probabilistic Information Theory: Discrete and Memoryless Models*. New York: McGraw-Hill, 1968.

[13] R. G. Gallager, *Information Theory and Reliable Communication*. New York: John Wiley & Sons, 1968.

[14] A. J. Viterbi and J. K. Omura, *Principles of Digital Communication and Coding*. New York: McGraw-Hill, 1979.

[15] R. Johannesson and K. S. Zigangirov, *Fundamentals of Convolutional Coding*. New York: Wiley-IEEE Press, 1999.

[16] J. M. Wozencraft and I. M. Jacobs, *Principles of Communication Engineering*. New York: John Wiley & Sons, 1965.

[17] 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.

[18] F. Jelinek, "Upper bounds on sequential decoding performance parameters," *IEEE Trans. Inf. Theory*, vol. 20, no. 2, pp. 227–239, Mar. 1974.

[19] 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.

[20] N. Shulman, "Coding theorems for structured code families," Master's thesis, Tel-Aviv University, Sep. 1995.

[21] A. Guillen i Fabregas, I. Land, and A. Martinez, "Extremes of error exponents," *IEEE Trans. Inf. Theory*, vol. 59, pp. 2201–2207, 2013.

[22] M. Alsan, "Extremality for Gallager's reliability function E_0," *IEEE Trans. Inf. Theory*, vol. 61, no. 8, pp. 4277–4292, Aug. 2015.

[23] N. Shulman and M. Feder, "The uniform distribution as a universal prior," *IEEE Trans. Inf. Theory*, vol. 50, no. 6, pp. 1356–1362, June 2004.

[24] E. E. Majani and H. Rumsey, "Two results on binary-input discrete memoryless channels," in *Proc. IEEE Int. Symp. on Inf. Theory (ISIT)*, Budapest, Hungary, June 1991, p. 104.

[25] S. C. Draper and A. Sahai, "Universal anytime coding," Limassol, Cyprus, Apr. 2007.

[26] E. Arıkan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. Inf. Theory*, vol. 55, pp. 3051–3073, July 2009.

[27] T. M. Cover and J. A. Thomas, *Elements of Information Theory, Second Edition*. New York: Wiley, 2006.

[28] R. L. Dobrushin, "Optimal information transmission over a channel with unknown parameters," (in Russian) *Radiotekh. i Elektron.*, vol. 4, no. 12, pp. 1951–1956, Dec. 1959.

[29] J. Wolfowitz, "Simultaneous channels," *Arch. Rational Mech. Anal.*, vol. 4, no. 1, pp. 371–386, Jan. 1959.

[30] D. Blackwell, L. Breiman, and A. J. Thomasian, "The capacity of a class of channels," *The Annals of Math. Stat.*, vol. 30, pp. 1229–1241, Dec. 1959.

[31] J. Wolfowitz, *Coding Theorems of Information Theory*. New York: Springer-Verlag, 1964.

[32] A. Lapidoth and P. Narayan, "Reliable communication under channel uncertainty," *IEEE Trans. Inf. Theory*, vol. 44, pp. 2148–2177, 1998.

[33] R. M. Fano, *Transmission of Information*. Cambridge:M.I.T. Press and New York: John Wiley and Sons, 1961.

[34] P. Elias, "Coding for noisy channels," *IRE Concention Record*, vol. 4, pp. 37–46, 1955. Also appears in *Key Papers in the Development of Information Theory*, Ed. D. Slepian, IEEE Press, 102-111, 1974.

[35] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inf. Theory*, vol. IT-20, pp. 284–287, Mar. 1974.

[36] A. Khina, W. Halbawi, and B. Hassibi, "Sequential stack decoder: Python implementation," Jan. 2016. [Online]. Available: http://www.its.caltech.edu/~khina/code/tree_codes.py

[37] E. Riedel Gårding, "Sequential stack decoder: Python implementation," Aug. 2017. [Online]. Available: https://github.com/eliasrg/SURF2017/tree/master/code/separate/coding/convolutional

[38] I. M. Jacobs and E. R. Berlekamp, "A lower bound to the distribution of computation for sequential decoding," *IEEE Trans. Inf. Theory*, vol. 13, no. 2, pp. 167–174, Apr. 1967.

[39] J. E. Savage, "Sequential decoding — the computation problem," *Bell Sys. Tech. Jour.*, vol. 45, no. 1, pp. 149–175, Jan. 1966.

[40] A. Khina, Y. Yona, and U. Erez, "LDPC ensembles that universally achieve capacity under BP decoding: A simple derivation," in *Proc. IEEE Int. Symp. on Inf. Theory (ISIT)*, Hong Kong, June 2015, pp. 1074–1078.

[41] S. Kudekar, T. J. Richardson, and R. Urbanke, "Spatially coupled ensembles universally achieve capacity under belief propagation," *IEEE Trans. Inf. Theory*, submitted, June 2013. [Online]. Available: http://arxiv.org/abs/1201.2999

[42] G. Franklin, J. D. Powell, and A. Emami-Naeini. New Jersey: Pearson Prentich Hall, 2006.

[43] 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.

[44] A. El Gamal and Y.-H. Kim, *Network Information Theory*. Cambridge University Press, 2011.

[45] S. Yüksel, "Stochastic stabilization of noisy linear systems with fixed-rate limited feedback," *IEEE Trans. Autom. Control*, vol. 55, no. 12, pp. 2847–2853, Dec. 2010.

[46] A. Khina, Y. Nakahira, Y. Su, and B. Hassibi, "Algorithms for optimal control with fixed-rate feedback," in *Proc. IEEE Conf. Decision and Control (CDC)*, Dec. 2018. [Online]. Available: www.its.caltech.edu/~khina/papers/conferences/fixed_rate_lqg_cdc2017.pdf

[47] A. Khina, G. M. Pettersson, V. Kostina, and B. Hassibi, "Multi-rate control over awgn channels via analog joint source-channel coding," in *Proc. IEEE Conf. Decision and Control (CDC)*, Las Vegas, NV, USA, Dec. 2017, pp. 5968–5973.