Error-Correcting Codes for Automatic Control

Rafail Ostrovsky* Yuval Rabani† Leonard J. Schulman‡

July 18, 2005

Abstract

In many control-theory applications one can classify all possible states of the device by an infinite state graph with polynomially-growing expansion. In order for a controller to control or estimate the state of such a device, it must receive reliable communications from its sensors; if there is channel noise, the encoding task is subject to a stringent real-time constraint. We show a constructive on-line error correcting code that works for this class of applications. Our code is is computationally efficient and enables on-line estimation and control in the presence of channel noise. It establishes a constructive (and optimal-within-constants) analog, for control applications, of the Shannon coding theorem.

*Computer Science Department, University of California at Los Angeles, 90095, USA. Part of this work was done while at the Institute for Pure and Applied Mathematics (IPAM). Supported in part by a gift from Teradata, Intel equipment grant, NSF Cybertrust grant, OKAWA research award and B. John Garrick Foundation. Email: rafail@cs.ucla.edu

†Computer Science Department, Technion — Israel Institute of Technology, Haifa 32000, Israel. Work supported by Israel Science Foundation grant number 52/03 and by United States-Israel Binational Science Foundation grant number 2002282. Part of this work was done while visiting the Institute for Pure and Applied Mathematics in the University of California at Los Angeles. Email: rabani@cs.technion.ac.il

‡California Institute of Technology. Supported in part by an NSF ITR and the Okawa Foundation. Email: schulman@caltech.edu

1 Introduction

Motivation. In many automatic control applications, a device (an engine, a terrestrial or aerial mobile robot, etc.) communicates with a base station that controls its actions. The communication may be wireless or wired, synchronous or packet-based. Typically the devices have a limited set of commands/ controls/ actions/ moves that they can execute. Actions by the devices combine with environmental disturbances, to cause a change in the parameters describing the state of the system (such as location, orientation, or temperature). Such devices need to communicate with the base station regarding their current state and get further instructions. Examples are numerous, and include remote mobility issues (such as space or submarine exploration) and web-based on-line control (such as camera and sensor distributed control) [9, 5].

If the controller is physically remote from the sensors or actuators, information flow between them can be subject to noise; if so, system performance depends upon encoding the transmissions against channel noise. In control applications, the encoding of communications against channel noise faces a special difficulty due to the need for real-time response to transmissions. The objective of the base station is to learn as precisely as possible the current state of each device in its parameter space. Naturally, there is a tradeoff between the amount of communication (and hence delay) and the accuracy and reliability of the information known at the base station. It is therefore a challenge to perform the channel coding subject to a channel capacity constraint.

The problem can be considered within a very general framework of interactive communication problems [11]; however, the best results in that literature remain nonconstructive. Fortunately, there is a feature of the control application that makes it easier than general interactive-communication problems, since the controlled devices can typically be described with a finite-dimensional parameter space. (Example: the location, orientation and engine RPM of an aerial drone.) What characterizes a typical parameter space is that the growth rate of the state space around any point is polynomially bounded.

At each step in its state-space the remote device wishes to send one (or a constant number) of bits to the base station to indicate its position/configuration. Despite channel-noise, the objective of the base-station is to determine, as accurately as possible, the location of the device in its state-space. Of course, one cannot ask that the base station already have high certainty about the real value of any measured bit, before a significant number of subsequent message bits have been received. More specifically, if the channel has a constant rate of stochastic noise, then the best one can hope for (on non-degenerate noisy channels) is that the base station have probability $\exp(-\Omega(n))$ of estimating incorrectly a particular state of a device, if all histories leading to that state diverge from the true history at least n steps previously. The meaningful question is: Can we achieve such a bound? Doing so demands that encoded characters convey information across all time scales. This is exactly what we achieve in this paper in a constructive fashion, as we explain below.

Problem statement and results. In this paper, we initiate the study of error-correcting codes for remote control of devices that move in a finite-dimensional parameter space. All of the communication systems we discuss share the following features. There is at least one transmitter and one receiver. The state of the transmitter at any time t is identified with a vertex (which we denote x_t) of a state graph (which we denote G); the graph (which may be directed or undirected and will typically have self-loops) is known to both parties, as is the initial state x_0 of the transmitter. In each round, the state of the transmitter shifts to an out-neighbor of the previous state. The transmitter can then use the channel once; the communicated character can depend upon the entire history of the transmitter. Our concern is the design of an efficient code for these communications.

For nodes $x, x' \in G$ let d_G be the length of a shortest path from x to x' in G. Let $B(x, \ell) = \{x' : d_G(x, x') \leq \ell\}$. The growth of G as a function of ℓ is the supremum over all x of $|B(x, \ell)|$. If this is bounded above by a polynomial in ℓ we say G has polynomial growth. Finite-dimensional grids have polynomial growth. We suppose that the alphabet of the channel is a finite set S. S^* denotes the set of finite words over S. If the greatest out-degree or in-degree of G is Δ, we say that the *rate* of the code is $\rho = (\log \Delta)/(\log |S|)$. (We assume below that $\Delta \geq 2$.) We expressly avoid tailoring our results to particular kinds of noisy channels. Our results are aimed at noisy but non-adversarial channels, in particular discrete memoryless channels, for which we assume only that the capacity is proportional to $\log |S|$. (We remark that, using cryptography, we can extend our results to polynomially-bounded adversarial channels [8].)

Based upon the code and upon the history of communications, the receiver has at time t a guess $\hat{x}_t$ of the current state of the transmitter. (We understand the code to include the estimation procedure used by the receiver.) We say that the code has *error exponent* κ if $P(d_G(x_t, \hat{x}_t) \geq \ell) \leq \exp(-\kappa\ell)\ \forall t, \ell$. We say that the code is *time-efficient* if the encoding and expected decoding times are $(\log t)^{O(1)}$. It is *time-and-space-efficient* if the space required for encoding, and the expected space required for decoding, are also $(\log t)^{O(1)}$.

We show the existence of asymptotically good error-correcting codes for every state graph G. Our main result is the *construction* of a code for communication in finite-dimensional grid graphs that has positive rate, positive error exponent, and is time-and-space-efficient. The method extends to other graphs with polynomial growth and simply-represented global structure, but we do not formalize this in this extended abstract.

Previous work. Standard error-correcting block codes are not satisfactory for remote control applications because low probabilities of error (necessary in order to maintain system performance over extended periods of time) can be achieved only using long blocks, and this introduces a time lag that degrades the effectiveness of controller response.

Two forms of affirmative answer to our problem are known. One is to use convolutional codes [18, 10, 3] or other randomized protocols [11]. This has a significant drawback: these "codes" are not strictly speaking codes, but probability distributions over codes, and require the encoder and decoder to use shared random bits. (Certain kinds of deterministic convolution codes are currently in wide use but these have only bounded support, i.e. fixed-time memory. They therefore cannot provide error-bounded performance for extended time periods.) The second answer is to use tree codes [12, 13]. These are true error correcting codes, but they too cannot presently be used in long protocols because no efficient algorithm is known for constructing them. (A similar situation reigned for block codes after Shannon's existence proof for asymptotically-good block codes [14] until explicit constructions were provided [6, 4].) There has recently been substantial progress in information-theoretic and rate-distortion bounds for control applications [17, 2, 15, 16, 7]; these works solve different problems than the one considered here. There does not appear to be a prior code for our problem that is efficient in both computation and communication.

2 Trajectory codes

Throughout, G is a graph with vertex set V, initial vertex $x_0 \in V$, and edge set $E \subseteq V \times V$. A trajectory γ of length $|\gamma| = t$ and which begins at time t_0 is a mapping from $\{t_0, \ldots, t_0+t\}$ to V for which all $(\gamma(i), \gamma(i+1)) \in E$. If two trajectories γ, γ' are of equal length, start at the same time t_0, and share the same start vertex (i.e., $\gamma(t_0) = \gamma'(t_0)$), we write $\gamma \sim \gamma'$. The distance τ between trajectories $\gamma \sim \gamma'$ of length t is $\tau(\gamma, \gamma') = |\{t_0 < i \leq t_0+t : \gamma(i) \neq \gamma'(i)\}|$.

A *trajectory code* is a mapping $\chi : V \times \{1, 2, \ldots\} \to S$, extended to a mapping from trajectories to S^* by concatenation: $\chi(\gamma) = (\chi(\gamma(t_0+1)), \ldots, \chi(\gamma(t_0+t)))$. Hamming distance between equal-length words in S^* is denoted h. The *relative distance* of the code is defined to be $\delta = \inf_{\gamma \sim \gamma'} \{h(\chi(\gamma), \chi(\gamma'))/\tau(\gamma, \gamma')\}$. A finite-time trajectory code is defined similarly by a mapping $\chi : V \times \{1, 2, \ldots, T\} \to S$.

We say that the code is *asymptotically good* if it has both positive rate ρ and positive relative distance δ.

Lemma 1. *If the t'th character of an asymptotically good code with alphabet S can be computed in time and space $(\log t)^{O(1)}$ then the code can be converted into another asymptotically good code that has positive error exponent and is time-and-space-efficient.*

Proof. The conversion is by simple repetition (the alphabet of the new code is S^k for constant k), and serves only to improve the error exponent. For sufficiently high error exponent, decoding by max-likelihood matching is exponentially unlikely to need to examine trajectories far away from that decoded in the previous round. Hence the expected time and space of the computation is $(\log t)^{O(1)}$. □

Our task therefore is to construct an asymptotically good trajectory code. The first problem is to show that such codes exist (Section 3). Interestingly, the only proof we know is non-constructive; however, with the aid of this proof we provide a constructive and time-and-space-efficient finite-time code for grids. (Section 4).

Comparison with tree-codes It is instructive to compare the present work with that on tree codes. In the terminology of the present paper, [12, 13] used the protocol tree of a given noiseless communication protocol in the role of our graph G; the tree code used in that work for a noisy-communication protocol is what we call the trajectory code on $V \times \{1, 2, \ldots\}$. The existence proof provided in that work relies on the tree structure of the graph, and does not apply to the more general case considered here. However, the purpose of the generalization is *not* just handling more difficult communication problems; the case that G is a tree is, in fact, the most difficult one. (Using tree codes enables eventual reconstruction of the entire history of the transmitter, not only reconstruction of a good estimate of the current state.) Instead, the purpose in our paper is to obtain a computationally effective solution using the special assumption that G has polynomial growth. This assumption is motivated by control applications, with G being a discretization of the finite-dimensional parameter space of the system. Thus, we circumvent the need to construct an explicit tree code and show that a different code which that works for the entire class of polynomial-growth graphs is sufficient.

3 Existence of asymptotically good trajectory codes

Theorem 2. *Every graph G possesses an asymptotically good trajectory code. Furthermore, every $\delta < 1$ is feasible as the relative distance of an asymptotically good code.*

Proof. To achieve positive rate we must label $V \times \{1, 2, \ldots\}$ with an alphabet S of size $\Delta^{O(1)}$. Consider choosing each label independently and uniformly. A code obtained in this way is almost-surely not asymptotically good. Nonetheless this probability space can be used for an existence proof.

Consider at first the finite-graph, finite-time restriction of the problem to $B(x_0, T) \times \{1, 2, \ldots, T\}$. Fix any desired relative distance bound δ. If $\gamma = (\gamma_1, \gamma_2)$ consists of two trajectories such that $\gamma_1 \sim \gamma_2$ and which share only their common start vertex (i.e., $\tau(\gamma_1, \gamma_2) = |\gamma_1|$), then we refer to γ as a pair of "twins" and write $|\gamma| = |\gamma_1|$ and $h\chi(\gamma) = h(\chi(\gamma_1), \chi(\gamma_2))$. Note that $\inf_{\gamma_1 \sim \gamma_2}(h(\chi(\gamma_1), \chi(\gamma_2))/\tau(\gamma_1, \gamma_2)) = \inf_{\text{twins } \gamma}(h\chi(\gamma)/|\gamma|)$. For a pair of twins γ let A_γ be the event that $h\chi(\gamma)/|\gamma| < \delta$. There is a positive c for which $P(A_\gamma) \leq |S|^{-c|\gamma|}$.

For twins $\gamma = (\gamma_1, \gamma_2)$ let $N_\gamma = \{\text{twins } \beta = (\beta_1, \beta_2) : \exists \epsilon_1, \epsilon_2 \in \{1, 2\}, j_1, j_2 > 0 \text{ such that } \gamma_{\epsilon_1}(j_1) = \beta_{\epsilon_2}(j_2)\}$.

Observe that A_γ is independent of the random variable $(A_\beta)_{\beta \notin N_\gamma}$.

The Lovász local lemma [1] ensures that $\bigcap \overline{A_\gamma} \neq \emptyset$ provided that there exist nonnegative reals $0 \leq x_\gamma < 1$ for which

$$x_\gamma \prod_{\beta \in N_\gamma} (1 - x_\beta) \geq P(A_\gamma).$$

Observe that $|\{\beta : \beta \in N_\gamma, |\beta| = \ell\}| \leq 4|\gamma|\ell\Delta^{2\ell}$. For c' to be determined set $x_\gamma = \Delta^{-c'|\gamma|}$. Now,

$$x_\gamma \prod_{\beta \in N_\gamma} (1 - x_\beta) \geq \Delta^{-c'|\gamma|} \prod_{\ell=1}^{\infty} (1 - \Delta^{-c'\ell})^{4|\gamma|\ell\Delta^{2\ell}}.$$

A sufficiently large c' ensures that for $\Delta \geq 2$, $1 - \Delta^{-c'\ell} \geq e^{-2\Delta^{-c'\ell}}$. So

$$\ldots \geq \Delta^{-c'|\gamma|} \prod_{\ell=1}^{\infty} e^{-8\Delta^{-c'\ell}|\gamma|\ell\Delta^{2\ell}} = \Delta^{-c'|\gamma|} e^{-8|\gamma| \sum_{\ell=1}^{\infty} \ell\Delta^{(2-c')\ell}}.$$

A sufficiently large c' ensures that for $\Delta \geq 2$, $\sum_{\ell=1}^{\infty} \ell \Delta^{(2-c')\ell} \leq 2$. So

$$\ldots \geq \Delta^{-c'|\gamma|} e^{-16|\gamma|}.$$

Since $P(A_\gamma) \leq |S|^{-c|\gamma|}$, the hypotheses of the local lemma are met with an alphabet of size $\Delta^{O(1)}$.

To extend the proof to the general case we apply a standard compactness argument (see [1]). For any T, the trajectory codes on $B(x_0, T) \times \{1, 2, \ldots, T\}$ ensured by the above argument form a finite nonempty set. Let C_T denote the set of codes on $V \times \{1, 2, \ldots\}$ which restrict to one of the trajectory codes on $B(x_0, T) \times \{1, 2, \ldots, T\}$. C_T is a nonempty set that is closed in the product topology on $S^{V \times \{1,2,\ldots\}}$. Note that $C_T \subseteq C_{T-1}$; the intersection of the sets C_T for any finite number of indices T is therefore nonempty. The set $\bigcap_{t \in \mathbb{N}} C_T$ is the desired set of trajectory codes with relative distance δ. By Tychonoff's Theorem, $S^{V \times \{1,2,\ldots\}}$ is compact. Therefore $\bigcap_{t \in \mathbb{N}} C_T \neq \emptyset$. □

4 Construction of trajectory codes for grids

We now construct an asymptotically good and time-and-space-efficient finite-time trajectory code, of any desired relative distance $\delta < 1$, for a grid graph of arbitrary finite dimension d.

Let P_n denote the path of length n, with vertices labeled $\{-n/2 + 1, \ldots, n/2\}$. Let G be the graph on vertex set $V_{n,d} = \{-n/2 + 1, \ldots, n/2\}^d$ with an edge from $(u_1, \ldots, u_d)$ to $(v_1, \ldots, v_d)$ if $|u_i - v_i| \leq 1 \forall i$. For simplicity we describe the construction for a time bound of $n/2$. So our task is to construct for a trajectory code $\chi : V_{n,d} \times \{1, \ldots, n/2\} \to S$ of relative distance δ.

The idea is to combine recursion with use of an explicit block code. Set $n_1 \in \Theta(\log n)$. Let $k \sim 1/(1 - \delta)$. For simplicity assume that kn_1 divides n.

4.1 Recursive construction

The block code: Let $\eta : V_{n,d} \to R_1^{n_1}$ (for a finite alphabet R_1) be an asymptotically good block code of relative distance $(1 + \delta)/2$, in which encoding and decoding can be performed in time $n_1^{O(1)}$. Rewrite η as a mapping $\eta_1 : V_{n,d} \times \{1, \ldots, n_1/2\} \to R_1$, so that for $x \in V_{n,d}$, $\eta(x) = (\eta_1(x, 1), \ldots, \eta_1(x, n_1/2))$.

The recursive code: Let $\chi_1 : V_{kn_1,d} \times \{1, \ldots, kn_1/2\} \to S_1$ (for a finite alphabet S_1) be a trajectory code of relative distance $(1 + \delta)/2$.

The basic idea is to cover $V_{n,d} \times \{1, \ldots, n/2\}$ by overlapping "shingles". Each shingle is "placed" at a specified $x \in V_{n,d} \times \{0, \ldots, n/2 - 1\}$, and is the following mapping:

$$\sigma_x : \left(\prod_{i=1}^{d} \{x_i - kn_1/2 + 1, \ldots, x_i + kn_1/2\} \right) \times (x_{d+1} + 1, \ldots, x_{d+1} + kn_1/2) \to S_1 \times R_1$$

$$\sigma_x(y) = (\chi_1(y - x), \eta_1(x_1, \ldots, x_d, (y_{d+1} - x_{d+1} \bmod n_1)))$$

The cover of $V_{n,d} \times \{1, \ldots, n/2\}$ by overlapping shingles will be described by a union of several covers, each of which is a tiling (a cover by nonoverlapping shingles). Each tiling is associated with a vector $(a_1, \ldots, a_{d+1}) \in \{-k/2+1, \ldots, k/2\}^d \times \{0, \ldots, k-1\}$. (Strictly speaking each tiling may fail to be a cover but only due to edge effects which we gloss over.) The collection of shingles associated with the label $(a_1, \ldots, a_{d+1})$ consists of those placed at x of the form

$$x = n_1(kz_1 + a_1, \ldots, kz_{d+1} + a_{d+1}),$$

for all $(z_1, \ldots, z_{d+1})$ of the form

$$(z_1, \ldots, z_{d+1}) \in \{-n/(2kn_1)+1, \ldots, n/(2kn_1)\}^d \times \{1, \ldots, n/(kn_1)\}$$

The tiling labeled $(a_1, \ldots, a_{d+1})$ therefore defines a mapping

$$\chi_{a_1,\ldots,a_{d+1}} : V_{n,d} \times \{1, \ldots, n/2\} \to S_1 \times R_1$$

by restriction (except possibly near the boundaries due to fencepost errors).

The trajectory code χ is the concatenation of the codes associated with each of the tilings:

$$\chi(y) = \left(\chi_{a_1,\ldots,a_{d+1}}(y)\right)_{a_1,\ldots,a_{d+1}}$$

Observe that the number of labels concatenated at each vertex is k^{d+1}.

Lemma 3. *χ achieves relative distance δ.*

Proof. Consider any twins (γ, γ'). Let $t = |\gamma|$ and let t_0 be the starting time of the pair of trajectories.

If $t \le (k-2)n_1/2$ then the pair (γ, γ') is contained entirely within a shingle. This implies relative distance at least $(1+\delta)/2$.

Otherwise, round $t_0 + t$ down to the preceding multiple of n_1: specifically, set $\tilde{t} = n_1\lfloor (t_0 + t)/n_1 \rfloor - t_0$. Partition the time period from t_0 to $t_0 + \tilde{t}$ into r epochs (for some r) using boundaries at times $t_1, \ldots, t_r$, defined inductively: if the set $\{t' : t' \ge t_{i-1} + (k-2)n_1, |\gamma(t') - \gamma'(t')| \le 2n_1\}$ is nonempty, t_i is its least element, otherwise $t_i = t_0 + \tilde{t}$ and $r = i$. For $i = 1, \ldots, r$ define words $\chi_{\gamma,i} = \chi(\gamma(t_{i-1}+1)) \cdots \chi(\gamma(t_i))$ and $\chi_{\gamma',i} = \chi(\gamma'(t_{i-1}+1)) \cdots \chi(\gamma'(t_i))$.

In the first epoch (just as in the case $t \le (k-2)n_1/2$), the relative distance is at least $(1+\delta)/2$. Hence $h(\chi_{\gamma,1}, \chi_{\gamma',1}) \ge \delta(t_1 - t_0) + n_1$. This excess is applied against the last epoch and against the rounding-down of t to $\tilde{t}$.

Next we show that in epochs $2, \ldots, r-1$ the relative distance is at least δ, while in the last epoch, the distance bound is $h(\chi_{\gamma,r}, \chi_{\gamma',r}) \ge ((1+\delta)/2)(t_r - t_{r-1}) - n_1$.

First consider the case $i \in \{2, \ldots, r-1\}$, or the case that $i = r$ and $t_r - t_{r-1} \ge (k-2)n_1$. Let $s_i = t_{i-1} + (k-2)n_i$. Partition $\chi_{\gamma,i}$ into two words $\chi_{\gamma,i,1}$ and $\chi_{\gamma,i,2}$ by $\chi_{\gamma,i,1} = \chi(\gamma(t_{i-1}+1)) \cdots \chi(\gamma(s_i))$ and $\chi_{\gamma,i,2} = \chi(\gamma(s_i)) \cdots \chi(\gamma(t_i))$. (The latter may be empty.) Partition $\chi_{\gamma',i}$ into $\chi_{\gamma',i,1}$ and $\chi_{\gamma',i,2}$ similarly.

First we claim that $h(\chi_{\gamma,i,1}, \chi_{\gamma',i,1}) \ge ((1+\delta)/2)(k-3)n_1$ by the following "virtual trajectory" argument. Choose a vertex $y = (y_1, \ldots, y_d) \in V_{n,d}$ such that both $d_G(y, \gamma(t_{i-1})) \le n_1$

and $d_G(y, \gamma'(t_{i-1})) \leq n_1$. Define $\tilde{y} \in V_{n,d} \times \{1, \ldots, n/2\}$ by $\tilde{y} = (y_1, \ldots, y_d, t_{i-1} - n_1)$. Construct a trajectory $\tilde{\gamma}$ with start time $t_{i-1} - n_1$ and length $n_1 + s_i - t_{i-1}$ by having it start at $\tilde{\gamma}(t_{i-1} - n_1) = \tilde{y}$, reach $\tilde{\gamma}(t_{i-1}) = \gamma(t_{i-1})$, and thereafter be identical to γ until time s_i. Similarly construct a trajectory $\tilde{\gamma}'$ with start time $t_{i-1} - n_1$ and length $n_1 + s_i - t_{i-1}$ by having it start at $\tilde{\gamma}'(t_{i-1} - n_1) = \tilde{y}$, reach $\tilde{\gamma}'(t_{i-1}) = \gamma'(t_{i-1})$, and thereafter be identical to γ' until time s_i. Observe that $\tilde{\gamma} \sim \tilde{\gamma}'$ and $|\tilde{\gamma}| = n_1 + s_i - t_{i-1} \leq (k-2)n_1$, so there is a shingle entirely containing the pair $(\tilde{\gamma}, \tilde{\gamma}')$. Hence the claim.

Second we claim that $h(\chi_{\gamma,i,2}, \chi_{\gamma',i,2}) \geq ((1+\delta)/2)(t_i - s_i) - n_1$. This is because in each complete segment of length n_1 these two trajectories pass through distinct codewords of η.

Combining the two claims and using the fact that $k \sim 1/(1-\delta)$ we conclude that $h(\chi_{\gamma,i}, \chi_{\gamma',i}) \geq \delta(t_i - t_{i-1})$.

The remaining case is that $i = r$ and $t_r - t_{r-1} < (k-2)n_1$. Then again by a virtual trajectory argument, $h(\chi_{\gamma,r}, \chi_{\gamma',r}) \geq ((1+\delta)/2)(t_r - t_{r-1}) - n_1$. □

4.2 The code

What is left unstated by the above construction, is how the code χ_1 on the shingles is constructed. The two extreme options are to pursue the whole construction recursively, or to construct χ_1 by exhaustive search. The former option is unsatisfactory because of the alphabet blow-up at each level of recursion. The latter option requires a one-time $n^{O(1)}$-time computation. Once χ_1 has been constructed, local look-up can be performed in time $\log^{O(1)} n$, hence achieving time-efficiency. In order to also achieve space-efficiency, we implement just one more level of recursion, constructing χ_1 out of a code χ_2 for shingles of size $\log\log n$, which is itself constructed by exhaustive search in time $\log^{O(1)} n$. Recall that by time-and-space efficient construction we mean that for any vertex in the state-space graph, we can compute χ in time and space polynomial in the length of the vertex label. Thus, we have:

Theorem 4. *The above construction of χ using χ_2 is time-and-space efficient, and achieves any required relative distance $\delta < 1$.*

Proof. The relative distance guarantee follows from section 4.1; the construction efficiency follows by combining the construction of section 4.1 with the double-recursion of section 4.2. □

5 Trajectory codes have an efficient verification procedure

In this section we show how to explicitly verify the distance property of any trajectory code using dynamic programming. This is in sharp contrast to tree codes, for which no such efficient verification procedure is known. Existence of an efficient verification procedure is important because our construction in the previous section has large constants. Using branch-and-bound methods, and the verification procedure, we can improve our constants further and bring the communication rate closer to the true but unknown optimum. (We suppress the optimization from this extended abstract.)

Let $G = (V, E)$ be a graph with polynomial growth rate p. We show an algorithm that verifies that a finite time trajectory code $\chi : V \times \{1, 2, \ldots, T\} \to S$ has relative distance at least δ. The running time of the algorithm is polynomial in T.

The algorithm is a simple dynamic program. The dynamic programming table D is indexed by quintuples. Valid quintuples (x, y, z, t_0, t) are those for which $x, y, z \in V$, $t_0 + t \leq T$, and there exists a pair of twin trajectories (γ, γ') which begin at time t_0 at x, and such that at time $t_0 + t$, γ ends at y while γ' ends at z. (In other words: $|\gamma| = |\gamma'| = t$, $\gamma(t_0) = \gamma'(t_0) = x$, $\gamma(t_0 + t) = y$, $\gamma'(t_0 + t) = z$, and for every $i > t_0$, $\gamma(i) \neq \gamma'(i)$.) We compute

$$D(x, y, z, t_0, t) = \min_{\text{twins } \gamma, \gamma'} h(\chi(\gamma), \chi(\gamma')).$$

Notice that the size of D can be loosely upper bounded by $(p(T))^3 T^2$ which is polynomial in T. Clearly, upon completion of the computation of D, the relative distance of the code can be verified by checking if

$$D(x, y, z, t_0, t) \geq \delta t,$$

for all valid quintuples (x, y, z, t_0, t).

The table D is computed by induction over t. For $t = 0$ the valid quintuples are $(x, x, x, t_0, 0)$ such that $t_0 \leq T$ and there is a length t_0 trajectory starting at x_0 and ending at x. For such valid quintuples we set $D(x, x, x, t_0, 0) = 0$. For $t > 0$, suppose we already computed all the valid entries of the form $(x, y, z, t_0, t-1)$. For every $t_0 \leq T - t$ and for every three distinct nodes $x, y, z \in B(x_0, T)$ we compute the following. Let $\varepsilon \in \{0, 1\}$ be the indicator of $\chi(y, t_0+t) \neq \chi(z, t_0+t)$. Consider all pairs of nodes y', z' such that $(y', y), (z', z) \in E$ and $(x, y', z', t_0, t-1)$ is a valid quintuple. If no such pair exists, then (x, y, z, t_0, t) is not a valid quintuple. Otherwise, put

$$D(x, y, z, t_0, t) = \varepsilon + \min_{y', z'} \{D(x, y', z', t_0, t-1)\}.$$

This completes the description of the dynamic program.

Theorem 5. *The dynamic program takes poly(T) time to execute, and it correctly computes $D(x, y, z, t_0, t)$ for all valid quintuples (x, y, z, t_0, t).*

Proof. The number of quintuples (x, y, z, t_0, t) (valid or not) that are checked is at most $|B(x_0, T)|^3 T^2 \leq (p(T))^3 T^2$. The number of pairs y', z' that need to be examined in order to compute $D(x, y, z, t_0, t)$ is at most twice the maximum in-degree in the subgraph induced by $B(x_0, T)$. The proof of correctness is a trivial induction on t. □

References

[1] N. Alon and J. H. Spencer. *The probabilistic method.* Wiley, 2nd edition, 2000.

[2] N. Elia and S. K. Mitter. Stabilization of linear system with limited information. *IEEE Transactions on Automatic Control*, 46(7):1384–1400, 2001.

[3] R. M. Fano. A heuristic discussion of probabilistic decoding. *IEEE Transactions on Information Theory*, pages 64–74, 1963.

[4] G. D. Forney. *Concatenated Codes*. MIT Press, 1966.

[5] K. Goldberg and R. Siegwart (editors). *Beyond webcams: an introduction to online robots*. MIT Press, Cambridge, MA, USA, 2002.

[6] J. Justesen. A class of constructive, asymptotically good algebraic codes. *IEEE Transactions on Information Theory*, IT-18:652–656, September 1972.

[7] N. C. Martins and M. A. Dahleh. Feedback control in the presence of noisy channels: "Bode-like" fundamental limitations of performance. Draft, 2004.

[8] S. Micali, C. Peikert, M. Sudan, and D. Wilson. Optimal error correction against computationally bounded noise. In *Second Theory of Cryptography Conference, TCC 2005, Cambridge, MA, USA, February 10-12, 2005, Proceedings*, volume 3378 of *Lecture Notes in Computer Science*, pages 1–16. Springer, 2005.

[9] R. M. Murray, editor. *Control in an information rich world: report of the panel on future directions in control, dynamics and systems*. AFOSR, 2002.

[10] B. Reiffen. Sequential encoding and decoding for the discrete memoryless channel. *Res. Lab. of Electronics, M.I.T. Technical Report*, 374, 1960.

[11] L. J. Schulman. Communication on noisy channels: A coding theorem for computation. In *Proceedings of the 33rd Annual Symposium on Foundations of Computer Science*, pages 724–733, 1992.

[12] L. J. Schulman. Deterministic coding for interactive communication. In *Proceedings of the 25th Annual Symposium on Theory of Computing*, pages 747–756, 1993.

[13] L. J. Schulman. Coding for interactive communication. *Special Issue on Codes and Complexity of the IEEE Transactions on Information Theory*, 42(6):1745–1756, November 1996.

[14] C. E. Shannon. A mathematical theory of communication. *Bell System Tech. J.*, 27:379–423; 623–656, 1948.

[15] S. Tatikonda and S. Mitter. Control over noisy channels. *IEEE Transactions on Automatic Control*, 49(7):1196–1201, 2004.

[16] S. Tatikonda, A. Sahai, and S. Mitter. Stochastic linear control over a communication channel. *IEEE Transactions on Automatic Control*, 49(9):1549–1561, 2004.

[17] S. C. Tatikonda. *Control under communication constraints*. PhD thesis, Massachusetts Institute of Technology, September 2000.

[18] J. M. Wozencraft. Sequential decoding for reliable communications. *Res. Lab. of Electronics, M.I.T. Technical Report*, 325, 1957.