Control over Gaussian Channels With and Without Source–Channel Separation

Elias Riedel Gårding*

Royal Institute of Technology, Stockholm

eliasrg@kth.se

November 1, 2017

Abstract

We consider the problem of controlling an unstable system over an additive Gaussian noise channel. To that end, much effort has been put into extending the tandem communications strategy, where the message is quantized into a stream of bits which are subsequently mapped into coded channel inputs. This is a conceptually attractive technique that was shown to be optimal in the limit of long messages. However, the real-time nature of networked control necessitates transmitting very short messages, for which the tandem strategy is not optimal. An alternative paradigm is that of analog joint source–channel coding (JSCC), where the message is mapped directly to an analog channel input, thereby bypassing the digital domain altogether. In this work, we develop schemes that follow both approaches and compare their control performance using numerical simulations. Specifically, we construct a tandem scheme using recently developed techniques for optimal quantization and anytime reliable codes in the context of control, and compare it to analog JSCC schemes employing Shannon–Kotel'nikov maps. We find that the JSCC schemes provide performance gains in both control cost and computational efficiency.

I. Introduction

Control theory and engineering are motivated by the need for robust and efficient control of inherently unstable real-world systems. Traditionally, the system ("plant" in control parlance) is supplemented with an observer and a controller between which perfect communication is implicitly assumed. This assumption is unproblematic for well-engineered, localized systems, but does not apply to wireless solutions, for which demand is constantly increasing. The emerging field of *networked control* introduces a communication link between the observer and the controller and combines control theory and information theory to explore the resulting scenario.

The traditional scenario has been thoroughly explored and optimal control strategies are known (see for example [1]). The chief difficulty of the networked control problem is therefore how to convey the observer's plant state estimate over the channel with minimal distortion, i.e. the information-theoretic problem of coding (although, as we shall see in Section V.iii, the choice of coding scheme will also affect the required control strategy).

II. Problem statement

The problem considered is the same as in [6]. For simplicity, we consider the scalar, discrete-time, linear plant

$$x_{t+1} = \alpha x_t + w_t + u_t \tag{1}$$

$$y_t = x_t + v_t \tag{2}$$

where (at time t) x_t is the plant state, y_t is the noisy measurement, u_t is the control signal, $w_t \sim \mathcal{N}(0, W)$ is the process noise and $v_t \sim \mathcal{N}(0, V)$ is the observation noise. To

*Supervised by Victoria Kostina and Anatoly Khina under the SURF programme at the California Institute of Technology.

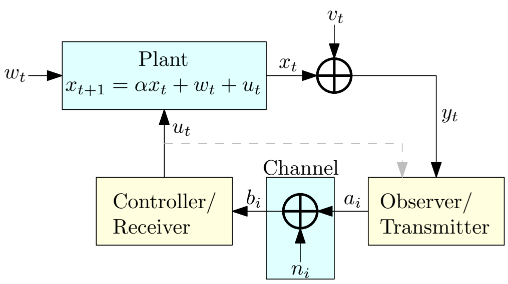

Figure 1: *The networked control system considered.*

communicate its measurement to the controller, the observer may, at each time step, generate K channel inputs a_i, subject to the unit power constraint

$$\sum_{i=i_t}^{i_t+K-1} \mathbb{E}\left[a_i^2\right] \leq K. \tag{3}$$

where $i_t \triangleq K(t-1)+1$. These are then transmitted across the channel to produce outputs

$$b_i = a_i + n_i \tag{4}$$

where $n_i \sim \mathcal{N}(0, \frac{1}{\text{SNR}})$ and the parameter SNR is the *signal-to-noise ratio* describing the channel. Lastly, the controller chooses a control action u_i based on the received signals b_i.

The goal is to minimize the linear quadratic Gaussian (LQG) cost function

$$\bar{J}_T \triangleq \frac{1}{T}\mathbb{E}\left[Fx_{T+1}^2 + \sum_{t=1}^{T}\left(Qx_t^2 + Ru_t^2\right)\right]. \tag{5}$$

We shall, for simplicity, focus on the cases $K = 1$ and $K = 2$, and assume that knowledge of all past control signals $u_{t-1}, u_{t-2}, \ldots$ is available to the observer at time t.

III. Solution approaches

The standard approach to coding in communication systems is to pass through the digital domain: An analog signal is quantized into a string of bits, which is subsequently protected from channel noise by a binary error-correcting code, producing a longer string of bits that is finally modulated onto the channel. The receiver must then perform the operations in reverse to unpack an estimate of the original signal. The advantage of this *tandem* strategy is that it separates the concerns of quantization (source coding) and error protection (channel coding). Its usage is justified by the information-theoretic *separation principle*, devised by Shannon [7], which states that such source–channel separation becomes optimal in the limit of long messages. However, the real-time nature of the control feedback scenario means that the messages must be short and frequent; a long message entails a large delay and therefore bad control performance. Therefore, there seems to be no *a priori* reason to expect that separation-based schemes should work well for control.

An alternative, more promising strategy is to employ joint source–channel coding (JSCC). Here, the digital domain is bypassed completely; the idea is to simply choose an appropriate mapping $\mathbb{R} \to \mathbb{R}^K$ taking y_t to $(a_{i_t}, \ldots, a_{i_t+K-1})$. In [6], the cases $K = 1$ and $K = 2$ were explored and good (for $K = 1$ in fact optimal) solutions were proposed.

The purpose of this work is to provide a practical comparison of these two approaches. Specifically, the JSCC-based scheme of [6] is numerically compared to a tandem scheme that combines the source coding algorithm from [5] with channel codes from [4].

IV. JSCC-based solutions

The control schemes based on joint source–channel coding described in [6] do not employ source–channel separation, but they nevertheless rely on the classical *separation principle of estimation and control*, which states that optimal control of a noisy plant consists of optimal estimation of the plant's state in tandem with optimal control as if the system were noiseless [1]. This means that the tasks of control and coding can be cleanly separated.

i. Control strategy

The control algorithm described in [6] is independent of the value of K. The observer (transmitter) and controller (receiver) construct MMSE[1] estimates $\hat{x}^{\mathrm{t}}_{t|t'}$ and $\hat{x}^{\mathrm{r}}_{t|t'}$ (where

[1] The (biased) MMSE estimate of a variable x given an observation y is, if x and y are jointly Gaussian, $\hat{x}_{\text{MMSE}}(y) = \mathbb{E}[x|y] = \mathbb{E}[x] + \frac{\mathbb{C}[x,y]}{\mathbb{V}[y]}(y - \mathbb{E}[y])$.

$\hat{x}_{t|t'}$ indicates the estimate of x_t made at time t') through Kalman filtering. The observer, having access to past control signals, keeps a simulation of the controller's estimate and transmits its best estimate of the controller's estimation error over the channel, normalized to account for the power constraint. The details are as follows:

Observer/transmitter:

- Given the previous estimate $\hat{x}^{\mathrm{t}}_{t-1|t-1}$ of variance $P^{\mathrm{t}}_{t-1|t-1}$, perform the Kalman filter prediction step:

$$\hat{x}^{\mathrm{t}}_{t|t-1} = \alpha \hat{x}^{\mathrm{t}}_{t-1|t-1} + u_{t-1} \quad (6)$$
$$P^{\mathrm{t}}_{t|t-1} = \alpha^2 P^{\mathrm{t}}_{t-1|t-1} + W. \quad (7)$$

- Perform the Kalman filter observation step to obtain the estimate $\hat{x}^{\mathrm{t}}_{t|t}$:

$$\hat{x}^{\mathrm{t}}_{t|t} = \hat{x}^{\mathrm{t}}_{t|t-1} + \frac{P^{\mathrm{t}}_{t|t-1}}{P^{\mathrm{t}}_{t|t-1} + V}(y - \hat{x}^{\mathrm{t}}_{t|t-1}) \quad (8)$$
$$P^{\mathrm{t}}_{t|t} = \frac{P^{\mathrm{t}}_{t|t-1} V}{P^{\mathrm{t}}_{t|t-1} + V}. \quad (9)$$

- Construct the error signal

$$s_t = \hat{x}^{\mathrm{t}}_{t|t} - \hat{x}^{\mathrm{r}}_{t|t-1} \quad (10)$$

(note that the observer knows $\hat{x}^{\mathrm{r}}_{t|t-1}$ because it knows the history of control signals $u_{t-1}, u_{t-2}, \ldots$). It may be seen by independence arguments that this signal has power (variance) $P^{\mathrm{r}}_{t|t-1} - P^{\mathrm{t}}_{t|t}$. Therefore, the normalized version (of power 1) is

$$\bar{s}_t = \frac{s_t}{\sqrt{P^{\mathrm{t}}_{t|t} - P^{\mathrm{r}}_{t|t-1}}}. \quad (11)$$

- Using a JSCC scheme $(\mathcal{E}, \mathcal{D})$ (where $\mathcal{E}\colon \mathbb{R} \to \mathbb{R}^K$ is the encoder and $\mathcal{D}\colon \mathbb{R}^K \to \mathbb{R}$ is the corresponding decoder) of (known) additive distortion power $\frac{1}{\mathrm{SDR}_0}$, construct the channel inputs

$$(a_{i_t}, \ldots, a_{i_t+K-1}) = \mathcal{E}(\bar{s}_t) \quad (12)$$

and transmit them over the channel.

Controller/receiver:

- Receive the channel outputs $(b_t, \ldots, b_{t+K-1})$ and decode them to form an estimate of $\bar{s}_t$:

$$\hat{\bar{s}}_t = \mathcal{D}(b_{t_i}, \ldots, b_{i_t+K-1}) \quad (13)$$

- Unnormalize $\hat{\bar{s}}_t$ to obtain

$$\hat{s}_t = \sqrt{P^{\mathrm{t}}_{t|t} - P^{\mathrm{r}}_{t|t-1}}\,\hat{\bar{s}}_t. \quad (14)$$

- Perform the Kalman filter prediction step:

$$\hat{x}^{\mathrm{r}}_{t|t-1} = \alpha \hat{x}^{\mathrm{r}}_{t-1|t-1} + u_{t-1} \quad (15)$$
$$P^{\mathrm{r}}_{t|t-1} = \alpha^2 P^{\mathrm{r}}_{t-1|t-1} + W. \quad (16)$$

- Perform the Kalman filter observation step:

$$\hat{x}^{\mathrm{r}}_{t|t} = \hat{x}^{\mathrm{r}}_{t|t-1} + \frac{\mathrm{SDR}_0}{1+\mathrm{SDR}_0}\hat{s}_t \quad (17)$$
$$P^{\mathrm{r}}_{t|t} = \frac{1}{1+\mathrm{SDR}_0}\left(P^{\mathrm{r}}_{t|t-1} + \mathrm{SDR}_0 P^{\mathrm{t}}_{t|t}\right). \quad (18)$$

The above expressions can be found using orthogonality arguments assuming that $\hat{\bar{s}}_t$ is a *correlation-sense unbiased estimator* (CUBE) of $\bar{s}_t$, meaning that $\mathbb{C}[\bar{s}_t, \bar{s}_t - \hat{\bar{s}}_t] = 0$.

- Output the control signal

$$u_t = -L_t \hat{x}^{\mathrm{r}}_{t|t} \quad (19)$$

where L_t is the optimal control gain, given (see e.g. [1]) by

$$L_t = \frac{\alpha S_{t+1}}{S_{t+1} + R} \quad (20a)$$
$$S_t = \frac{\alpha^2 S_{t+1}}{S_{t+1} + R} + Q \quad (20b)$$
$$S_{T+1} = F. \quad (20c)$$

ii. Coding schemes

It remains to choose appropriate encoding and decoding maps $\mathcal{E}$ and $\mathcal{D}$ as referenced in (12) and (13) above. The goal is simply to produce maps that minimize the distortion

$$D \triangleq \mathbb{E}[(\bar{s} - \hat{\bar{s}})^2] \quad (21)$$

subject to the constraint that $\hat{\bar{s}}$ is a CUBE of $\bar{s}$.

In the case of $K = 1$, it has been shown that choosing the identity map, $\mathcal{E} = \mathcal{D} = \mathrm{id}_{\mathbb{R}}$, gives *optimal* control performance.

In the case $K = 2$, the maps to be optimized are $\mathcal{E}\colon \mathbb{R} \to \mathbb{R}^2$ and $\mathcal{D}\colon \mathbb{R}^2 \to \mathbb{R}$. A natural choice might be the naïve repetition

$$\mathcal{E}(\bar{s}) = (\bar{s}, \bar{s}) \quad (22)$$

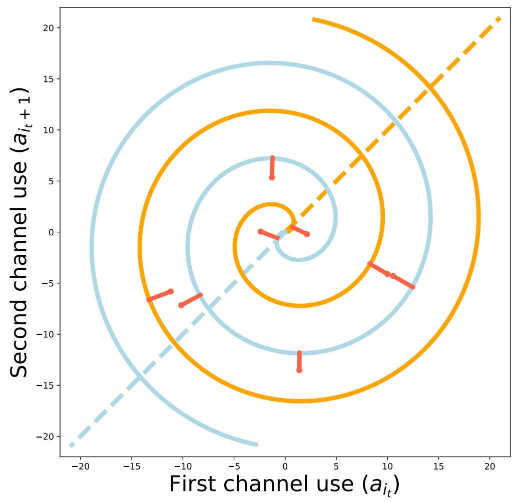

Figure 2: *Linear repetition and the spiral map, with example runs of the ML decoder.*

with the corresponding maximum-likelihood (ML, in this case minimum-distance) decoder

$$\mathcal{D}(\mathbf{b}) = \frac{(1,1)}{2} \cdot \mathbf{b}. \tag{23}$$

However, as Figure 2 shows, this map does not make use of the two dimensions of the available space, and the displayed *Archimedean bi-spiral*

$$\mathcal{E}(\bar{s}) = c\bar{s}\big(\cos(\omega|\bar{s}|), \sin(\omega|\bar{s}|)\big) \tag{24}$$

should provide a basis for a better JSCC scheme. This type of map is referred to as a *Shannon–Kotel'nikov map*.

The constant c is chosen to satisfy the power constraint (3); because $\|\mathcal{E}(\bar{s})\|^2 = c^2\bar{s}^2$ and $\bar{s}$ is normalized such that $\mathbb{E}[\bar{s}^2] = 1$, we must choose $c = \sqrt{2}$. The optimal choice for the constant ω must be experimentally determined (see [6] for details). The construction of an ML decoder (exemplified in Figure 2) is discussed in Appendix A.

V. Separation-based solutions

The tandem scheme, similarly to the JSCC-based one, makes use of the separation principle of estimation and control. The *information-theoretic* (source–channel) separation takes the form of a further split of the task of estimation, specifically coding, into source and channel coding steps.

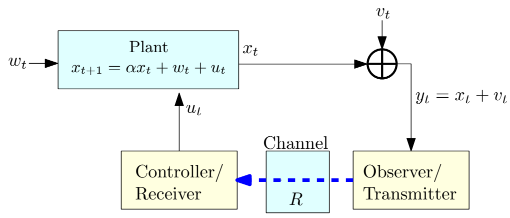

Figure 3: *The system with a noiseless digital channel.*

i. Optimal quantization

As the separation approach is to isolate the channel noise and treat it separately, the source coding part assumes that channel coding has reduced the noisy analog channel to a noiseless digital channel with a rate limit R (meaning that R bits can be transmitted without error at each channel use). The resulting control problem, depicted in Figure 3, has been solved optimally in [5]. In addition to the globally optimal control algorithm, the optimal *greedy* algorithm, which is less complex, has been shown to be very nearly optimal. In view of this, our work uses the greedy variant.

The scheme laid out in [5] is based around both the observer and the controller keeping track of the probability distribution of x_t given the controller's received signals, and using this knowledge to communicate using the optimal (that is, minimum-distortion) quantizer, which can be generated by the Lloyd–Max algorithm (see for example [9]).

For simplicity, assume that the system is perfectly observed ($V = 0$). At each time step, the observer transmits R bits, or equivalently an integer $0 \leq \ell_t < 2^R$ over the channel.

Given a known probability distribution $f_x\colon \mathbb{R} \to \mathbb{R}$, the Lloyd–Max algorithm partitions $\mathbb{R}$ into 2^R intervals, each containing a single reproduction value. The intervals may be represented by their borders $(p_0, \ldots, p_{2^R})$ where $p_0 = -\infty$ and $p_{2^R} = \infty$. Two functions $\mathcal{E}\colon \mathbb{R} \to \{0, \ldots, 2^R - 1\}$ and $\mathcal{D}\colon \{0, \ldots, 2^R - 1\} \to \mathbb{R}$ are thus obtained such that $\mathcal{E}(x)$ is the index of the interval containing x and $\mathcal{D}(\ell)$ is the reproduction value in interval number ℓ. The algorithm ensures (given that $\log \circ f_x$ is concave) that the expected distortion $\mathbb{E}\Big[\{x - \mathcal{D}(\mathcal{E}(x))\}^2\Big]$ is minimized.

Optimal communication can now be achieved by letting both sides run the Lloyd–Max algorithm and transmitting the quantization index $\ell_t \triangleq \mathcal{E}_t(x_t)$. This requires calculating the probability distribution $f_{x_t|\ell^t}$ where $\ell^t \triangleq (\ell_1, \ldots, \ell_t)$. Given $f_{x_t|\ell^{t-1}}$, this amounts to cutting and renormalizing the distribution:

$$f_{x_t|\ell^t}(x) = \begin{cases} \dfrac{f_{x_t|\ell^{t-1}}(x)}{\gamma} & \text{if } p_{\ell_t} \le x < p_{\ell_{t+1}} \\ 0 & \text{otherwise} \end{cases} \tag{25}$$

where

$$\gamma \triangleq \int_{p_{\ell_t}}^{p_{\ell_{t+1}}} f_{x_t|\ell^{t-1}}(x)\,\mathrm{d}x\,. \tag{26}$$

Next, the prediction of x_{t+1} is computed using $x_{t+1} = \alpha x_t + w_t + u_t$:

$$f_{x_{t+1}|\ell^t}(x) = \frac{1}{|\alpha|} f_{x_t|\ell^t}\left(\frac{x-u_t}{\alpha}\right) * f_{w_t}(x) \tag{27}$$

where $*$ denotes the convolution

$$f(x) * g(x) \triangleq \int_{-\infty}^{\infty} f(y) g(x-y)\,\mathrm{d}y \tag{28}$$

(originating from adding independent random variables) and f_{w_t} is the Gaussian probability distribution function (using $\tau \triangleq 2\pi$)

$$f_{w_t}(w) = \frac{1}{\sqrt{\tau W}} e^{-\frac{w^2}{2W}}. \tag{29}$$

The control strategy is, owing to the separation principle of estimation and control, to simply set $u_t = -L_t \hat{x}_t$ where $\hat{x}_t \triangleq \mathcal{D}_t(\ell_t)$.

ii. Tree codes

The channel noise is handled via so-called *tree codes*, which are a generalization of convolutional codes motivated by the concept of *anytime reliability*. They are described in detail in [4], and more information can be found in [3], [8] or [2].

A (linear, time-invariant) tree code encodes a binary stream incrementally by, at each time step, consuming k input bits $\mathbf{q}_t \in \mathbb{Z}_2^k$ and producing n output bits $\mathbf{c}_t \in \mathbb{Z}_2^n$. It is characterized by a sequence of binary matrices $G_0, G_1, G_2, \cdots \in \mathbb{Z}_2^{n\times k}$ such that

$$\mathbf{c}_t = G_0\mathbf{q}_t + G_1\mathbf{q}_{t-1} + \cdots + G_{t-1}\mathbf{q}_1 \tag{30}$$

where addition happens modulo 2 (that is, it is the same as the logical exclusive or commonly written $\oplus$).

The encoding step of a tree code is as straightforward as evaluating (30). Decoding is the difficult problem. The n coded bits $\mathbf{c}_t$ are modulated into K channel inputs $\mathbf{a}_t$, which are then transmitted across the channel, emerging as noisy versions $\mathbf{b}_t$. The decoding task consists of reconstructing $\mathbf{q}_t$ given $\mathbf{b}_t$, specifically, to find the input bits $\mathbf{q}^t \triangleq (\mathbf{q}_1, \ldots, \mathbf{q}_t)$ that maximize the probability $p(\mathbf{b}^t|\mathbf{q}^t)$ of receiving the channel outputs $\mathbf{b}^t \triangleq (\mathbf{b}_1, \ldots, \mathbf{b}_t)$ if the input bit sequence was $\mathbf{q}^t$. The analysis will be based on knowledge of $p(\mathbf{b}_t|\mathbf{c}_t)$. The form of this quantity depends on the details of the channel and the modulation scheme used. We consider the simplest case of 2-PAM (pulse amplitude modulation) over the Gaussian channel: $a_i = (-1)^{c_i}$. Then,

$$p(b_i|c_i) = p_{n_i}(b_i - (-1)^{c_i}) \tag{31}$$

$$p(b_i) = \frac{p_{n_i}(b_i - 1) + p_{n_i}(b_i + 1)}{2} \tag{32}$$

where p_{n_i} is the Gaussian p.d.f.

$$p_{n_i}(n) = \sqrt{\frac{\mathrm{SNR}}{\tau}} e^{-\frac{\mathrm{SNR}}{2}n^2}. \tag{33}$$

Precisely speaking, we pose the problem of minimizing the *Fano metric*

$$M(\mathbf{c}^t) \triangleq \sum_{\tau=1}^{t} \left[\log \frac{p(\mathbf{b}_\tau|\mathbf{c}_\tau)}{p(\mathbf{b}_\tau)} - B\right] \tag{34}$$

where B is a *bias* term that penalizes longer sequences.

For optimal (ML) decoding, the value of the bias term B in (34) is immaterial, as all possible input sequences have the same length t. However, the time (and, depending on the implementation, memory) complexity of ML decoding are $\mathcal{O}(2^{kt})$, because all 2^{kt} possible input sequences must be compared. In practice, one must use faster algorithms that approximate ML decoding. A simple such algorithm, the so-called *stack algorithm* (an example of a *sequential decoding* algorithm) is summarized as Algorithm 1. Its name can be misleading as it is best described in terms of the standard priority queue data structure

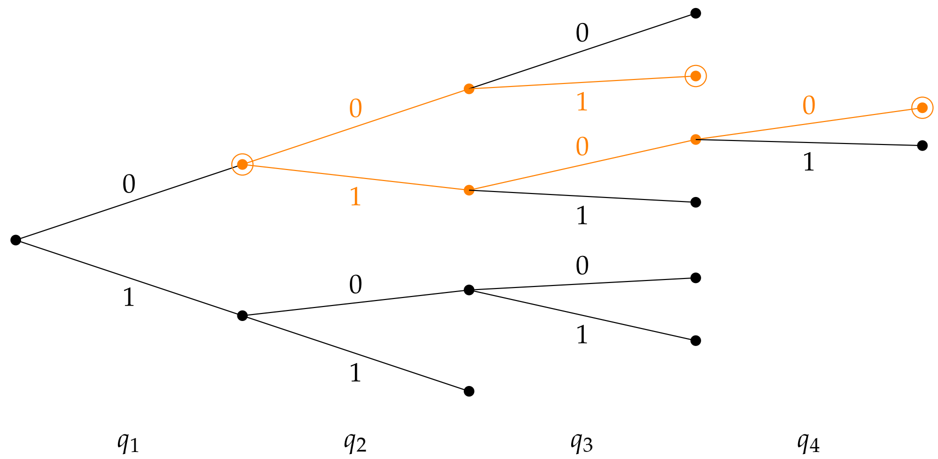

Figure 4: *A partially explored search tree for $k = 1$ and a common ancestor computation.*

Algorithm 1 The stack algorithm for sequential decoding.

$Q \leftarrow$ `MaxPriorityQueue(node` $\mapsto$ `node.metric)` ▷ Leaf nodes, ordered by Fano metric
Q`.put(root)`
while Q`.top.depth` $< t$ **do** ▷ Stop at the first sequence to reach full length
  `node` $\leftarrow Q$`.pop()` ▷ Take the sequence with the largest metric
  for `child` $\in$ `node.create_children()` **do** ▷ Replace it with its 2^k extensions
    Q`.put(child)`
return Q`.top.input_sequence()` ▷ Reconstruct the input sequence by backtracking

(for which, however, the name *priority stack* would be just as good).

The stack algorithm explicitly considers the trie (prefix tree) structure (see Figure 4) of the set of all possible input sequences and performs a partial tree search, iteratively extending the leaf node corresponding to the sequence with the largest Fano metric. Thus, the Fano metric serves as a measure of how promising a partially explored possible input sequence is, and the value of B determines the importance of the length of such a sequence; see [4] for further details.

iii. Control using tree codes

The full separation-based control scheme works as follows. At the observer, the source coding step described in Section V.i (with a certain rate R) is run to produce a quantization index ℓ_t. This index is subsequently encoded as a word $\mathbf{q}_t$ of $k \triangleq R$ bits, which are encoded according to (30) to form a word $\mathbf{c}_t$ of n coded bits, which are then modulated into K channel inputs $\mathbf{a}_t$ and transmitted across the channel.

The controller receives noisy channel outputs $\mathbf{b}_t$ and runs the stack algorithm to decode them. More precisely, let $\mathtt{n}_t$ be the first node in the tth layer (corresponding to an input sequence of length t) to be reached by the stack algorithm. At time t, this node provides the decoded input sequence $\hat{\mathbf{q}}^t_{|t} \triangleq \mathtt{n}_t\mathtt{.input\_sequence()}$. This sequence maps to a sequence of received quantization indices $\hat{\ell}^t_{|t}$, which in turn gives rise to a sequence $(\hat{\mathcal{E}}, \hat{\mathcal{D}})^t_{|t}$ of Lloyd–Max quantizers.

In the ideal case where no decoding errors happen, $\hat{\ell}_{\tau|t} = \ell_\tau$ for all $\tau \le t$, and so the optimal control policy is once again simply $u_t = -L_t \hat{x}_{t|t}$ where $\hat{x}_{t|t} \triangleq \hat{\mathcal{D}}_{t|t}(\hat{\ell}_{t|t})$. In this case, each decoded $\hat{\mathbf{q}}^\tau_{|\tau}$ will be a prefix of $\hat{\mathbf{q}}^t_{|t}$ for $\tau \le t$, meaning that $\mathtt{n}_\tau\mathtt{.parent} = \mathtt{n}_{\tau-1}$, that is, the stack algorithm honed in on the correct path from the beginning. However, if decoding errors are present, it may happen that $\mathtt{n}_t\mathtt{.parent} \neq \mathtt{n}_{t-1}$. Assuming that $\mathtt{n}_t$ is on the correct path ($\hat{\mathbf{q}}^t_{|t} = \mathbf{q}^t$), this means that a decoding error was made when the paths started diverging (see Figure 4), namely at time $t_0 \triangleq \textsc{CommonAncestor}(\mathtt{n}_t, \mathtt{n}_{t-1}).\mathtt{depth} + 1$ (efficiently computed by Algorithm 2).

Algorithm 2 First common ancestor.

```
function CommonAncestor(a, b)
    if a.depth < b.depth then
        Swap a and b
    while a.depth > b.depth do
        a ← a.parent
    while a ≠ b do
        a ← a.parent
        b ← b.parent
    return a
```

The controller, having discovered this error, must then correct for it. The approach taken in this work requires a modification to the observer in addition to the controller, namely making it actively ignore any decoding errors. That is, the observer internally simulates how the plant would evolve if the channel were noiseless, as in Section V.i. This can be accomplished because the observer can detect decoding errors through its assumed knowledge of u. More precisely, the linear plant (1) may be split as $x_t = x_t^w + x_t^u$ where

$$x_{t+1}^w = \alpha x_t^w + w_t \tag{35}$$

$$x_{t+1}^u = \alpha x_t^u + u_t. \tag{36}$$

The plant state without noise would be determined by

$$x_{t+1}^{\text{ideal}} = \alpha x_t^{\text{ideal}} + w_t + u_t^{\text{ideal}} \tag{37}$$

$$u_t^{\text{ideal}} \triangleq -L_t \mathcal{D}_t(\mathcal{E}_t(x_t^{\text{ideal}})) \tag{38}$$

where $(\mathcal{E}_t, \mathcal{D}_t)$ is the observer's Lloyd–Max quantizer (that is, without the step $\ell_t \mapsto \hat{\ell}_{t|t}$). Splitting this as $x_t^{\text{ideal}} = x_t^w + x_t^{u^{\text{ideal}}}$ shows that x^{ideal} can be simulated via

$$x_t^{\text{ideal}} = x_t - x_t^u + x_t^{u^{\text{ideal}}} \tag{39}$$

$$= x_t - x_t^{u - u^{\text{ideal}}} \tag{40}$$

(or, if V is nonzero, $y_t^{\text{ideal}} = y_t - x_t^u + x_t^{u^{\text{ideal}}}$). The policy of the observer is then simply to quantize and encode x_t^{ideal} instead of x_t.

The responsibility of acting on decoding errors now lies entirely with the controller. The controller lacks perfect knowledge of u^{ideal}, and so must keep estimates $\hat{u}_{|t}^t$ of it. Unless the controller detects (via the stack algorithm) a previous decoding error, the control policy is simply $u_t = \hat{u}_{t|t}$. If, on the contrary, the controller judges at time t that an error was made at a previous time t_0, the sequences $\hat{u}_{|t-1}^{t-1}$ and $\hat{u}_{|t}^{t-1}$ will differ at positions $t_0, \ldots, t-1$. The controller then adds to its control signal the required impulse to immediately bring the plant to the trajectory that is currently believed to be ideal:

$$u_t = -\alpha x_t^{\hat{u}_{|t-1}} + x_{t+1}^{\hat{u}_{|t}} \tag{41}$$

$$= \alpha(x_t^{\hat{u}_{|t}} - x_t^{\hat{u}_{|t-1}}) + \hat{u}_{t|t} \tag{42}$$

where $\hat{u}_{t|t} = -L_t \hat{\mathcal{D}}_{t|t}(\hat{\ell}_{t|t})$. The error $\hat{\tilde{x}}_{t_0 \ldots t|t} \triangleq x_t^{\hat{u}_{|t}} - x_t^{\hat{u}_{|t-1}}$ can be computed via the recursion

$$\hat{\tilde{x}}_{t_0 \ldots t_0-1|t} = 0 \tag{43}$$

$$\hat{\tilde{x}}_{t_0 \ldots \tau+1|t} = \alpha \hat{\tilde{x}}_{t_0 \ldots \tau|t} + \hat{u}_{\tau|t} - \hat{u}_{\tau|t-1}. \tag{44}$$

This strategy potentially totally nullifies the effects of decoding errors as soon as they are discovered, albeit at the price of a large contribution to the cost function.

iv. Nontrivial PAM constellations

In the discussion in Section V.ii, an elementary 2-PAM ($a_i = (-1)^{c_i}$) was assumed for simplicity. This corresponds to $n = K$. For the more general case of $n = K, 2K, 3K, \ldots$ one needs to select a mapping $m\colon \mathbb{Z}_2^n \to \mathbb{R}^K$ that takes $\mathbf{c}_t$ to $\mathbf{a}_t$, subject to the power constraint (3). It is simplest to choose a $2^{n/K}$-PAM mapping $a\colon \mathbb{Z}_2^{n/K} \to \mathbb{R}$ and apply it blockwise (meaning that m is a Cartesian power of the smaller constellation a). The generalizations of (31), (32) and (33) are

$$p(\mathbf{b}_t|\mathbf{c}_t) = p_{\mathbf{n}_t}(\mathbf{b}_t - m(\mathbf{c}_t)) \tag{45}$$

$$p(\mathbf{b}_t) = \frac{1}{2^n} \sum_{\mathbf{c} \in \mathbb{Z}_2^n} p_{\mathbf{n}_t}(\mathbf{b}_t - m(\mathbf{c})) \tag{46}$$

$$p_{\mathbf{n}_t}(\mathbf{n}) = \left(\frac{\text{SNR}}{\tau}\right)^{n/2} e^{-\frac{\text{SNR}}{2}\|\mathbf{n}\|^2}. \tag{47}$$

In the present work, some simulations (see Section VI for the results) were made for a 4-PAM

$$a(c_i, c_{i+1}) = \frac{2f(c_i, c_{i+1}) - 3}{\sqrt{5}} \tag{48}$$

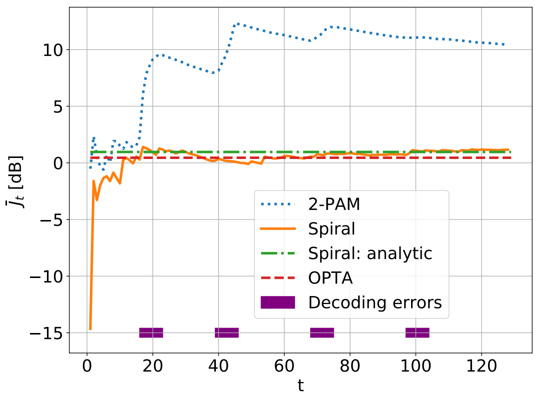

Figure 5: *Performance comparison for* SNR = 4.5 dB; $\alpha = 1.2$; $K = 2$; $Q = F = 1$; $R = 0$.

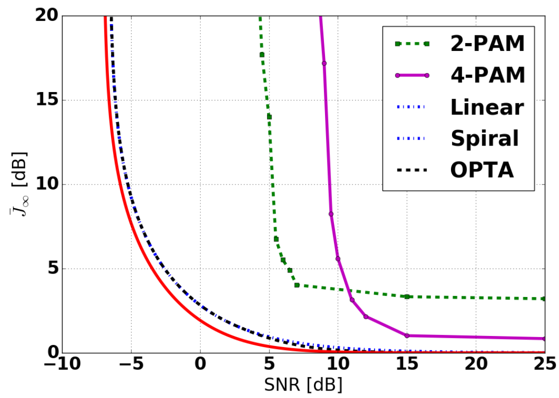

Figure 6: *Dependence of cost on* SNR *for* $\alpha = 1.2$; $K = 2$; $Q = F = 1$; $R = 0$.

where f maps bit strings to natural numbers in the canonical way (here, $f(c_i, c_{i+1}) = 2c_i + c_{i+1}$).

VI. Simulations

The two schemes described above were implemented[2][3] in Python 3 with NumPy and SciPy. Figure 5 shows a comparison of the control costs $\bar{J}_t$ achieved by both schemes for a perfectly observed scalar plant with $\alpha = 1.2$ over an AWGN channel with SNR = 4.5 dB. The two schemes are compared with identical noise realizations. Furthermore, the average control cost over 256 runs of the JSCC-based scheme on different noise realizations is plotted, and agrees with the theoretical prediction [6, Eq. 17a], which is just slightly above the optimum performance theoretically achievable (OPTA). Clearly, the JSCC-based scheme outperforms the tandem scheme by a large margin.

For the tandem scheme, decoding errors (times when $\hat{\mathbf{q}}^t_{|t} \neq \mathbf{q}^t$) are highlighted. Their impact is clear: while the decoder is in error, it applies the wrong control signals, causing the cost function to explode. In the instance shown, these decoding errors clearly are the major factor degrading performance.

Figure 6 shows the dependence of control cost $\bar{J}_\infty \triangleq \lim_{t\to\infty} \bar{J}_t$ on SNR, theoretically derived for the JSCC scheme and numerically simulated in the case of separation-based schemes. The 4-PAM (48) performs better than 2-PAM at high SNR and worse at low SNR, which is to be expected.

We conclude that in addition to demanding far less computation time and memory and being considerably simpler to implement than tandem schemes, the JSCC-based scheme also performs much better in terms of control cost.

VII. Further directions

The implementation of the separation-based scheme assumes a perfectly observed plant ($V = 0$). This restriction was inherited from [5], where it was made for simplicity. Lifting it should be a fairly straightforward matter of modifying (25) and (27) to accommodate the observation noise.

The investigation of larger PAM constellations in Section V.iv was intended as a proof of concept, and the 4-PAM constellation (48) used is simple. The general pattern that larger constellations perform better at high SNR, but worse at low SNR, is expected to hold for all constellations, so it seems clear that at low SNR, the JSCC schemes will outperform the tandem scheme regardless of what constellation is chosen. Nevertheless, for higher SNR, optimizing the constellation could yield some (slight) improvements in performance.

These generalizations aside, the results strongly suggest that further research should shift the focus from separation-based solutions to JSCC-based ones. The largest open

[2]The implementation is available under the MIT license at `https://github.com/eliasrg/SURF2017`.

[3]Thanks to Hikmet Yıldız for contributing an implementation of the optimal quantization scheme.

problem may be to devise suitable Shannon–Kotel'nikov maps for higher dimensions ($K > 2$), but alternatives to the two-dimensional bi-spiral map (24) should also be investigated.

A. ML decoder for the spiral

This appendix describes the construction of an ML decoder for the Archimedean bi-spiral Shannon–Kotel'nikov map (24). For convenience, we shall use complex notation and omit the bar on $\bar{s}$;

$$\mathcal{E}(s) = cse^{i\omega|s|}. \tag{49}$$

The goal is now to design the function $\mathcal{D}\colon \mathbb{C} \to \mathbb{R}$ such that $\mathcal{D}(b)$ is the value of s that minimizes the distance between $\mathcal{E}(s)$ and b;

$$\mathcal{D}(b) = \arg\min_{s} |b - \mathcal{E}(s)|. \tag{50}$$

Now, if $\mathcal{E}(s)$ is the closest point to b on the spiral, then the displacement $b - \mathcal{E}(s)$ must be orthogonal to the tangent vector $\mathcal{E}'(s)$. Using the expression $z \cdot w = \operatorname{Re}\{z^* w\}$ for the $\mathbb{R}^2$ dot product between complex numbers yields the symbolic criterion

$$\operatorname{Re}\big\{[b - \mathcal{E}(s)]^* \mathcal{E}'(s)\big\} = 0. \tag{51}$$

Letting $\sigma \triangleq \operatorname{sign} s$, we have that

$$\mathcal{E}(s) = cse^{i\sigma\omega s}. \tag{52}$$

Differentiating gives

$$\mathcal{E}'(s) = c(1 + i\sigma\omega s)e^{i\sigma\omega s}, \tag{53}$$

whence

$$\begin{aligned}\operatorname{Re}&\big\{[b - \mathcal{E}(s)]^* \mathcal{E}'(s)\big\} \\ &= c\Big(b^* - cse^{-i\sigma\omega s}\Big)(1 + i\sigma\omega s)e^{i\sigma\omega s} \\ &= c\Big(b^* e^{i\sigma\omega s} - cs\Big)(1 + i\sigma\omega s). \end{aligned} \tag{54}$$

Letting $g(s) \triangleq \frac{1}{c}[b - \mathcal{E}(s)]^* \mathcal{E}'(s)$ we therefore have

$$\begin{aligned} g'(s) = &\Big(i\sigma\omega b^* e^{i\sigma\omega s} - c\Big)(1 + i\sigma\omega s) \\ &+ i\sigma\omega\Big(b^* e^{i\sigma\omega s} - cs\Big). \end{aligned} \tag{55}$$

Letting $f(s) \triangleq \operatorname{Re} g(s)$ entails that $f'(s) = \operatorname{Re} g'(s)$. As (51) is equivalent to $f(s) = 0$, the problem may be solved by supplying f and f' as inputs to Newton's method twice; once for each $\sigma \in \{1, -1\}$.

There is, of course, the question of which starting guess to supply. It was found that picking a range of values centred on

$$\begin{aligned} s_{\text{start}} \triangleq \frac{1}{\sigma\omega}\bigg[&\tau\left\lfloor \frac{\omega|b|}{\tau c} \right\rfloor \\ &+ \operatorname{atan}_2(\sigma \operatorname{Im} b, \sigma \operatorname{Re} b) \bmod \tau\bigg] \end{aligned} \tag{56}$$

(a guess such that $\mathcal{E}(s_{\text{start}}) \parallel b$), where $\tau \triangleq 2\pi$, resulted in a functioning decoder (shown in Figure 2).

References

[1] D. P. Bertsekas. *Dynamic programming and optimal control*. Belmont: Athena Scientific, 1995. ISBN: 1-886529-11-6.

[2] R. G Gallager. *Information Theory and Reliable Communication*. John Wiley & Sons, Inc., 1968. ISBN: W-471-29048-3.

[3] F. Jelinek. *Probabilistic Information Theory. Discrete and Memoryless Models*. McGraw–Hill, Inc., 1968.

[4] A. Khina, W. Halbawi, and B. Hassibi. "(Almost) practical tree codes". In: *2016 IEEE International Symposium on Information Theory (ISIT)*. Extended version available at `http://www.its.caltech.edu/~khina/papers/conferences/tree_codes_isit2016.pdf`. July 2016, pp. 2404–2408. DOI: `10.1109/ISIT.2016.7541730`. URL: `http://www.its.caltech.edu/~khina/papers/conferences/tree_codes_isit2016_short.pdf`.

[5] A. Khina, Y. Nakahira, Y. Su, and B. Hassibi. "Algorithms for Optimal Control with Fixed-Rate Feedback". Submitted to *IEEE Conference on Decision and Control*. Mar. 2017.

[6] A. Khina, G. M. Pettersson, V. Kostina, and B. Hassibi. "Multi-rate control over AWGN channels via analog joint source–channel coding". In: *IEEE Conference on Decision and Control (CDC)*. Extended version available at `http://www.`

its.caltech.edu/~khina/papers/conferences/jscc4control_cdc2016.pdf. June 2016. URL: http://www.its.caltech.edu/~khina/papers/conferences/jscc4control_cdc2016_short.pdf.

[7] C. E. Shannon. “A Mathematical Theory of Communication”. In: *SIGMOBILE Mob. Comput. Commun. Rev.* 5.1 (Jan. 2001), pp. 3–55. ISSN: 1559-1662. DOI: 10.1145/584091.584093.

[8] A. J. Viterbi and J. K. Omura. *Principles of Digital Communication and Coding*. Ed. by F. J. Cerra and J. W Masel. McGraw–Hill, Inc., 1979. ISBN: 0-07-067516-3.

[9] Y. You. “Scalar Quantization”. In: *Audio Coding: Theory and Applications*. Boston, MA: Springer US, 2010, pp. 19–42. ISBN: 978-1-4419-1754-6. DOI: 10.1007/978-1-4419-1754-6_2. URL: http://dx.doi.org/10.1007/978-1-4419-1754-6_2.