On the Capacity of Precision-Resolution Constrained Systems

Moshe Schwartz
California Institute of Technology
1200 E California Blvd., Mail Code 136-93
Pasadena, CA 91125, U.S.A.
moosh@paradise.caltech.edu

Jehoshua Bruck
California Institute of Technology
1200 E California Blvd., Mail Code 136-93
Pasadena, CA 91125, U.S.A.
bruck@paradise.caltech.edu

Abstract— **Arguably, the most famous constrained system is the (d,k)-RLL (Run-Length Limited), in which a stream of bits obeys the constraint that every two 1's are separated by at least d 0's, and there are no more than k consecutive 0's anywhere in the stream. The motivation for this scheme comes from the fact that certain sensor characteristics restrict the minimum time between adjacent 1's or else the two will be merged in the receiver, while a clock drift between transmitter and receiver may cause spurious 0's or missing 0's at the receiver if too many appear consecutively.**

The interval-modulation scheme introduced by Mukhtar and Bruck extends the RLL constraint and implicitly suggests a way of taking advantage of higher-precision clocks. Their work however, deals only with an encoder/decoder construction.

In this work we introduce a more general framework which we call the precision-resolution (PR) constrained system. In PR systems, the encoder has precision constraints, while the decoder has resolution constraints. We examine the capacity of PR systems and show the gain in the presence of a high-precision encoder (thus, we place the PR system with integral encoder, $(p{=}1,\alpha,\theta)$-PR, which turns out to be a simple extension of RLL, and the PR system with infinite-precision encoder, (∞,α,θ)-PR, on two ends of a continuum). We derive an exact expression for their capacity in terms of the precision p, the minimal resolvable measurement at the decoder α, and the decoder resolution factor θ. In an analogy to the RLL terminology these are the clock precision, the minimal time between peaks, and the clock drift. Surprisingly, even with an infinite-precision encoder, the capacity is finite.

I. Introduction

The (d,k)-RLL is perhaps the most commonly used constrained system, appearing in various contexts and applications, including (but not restricted to) magnetic and optical storage, holographic storage, and wireless and fiber-optic communication. It is however easy to forget that RLL coding is but one possible solution to a set of restrictions imposed by the medium, by current technology, or by our own choice. To better understand any extension to it, we should first describe these restrictions.

In the context of storage applications, for example, one wants to write a stream of bits. Due to the nature of the recording devices, when reading back the stream, we receive a series of analog peaks which correspond to the **1**'s in the original bit stream. The first restriction imposed by such a system is a function of the sensors used for the reading which may cause adjacent peaks to merge if these are positioned too close to each other. In the RLL constraint, this restriction is upheld by requiring that in the bit stream we write, every two adjacent **1**'s are separated by at least $d \geqslant 0$ consecutive **0**'s.

On the other hand, the decoding mechanism attempts to acquire the correct number of **0**'s in a run by measuring the time between two peaks, and dividing it by the duration of a single **0**. Since no two clocks are exactly the same, the difference in their frequencies causes a drift. If this drift is bounded by $0 < \delta < 1$, then a written run of **0**'s of duration $t \in R$, may actually end up being decoded as a run of duration $(1-\delta)t < t' < (1+\delta)t$. Obviously, the longer the run of **0**'s, the more spurious or missing **0**'s may occur in the decoding. To avoid such a problem, the RLL-constraint chooses to limit the maximum length of any run of **0**'s to k, where $k \in \mathbb{N}$ is the largest integer for which $(1+\delta)k \leqslant (1-\delta)(k+1)$. We note that since the **1**'s are used only to punctuate the stream, and we measure the time duration between peaks, the duration of the run includes the duration of the consecutive **0**'s and the single following **1**. Thus, for example, any value of $\delta \in (\frac{1}{23}, \frac{1}{21}]$ results in $k = 10$ found in the $(2,10)$-RLL of CD-ROMs and DVDs.

In order to get a correct **resolution** at the decoder we need the two restrictions mentioned above. But another arbitrary implicit restriction is part of the RLL constraint, and that is the requirement that the set of valid intervals form a sequence of consecutive integers. Having similar motivation, Funk [2] devised an improvement to the RLL scheme called RLL-MS, by replacing this requirement with another equally arbitrary requirement that the set of valid intervals form an arithmetic progression. We remove this requirement altogether, while also parting ways with the restriction that the time between adjacent peaks be an integer. Exposing and parameterizing this **precision** constraint, will bring us to define a sequence of ever more precise systems. These are able to measure small discrete time intervals, and in the limit, reach a system with infinite-precision measurement.

We therefore introduce the (p,α,θ)*-PR (precision-resolution) framework*. In this framework (see Figure 1), an *encoder* receives a stream of binary bits, transforms them

This work was supported in part by the Caltech Lee Center for Advanced Networking and by NSF grant ANI-0322475.

1-4244-0504-1/06/$20.00 ©2006 IEEE

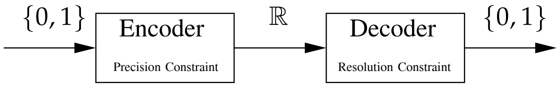

Fig. 1. The precision-resolution framework

into a stream of real numbers which are sent over a noiseless channel to a *decoder*, which then converts them back to the original binary stream. However, both sides suffer from further constraints:

Encoder Constraint — The encoder has constrained precision. Each transmitted symbol $t \in \mathbb{Q}$ is some $t = n/p$ where $n, p \in \mathbb{N}$. We call p the *precision* of the encoder. We denote infinite precision by $p = \infty$, i.e., the framework in which the encoder can send any $t \in \mathbb{R}$.

Decoder Constraints — The decoder suffers from resolution constraints. We may define several different sets of constraints which correspond to different applications. In this paper we consider a set which is motivated mainly (but not only) by storage applications.

- The decoder cannot correctly recover any $t < \alpha$, where $\alpha \in \mathbb{R}$, $\alpha > 0$, is called the *minimum resolution*.
- If $t \in \mathbb{R}$ was transmitted, then the received $t' \in \mathbb{R}$ obeys $(1-\delta)t < t' < (1+\delta)t$, where $\delta \in \mathbb{R}$, $0 < \delta < 1$, is another parameter of the framework. For convenience, we define $\theta = \frac{1+\delta}{1-\delta}$ and call it the *resolution factor*.

We note that the RLL constraint is but one solution to a precision-resolution framework. In the following sections we will present the *precision-resolution constraint* which has a higher capacity than its RLL and RLL-MS counterparts.

In his seminal work, Shannon [10] defined the capacity of a constrained system S as

$$\mathsf{cap}(S) \overset{\text{def}}{=} \lim_{n\to\infty} \frac{\log_2 |S(n)|}{n},$$

where $S(n)$, $n \in \mathbb{N}$, denotes the set of strings of length n in S. Further advances in the subject include the work of Adler, Coppersmith, and Hassner [1], who applied symbolic dynamics [5] to the analysis and construction of sliding-block encoders for constrained systems. More extensions may be found in works such as Heegard, Marcus, and Siegel [3], Marcus, Siegel, and Wolf [6], and a review in Marcus, Roth, and Siegel [7].

Since the definition of the RLL constraint in [4], [11], several variants to the RLL constraint were suggested. However, the basic premise that we may use ever growing runs of **0**'s until we reach a run length for which the clock drift may produce a spurious **0**, was never changed. This paradigm was shifted in the work of Mukhtar and Bruck [8], [9] which described an extension to the RLL constraint and gave a variable-bit-rate to variable-bit-rate encoder/decoder scheme for it. In Section II, we describe this extension in the precision-resolution framework, discuss the integral-precision constraint, $(1, \alpha, \theta)$-PR, and calculate its capacity. In Section III we consider the sequence of constraints (p, α, θ)-PR, where $p = 1, 2, \dots$, and show how it gives a significant improvement to the capacity while the RLL constraint fails. We then find the capacity of the infinite-precision constraint, (∞, α, θ)-PR, and show that it is actually the limit as $p \to \infty$ of the capacity of (p, α, θ)-PR. As a result, we can compare, for example, the capacity of $(2, 10)$-RLL used in CD-ROMs and DVDs and is approximately 0.5418, with the capacity of the corresponding $(\infty, 3, 1.1)$-PR which is approximately 0.7725, thus giving the hope for an increase in capacity of more than 40% in this case. We summarize our results in Section IV.

II. The $(1, \alpha, \theta)$-PR Constraint

The (d, k)-RLL constrained system is, equivalently, the set of all possible strings which are formed by concatenating strings from the set

$$\mathcal{S}_{d,k} \overset{\text{def}}{=} \left\{ 0^d 1, 0^{d+1} 1, 0^{d+2} 1, \dots, 0^k 1 \right\}.$$

Because of this form of representation, it was shown in [10] that the capacity of the (d, k)-RLL constraint is $\log_2 r_{d,k}$, where $r_{d,k} \in \mathbb{R}$ is the unique positive root of the equation

$$x^{-(d+1)} + x^{-(d+2)} + x^{-(d+3)} + \cdots + x^{-(k+1)} = 1.$$

We may also think of the strings in $\mathcal{S}_{d,k}$ as a set of symbols which differ only in their duration. For (d, k)-RLL this set is

$$\hat{\mathcal{I}}_{d,k} \overset{\text{def}}{=} \{d+1, d+2, \dots, k+1\}.$$

For the decoder of a (d, k)-RLL stream there is no ambiguity. Any time duration read t', is some $(1-\delta)t < t' < (1+\delta)t$, where $t \in \hat{\mathcal{I}}_{d,k}$ is the written duration and $0 < \delta < 1$ is the clock drift. But since k was chosen such that $(1+\delta)k \leqslant (1-\delta)(k+1)$, for any two distinct elements $t_1, t_2 \in \hat{\mathcal{I}}_{d,k}$ there is no intersection in their δ-neighborhoods: $((1-\delta)t_1, (1+\delta)t_1)$ and $((1-\delta)t_2, (1+\delta)t_2)$.

Essentially, the (d, k)-RLL constraint starts with the minimal time interval allowed by the decoder, $d+1$, and continues to add successive lengths $d+2, d+3, \dots$ as long as their δ-neighborhoods are disjoint. It stops at the first length $k+1$ whose δ-neighborhood intersects that of $k+2$.

An obvious question asked by Mukhtar and Bruck [8], is why restrict ourselves to successive time durations? We may skip $k+2$ but still be able to use $k+3$ or some other longer length. In [8] it is assumed that some digital clock is governing the system, thus restricting all time durations measured to be integral multiples of the clock period. This brings us to define the integral precision-resolution constrained system.

Definition 1. *Let $\alpha \in \mathbb{R}$, $\alpha > 0$, be the minimum resolution, and let $\theta = \frac{1+\delta}{1-\delta} > 1$ be the resolution factor, where $\delta \in \mathbb{R}$, $0 < \delta < 1$. Then the $(1, \alpha, \theta)$-PR constraint is the set of all streams with symbols from the set*

$$\hat{\mathcal{I}}_{\alpha,\theta} \overset{\text{def}}{=} \left\{ \lceil \alpha, \theta \rceil^i \mid i = 0, 1, 2, \dots \right\} \subset \mathbb{N}$$

where we define

$$\lceil \alpha, \theta \rceil^i \overset{\text{def}}{=} \lceil \dots \lceil \lceil \lceil \alpha \rceil \overbrace{\theta \rceil \theta \rceil \dots \theta}^{i} \rceil.$$

Example 1. *For $\alpha = 2$ and $\theta = 1.5$, we have*

$$\hat{\mathcal{I}}_{\alpha,\theta} = \{2, 3, 5, 8, 12, 18, 27, \dots\}\,.$$

Obviously, the $(1, \alpha, \theta)$-PR constraint obeys the $(1, \alpha, \theta)$-PR framework since all symbols are integers, the smallest one is α, and

$$\lceil \alpha, \theta \rceil^{i+1} = \left\lceil \theta \lceil \alpha, \theta \rceil^{i} \right\rceil \geqslant \theta \lceil \alpha, \theta \rceil^{i}$$

ensures that the δ-neighborhoods of elements in $\hat{\mathcal{I}}_{\alpha,\theta}$ are disjoint. For a practical implementation we may want to limit the size of elements in $\hat{\mathcal{I}}_{\alpha,\theta}$ and so we define

$$\hat{\mathcal{I}}^{m}_{\alpha,\theta} \overset{\text{def}}{=} \left\{\tau \in \hat{\mathcal{I}}_{\alpha,\theta} \mid \tau \leqslant m\right\}.$$

Furthermore, if we define

$$k_\theta = \max\left\{k \in \mathbb{Z} \mid \theta(k-1) \leqslant k\right\},$$

then $\hat{\mathcal{I}}^{k_\theta}_{\alpha,\theta}$ is exactly the set used in $(\lceil \alpha \rceil - 1, k_\theta - 1)$-RLL.

Theorem 1. *Let $\hat{S}^{m}_{\alpha,\theta}$ denote the constraint with intervals from $\hat{\mathcal{I}}^{m}_{\alpha,\theta}$. Then for every $m < m'$ such that $\left|\hat{\mathcal{I}}^{m}_{\alpha,\theta}\right| < \left|\hat{\mathcal{I}}^{m'}_{\alpha,\theta}\right|$,*

$$\mathsf{cap}(\hat{S}^{m}_{\alpha,\theta}) < \mathsf{cap}(\hat{S}^{m'}_{\alpha,\theta}) < 1.$$

We therefore conclude that adding more symbols to the constraint, strictly increases the capacity. If we consider the infinite set $\hat{\mathcal{I}}_{\alpha,\theta}$, we get the following theorem.

Theorem 2. *Let $\hat{S}_{\alpha,\theta}$ denote the $(1, \alpha, \theta)$-PR constrained system. Then*

$$\mathsf{cap}(\hat{S}_{\alpha,\theta}) = \log_2 \hat{r}_{\alpha,\theta} < 1$$

where $\hat{r}_{\alpha,\theta}$ is the unique solution in $(1, \infty)$ to the equation $\hat{f}_{\alpha,\theta}(x) = 1$ and

$$\hat{f}_{\alpha,\theta}(x) \overset{\text{def}}{=} \sum_{i \geqslant 0} x^{-\lceil \alpha, \theta \rceil^{i}}.$$

We see in Figure 2, for $1 < \theta \leqslant 2$, a comparison of the resulting $(0, k_\theta - 1)$-RLL constraint with its extension, the $(1, 1, \theta)$-PR constraint. To plot this graph, an approximation was needed since no simple closed form is known for $\hat{f}_{\alpha,\theta}(x)$. Specifically, we added only the first five monomials beyond those of the corresponding RLL constraint.

III. The General (p, α, θ)-PR Constraint

It is intuitively clear that the rounding up of the symbols of the $(1, \alpha, \theta)$-PR constraint, incurs a penalty in capacity, especially for small values of θ. So while the δ-neighborhoods of $\lceil \alpha \rceil$ and $\lceil \lceil \alpha \rceil \theta \rceil$ are disjoint, the unused gap between them might be fairly large.

All of this is caused by the fact that we restricted ourselves to symbols which are integers. But suppose we are given a system which is $p \in \mathbb{N}$ times more precise[1], i.e., it is able to transmit symbols which are multiples of $1/p$. So now, instead of rounding up the symbols of our constrained system to the nearest integer, we can round them up to the nearest multiple of $1/p$.

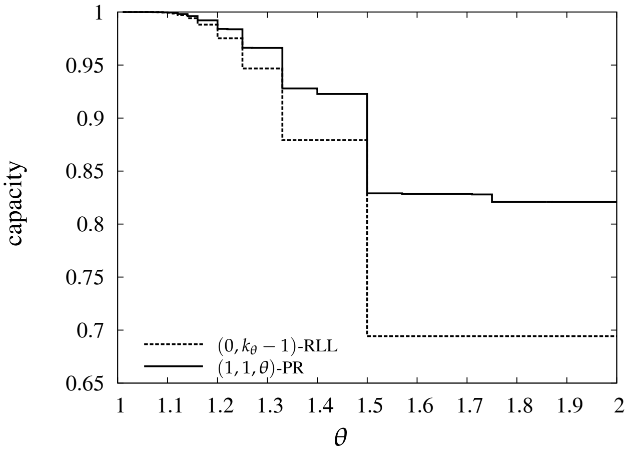

Fig. 2. The capacity of $(0, k_\theta - 1)$-RLL vs. the capacity of $(1, 1, \theta)$-PR

Definition 2. *Let $p \in \mathbb{N}$ be the precision, $\alpha \in \mathbb{R}$, $\alpha > 0$, be the minimum resolution, and let $\theta = \frac{1+\delta}{1-\delta} > 1$ be the resolution factor, where $\delta \in \mathbb{R}$, $0 < \delta < 1$. Then the (p, α, θ)-PR* constraint *is the set of all streams with symbols from the set*

$$\hat{\mathcal{I}}_{p,\alpha,\theta} \overset{\text{def}}{=} \left\{ \lceil p\alpha, \theta \rceil^{i} / p \;\middle|\; i = 0, 1, 2, \dots \right\}.$$

Example 2. *For $p = 2$, $\alpha = 2$ and $\theta = 1.5$, we have*

$$\hat{\mathcal{I}}_{p,\alpha,\theta} = \{2, 3, 4.5, 7, 10.5, 16, 24, \dots\}\,.$$

The following theorem shows that we never lose capacity when using a constraint which is p times more precise.

Theorem 3. *Let $p \in \mathbb{N}$, $\alpha, \theta \in \mathbb{R}$, $\alpha > 0$, $\theta > 1$. Then*

$$\mathsf{cap}(\hat{S}_{1,\alpha,\theta}) \leqslant \mathsf{cap}(\hat{S}_{p,\alpha,\theta}) = p \cdot \mathsf{cap}(\hat{S}_{1,p\alpha,\theta})$$

which holds in equality iff $p \lceil \alpha, \theta \rceil^{i} = \lceil p\alpha, \theta \rceil^{i}$ for all $i \geqslant 0$.

We note that the requirement that p be an integer is necessary for Theorem 3. For example, there are values of θ for which we get $\mathsf{cap}(\hat{S}_{2,1,\theta}) > \mathsf{cap}(\hat{S}_{3,1,\theta})$. In Figure 3, we see $\mathsf{cap}(\hat{S}_{p,1,\theta})$ for $p \in \{1, 2, 4, 8\}$.

Returning to the RLL constraint, being p times more precise usually translates into having a digital clock which runs p times faster. While we gain from faster clocks when using (p, α, θ)-PR, the case is often worse in (d, k)-RLL. This is because when using a clock which runs p times faster, we have to use $(p(d+1) - 1, k)$-RLL since k is not affected by the speed of the clock, but the minimum time between adjacent peaks becomes $p(d+1)$ clock ticks. However, we have $p \cdot \mathsf{cap}(S_{p(d+1)-1,k}) = 0$ when $p(d+1) - 1 \geqslant k$, which for most practical values of p, d, and k, is the case.

If we look at Figure 3 there appears to be an upper bound on the capacity as we continue to increase the precision. This

[1] In practical terms, being p times more precise means the encoder has a clock with p times the original frequency, which allows it to align the peaks to multiples of $1/p$. This does not mean that the encoder writes more densely since the decoder's minimum resolution does not depend on the precision p.

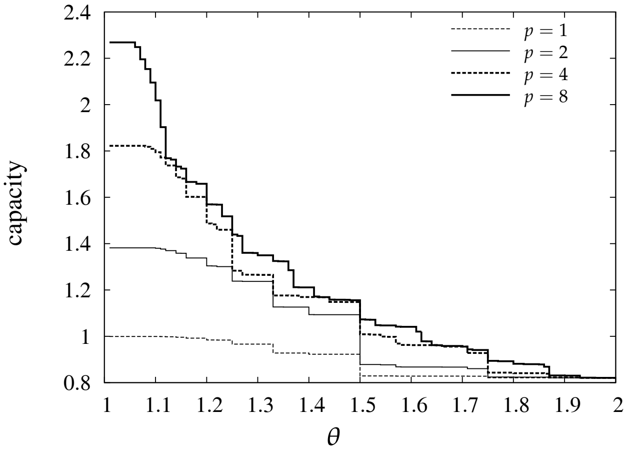

Fig. 3. The capacity of $(p,1,\theta)$-PR, for $p = 1,2,4,8$.

is indeed the case as will shall prove shortly. First, we define the appropriate constrained system with infinite-precision.

Definition 3. *Let $\alpha \in \mathbb{R}$, $\alpha > 0$, be the minimum resolution, and let $\theta = \frac{1+\delta}{1-\delta} > 1$ be the resolution factor, where $\delta \in \mathbb{R}$, $0 < \delta < 1$. Then the (∞,α,θ)*-PR constraint *is the set of all streams with symbols from the set*

$$\mathcal{I}_{\alpha,\theta} \stackrel{\text{def}}{=} \left\{ \alpha\theta^i \mid i = 0,1,2,\dots \right\} \subset \mathbb{R}.$$

Example 3. *For $\alpha = 2$ and $\theta = 1.5$, we have*

$$\mathcal{I}_{\alpha,\theta} = \{2,3,4.5,6.75,10.125,15.1875,22.78125,\dots\}.$$

Since an (∞,α,θ)-PR system is no longer made up of discrete bits, we need the analog of length n strings. We define the *τ-header*, $\tau \in \mathbb{R}$, of such a stream, as the ordered set of symbols appearing between the beginning of the stream and time τ. If we observe the τ-header, we may see several symbols encoded in it one after the other. The last complete symbol may not necessarily reach time τ exactly. The remaining part of the time interval is the beginning of another encoded symbol whose end we do not see, and which we call the *tail*.

Example 4. *For $\alpha = 1$, $\theta = 1.5$ the allowed symbols are of duration $1 = \alpha\theta^0, 1.5 = \alpha\theta^1, 2.25 = \alpha\theta^2,\dots$, and so on. The following 3-headers may be seen: $(1,1,1)$, $(1.5,1.5)$, $(1,1.5,-)$, $(1.5,1,-)$, $(1,1,-)$, $(2.25,-)$, $(1.5,-)$, $(1,-)$, and $(-)$. The $-$ sign denotes the tail.*

We can now proceed with calculating the capacity of the (∞,α,θ)-PR constraint. For simplicity, we handle the $\alpha = 1$ case first, and then prove the general case.

Theorem 4. *Let S_θ denote the $(\infty,1,\theta)$-PR constrained system, $\theta > 1$. Then*

$$\mathsf{cap}(S_\theta) = \log_2 r_\theta$$

where r_θ is the unique solution in $(1,\infty)$ to the equation $f_\theta(x) = 1$ and

$$f_\theta(x) \stackrel{\text{def}}{=} \sum_{i \geqslant 0} x^{-\theta^i}.$$

Proof: We omit the proof that r_θ actually exists and is unique. Let us denote by $S_\theta(\tau)$ the set of all τ-headers of S_θ. Obviously,

$$|S_\theta(\tau)| = \begin{cases} 1 + \sum_{i=0}^{\infty} \left|S_\theta(\tau - \theta^i)\right| & \tau \geqslant 0 \\ 0 & \tau < 0 \end{cases}$$

For technical simplicity let us define $\overline{S}_\theta(\tau)$ to be the set of all τ-headers of S_θ with tail strictly smaller than 1. It is easy to see that

$$\left|\overline{S}_\theta(\tau)\right| = \begin{cases} \sum_{i=0}^{\infty} \left|\overline{S}_\theta(\tau - \theta^i)\right| & \tau \geqslant 1 \\ 1 & 0 \leqslant \tau < 1 \\ 0 & \tau < 0 \end{cases}$$

We contend that $|S_\theta(\tau)| \leqslant \lfloor\tau\rfloor \left|\overline{S}_\theta(\tau)\right|$ for $\tau \geqslant 1$. This is because any τ-header of $\overline{S}_\theta(\tau)$ contains at most $\lfloor\tau\rfloor$ complete symbols, and since its tail is strictly shorter than 1, we can create at most $\lfloor\tau\rfloor$ distinct τ-headers in $S_\theta(\tau)$ by repeatedly merging the tail with the last complete symbol. We can get any τ-header of $S_\theta(\tau)$ that way because we can take any τ-header with tail of length 1 or more, and repeatedly add complete symbols of length 1 while reducing the tail by 1, to get a τ-header from $\overline{S}_\theta(\tau)$.

We now prove by inductionthat $\left|\overline{S}_\theta(\tau)\right\lfloor \leqslant r_\theta^\tau$. For the induction base we point out that obviously $\left|\overline{S}_\theta(\tau)\right| = 0 \leqslant r_\theta^\tau$ for all $\tau \in (-\infty,0)$ by definition. Furthermore, since $r_\theta > 1$, $\left|\overline{S}_\theta(\tau)\right| = 1 \leqslant r_\theta^\tau$ for all $\tau \in [0,1)$. For the induction hypothesis, let us assume that $\left|\overline{S}_\theta(\tau)\right| \leqslant r_\theta^\tau$ for all $\tau \in (-\infty,n)$, $n \in \mathbb{N}$. We then prove the claim also holds for $\tau \in [n,n+1)$. Let $\tau \in [n,n+1)$, then

$$\left|\overline{S}_\theta(\tau)\right| = \sum_{i=0}^{\infty} \left|\overline{S}_\theta(\tau - \theta^i)\right| \leqslant \sum_{i=0}^{\infty} r_\theta^{\tau-\theta^i} = r_\theta^\tau \sum_{i=0}^{\infty} r_\theta^{-\theta^i} = r_\theta^\tau$$

where we are able to use the induction hypothesis since $\theta^i \geqslant 1$ and so $\tau - \theta^i \in (-\infty,n)$. It follows that

$$\mathsf{cap}(S_\theta) =\leqslant \lim_{\tau\to\infty} \frac{\log_2\left(\lfloor\tau\rfloor \left|\overline{S}_\theta(\tau)\right|\right)}{\tau} \leqslant \log_2 r_\theta.$$

We now want to prove that $\mathsf{cap}(S_\theta) \geqslant \log_2 r_\theta$. We define the positive real constant $\gamma = \min\left\{1 - r_\theta^{1-\theta}, r_\theta^{-1}\right\}$ and contend that $|S_\theta(\tau)| \geqslant \gamma r_\theta^\tau$ for all $\tau \geqslant 0$. We do this again by induction. For the induction base take any $\tau \in [0,1)$ and then $|S_\theta(\tau)| = 1 \geqslant r_\theta^{-1} r_\theta^\tau \geqslant \gamma r_\theta^\tau$. For the induction hypothesis we assume that $|S_\theta(\tau)| \geqslant \gamma r_\theta^\tau$ for all $\tau \in [0,n)$, $n \in \mathbb{N}$. We prove the claim also holds for $\tau \in [n,n+1)$. Taking $\tau \in [n,n+1)$

it follows that,

$$\begin{aligned}
|S_\theta(\tau)| &= 1 + \sum_{i=0}^{\infty} \left|S_\theta(\tau-\theta^i)\right| = 1 + \sum_{i=0}^{\lfloor \log_\theta \tau \rfloor} \left|S_\theta(\tau-\theta^i)\right| \\
&\geqslant 1 + \sum_{i=0}^{\lfloor \log_\theta \tau \rfloor} \gamma r_\theta^{\tau-\theta^i} \\
&= 1 + \sum_{i=0}^{\infty} \gamma r_\theta^{\tau-\theta^i} - \sum_{i=\lfloor \log_\theta \tau \rfloor+1}^{\infty} \gamma r_\theta^{\tau-\theta^i} \\
&= \gamma r_\theta^{\tau} + 1 - \sum_{i=\lfloor \log_\theta \tau \rfloor+1}^{\infty} \gamma r_\theta^{\tau-\theta^i} \\
&\geqslant \gamma r_\theta^{\tau} + 1 - \sum_{i=0}^{\infty} \gamma r_\theta^{-\theta^i} \geqslant \gamma r_\theta^{\tau} + 1 - \sum_{i=0}^{\infty} \gamma r_\theta^{-i(\theta-1)} \\
&= \gamma r_\theta^{\tau} + 1 - \gamma \frac{1}{1-r_\theta^{1-\theta}} \geqslant \gamma r_\theta^{\tau}.
\end{aligned}$$

Thus we get

$$\mathsf{cap}(S_\theta) = \lim_{\tau\to\infty} \frac{\log_2 |S_\theta(\tau)|}{\tau} \geqslant \log_2 r_\theta.$$

which completes the proof. ■

To translate the capacity of $(\infty, 1, \theta)$-PR to the capacity of (∞, α, θ)-PR with any $\alpha > 0$, we have the following theorem.

Theorem 5. *Let $S_{\alpha,\theta}$ denote the (∞, α, θ)-PR constrained system, $\alpha > 0, \theta > 1$. Then $\mathsf{cap}(S_{\alpha,\theta}) = \frac{1}{\alpha} \cdot \mathsf{cap}(S_{1,\theta})$.*

The capacity of (∞, α, θ)-PR is an upper bound on the capacity of any (p, α, θ)-PR, $p \in \mathbb{N}$, and is actually the limit as $p \to \infty$. This is shown in the next two theorems.

Theorem 6. *For any $p \in \mathbb{N}$, $\alpha, \theta \in \mathbb{R}$, $\alpha > 0$, $\theta > 1$, let $\hat{S}_{p,\alpha,\theta}$ denote the (p, α, θ)-PR constraint, and $S_{\alpha,\theta}$ denote the (∞, α, θ)-PR constraint. Then*

$$\mathsf{cap}(\hat{S}_{p,\alpha,\theta}) \leqslant \mathsf{cap}(S_{\alpha,\theta}).$$

Theorem 7. *With notation as before,*

$$\lim_{p\to\infty} \mathsf{cap}(\hat{S}_{p,\alpha,\theta}) = \mathsf{cap}(S_{\alpha,\theta}).$$

The capacity of the $(\infty, 1, \theta)$-PR constraint is shown in Figure 4. Simple root-finding techniques were used together with limited precision to approximate the roots of $f_\theta(x) = 1$.

IV. Results Summary

We started by exploring the $(1, \alpha, \theta)$-PR constraint which is a natural extension to (d, k)-RLL. The extension allows more run-lengths to be written than (d, k)-RLL admits, subject to the restriction that the decoder, which suffers from a bounded clock drift, can get no ambiguous reading. In light of this, the RLL restriction to successive run-lengths seems arbitrary.

We further generalized our setting by defining the (p, α, θ)-PR framework, and the (p, α, θ)-PR constraint which realizes the framework. We then showed that as $p \to \infty$, the capacity of (p, α, θ)-PR approaches that of (∞, α, θ)-PR from below.

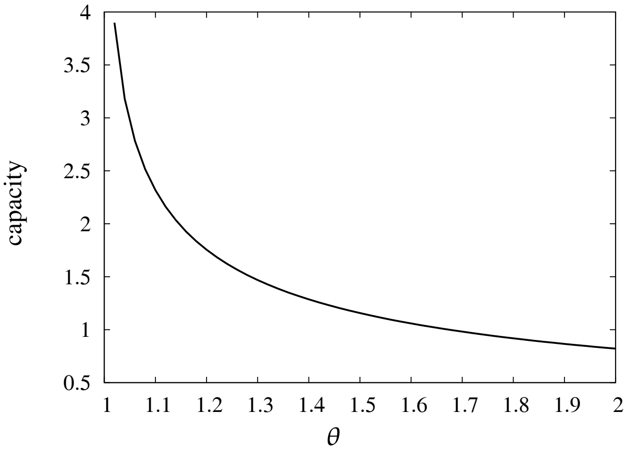

Fig. 4. The capacity of $(\infty, 1, \theta)$-PR

Though the RLL constraint also fits the framework's requirements, the (p, α, θ)-PR constraint seems more natural, and it achieves a higher capacity which does not vanish for $p > 1$ as in the case of RLL. This is especially appealing for engineering reasons: having only an RLL solution, any improvement in precision, e.g., faster clocks, requires a lower clock-drift or else the capacity may drop to zero. However, in the (p, α, θ)-PR constraint, the parameters of precision and resolution are independent. Thus, they allow improvement of one without the other, and offer an optimizable trade-off. In addition, future work may merge PRML into the precision-resolution framework by considering PRML parameters, e.g., sampling rate, as a set of resolution constraints, perhaps achieving another increase in capacity.

References

[1] R. L. Adler, D. Coppersmith, and M. Hassner, "Algorithms for sliding block codes – an application of symbolic dynamics to information theory," *IEEE Trans. on Inform. Theory*, vol. 29, pp. 5–22, 1983.

[2] P. Funk, "Run-length-limited codes with multiple spacing," *IEEE Trans. on Magnetics*, vol. MAG-18, no. 2, pp. 772–775, Mar. 1982.

[3] C. D. Heegard, B. H. Marcus, and P. H. Siegel, "Variable length state splitting with applications to average runlength-constrained (ARC) codes," *IEEE Trans. on Inform. Theory*, vol. 37, pp. 759–777, 1991.

[4] W. H. Kautz, "Fibonacci codes for synchronization control," *IEEE Trans. on Inform. Theory*, vol. IT-11, pp. 284–292, Apr. 1965.

[5] D. Lind and B. H. Marcus, *An Introduction to Symbolic Dynamics and Coding*. Cambridge University Press, 1985.

[6] B. H. Marcus, P. H. Siegel, and J. K. Wolf, "Finite-state modulation codes for data storage," *IEEE J. Select. Areas Commun.*, vol. 10, pp. 5–37, Jan. 1992.

[7] B. H. Marcus, R. M. Roth, and P. H. Siegel, *Constrained systems and coding for recording channels*. V. S. Pless and W. C. Huffman (Editors), Elsevier, Amsterdam, 1998.

[8] S. Mukhtar and J. Bruck, "Interval modulation coding," Paradise Laboratory, California Institute of Technology, Tech. Rep. ETR040, Oct. 2001. [Online]. Available: http://www.paradise.caltech.edu/papers/etr040.pdf

[9] ——, "Interval modulation coding," in *Proceedings of the 2002 IEEE International Symposium on Information Theory, ISIT2002, Lausanne, Switzerland*, June 2002, p. 327.

[10] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423, July 1948.

[11] D. T. Tang and L. R. Bahl, "Block codes for a class of constrained noiseless channels," *Inform. and Control*, vol. 17, pp. 436–461, Dec. 1970.