

Digital Arithmetic in Nature: Continuous-Digit RNS

BEHROOZ PARHAMI*

University of California, Santa Barbara, CA 93106-9560, USA

*Corresponding author: parhami@ece.ucsb.edu

It has been reported in the literature on computational neuroscience that a rat's uncanny ability to dash back to a home position in the absence of any visual clues (or in total darkness, for that matter) may stem from its distinctive method of position representation. More specifically, it is hypothesized that the rat uses a multimodular method akin to residue number system (RNS), but with continuous residues or digits, to encode position information. After a brief review of the evidence in support of this hypothesis, and how it relates to RNS, we discuss the properties of continuous-digit RNS, and derive results on the dynamic range, representational accuracy and factors affecting the choice of the moduli, which are themselves real numbers. We conclude with suggestions for further research on important open problems concerning the process of selection, or evolutionary refinement, of the set of moduli in such a representation.

Keywords: arithmetic algorithms; continuous-valued digits; hybrid analog/digital encoding; number representation

Received 30 July 2013; revised 06 May 2014

Handling editor: Fionn Murtagh

1. INTRODUCTION

Neurobiologists have long been in awe of the common rat's ability to return to its starting position in a straight line, after a long sequence of movements in different directions. This ability (Fig. 1a) persists in the absence of visual and other environmental clues. Experiments in total darkness, and in locations not previously visited by the rat, have confirmed that this is an innate ability resulting, in part, from internal neural activity as opposed to the processing of environmental clues. Of course, there is no reason to believe that the rat's nervous system operates in a manner that is fundamentally different from other animals, or even humans. So, a study of rat's navigation mechanisms, and the attendant encoding of position information, is quite important to neurobiology.

We will not enter into a detailed discussion of the biological bases of the rat's ability to localize within an environment that could span 10s or 100s of meters. Rather, we touch upon the most important of the supporting evidence, reported recently by Fiete *et al.* [1], to provide motivation for our study of residue number systems (RNS) with continuous, real-valued digits (residues) and moduli.

Briefly, a landmark paper by Hafting *et al.* [2], established that in addition to 'place cells,' which are activated when the

rat is in specific locations, thus encoding location information to help with the acquisition of memory, there exist 'grid cells' whose firing is not linked to specific locations, but rather to the rat's relative in-cell position within a periodic, hexagonal grid (Fig. 1b). When the rat moves to the vicinity of the grid point A, say, a specific cell associated with that grid fires. Thus, firing of a grid cell provides partial information about the rat's location (Fig. 1c [3]), in the same manner that knowing the residue of an integer modulo m supplies partial information about its value. Such partial information, obtained from several grids of different resolutions and orientations (Fig. 1d [3]), would, in principle, allow the rat to derive full location information. Again, the analog in digital arithmetic is the identification of an integer R from several residues, via the Chinese remainder theorem (CRT) [4].

For simplicity, let us ignore the 2D (or even 3D) nature of a rat's environs and focus on a 1D model at the outset. As shown in Fig. 2, given a 1D grid x , the real value R might be specified by a pair (i, ϕ) , where i is an integer identifier of a grid cell or interval and ϕ is a real-valued 'phase' or displacement within the cell. Similarly, R can be specified by the pair (j, ψ) in relation to a second 1D grid y . Now, given only the phases ϕ and ψ within the grids x and y , we may be able to deduce R . The

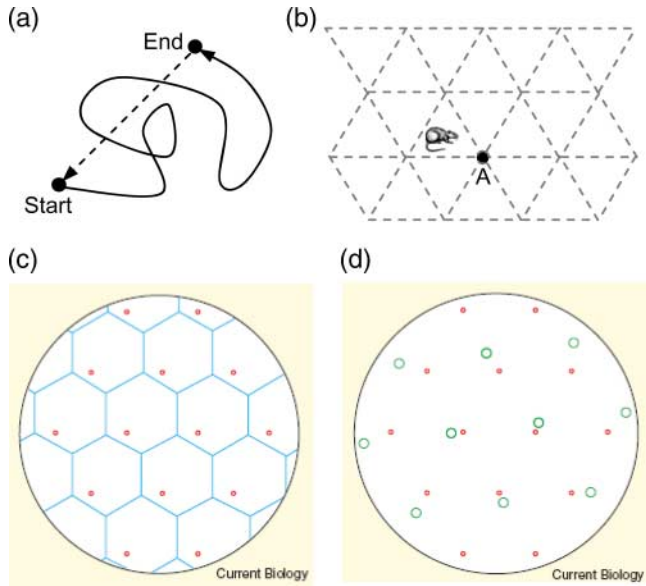


FIGURE 1. The notion of ‘grid cells’ in a rat’s sense of location. (a) Travel and return paths, (b) rat’s hexagonal grid, (c) firings and locations [3] and (d) two hexagonal grids [3].

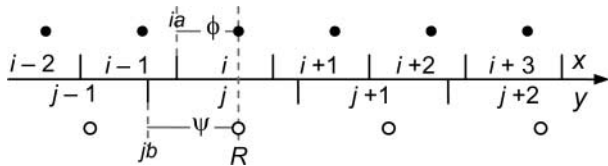


FIGURE 2. Localization with two grids in 1D space.

phase ϕ within grid x represents any of the heavy dots in Fig. 2, whereas the phase ψ within grid y is associated with any of the values marked by small, hollow circles. Given a limited range of values for the indices i and j , the two phases could pinpoint the value R in the 1D space, depending on the accuracy with which ϕ and ψ are known.

Assuming that the cell indexed 0 in each of the two grids x and y begins at 0, grid cell widths are a and b , respectively, and the phases ϕ and ψ are known exactly, the location of R is obtained by solving the linear Diophantine equation $ia + \phi = jb + \psi$ for integer indices i and j . Because a linear Diophantine equation has either no solution or an infinite number of solutions [5], ensuring that the solution $R = ia + \phi$ is unique requires the placement of bounds on the values of i and j , thus restricting the dynamic range. The effect of imprecision in the phase values ϕ and ψ is to further restrict the dynamic range of the representation scheme, as we will see later.

In this paper, we study continuous-digit RNS (CD-RNS) as a model for the aforementioned grid-cell-based localization, solving an open problem posed in [1] regarding the range of number representation. After a review of RNS and the definition

of CD-RNS in Sections 2 and 3, we derive the representation range as a function of accuracy in Section 4. We then discuss moduli selection, arithmetic and extensions in Sections 5, 6 and 7, respectively. Conclusions and some directions for future work are presented in Section 8.

2. RESIDUE NUMBER SYSTEMS

RNS have been studied since the early days of digital computers. Initial hopes of superfast arithmetic, arising from independent operations on residues within parallel channels, were shattered by the realization that certain difficult arithmetic operations, such as division, and commonly used decision processes, like sign or overflow detection, can nullify much or all of the gain from parallelism. Consensus thus developed that only certain signal processing tasks that are dominated by additions and multiplications are appropriate candidates for RNS implementation. An extensive body of techniques and applications in this regard was already developed by 1986 [6]. More recently, the potential of RNS for low-power arithmetic has gained attention (e.g. [7, 8]).

In the context discussed above, RNS is used to represent integers or scaled fixed-point values. Given pairwise relatively prime integer moduli $m_{k-1} > \dots > m_1 > m_0$, an integer R is represented by its ordered set of residues $(r_{k-1}, \dots, r_1, r_0)$ with respect to the k moduli. The dynamic range of this number system, that is, the number of distinct integers that are uniquely represented, is $M = m_{k-1} \dots m_1 m_0$. This range can be used for the unsigned values $[0, M - 1]$, the symmetric range $[-M/2, M/2]$ or any other set of M consecutive integer values.

In RNS, addition, subtraction and multiplication operations are performed independently on each residue. Given the RNS operands $S \equiv (s_{k-1}, \dots, s_1, s_0)$ and $T \equiv (t_{k-1}, \dots, t_1, t_0)$, their sum/difference and product are given by

$$S \pm T \equiv (|s_{k-1} \pm t_{k-1}|_{m_{k-1}}, \dots, |s_1 \pm t_1|_{m_1}, |s_0 \pm t_0|_{m_0})$$

$$S \times T \equiv (|s_{k-1} \times t_{k-1}|_{m_{k-1}}, \dots, |s_1 \times t_1|_{m_1}, |s_0 \times t_0|_{m_0})$$

Notationally, $|z|_m$ stands for $z \bmod m$. Division is in general difficult, making RNS unsuitable for applications involving even moderate use of division. Division by a modulus, known as scaling, can be performed fairly efficiently, however.

Problems to be addressed in applying RNS in practice include choice of the moduli (how many, and of what values), conversions from/to binary format for input/output, and determination of when and how to scale the intermediate results to avoid overflow. Moduli of the form $2^i \pm 1$ have received much attention from researchers and practitioners [9], given that such moduli vastly simplify both modular operations and input/output conversions. By including a single modulus 2^j among the moduli of the form $2^i \pm 1$, various special sets of moduli are derived (e.g. [10, 11]). Readers interested in learning more about the theoretical and practical aspects of RNS, as well

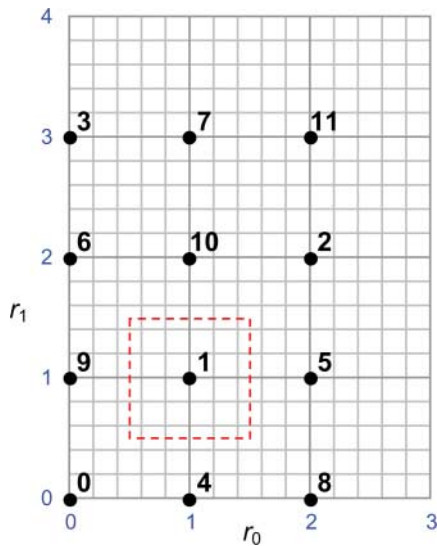


FIGURE 3. Conventional RNS with the moduli 3 and 4.

as some of its applications, can consult the book by Omondi and Premkumar [4].

In the rest of this section, we focus only on the dynamic range and its properties, in preparation for our discussion of the CD case in Section 3. The example RNS with moduli $m_1 = 4$ and $m_0 = 3$ has a dynamic range of 12, which can be used to represent the natural numbers 0–11, as depicted in Fig. 3. The next natural number, 12, has the same representation as 0, so 11 is as high as we should go.

Now, forgetting for the moment that the residues r_1 and r_0 are restricted to natural numbers, let us ponder the effect of having real-valued residues, represented, for example, by analog signals. If we change $(r_1, r_0) = (1.0, 1.0)$ to $(r'_1, r'_0) = (1.0 + \epsilon_1, 1.0 + \epsilon_0)$, where ϵ_1 and ϵ_0 are absolute errors, we can still correctly decode the number, as long as both ϵ_1 and ϵ_0 are <0.5 . We see that the horizontal/vertical separation of codes in Fig. 3 by a minimum of 1 unit allows us to correct any error of <0.5 in residues. In other words, r'_1 and r'_0 can vary anywhere inside the dashed box surrounding the representation of 1 in Fig. 3, without causing a decoding error. Of course, if we were interested in error detection, rather than correction, then any error of <1 would be detectable. Similar error detection and correction capabilities may be provided by additional or redundant moduli, which lead to the extension of dynamic range, without actually utilizing all the values that would be representable [12].

3. CONTINUOUS-DIGIT RNS

The RNS of Fig. 3, with continuous or analog digits, can be viewed as representing real numbers in the range $[0.0, \mu)$, where μ is the dynamic range. But what would be an appropriate value

for μ ? For ease of future reference, we tackle this question in Example 1. Notationally, we switch to the use of μ for dynamic range and ρ for residues, to signify that they are real valued rather than integers.

Example 1. If, in Fig. 3, we drew lines with the slope of 1 to connect all the values 0–11 in ascending order, intermediate points on these line would correspond to representations of nonintegers. Thus, the value 6.3 yields $(\rho_1, \rho_0) = (0.3, 2.3)$, a point on the sloped line between 6 and 7. It is readily seen that we can go beyond 11 in this case. If the real-valued residues are exact, the representable range will be $[0.0, 12.0)$; otherwise, if ϵ_{\max} denotes the maximum error in ρ_1 or ρ_0 , that is, $\epsilon_{\max} = \max((\max \epsilon_1), (\max \epsilon_0))$, then the range that ensures correct decoding, within an error of ϵ_{\max} , is $[0.0, 12.0 - \epsilon_{\max})$.

At any rate, the dynamic range is on the order of $m_1 m_0$. Note that even though the residues ρ_1 and ρ_0 are now real valued, m_1 and m_0 being relatively prime is still a precondition for maximizing the dynamic range. More importantly, we cannot increase the dynamic range beyond $m_1 m_0$, even if the real-valued residues are known to be very accurate. This is because the wrap arounds in Fig. 3 occur at precisely the same points for both real valued and integer residues.

Consider the following analysis as a starting point for solving the limited dynamic range problem. Given the RNS encoding of a value R , with the two residues containing (signed) errors of ϵ_1 and ϵ_0 , the erroneous residues specify a point R' on the 2D residue plane (Fig. 4). To decode this number, we might attempt to find the closest point R'' to it on one of the slanted lines of Fig. 3. It is readily seen that, as long as ϵ_1 and ϵ_0 are <0.5 , the closest line to R' would be the same line that contains R and that the error in the decoded value R'' is at most ϵ_{\max} . This is because the coordinates of R'' in Fig. 4 are $\rho''_1 = \rho_1 + (\epsilon_1 + \epsilon_0)/2$ and $\rho''_0 = \rho_0 + (\epsilon_1 + \epsilon_0)/2$, leading to the conclusion $R - \epsilon_{\max} \leq R'' \leq R + \epsilon_{\max}$. Note that in Fig. 4, we

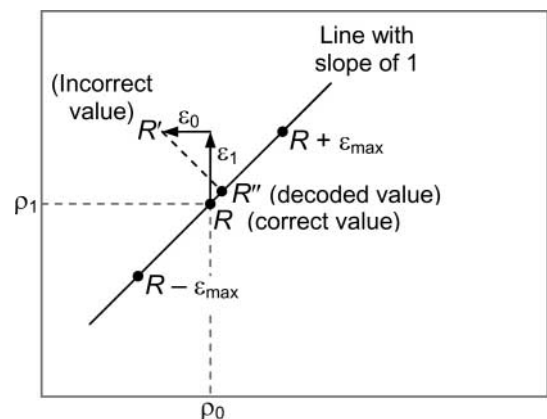


FIGURE 4. Decoding error with CD-RNS.

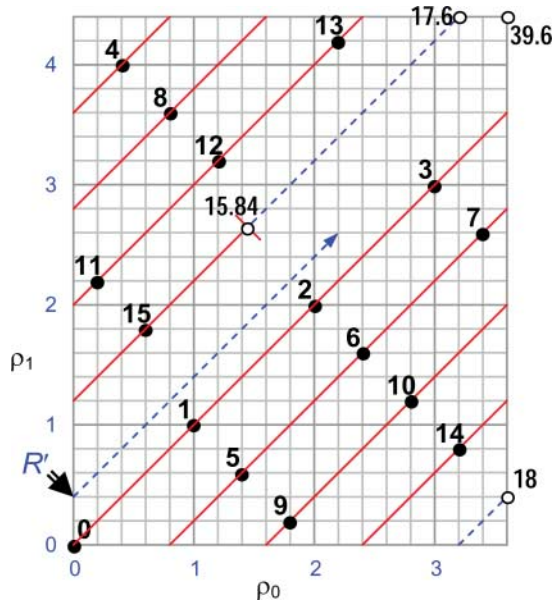


FIGURE 5. CD-RNS with the moduli 3.6 and 4.4.

have $\varepsilon_1 > 0$ and $\varepsilon_0 < 0$, which do not represent the worst-case error conditions.

Now that we have opted for continuous digits (residues), there is no good reason to insist that the moduli m_1 and m_0 be whole numbers in a CD-RNS.

Example 2. Consider a CD-RNS with $\mu_1 = 4.4$ and $\mu_0 = 3.6$ (Fig. 5). The set of values represented in this RNS fall on lines of slope 1, starting with 0 at the origin and ending at the dynamic range μ , as yet to be determined. The product of the two moduli, that is, $15.84 = 4.4 \times 3.6$, would be a reasonable first guess. But, we see from Fig. 5 that we can go past 15.84, all the way to 18.0 (see the lower right corner of Fig. 5), without the horizontal or vertical line separations becoming < 0.8 . Actually, if we want an error of ± 0.4 in either residue to be correctable, we have to stop just under 17.6, a value represented with $\rho_1 = 0$ and $\rho_0 = 3.2$. This is because an error of $+0.4$ in both these residues at $R = 17.6$ will yield $(0.4, 0.0)$ as the residue representation, which defines Point R' , thus causing the value to be interpreted as $R'' = 0.2$ after correction, rather than $R = 17.6$, per our procedure of Fig. 4.

Example 2 shows that the same error tolerance is provided with the extended range $[0.0, 17.6)$, in lieu of $[0.0, 15.84)$. In other words, reducing the line separations from 1.0 in Fig. 3 to 0.8 in Fig. 5, a reduction factor of 1.25, has allowed us to extend the dynamic range by $\sim 11\%$, from 15.84 to 17.6. Intuitively, the reason for the dynamic range not increasing by a factor of 1.25 is the unevenness of line spacings in Fig. 5, which leads to some waste.

Example 3. If we decide to go beyond 18.0 in the CD-RNS of Example 2, the continuation of the dashed line from point R' in Fig. 5 suggests that we can do so, provided that the error tolerance is reduced from 0.4 to 0.2. In fact, we can go all the way to 39.6, represented by the point at the upper right corner of Fig. 5, after wrapping around at 21.6, 22.0, 25.2, 26.4, 28.8, 30.8, 32.4, 35.2 and 36.0, successively. Again, to ensure that an error of ± 0.2 in either residue is correctable, we have to end the dynamic range just under 39.4.

In the context of a rat's position encoding, as discussed in Section 1, if 1D grids with cell widths of 44 and 36 cm ($\mu_1 = 4.4$ and $\mu_0 = 3.6$, in units of 10 cm) are used, and assuming that the rat knows its phase inside each of the two 1D cell types within ± 2 cm, then a dynamic range of 39.4×10 cm ≈ 4 m is available (the number 39.4 was derived in Example 3). The example above might be considered realistic, given that measured grid distances tend to be ~ 2 – 3 times the length of the animal [3].

In the preceding example, what limits the dynamic range to ~ 4 m is a combination of the assumed error bound of $\varepsilon_{\max} = 0.2$ and the chosen moduli $\mu_1 = 4.4$ and $\mu_0 = 3.6$. If we reduce the error bound to 0.1 (± 1 cm), the dynamic range will remain the same, however, because the value $39.6 + \beta$ has the exact same representation as $0 + \beta$. So, even much greater accuracy would not help extend the dynamic range. On the other hand, if we modify one of the moduli slightly, the final wrap around will not lead to a line that exactly overlaps the line beginning at 0, and we may be able to accommodate a wider dynamic range. Relationships between the moduli, precision and dynamic range are explored in the next section.

4. DYNAMIC RANGE AND ACCURACY

We continue with the assumption of an RNS with only two moduli μ_1 and μ_0 , and further assume, at the outset, that residues are accurate to within $\pm(\mu_1 \bmod \mu_0)/2$, which is dependent on the moduli chosen. We will later consider the effect of precision as an independent parameter.

THEOREM 1. Let $\sigma_1 = \lfloor \mu_1 / \mu_0 \rfloor$. If the residues ρ_1 and ρ_0 are accurate to within $\pm \varepsilon_{\max} = \pm \sigma_1 / 2$, then the dynamic range of a CD-RNS with $\mu_1 > \mu_0$ is $\mu = \mu_0(1 + \lfloor \mu_1 / \mu_0 \rfloor \lfloor \mu_0 / \sigma_1 \rfloor)$.

Proof of theorems appear in Appendix.

Note that in Example 1, we derived the dynamic range as $[0.0, 12.0 - \varepsilon_{\max})$. Similarly, in Examples 2 and 3, we subtracted ε_{\max} from the upper end of a computed range to ensure that the last number included within the range is distinguishable from very small values at the start of the range. In Theorem 1, and henceforth, we omit the very small $-\varepsilon_{\max}$ adjustment, in an effort to simplify the expressions involved. Thus, the statement of Theorem 1 should really entail the expression $\mu_0(1 + \lfloor \mu_1 / \mu_0 \rfloor \lfloor \mu_0 / \sigma_1 \rfloor) - \varepsilon_{\max}$.

When μ_1 is not close to a multiple of μ_0 , Theorem 1 essentially establishes a lower bound on the available dynamic range, given that its accuracy requirement for the residues is quite modest.

Example 4. With μ_1 and μ_0 integers and $\sigma_1 = 1$, the dynamic range of Theorem 1 will extend to $\mu = \mu_0(1 + \lfloor \mu_1/\mu_0 \rfloor \mu_0) = \mu_0\mu_1$, provided residues are accurate to within ± 0.5 . This is consistent with the dynamic range of $[0.0, 11.5)$, which turns into $[0.0, 12.0)$ when ignoring $-\varepsilon_{\max}$, obtained in Example 1.

Example 5. For $\mu_1 = 4.4$, $\mu_0 = 3.6$, leading to $\sigma_1 = 0.8$, an accuracy of ± 0.4 in the residues would guarantee a dynamic range of $[0.0, 18.0)$, matching to within the ignored $-\varepsilon_{\max}$ term the result $[0.0, 17.6)$ that we obtained in Example 2.

Now, given an arbitrary precision $\pm \varepsilon_{\max}$ for the two residues, deriving the dynamic range is a bit more complicated. First, we note that if either of the moduli is an integral multiple of σ_1 , then the dynamic range will be limited to that given in Theorem 1, provided that $\varepsilon_{\max} \leq \sigma_1/2$. This is because the sloped lines of Fig. 4 will eventually wrap around and trace the same path as that beginning at point 0. This occurs in Fig. 3, for example, limiting the dynamic range to $[0, 12.0)$. On the other hand, in Fig. 5, neither μ_1 nor μ_0 is divisible by $\sigma_1 = 0.8$, allowing the dynamic range to extend beyond $[0, 18.0)$, provided that $\varepsilon_{\max} < 0.2$. Generally speaking, this extension is possible if the residues of μ_1 and $\mu_0 \bmod \sigma_1$ are at least $2\varepsilon_{\max}$. This is formalized as follows.

THEOREM 2. Let $\sigma_{-1} = \mu_1$, $\sigma_0 = \mu_0$ and, for $0 \leq i \leq j$, $\sigma_{i+1} = \min(|\sigma_{i-1}|_{\sigma_i}, \sigma_i - |\sigma_{i-1}|_{\sigma_i})$, where j is the largest possible index for which $\sigma_j \geq 2\varepsilon_{\max}$. If the residues ρ_1 and ρ_0 are accurate to within $\pm \varepsilon_{\max}$, then the dynamic range μ of a CD-RNS with the moduli μ_1 and μ_0 is lower bounded by

$$\mu_0(1 + \lfloor \mu_1/\mu_0 \rfloor \lfloor \mu_0/\sigma_1 \rfloor) \lfloor \sigma_1/\sigma_2 \rfloor \lfloor \sigma_2/\sigma_3 \rfloor \dots \lfloor \sigma_{j-1}/\sigma_j \rfloor$$

Given that the expression in the statement of Theorem 2 is rather complicated, an intuitive explanation that supplements the formal proof in Appendix might be helpful. If we remove the floor operators (which essentially model the waste due to unequal line spacings of the types seen in Figs 5 and 6) and assume $\sigma_j = 2\varepsilon_{\max}$, both bounds reduce to $\mu_0\mu_1/(2\varepsilon_{\max})$, indicating that under ideal conditions, the allowed dynamic range is directly proportional to the two moduli μ_1 and μ_0 and varies inversely with the maximum representation error ε_{\max} . One implication of this result is that the same pair μ_1 and μ_0 of moduli can be used to cover a wider dynamic range with more accurate residues, provided certain conditions on the parameters σ_i are met.

THEOREM 3. Define $\gamma = \max(2\varepsilon_{\max}, \delta)$, where δ is the largest number that exactly divides both μ_0 and μ_1 , if such a number

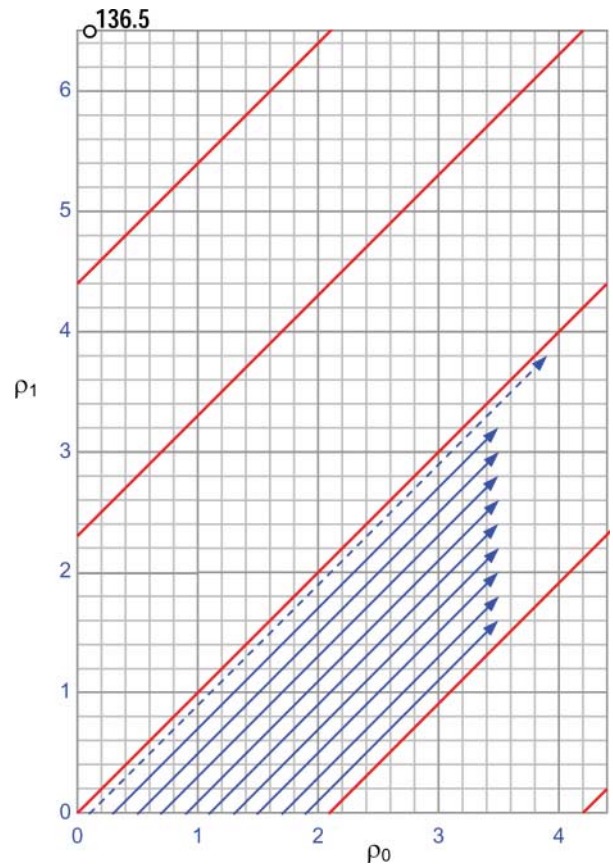


FIGURE 6. CD-RNS with the moduli 6.5 and 4.4.

exists, and 0 otherwise. Then, the dynamic range μ of a CD-RNS with the moduli μ_1 and μ_0 is upper bounded by

$$\max(\mu_0 \lfloor \mu_1/\gamma \rfloor, \mu_1 \lfloor \mu_0/\gamma \rfloor)$$

The following corollary entails a useful variation of the result of Theorem 3.

Corollary 1. To attain a dynamic range μ with a maximum residue error ε_{\max} , we must have $\mu_0\mu_1 \geq 2\mu\varepsilon_{\max}$.

When the moduli μ_0 and μ_1 are comparable in magnitude, the lower and upper bounds of Theorems 2 and 3 are close to each other, thus providing a fairly accurate estimate for the dynamic range μ . We will see shortly that optimal choice of the moduli essentially entails an attempt to make the two bounds approach each other. Put another way, the most efficient CD-RNS representations tend to have moduli that are fairly close in magnitude. Interestingly, in ordinary (integer-valued) RNS, moduli that are comparable in magnitude are also desirable because they lead to more efficient and uniform circuitry; speed is determined by the largest modulus, so making the other ones larger entails no speed penalty.

Example 6. Assuming $\varepsilon_{\max} = 0.2$ in the CD-RNS of Fig. 5, we obtain $\sigma_1 = 0.8$ and $\sigma_2 = 0.4$, with $\sigma_3 = 0.0$ violating the requirement that the last σ_j not be $< 2\varepsilon_{\max}$. The lower- and upper-bound expressions of Theorems 2 and 3 evaluate to 36.0 and 39.6, respectively, indicating a dynamic range of at least $[0.0, 36.0)$. Note that the true range is $[0.0, 39.6)$, with its top end shown at the upper right corner of Fig. 5, that is, the point where full wrap around from 39.6 to 0 occurs.

Example 7. Raising the accuracy of residues in Example 6 to $\varepsilon_{\max} = 0.1$ would not increase the dynamic range, given the full wrap around at 39.6. The pessimistic lower-bound expression of Theorem 2 reflects this limitation, as its value remains at 36.0 for $\varepsilon_{\max} < 0.2$. On the other hand, the optimistic upper bound of Theorem 3 indicates the wider range $[0.0, 79.2)$, which is clearly unattainable. Slightly different values for the moduli μ_1 and μ_0 could fix this problem, however.

Example 8. We consider a case where σ_1 is not an integral multiple of σ_2 . For the CD-RNS of Fig. 6, we have $\sigma_{-1} = \mu_1 = 6.5$, $\sigma_0 = \mu_0 = 4.4$, $\sigma_1 = |\mu_1|_{\mu_0} = 2.1$ and $\sigma_2 = |\mu_0|_{\sigma_1} = 0.2$. Assuming $\varepsilon_{\max} = 0.1$, Theorem 2 yields the dynamic range lower bound 132.0. The upper end of the derived range is less than the ideal value $\mu_1\mu_0/(2\varepsilon_{\max}) = 143.0$ because some of the potential dynamic range is lost to the unequal spacing between lines of slope 1 in Fig. 6 and some to the fact that the lower bound is generally less than the true value. Note, in particular, that the solid line just below the one originating at 0 in Fig. 6 has a horizontal and vertical separations of 0.3, instead of 0.2. The next line after wrap around, the dashed one in Fig. 6, will have horizontal and vertical separations of 0.1, which is not allowed with $\varepsilon_{\max} = 0.1$. The actual dynamic range is 136.5.

Example 9. Consider the smaller error bound $\varepsilon_{\max} = 0.05$ in Fig. 6. The dashed line, along with its (wrapped) successors, will then be allowed, leading to increases in the dynamic range lower/upper bounds provided by Theorems 2 and 3 (to 264.0 and 286.0, respectively); the actual range is $[0, 286.0)$.

A natural question at this point is whether we can derive an exact formula for the range of a CD-RNS, rather than rely only on the bounds provided by Theorems 2 and 3. The author's attempts in this direction have not been successful, but this does not necessarily imply that a closed-form formula does not exist. To provide a sense of the difficulty of deriving an exact formula, we construct an example showing the relationship between the bounds and the exact dynamic range for a particular family of CD-RNS representations.

Example 10. Let us fix μ_1 at 4.4 and ε_{\max} at 0.2, and study how the dynamic range varies as we pick different μ_0 values in the range 2.5–4.3. Lower and upper bounds of Theorems 2 and 3 are shown as white and black circles, respectively, in Fig. 7. The lower bound is shown only for μ_0 in $[2.5, 4.0]$ because

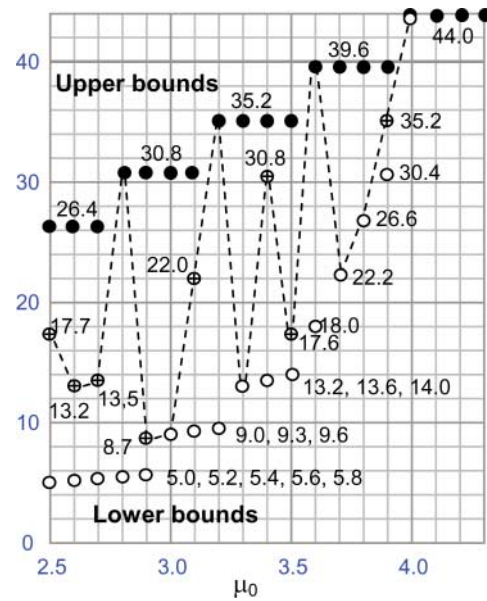


FIGURE 7. Dynamic range and its bounds for $\mu_1 = 4.4$, as μ_0 varies.

beyond that the condition for Theorem 1 is not met. The actual dynamic range, obtained via manual calculation, is shown as the dashed line, with XOR (\oplus) symbols denoting points that do not coincide with the lower or upper bound. Chaotic variation of the dynamic range is a good indication that an exact formula, if it exists, will be quite complicated.

Note that in Example 10, upper and lower bounds approach each other as the smaller modulus gets closer to the larger one, with the bounds coinciding when μ_0 reaches $\mu_1 - 2\varepsilon_{\max}$. The latter choice also maximizes the dynamic range and the lower and upper bounds for it. This observation leads to the general result in Theorem 4.

In conventional RNS, the moduli being pairwise relatively prime lead to maximization of the dynamic range. The corresponding result for CD-RNS is as follows.

THEOREM 4. Let $\sigma_{-1} = \mu_1$, $\sigma_0 = \mu_0$, and, for $0 \leq i \leq j$, $\sigma_{i+1} = \min(|\sigma_{i-1}|_{\sigma_i}, \sigma_i - |\sigma_{i-1}|_{\sigma_i})$, where j is the largest possible index for which $\sigma_j \geq 2\varepsilon_{\max}$. The maximum possible dynamic range of $\mu_0\mu_1/(2\varepsilon_{\max})$ is obtained if $\sigma_j = 2\varepsilon_{\max}$.

Note that the maximum range specified by Theorem 4 is achieved in Examples 1, 3, 4, 6 and 9.

5. CHOOSING THE CD-RNS MODULI

Based on Theorem 1, given a maximum error $\pm\varepsilon_{\max}$ for our residues, we can choose the moduli to be $\mu_0 + 2\varepsilon_{\max}$ and μ_0 , leading to the dynamic range $\mu_0\lfloor\mu_0/(2\varepsilon_{\max})\rfloor + \mu_0$. This result suggests that to cover a desired dynamic range $[0, \mu)$ with

residues that are accurate to $\pm\epsilon_{\max}$ and moduli that are as small as possible, the moduli should approximately equal $(2\mu\epsilon_{\max})^{1/2}$ and they should differ by $2\epsilon_{\max}$.

Example 11. A dynamic range of $[0, 100.0)$ with $\epsilon_{\max} = 0.1$ translates to moduli on the order of 4.5. Choosing $\mu_1 = 4.6$ and $\mu_0 = 4.4$ yields the dynamic range $[0.0, 101.2)$.

Note that the dynamic range of Example 11 could have been computed by scaling up ϵ_{\max} to 1, scaling up the two moduli by the same factor of 10 to integer values $m_1 = 46$ and $m_0 = 44$, and then using the rules for ordinary RNS. This would have yielded $M = m_1m_0/\gcd(m_1, m_0) = 1012$. Scaling back down by a factor of 10 confirms the derived range of $[0.0, 101.2)$. However, the same could not be done when 1 is not an integral multiple of ϵ_{\max} .

Example 12. To attain a dynamic range of $\mu = 100.0$ with $\epsilon_{\max} = 0.15$, the two moduli must be on the order of $(2 \times 100.0 \times 0.15)^{1/2} = 5.5$. Choosing $\mu_1 = 5.6$ and $\mu_0 = 5.4$ yields the dynamic range $[0, 102.6)$. There is no direct counterpart to this latter CD-RNS among ordinary RNS.

Now consider the case when a rough magnitude for one of the moduli is known. Without loss of generality, let the smaller of the two moduli μ_0 be known. Further, assume that the given μ_0 is an integral multiple of $2\epsilon_{\max}$ (if not, minor adjustment to μ_0 will make it so). Then, to achieve a dynamic range of μ , Corollary 1 suggests that μ_1 must be at least $2\mu\epsilon_{\max}/\mu_0$. One way to ensure the adequacy of a near-minimal value for μ_1 is to adjust it upward, if necessary, until it is a multiple of $2\epsilon_{\max}$, compute the lower bound of Theorem 2, and continue adjusting upward by $2\epsilon_{\max}$ until the lower bound equals or exceeds the desired dynamic range. Alternatively, one could aim for satisfying the condition of Theorem 4.

Example 13. Consider a goal of $\mu \geq 100.0$, with $\epsilon_{\max} = 0.15$ and $\mu_0 \approx 3.5$. We take $\mu_0 = 3.6$, which is a multiple of $2\epsilon_{\max}$. Corollary 1 yields $\mu_1 \geq 8.33$. Taking $\mu_1 = 8.4$ leads to the lower bound of 25.2, which is too small. Adding $= 0.3$ to make $\mu_1 = 8.7$ yields the lower bound 72.0, which is still inadequate. We could continue with increasing μ_1 to 9.0 and beyond, but Theorem 4 guarantees a dynamic range of 104.4 with $\mu_0 = 3.6$ and $\mu_1 = 8.7$, given that $\sigma_3 = 0.3$.

A general algorithm for choosing an optimal moduli set for a CD-RNS of a desired dynamic range is unlikely to exist, given that for ordinary RNS, constituting special cases of CD-RNS, such a tool is not available. At this juncture, the best we have for ordinary RNS are trial-and-error methods aided by certain mathematical results and heuristics to limit the search space [13]. More work remains to be done in developing corresponding tools for CD-RNS.

6. ARITHMETIC IN CD-RNS

The main aspects of arithmetic discussed in this section pertain to forward and reverse conversions; that is, conversion from binary to CD-RNS and vice versa. CD-RNS arithmetic operations, such as addition, subtraction and multiplication, are identical to those of conventional RNS [13]. In the context of a rat's sense of location, discussed in Section 1, updating of position may entail adding appropriate offsets to a previously stored location vector [1]. The addition occurs independently for each component (residue) in the vector and entails a very limited range of values in analog form.

Forward conversion to CD-RNS is similar to binary-to-RNS conversion in all respects. Thus, it can be accomplished by standard table-lookup, followed by modular multioperand addition [4]. The fact that the moduli may be nonintegers does not complicate either of the two steps. At the circuit level, designs similar to A/D converters can be used to transform an analog input to modulo- μ_i residues. For both approaches, the entire process, including the final multioperand addition, if any, can be mapped onto a neural network.

The reverse conversion is perhaps more important in the biological context, because some type of absolute positional information is needed to allow the rat to zip back to its starting point in a straight line. How the post-conversion absolute coordinate data might be stored or processed by the rat is not elaborated upon in neurobiological literature available to us. We will thus not postulate on the practical application of our reverse conversion process, focusing solely on how it might be accomplished in a feasible way.

In the case of ordinary RNS, the CRT or its various derivatives [4, 13] are used for reverse conversion. These methods essentially compute a weight w_i corresponding to each modulus m_i , with the converted value x then derived as

$$x = w_{k-1}r_{k-1} + \dots + w_1r_1 + w_0r_0 \pmod{M}$$

The weight w_i associated with m_i is the smallest multiple of M/m_i that is 1 mod m_i .

Example 14. Consider the conventional RNS with $m_1 = 4$, $m_0 = 3$ and $M = m_1m_0 = 12$ (Fig. 3). The weight w_1 is a multiple of $M/m_1 = 3$ such that $w_1 = 1 \pmod{4}$; thus, $w_1 = 9$. Similarly, w_0 is a multiple of $M/m_0 = 4$ such that $w_0 = 1 \pmod{3}$; thus, $w_0 = 4$. Given the representation (r_1, r_0) in our $(4, 3)$ RNS, it can be converted to decimal via the weighted-sum computation $x = 9r_1 + 4r_0 \pmod{12}$. For example, the residue pair $(3, 1)$ represents $9 \times 3 + 4 \times 1 \pmod{12} = 7$.

Unfortunately, standard methods based on the CRT and its derivatives [4, 13] are inapplicable to CD-RNS. It suffices to show the inapplicability of CRT in the special case of a CD-RNS using moduli that are scaled-down versions of a conventional RNS with integer moduli. We do this through an example.

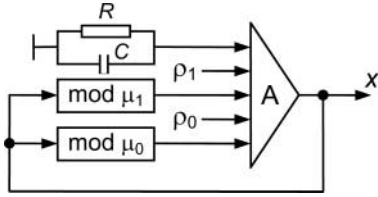


FIGURE 8. Circuit for CD-RNS to binary conversion.

Example 15. Consider the CD-RNS of Example 11 and its conventional RNS counterpart with $m_1 = 44$ and $m_0 = 23$, having a dynamic range of 1012 (we have divided the larger modulus 46 by 2 to make the moduli relatively prime for this demonstration). The CRT weights associated with the two moduli are 529 and 484, respectively. Thus, the RNS number (r_1, r_0) represents $x = 529r_1 + 484r_0 \bmod 1012$, or in scaled form [14, 15], $x/1012 \approx 0.522727r_1 + 0.478261r_0 \bmod 1$. Given that we have $\varepsilon_{\max} = 0.5$ in our conventional RNS formulation, the error in the latter expression can be as large as $(0.522727 + 0.478261) \times 0.5 \approx 0.5$, indicating a relative error of $>50\%$ in the scaled conversion process.

Example 15 demonstrates that small errors in the residues may be amplified through the conversion process, leading to useless results. One way out is to view the reverse conversion process as a nonlinear optimization problem [16]. As shown schematically in Fig. 8, a single-layer feedback network may be utilized whose inputs are the given residues ρ_1 and ρ_0 , along with the residues $|x|_{\mu_1}$ and $|x|_{\mu_0}$ of the output. Note that the latter values can be formed using a forward converter. The resulting arrangement can be shown to converge to the correct result within the RC time constant of the circuit.

We used Example 15 to demonstrate that CRT-like methods used in connection with conventional RNS do not carry over to CD-RNS. This negative conclusion should not be interpreted as an impossibility result. Further research may reveal ways of adapting CRT-like methods to CD-RNS.

7. EXTENSIONS AND GENERALIZATIONS

Thus far, we have limited our discussion to CD-RNS with two moduli μ_1 and μ_0 . Whereas a conventional RNS with three pairwise relatively prime moduli m_2 , m_1 and m_0 can be studied as one with the two moduli m_1m_0 and m_2 , given that the latter two values are also relatively prime, the same method of extension cannot be applied to CD-RNS.

We leave extension of our results in a general case to future work. Here, we note that when we can choose ε_{\max} or the moduli μ_i such that $m_i = \mu_i / (2\varepsilon_{\max})$ is an integer, and the resulting m_i 's are pairwise relatively prime, then our results are readily extended as in the case of conventional RNS.

THEOREM 5. *The dynamic range of a multimodular CD-RNS whose moduli are pairwise relatively prime integral multiples of $2\varepsilon_{\max}$ is $\mu = \mu_0 \prod_{1 \leq i \leq k-1} [\mu_i / (2\varepsilon_{\max})]$.*

Example 16. Let $\varepsilon_{\max} = 0.1$, and consider a CD-RNS with the moduli $\mu_2 = 3.0$, $\mu_1 = 2.6$ and $\mu_0 = 2.2$. According to Theorem 5, the dynamic range of this CD-RNS is $\mu = 429.0$, given that $m_2 = 3.0/0.2 = 15$, $m_1 = 2.6/0.2 = 13$ and $m_2 = 2.2/0.2 = 11$ are pairwise relatively prime integers.

In the context of biological computing, justifying moduli choices that satisfy the conditions of Theorem 5 may seem difficult. On the other hand, given a particular set of moduli, a slight reduction in the error bound ε_{\max} may bring the system close to satisfying the required conditions. Such an improvement in the accuracy of the residues, which increases the dynamic range of the associated CD-RNS, may give the rat a survival advantage. Thus, it is conceivable that biological systems would converge on such optimal combinations of values in an evolutionary way.

Note that the result of Theorem 5 confirms the numerical simulations of Fiete *et al.* [1], which indicate the range to be exponentially increasing with $1/\varepsilon_{\max}$, with the exponent being slightly less than the number of moduli (they find an exponent of ~ 10.7 with a particular set of 12 moduli (Theorem 5 suggests an exponent of 11, given an ideal set of 12 moduli). In actuality, Fiete *et al.*'s results are based on a relative formulation of accuracy, as opposed to our absolute view (i.e. their error bound is greater for larger moduli); so the relationship of our work to their work is not quite direct.

Our discussion thus far has entailed a second limitation. Two or more residues can be used to represent a single value and thus may convey positional information in one dimension. Animals, on the other hand, live and move in a 3D world. At the very least, two dimensions must be considered in any practical context. Extending our results to multiple dimensions can be achieved in a number of ways. The use of a Cartesian coordinates system is one possible approach. However, the use of a coordinate system specifically attuned to the hex mesh may be more appropriate, given our discussion of neuroscience research in Section 1.

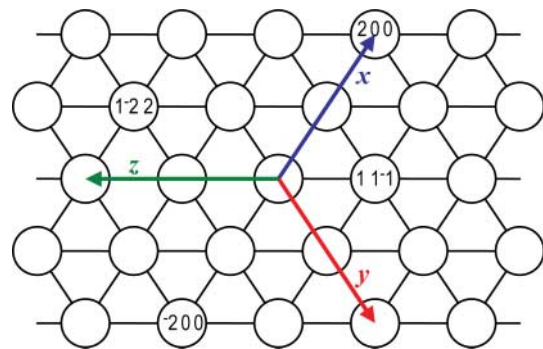


FIGURE 9. Hex-grid coordinate system.

It is well known [17] that cells in a hex mesh can be indexed by using three coordinates, as shown in Fig. 9. The triple of numbers in some of the nodes denote the xyz index of the associated node. This coordinate system is redundant, in the sense that any two of the displacements suffices for determining the third. This form of redundant representation can compensate for inaccuracies in the phase information and may be utilized in addition to the type of redundancy and error resilience that having more moduli can provide [1, 12].

8. CONCLUSION

Our results provide a digital arithmetic perspective on the work of computational neuroscientists who have hypothesized the use of modular number representation by rats and have verified the feasibility of this type of representation, as well as adequacy of the resulting dynamic range, via numerical simulation. We have shown that the aforementioned decade-old hypothesis and its simulation-based verification are supported by theoretical results on CD-RNS that parallel those of conventional RNS representation of integers.

Further research can be undertaken in several different directions. In the domain of digital arithmetic, there is a need to complete and expand the theory presented here by removing some of the restrictions under which certain results were obtained. Methods for selecting the moduli need to be improved and generalized. Expanding our discussions of arithmetic operations, particularly those of forward/reverse conversion, constitute other fruitful areas for further work.

In the domain of biological applications, one may endeavor to determine how close the actual moduli associated with a rat's navigation system are to optimal values that maximize the dynamic range, and whether there is any evidence of evolutionary refinement of the moduli or of the accuracy with which phase information is maintained.

Study of the robustness of CD-RNS representations, including suitable error detection/correction schemes, falls in the interface between the two disciplines. Such properties can be dealt with in isolation or in conjunction with the use of supplemental environmental clues (when available).

ACKNOWLEDGEMENTS

The problem discussed in this paper was first brought to the author's attention by Dr Ila Fiete in April 2006. At that time, we did not make much progress on formulating a solution, and the problem lay dormant for a while. Subsequently, Dr Fiete and co-workers reported on the neurobiological context of this problem in a series of publications, leading to [1] in July 2008. After reading Ref. [1], the author revisited the problem and derived the results presented in [18] and herein.

REFERENCES

- [1] Fiete, I.R., Burak, Y. and Brookings, T. (2008) What grid cells convey about rat location. *J. Neurosci.*, **28**, 6858–6871.
- [2] Hafting, T., Fyhn, M., Molden, S., Moser, M.-B. and Moser, E.I. (2005) Microstructure of a spatial map in the entorhinal cortex. *Nature*, **436**, 801–806.
- [3] O'Flanagan, R.A. and Stevens, C.F. (2005) Neural encoding: the brain's representation of space. *Curr. Biol.*, **15**, R628–R630.
- [4] Omondi, A. and Premkumar, B. (2007) *Residue Number Systems: Theory and Implementation*. Imperial College Press, London.
- [5] Koshy, T. (2002) *Elementary Number Theory with Applications*. Harcourt Publishers, San Diego.
- [6] Soderstrand, M.A., Jenkins, W.K., Jullien, G.A. and Taylor, F.J. (eds) (1986) *Residue Number System Arithmetic*. IEEE Press, Piscataway, NJ.
- [7] Bernocchi, G.L., Cardarilli, G.C., Del Re, A., Nannarelli, A. and Re, M. (2007) Low-power adaptive filter based on RNS components. *Proc. Int. Symposium on Circuits and Systems*, May 27–30, pp. 3211–3214.
- [8] Cardarilli, G.C., Nannarelli, A. and Re, M. (2007) Residue number system for low-power DSP applications. *Proc. 41st Asilomar Conf. Signals, Systems, and Computers*, November 4–7, pp. 1412–1416.
- [9] Jaberipur, G. and Parhami, B. (2009) Unified approach to the design of modulo- $(2^n \pm 1)$ adders based on signed-LSB representation of residues. *Proc. 19th Int. Symposium on Computer Arithmetic*, June 8–10, pp. 57–64.
- [10] Abdallah, M. and Skavantzios, A. (2005) On multi-moduli residue number systems with moduli of forms r^a , $r^b - 1$, $r^c + 1$. *IEEE Trans. Circuits Syst. I*, **52**, 1253–1266.
- [11] Parhami, B. (2010) On equivalences and fair comparisons among residue number systems with special moduli. *Proc. 44th Asilomar Conf. Signals, Systems, and Computers*, November 7–10, pp. 1690–1694.
- [12] Barsi, F. and Maestrini, P. (1973) Error correcting properties of redundant residue numbers systems. *IEEE Trans. Comput.*, **22**, 307–315.
- [13] Parhami, B. (2010) *Computer Arithmetic: Algorithms and Hardware Designs* (2nd edn). Oxford University Press, New York.
- [14] Kim, J.Y., Park, K.H. and Lee, H.S. (1991) Efficient residue-to-binary conversion technique with rounding error compensation. *IEEE Trans. Circuits Syst.*, **38**, 315–317.
- [15] Hung, C.Y. and Parhami, B. (1995) Error analysis of approximate Chinese-remainder-theorem decoding. *IEEE Trans. Comput.*, **44**, 1344–1348.
- [16] Sun, H. and Yao, T.-R. (1994) A neural-like network approach to residue-to-decimal conversion. *Proc. IEEE Int. Conf. Neural Networks*, June 27–29, pp. 3883–3887.
- [17] Nocetti, F.G., Stojmenovic, I. and Zhang, J. (2002) Addressing and routing in hexagonal networks with applications for tracking mobile users and connection rerouting in cellular networks. *IEEE Trans. Parallel Distrib. Syst.*, **13**, 963–971.
- [18] Parhami, B. (2009) Digital/analog arithmetic with continuous residues. *Proc. 43rd Asilomar Conf. Signals, Systems, and Computers*, November 1–4, pp. 1787–1791.

APPENDIX: PROOFS OF THEOREMS

Proof of Theorem 1. Consider the line of slope 1 in Fig. 10, beginning at the origin and continuing until the upper end of the dynamic range has been reached. This line wraps around horizontally and vertically. There is a horizontal wraparound for every μ_0 increase in dynamic range and a vertical one for every μ_1 increase. The separation of lines after the first vertical wrap around is $\sigma_1 = \lfloor \mu_1 / \mu_0 \rfloor \mu_0$, so the maximum possible number of wrap arounds ($\lfloor \mu_1 / \mu_0 \rfloor \lfloor \mu_0 / \sigma_1 \rfloor + 1$ horizontally and $\lfloor \mu_0 / \sigma_1 \rfloor$ vertically) are accommodated without the lines getting too close to each other. The dynamic range is thus $\max(\mu_0(\lfloor \mu_1 / \mu_0 \rfloor \lfloor \mu_0 / \sigma_1 \rfloor + 1), \mu_1 \lfloor \mu_0 / \sigma_1 \rfloor)$. Note that the number of vertical wrap arounds is $\lfloor \mu_0 / \sigma_1 \rfloor$, whereas for horizontal wrap arounds, the number $\lfloor \mu_0 / \sigma_1 \rfloor$ within each span of μ_0 is multiplied by $\lfloor \mu_1 / \mu_0 \rfloor$ and the result incremented by 1 to account for an extra wrap around. Our proof is complete upon noting that $\mu_0(\lfloor \mu_1 / \mu_0 \rfloor \lfloor \mu_0 / \sigma_1 \rfloor + 1) \geq \mu_1 \lfloor \mu_0 / \sigma_1 \rfloor$. This inequality holds because $\mu_1 = h\mu_0 + \sigma_1$ for some $h \geq 1$, $\lfloor \mu_1 / \mu_0 \rfloor = h$ and

$$\begin{aligned} \mu_0(\lfloor \mu_1 / \mu_0 \rfloor \lfloor \mu_0 / \sigma_1 \rfloor + 1) &= \mu_0(h \lfloor \mu_0 / \sigma_1 \rfloor + 1) \\ &= (h\mu_0 + \sigma_1) \lfloor \mu_0 / \sigma_1 \rfloor \\ &\quad + (\mu_0 - \sigma_1 \lfloor \mu_0 / \sigma_1 \rfloor) \\ &= \mu_1 \lfloor \mu_0 / \sigma_1 \rfloor + (\mu_0 - \sigma_1 \lfloor \mu_0 / \sigma_1 \rfloor) \\ &\geq \mu_1 \lfloor \mu_0 / \sigma_1 \rfloor \quad \square \end{aligned}$$

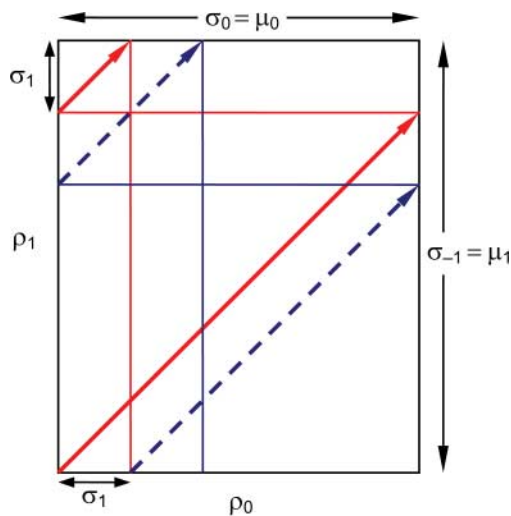


FIGURE 10. Illustrating the wrap arounds encountered in the course of spanning the dynamic range of a CD-RNS.

Proof of Theorem 2. Referring to Fig. 10, the lines of slope 1 starting at the origin (solid, then dashed) will continue shifting to the right, with the ρ_0 intercept moving by σ_1 each time. After $\lfloor \mu_0 / \sigma_1 \rfloor$ repetitions in this first set of slanted lines, the last ρ_0 intercept will be close to the lower right corner of the diagram and at a distance of $\lfloor \mu_0 / \sigma_1 \rfloor \sigma_1 = \lfloor \mu_0 / \sigma_1 \rfloor \mu_0$ from it. The next wrap around will take the line near the lower left corner, being placed above the slanted solid line at the origin and vertically separated by $\lfloor \mu_0 / \sigma_1 \rfloor \mu_0$ from it. The next vertical wrap around occurs at a horizontal distance of $\sigma_1 - \lfloor \mu_0 / \sigma_1 \rfloor \mu_0$ from the origin. Thus, $\sigma_1 = \min(\lfloor \mu_0 / \sigma_1 \rfloor \mu_0, \sigma_1 - \lfloor \mu_0 / \sigma_1 \rfloor \mu_0)$ denotes the smaller distance between the new set of slanted lines and the first set. The factor $\lfloor \sigma_1 / \sigma_2 \rfloor$ is the number of repetitions in the second set, as the lines spaced σ_2 apart fill the spacing between lines of distance σ_1 . The other floored ratios are similarly justified. \square

Proof of Theorem 3. If δ , satisfying $\delta > 2\varepsilon_{\max}$, exactly divides μ_0 and μ_1 , then the sequence of values $\sigma_1, \sigma_2, \dots$, defined in the statement of Theorem 2, will converge to $\sigma_j = \delta$. Thus, we will have at most μ_1 / δ horizontal and μ_0 / δ vertical wrap arounds, before a full wrap around, from the upper right corner of the diagram to the origin, occurs. The two terms $\mu_0 \lfloor \mu_1 / \delta \rfloor$ and $\mu_1 \lfloor \mu_0 / \delta \rfloor$ will be equal in this case and denote the total span of the slanted lines (or the dynamic range) in the ideal case. When $\delta < 2\varepsilon_{\max}$, including when $\delta = 0$, the best-case scenario is to have slanted lines cover the entire $\mu_0 \times \mu_1$ rectangular space, without any wasted space, except possibly at the very end (owing to μ_0 or μ_1 not being an integral multiple of $2\varepsilon_{\max}$). The larger of the two terms $\mu_0 \lfloor \mu_1 / (2\varepsilon_{\max}) \rfloor$ and $\mu_1 \lfloor \mu_0 / (2\varepsilon_{\max}) \rfloor$ is an upper bound for the total length of the slanted lines in this case. \square

Proof of Theorem 4. The proof parallels that of the first part of Theorem 3, with δ replaced by $2\varepsilon_{\max}$. \square

Proof of Theorem 5. We use an inductive proof. As the induction basis, the result holds for $k = 2$ by Theorem 4. Now, assume that the result is valid for $k - 1$ moduli, that is, the dynamic range $\mu^{(k-1)} = \mu_0 \prod_{1 \leq i \leq k-2} \lfloor \mu_i / (2\varepsilon_{\max}) \rfloor$ is attained with the moduli $\mu_0, \mu_1, \dots, \mu_{k-2}$ satisfying the postulated conditions. Upon adding μ_{k-1} to the existing set of moduli, consider the resulting CD-RNS as a two-modulus system, with the moduli $\mu' = \mu_0 \prod_{1 \leq i \leq k-2} \lfloor \mu_i / (2\varepsilon_{\max}) \rfloor$ and μ_{k-1} . It is readily seen that $\mu' / (2\varepsilon_{\max})$ and $\mu_{k-1} / (2\varepsilon_{\max})$ are relatively prime. Thus, applying Theorem 4 one more time, we find that the overall dynamic range is $\mu = \mu^{(k)} = \mu' \lfloor \mu_{k-1} / (2\varepsilon_{\max}) \rfloor = \mu_0 \prod_{1 \leq i \leq k-1} \lfloor \mu_i / (2\varepsilon_{\max}) \rfloor$. \square