PRIME NUMBERS AS POTENTIAL PSEUDO-RANDOM CODE FOR GPS SIGNALS

Números primos para garantir códigos pseudo-aleatórios para sinais de GPS

JÂNIA DUHA

Department of Physics
University of Maryland at College Park
20742-4111 Maryland, USA
jduha@umd.edu

RESUMO

O sistema de codificação GPS baseia-se em certas características importantes de números primos. De acordo com a teoria, números primos muito grandes são difíceis de encontrar, testar e prever. Essas características especiais estão, diretamente, relacionadas a segurança do sistema. Neste trabalho é apresentado um novo modelo para números primos, que evidencia um comportamento extremamente previsível com periodicidades claras, o que aponta para o fato, de que para garantir a segurança dos sistemas de codificação, assim como sua otimização no futuro, torna-se cada vez mais necessário buscar um entendimento maior sobre os padrões de lógica que regulam o comportamento desses números.

ABSTRACT

The GPS encoding system relies on some important features of prime numbers. According to prime numbers theory large primes are difficult to find, to test and to predict. These special features are connected to the security of the system. In this work we present a new model for prime numbers, that points to a very predictable behavior with clear periodicities, what points to the fact that a better knowledge about prime numbers is needed, in order to optimize and guarantee the security of the encoding systems.

1. INTRODUCTION

Pseudo-Random Noise (PRN) signals are a form of carrier modulation and encoding used by GPS satellites. The GPS satellites broadcast the PRN codes mixed with other information like GPS ephemeris, clock-parameters and also parameters concerning the other satellites. By mixing the PRN-code with the 50 Hz data the

total signal is spread out over a broad part of the spectrum. This technique is called spread spectrum.

PRN signals provide vastly superior noise, jamming and interference resistance, from continuous-wave noise sources. In fact, the PRN signals phase-shift-keying has very interesting noise-suppression properties.

The GPS was designed to work with a very low signal power, because, it is, in fact, this property of being under the thermal noise floor (and very similar to pure, unadulterated thermal noise) that makes PRN modulation a good encoding system. As a consequence, it is very hard to distinguish the signal from noise.

GPS encoding system allows satellites to transmit at power levels 30 dB what means that the data rate is vastly reduced to approximately 50-bits-per-second, but the encoding keys can be arbitrarily long and cryptographically secure preventing non-military users from getting the more accurate positions that the system is capable of providing (*Langley*, 1991; 1993).

The codes-pattern used for GPS consists of a long series of bits (0's and 1's). They repeat themselves after the 1023rd bit (*Kaplan*, 1996; *Seeber*, 1993). Of course, there is a regular pattern in the bits that cannot be easily detected. The random string of bits that act as the encryption key is 1023 bits long. The key has nearly equal numbers of ones and zeros in it, and the distribution of pairs of ones, pairs of zeros, etc. falls off rapidly, as it would in a random (white-noise) spectrum. Each GPS satellite has its own unique key; the keys are published and well known and serve to identify the transmitting satellite. Short keys are much easier to detect, but offer less noise-suppression. In fact, the process gain is just the logarithm of the length of the repeating pseudo-random noise sequence. The PRN length determines how much thermal noise can be overcome.

The only interesting PRN sequences are those with N (N = PRN Sequence Length) being a large prime number, as these are the ones that spread out the spectrum the most. By contrast, if N is a product of small primes, then several spectral lines are absent.

However, according to the prime numbers theory, large primes are difficult to find, to test and to predict. And this is only one of several reasons that make the understanding of the rule of prime numbers, on the natural numbers scenario, one of the most interesting tasks today.

How to get the best of PRN codes without understanding prime numbers? New code patterns and better management of the old ones depends on the understanding of these numbers. But, even today, after so many years of huge efforts, prime numbers are seen as a mystery. The history of prime numbers is linked to a broad range of research areas, along the last 2000 years, achieving the goal of remain untouched in its essence. They are known as random numbers for there isn't enough knowledge about these numbers that don’t present any kind of pattern, at least at first sight.

Eratosthenes developed one of the most important works on number theory at nearly two centuries BC. He is remembered for his prime number sieve, which is still an important tool in number theory research. He was a pioneer in geodesy and made the first accurate measurement of the circumference of the Earth. There are several questions about the rule of prime numbers on PRN codes, but one of them is how periodicities on prime numbers will impact the encoding system.

However, to answer this question one has, first, to answer another question: Are there interesting periodicities on prime numbers? To answer this question, let us seek the same path of Eratosthenes, more than 2000 years ago. Let us think about the Eratosthenes sieve, but with a different approach and modeling that will allow us to understand where prime numbers can be found among the natural sequence of numbers; how one can be 100% sure that the output is a prime number; and which are the patterns that rules its behavior?

2. PRIME NUMBERS FROM THE BEGINNING

Natural numbers are a very simple sequence of all possible numbers that you will find if you add 1 to the previous number in the sequence. The first number is one, the second is 1+1=2, the third is 2+1=3, and so on:

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, …

The number one is a construct number. By adding one successively you construct all the sequence. But the second and the third number can provide a good part of the sequence too, as shown on Tables 1 and 2.

Table 1. Natural numbers sequence – construct number: 2

	2		**4**		**6**		**8**		**10**		**12**		**14**		**16**		**18**	
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

Table 2. Natural numbers sequence – construct numbers: 2 and 3

	2		**4**		**6**		**8**		**10**		**12**		**14**		**16**		**18**	
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
		3			**6**			**9**			**12**			**15**			**18**	

But, note that the two and the three provides some numbers simultaneously. We will call these repeated numbers (6, 12, 18, 24, 30…) as the "knots" of the sequence (in shadow) in the Table 3.

Table 3. Natural numbers sequence – knots

	2		**4**		**6**		**8**		**10**		**12**		**14**		**16**		**18**	
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
		3			**6**			**9**			**12**			**15**			**18**	

Note also, that these knots have always an odd number on the left and an odd number on the right side that cannot be provided by the two or the three. Let us call this special class of odd numbers around the knot as "knot-numbers". If you look more carefully you will note that some of these numbers are *simple* (can be divided only by 1 and by their selves) and some are *composite*, or else, they are multiples of other knot-numbers. So, the numbers around the knots can be "simple" (*ks*) or "composite" (*kc*). Note that, the *ks* numbers are also known as prime numbers.

The *kc* numbers are composed by the multiplication of knot-numbers solely, or else: 5x5=25, 5x7=35, 5x11=55, 5x13=65, 5x17=85, etc.; and 7x7=49, 7x11=77, 7x13=91, etc.; and 11x11=121, 11x13=143, etc.; and so on, for all the sequence of knot-numbers.

Table 4 – Composite knot-numbers.

	5	**7**	**11**	**13**	**17**	**19**	**23**	**25**	**29**	***Periodicity***
5	25		55		85		115		145	**30** 10-20
		35		65		95		125		
7		49		91		133		175		**42** 28-14
			77		119		161		203	
11			121		187		253		319	**66** 22-44
				143		209		275		
13				169		247		325		**78** 52-26
					221		299		377	
17					289		391		493	**102** 34-68
						323		425		
19						361		475		**114** 76-38
							437		551	

These composite numbers are very easy to calculate and it is also very easy to predict their behavior through the infinity because they present a clear periodicity as shown in the Table 4. As an example, the first knot number, 5, generates the following sequence of numbers: 25, 25+10=35, 35+20=55, 55+10=65, 65+20=85, and so on. But, this alternating periodicity (10-20) can be reduced to a single period,

10+20=30, when considering two starting numbers (25, 35), instead of one (25). Then, we will have: 25, 25+30=55, 55+30=85, etc. and also, 35, 35+30=65, 65+30=95, etc., as shown on Table 4.

The sequence of knot-numbers with their respective knots is even easier to calculate and to predict. The knots start at 6 and will appear always at intervals of six also: 6, 6+6=12, 12+6=18, 18+6=24, 24+6=30, etc. You can be 100% sure that on the left and on the right side of a knot you will find a *k* number. But if you want a simple *k* number (*ks* = prime number), what you need to do it's only to identify the composite ones and discard them. Note also, that one is never going to find a prime number out of the knot vicinity. All prime numbers are "knot-numbers", but the opposite is not true.

So, now we point to the fact that if the knot and the *kc* numbers have a predictable behavior (with clear periodicities), the *ks* numbers should be predictable, too. Next, we will show the knots (shadow), the *kc* numbers (bolt) and the *ks* numbers (bolt underlined) form 5 to 100 in Table 5.

Table 5. Natural numbers – knots, *kc* and *ks* numbers.

5	6	**7**	8	9	10	**11**	12	**13**	14	15	16
17	18	**19**	20	21	22	**23**	24	**25**	26	27	28
29	30	**31**	32	33	34	**35**	36	**37**	38	39	40
41	42	**43**	44	45	46	**47**	48	**49**	50	51	52
53	54	**55**	56	57	58	**59**	60	**61**	62	63	64
65	66	**67**	68	69	70	**71**	72	**73**	74	75	76
77	78	**79**	80	81	82	**83**	84	**85**	86	87	88
89	90	**91**	92	93	94	**95**	96	**97**	98	99	100

To obtain all possible prime numbers (*ks*) from one to a number as big as one can possible image, you have only to follow the simple rules described above. One by one, the prime numbers will appear with 100% of accuracy without the need of any proof of primarility. Knot-numbers, as defined above in this work are

$$k_1 = 5\ ,\ k_2 = 7\ ,\ k_3 = 11\ ,\ k_4 = 13\ ,\ k_5 = 17\ ,\ k_6 = 19,\ k_7 = 23,\ k_8 = 25,\ \infty,$$

and, they can be obtained through the following simple relation: $k = 6i \pm 1$, with i = 1, 2, 3, 4.... , or else $k_{2i} = 6i + 1$, on the right side, and $k_{2i-1} = 6i - 1$, on the left side.

All possible composite knot-numbers are obtained through multiplying all possible *k* numbers $kc_{1n} = k_1 \,.\, k_n$, with n = 1, 2, 3, ...; $kc_{2n} = k_2 \,.\, k_n$, with n = 2, 3, 4, ...; $kc_{3n} = k_3 \,.\, k_n$, with n = 3, 4, 5, ...; and so on.

Prime numbers are all the *k* numbers, as defined above, except the composite ones. In order to calculate all possible *ks* numbers from 5 to 3,789,547, or any number as big as one can imagine, one by one, with 100% of accuracy and no need of any

proof of primarility, first, we compute the *k* numbers, second, we compute de *kc* numbers and subtract them from the set of original *k* numbers. This will work well for big primes as well as for the smaller ones.

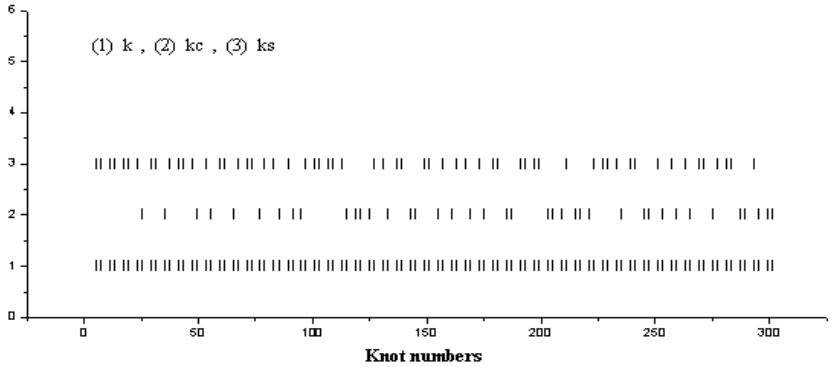
3. PRIME NUMBERS PATTERNS

The pattern of the knot-numbers is very simple as shown in the Figure 1, where the type (1) (first set from the bottom) are the *k* numbers. The type (2) are the *kc* numbers that must be discarded, and the type (3) are the *ks* numbers, or prime numbers (on the top). The periodic behavior of the *ks* numbers is not so easy to see, because every time you add the contribution of a new *kc* to the sequence, the period change and greatly increases.

This is why prime numbers looks like random ones. But there is a pattern there, and you will be able to work with it. While the period of *k* numbers is 6 for all intervals you consider, the period of the *kc* numbers depends on the maximum value you are considering (in this case, 300).

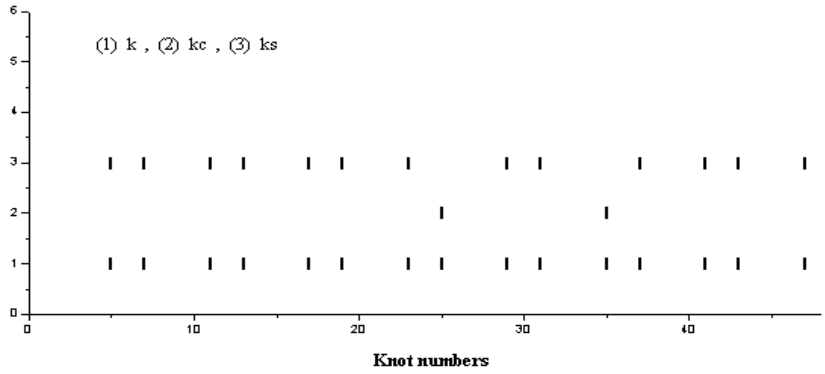
At this point two important questions rise up: How these patterns will change the scenario of the encoding systems? And how to use these periodicities to achieve better codes?

Figure 1 Prime numbers (*ks*) distribution from 5 to 300.



To understand the periodicity of *kc* numbers is important because if you have this information you can understand and predict the periodicity of prime numbers (*ks*), too. Let us begin with small numbers and analyze the distribution from 5 to 48, for example. This distribution has only one number contributing to the set of *kc* numbers that must be discarded: the 5, because the first contribution of the 7 is 7x7=49, the 11 is 11x11=121, and so on. So we have the following numbers that are composite: 5x5=25 and 5x7=35 as shown in the Figure 2.

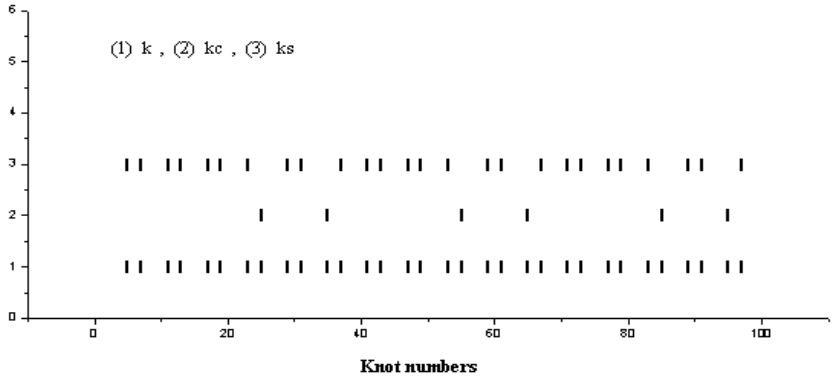
Figure 2 Prime numbers distribution from 5 to 48.



In the Figure 1 and 2 the type (3) are prime numbers with 100% accuracy. And, if you take into account all the numbers that generates the set of *kc* numbers that appears up to the number you want achieve, e.g., 300 in Figure 1 and 48 in Figure 2, you will always have 100% of accuracy. However, when talking about large numbers it may be reasonable to take into account only the first *kc* primes in order to maintain the periodicity easy to control. Because, the more numbers you take into account to construct the set of *kc* numbers, bigger will be the period that will rule the *ks* behavior. Lets say that you don't want too much work so, to predict the position of *ks* numbers from 5 to 100 you, will take into account only the five: 5x5=25, 5x7=35, 5x11=55, 5x13=65, 5x17=85 and 5x19=95. While you should take into account also the contribution of the seven: 7x7=49, 7x11=77, 7x13=91.

In this case, we don't have 100% of accuracy, but we can find a pattern for the prime numbers with period $T_{ks} = 30$ ($T_k = 6$, $T_{kc} = 30$).

Figure 3 Prime numbers distribution from 5 to 100.



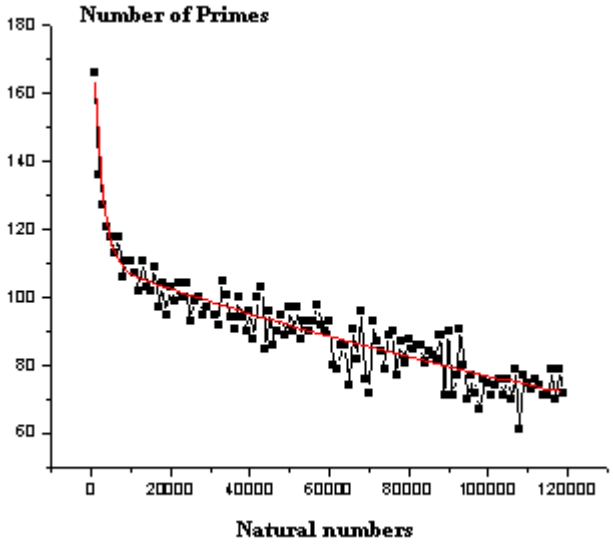
This approximate approach will work even better for big numbers, because the great majority of composite knot-numbers yields from the smallest numbers of the sequence. The probability of finding the contribution of a big *kc* number in the pattern decreases dramatically when moving towards big numbers. This means that the pattern provided by a reasonable set of *kc* numbers will provide a good periodic pattern that will point to the position of the prime numbers, without any need of further calculations.

4. DISCUSSION

One of the main conclusions of this work is that there is a pattern for prime numbers. But, it's a non-steady pattern that changes and increases when a new *kc* number joins the group. In order to obtain a steady pattern to be used as a tool for understand and find prime numbers, one has to stop to consider the *kc* numbers at some point of the distribution. This will fix the "movie" at one "frame", and will allow us to see the pattern in it. But, where is a good point to stop the process? The answer, of course, depends, on how much accuracy you need and how big are the prime numbers you are working with. PRN codes works with numbers 1023 bits long. At this range it's possible to work with prime number patterns, and to improve the encoding system.

Figure 4 shows how the number of primes (per thousand) decreases for every thousand you consider. All odd numbers around the knots are possible primes numbers. This means that we have 333 possible primes for every thousand. But when you start to cut of the composite ones this number decreases exponentially. The first thousand (5 to 1005) has 166 primes; the second thousand (1005 to 2005) has 136 primes, and so on.

Figure 4 Number of Primes - *ks*



All odd numbers around the knots are possible primes numbers. This means that we have 333 possible primes for every thousand. But when you start to cut of the composite ones this number decreases exponentially. The first thousand (5 to 1005) has 166 primes; the second thousand (1005 to 2005) has 136 primes, and so on. After 120 thousands (120000) there are, only, 70 possible primes to be considered (20% of the initial number). And if you let your computer work a little more, this number will decrease even more.

It's important to remark that, even if, one chooses to work with very large numbers, such as, numbers between, 2×10^{1234567} and 2×10^{2234567} , for example, there is no need to calculate all *kc* numbers before that. It's possible to achieve very accurate results considering only the contribution of a set of *kc* big enough to minimize the number of "possible" *ks* (prime numbers) numbers. If you stop to consider the *kc* numbers at some point, this will provide you a pattern for the primes, that will repeat it self through the infinity with periodicity *T*, and that will point to the few numbers that are able to be prime numbers.

Finally, we conclude that, the prime number patterns should be taken into account when working with encoding systems, because these patterns are very predictable and cannot be ignored by a system where the security is a major concern.

REFERENCES

KAPLAN, E. **Understanding GPS: Principles & Applications**. Artech House Publishers. Boston - London: 1996, 554pp.

LANGLEY, R.B. **The GPS observables**. GPS World. 4(4), 52-59 , 1993.

LANGLEY, R.B. **The mathematics of GPS**. GPS World. 2(7), 45-50, 1991.

SEEBER, G. **Satellite Geodesy: Foundations, Methods & Applications**. Walter de Gruyter. Berlin - New York: 1993, 531pp.

(Recebido em maio/04. Aceito em novembro/04.)