Chapter 7

Exploitation of Chaotic and Synchronization Properties of Logistic Maps for Application in Wireless Communication


7.1. Introduction:

Fading is a significant aspect in wireless communications and occurs due to multiple reflections, refraction, scattering and diffraction of the transmitted signal while propagating through the medium [100]. Fading generates rapid variations in the magnitude and phase of the original signal. Also, channel and background noise corrupts the signal considerably. To ensure proper quality of service (QoS) of any system while executing a communication process through such a medium needs proper modeling, understanding and design of appropriate mechanisms for such a situation [102].

Spread spectrum modulation (SSM) is an important class of techniques used extensively in wireless systems for obtaining proper QoS. The primary advantage of SSM is the fact that it uses a noise like rapidly fluctuating signal to spread the bandwidth of the signal much more than the normal requirement during transmission and just the reverse process during recovery. As a result, signal components are spread throughout the channel and specific portions of the signal cannot be corrupted by propagation related variations and noise. Normally, there
are two major techniques by which spreading the spectrum is achieved; one is the frequency hopping (FH) technique, which makes the narrow band signal jump in random restricted bands within a larger bandwidth and the other one is the direct sequence (DS) method which introduces rapid phase transitions to the data to make it larger in bandwidth. The spreading factor is obtained by using a pseudo-noise (PN) generator. In some cases Gold codes are also used. One of the critical disadvantages of PN and Gold code generators are their limited sequence lengths. It is directly dependent on the physical size of the generator. But practical communication conditions may demand varying or increasing sequence lengths. It has been observed that increasing sequence lengths in wireless communication serves as an effective aid to fight propagation related variations and contributes towards better QoS [94].

The chaotic behavior observed in specific operations region of the logistic map has been effectively used for such a purpose [89, 94]. The logistic map provides non-linear recursion which generates iterative values with a definite exhibition of deterministic chaos [80]. Another property of the logistic map namely synchronization can be effectively used in wireless communication to provide better QoS while used as part of SSM systems. In this work, we use the chaotic behaviour and the synchronization property of the logistic map as an aid to SSM in a wireless setup. Chaotic sequences of varied lengths are generated using logistic map for use as spreading factor of a spread spectrum modulation system. Next, the synchronization property of the logistic map is used to provide proper recovery of a sequence of bits in the identified modulation system. Experimental results show that the proposed method is effective in wireless channels and can provide proper QoS during recovery of signals.
The rest of the description is organized as follows. Section 7.2 gives a description of the theoretical considerations involved, Section 7.3 shows the experimental details and results obtained and Section 7.4 concludes.

### 7.2. Logistic Map as Chaos Generator:

A logistic map is a polynomial mapping having a degree of 2. It gives the idea of how a very complex, chaotic behaviour can occur from very simple non-linear dynamical equations. Prediction thus becomes impossible, and then the system behaves randomly [83, 95]. Mathematically, the logistic map is written as

\[
x_{n+1} = \mu \times x_n (1 - x_n)
\]

(1)

where \(x_n\) lies between 0 and 1.

The map behaviour is dependent on \(\mu\), which is clearly seen in the Table 7.1.

**Table 7.1. Behaviour dependent on \(\mu\)**

<table>
<thead>
<tr>
<th>Range of (\mu)</th>
<th>Behaviour of population</th>
</tr>
</thead>
<tbody>
<tr>
<td>Between 0 and 1</td>
<td>Independent of the initial population</td>
</tr>
<tr>
<td>Between 1 and 2</td>
<td>Independent of the initial population</td>
</tr>
<tr>
<td>Between 2 and 3</td>
<td>Fluctuate around the value (\mu - \frac{1}{\mu}) for some time</td>
</tr>
<tr>
<td>Greater than 3</td>
<td>Dependent of the initial population</td>
</tr>
</tbody>
</table>

Maps arise in various ways:

- As tools for analysing differential equations (e.g. Poincare and Lorenz)
- As models of natural phenomena (in economics and finance)
- As simple examples of chaos
Maps are capable of much wilder behaviour than differential equations because the points $x_n$ hop discontinuously along their orbits rather than flow continuously.

**For fixed point:**

An equation $x_{n+1} = f(x_n)$ is considered, where $f(.)$ is a smooth function from the real line onto itself.

Suppose $x^*$ satisfies $f(x^*) = x^* \Rightarrow x^*$ is a fixed point of the map. Its stability is determined by considering a nearby orbit $x_n = x^* + \eta_n$. Thus

$$x^* + \eta_{n+1} = x_{n+1} = f(x^*) + f'(x^*)\eta_n + O(\eta_n^2)$$

Since $f(x^*) = x^*$,

$$\eta_{n+1} = f'(x^*)\eta_n$$

is the linearized map and $\lambda = f'(x^*)$ is the eigen value or multiplier.

- If $|\lambda| = |f'(x^*)| < 1$ then $\eta_n \to 0$ as $n \to \infty \Rightarrow x^*$ is linearly stable.
- If $|\lambda| = |f'(x^*)| > 1$ then $x^*$ becomes unstable.
- If $|\lambda| = |f'(x^*)| = 1$ then the terms $O(\eta_n^2)$ have to be considered.

Similarly, if we consider $x_{n+1} = x_n^2$, now for fixed points at $x^* = (x^*)^2$

$$\Rightarrow x^* = 0, 1.$$ 

$\lambda = f(x^*) = 2(x^*) \Rightarrow x^* = 0$ is stable and $x^* = 1$ is unstable. Cobwebs allow us to see global behavior at a glance.

Considering the equation of logistic map, let $0 \leq \mu \leq 4$, $0 \leq x \leq 1 \Rightarrow$ map is a parabola with maximum value of $\mu/4$ at $x = 0.5$.

- For $r < 1$, $x_n \to 0$ as $n \to \infty$
• For $1 < \mu < 3$, $x_n$ grows as $n$ increases, reaching a non-zero steady state.

• For larger $\mu$ (e.g. $\mu = 3.3$) $x_n$ eventually oscillates about the former steady state $\Rightarrow$ period 2 cycle.

• At still larger $\mu$ (e.g. $\mu = 3.5$), $x_n$ approaches a cycle which repeats every 4 generations $\Rightarrow$ period 4 cycle.

• Further period doublings to cycles of period 8, 16, 32 etc occur when $\mu$ increases. Hence, $\mu_1 = 3$ (period 2 is generated); $\mu_2 = 3.449$ (period 4 is created); $\mu_3 = 3.54409$ (period 8 is obtained); $\mu_4 = 3.5644$ (period 16 is found); ...; $\mu_\infty = 3.569946$ (period $2^\infty$ is born);

• Successive bifurcations become faster and faster as $\mu$ increases.

• The $\mu_n$ converge to a limiting value $\mu_\infty$.

• For large $n$, the distance between successive transitions shrinks by a constant factor

$$\delta = \lim_{n \to \infty} \frac{\mu_{n+1} - \mu_n}{\mu_{n+1} - \mu_n} = 4.669 \ldots (3)$$

When $\mu > \mu_\infty$, for many values of $\mu$, the sequence $\{x_n\}$ never settles down to a fixed point or a periodic orbit, the long term behavior is aperiodic. It is expected that the system would become more and more chaotic as $r$ increases, but in fact the dynamics are more subtle.

At $\mu = 3.4$ the attractor is a period 2 cycle. As $\mu$ increases, both branches split, giving a period 4 cycle i.e. a period-doubling bifurcation has occurred. A cascade of further period-doublings occurs as $r$ increases, until at $\mu = \mu_\infty \approx 3.57$, the map becomes chaotic and the attractor changes from a finite to an infinite set of points.

When $\mu > \mu_\infty$, the orbit reveals a mixture of order and chaos, with periodic
windows interspersed with chaotic clouds of dots. When, $\mu \approx 3.83$, then there is a stable period 3 cycle.

Now, considering the logistic map equation $x_{n+1} = \mu x_n (1 - x_n); 0 \leq \mu \leq 4,$ and $0 \leq x_n \leq 1$

For fixed point, $x^* = f(x^*) = \mu x^* (1 - x^*)$

$\Rightarrow x^* = 0$ or $1 - 1/\mu$, where

$x^* = 0$ is a fixed for all $\mu$, and $x^* = 1 - 1/\mu$ only if $\mu \geq 1$; (since $0 \leq x_n \leq 1$)

Stability depends on $f'(x^*) = \mu - 2\mu x^*$

$x^* = 0$ is stable for $\mu < 1$ and unstable for $\mu > 1$

$x^* = 1 - 1/\mu$ is stable for $-1 < (2 - \mu) < 1$, i.e. $1 < \mu < 3$ and unstable for $\mu > 3$;

Therefore, at $\mu = 1, x^*$ bifurcates from the origin in a transcritical bifurcation. As $\mu$ increases beyond 1, the slope at $x^*$ gets increasingly steep. The critical slope $f'(x^*) = -1$ is attained when $\mu = 3$, the resulting bifurcation is called a flip bifurcation that means two cycle.

Now, logistic map has a 2-cycle for all $\mu > 3$, which is clear from the following description:

A 2-cycle exists if and only if there are two points $p$ and $q$ such that $f(p) = q$ and $f(q) = p$. Equivalently, such a $p$ must satisfy $f(f(p)) = p$ where $f(x) = \mu x (1 - x)$.

Hence, $p$ is a fixed point of the second iterate map $f^2(x) = f(f(x))$. Since $f(x)$ is a quadratic map and $f^2(x)$ is a bi-quadratic polynomial.

Now, to solve $f^4(x) = x$, $x^* = 0$ and $x^* = 1 - 1/\mu$ are trivial solutions. The other two solutions are
\[ p, q = \frac{\mu + 1 \pm \sqrt{1 - 3 (\mu - 1) \mu}}{2 \mu} \quad (4) \]

Which are real for \( \mu > 3 \). Hence a 2 cycle exists for all \( \mu > 3 \) for the logistic map \[4\].

### 7.3 Synchronization in Logistic Map:

In logistic maps, synchronization is related to coupled maps. Chaotic behaviour of logistic maps is considered as random oscillation generations. In a coupled system, if amplitude, frequency and phase of two logistic maps share compatible attributes, these are called to be synchronized maps [83, 95]. Two logistic maps can be shown to be synchronized as below:

\[ y_{n+1} = x_n \quad (5) \]

\[ x_{n+1} = y_n \quad (6) \]

Replacing the above with respective logistic map gives

\[ y_n = \mu x_n (1 - x_n) \quad (7) \]

\[ x_n = \mu q (1 - q) \quad (8) \]

where

\[ q = ax_n + (1 - a)y_n \quad (9). \]

Taking \( \mu = 4 \) for \( x \) and \( y \) gives three solutions which further yields \( a = \frac{1}{2} \) and

\[ a = \frac{5}{4} \]

with the later discarded for realistic cases [80, 92]. For a symmetrically coupled case \( a = \frac{1}{4} \).
7.4. Proposed Method of using Varying Length Chaotic Sequences and Synchronised Logistic Map as part of DS-SSM:

Here, we discuss the generation of binary chaos codes of varying length for use in a DS-SSM system in certain wireless communication setup and show how synchronization can be achieved to recover transmitted data bits.

A. Binary spreading sequence generation using logistic map

Two methods are adopted to generate the binary spreading sequence using logistic chaotic map. These are

- Binary sequence generation using thresholding method.
- Binary sequence generation using floating point to bit conversion method.

Binary sequence is generated using logistic map as shown in the Fig. 7.1. Four tests namely mono bit test, run length test, correlation property and computational time are checked while ascertaining the validity of the generated bits.

![Flow diagram of both thresholding method and integer to bit conversion method](image)

*Figure 7.1: Flow diagram of both thresholding method and integer to bit conversion method*
Table 7.2. Sequences generated by logistic map for $\mu=3.61$, 3.65 and 3.69

<table>
<thead>
<tr>
<th>Value of $\mu$</th>
<th>Binary sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.61</td>
<td>00111000011010010100</td>
</tr>
<tr>
<td>3.65</td>
<td>01001001010010010101</td>
</tr>
<tr>
<td>3.69</td>
<td>01101001001110000110</td>
</tr>
</tbody>
</table>

Figure 7.2: System model. The generated sequences for three different values of $\mu$ are shown in the Table 7.2.

B. System Model of using varying length chaos code as part of a DS-SSM

Certain experiments are performed as per the logical flow outlined in Figure 7.2. For fixed sized binary data blocks generated by following the process logic outlined in Figure 7.1, chaotic codes of varying length are generated using a logistic map generator. The basic system is a direct sequence SSM (DS-SSM). In traditional cases, PN sequences and Gold codes are used as spreading/ de-spreading factors. Here, the chaotic code acts as a spreading factor in the transmitter and is considered for de-spreading in the receiver. Depending upon the fading observed in the channel, chaotic sequences of varying lengths are generated. Fading increases with rise in speed of transmitters and receivers and/or either one or both. So depending upon the channel condition, the logistic map generator produces chaos codes of varying length. In case of traditional coding techniques
like PN sequence or Gold code generators, the size of the spreading factor is dependent on the physical length of the register used for the purpose which is not the case with the chaos code. As varying sequence lengths of the spreading factor provides better QoS in a SSM [86], hence certain experiments are performed to check the suitability of the chaos code in faded channels generated due to varying speeds of transmitter, receiver and/or either one or both of the two.

Inside the DS-SSM, data blocks are passed through binary phase shift keying (BPSK) where the chaotic code provides spectrum spreading and de-spreading. Unlike PN sequence and Gold code, the chaotic sequence length variation is achieved by changing the iteration count within the logistic map generator. This is a major advantage which removes the dependence on the physical devices. For a given data block, as per the instantaneous channel state, a unique chaotic code is generated during transmission which is also made available at the receiver side during recovery of the content.

![Figure 7.3: Coupled logistic map chaotic code generator for DS-SSM](image-url)
C. Achieving synchronization between coupled logistic maps for data recovery in a DS-SSM.

Another set of experiments are performed as per the logical flow shown in Figure 7.3. Two coupled logistic maps are considered to execute the spread factor generation for bits 1 and 0 as part of a DS-SSM. Data coming from a source are bifurcated into streams of 1 and 0 bits after passing through a bit selector block. Two coupled logistic maps are used to provide the spreading factors for generated DS-SSM-ed forms for both the streams. A coupler combines the output of both the DS-SSM blocks as per a clocking cycle used during bit separation. The coupled output is passed through the faded channel.

At the receiver side, two logistic maps attempt to achieve synchronization for recovery of the bits 1 and 0. The received signal is given as

\[ y_{m,n}(x, \theta) = f_1(x(n), c(n)).S_n * H_{m,n}(x, \theta) + N \]  

where,

- \( f_1(x(n), c(n)) \) is a modulation process with spreading factor \( c(n) \) and bit sequence \( x(n) \),
- \( H_{m,n}(x, \theta) = Rayleigh,Rician_{m,n} \) is the channel matrix for \( x \) sample value and \( \theta \) phase,
- \( S_n = Step(l_n) \) with \( l_{n+1} = \mu * l_n (1 - l_n) \) is a binary chaos sequence obtained with \( \mu = 3.582 \).

Two different logistic maps try to attain synchronization using the received signal \( y \), and a feedback as per the following arrangement:
\( x(n + 1) = f\{(1 - \epsilon)x(n) + \epsilon y(n)\} \quad (11) \)

with

\[ \epsilon \in \left[ \frac{1+a}{2}, \frac{3-a}{2} \right] \quad (12) \]

Synchronization is achieved when

\[ e(n) = |x(n) - y(n)| \quad (13) \]

is minimized.

**Table 7.3. Simulation parameters**

<table>
<thead>
<tr>
<th>Sl. No.</th>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Modulation type</td>
<td>DS-SSM with BPSK</td>
</tr>
<tr>
<td>2</td>
<td>Data block size</td>
<td>Between 1000 to 10,000</td>
</tr>
<tr>
<td>3</td>
<td>Coders</td>
<td>PN Sequence, Gold code, Logistic Map,</td>
</tr>
<tr>
<td>4</td>
<td>Channel types</td>
<td>Rayleigh, Rician</td>
</tr>
<tr>
<td>5</td>
<td>No. of trials per sequence length</td>
<td>at least 10</td>
</tr>
</tbody>
</table>
Table 7.4. BER v/s SNR plot of varying sequence lengths of PN, Gold and Chaos codes

<table>
<thead>
<tr>
<th>Sl no</th>
<th>Sequence Length</th>
<th>SNR in dB</th>
<th>PN Sequence</th>
<th>Gold Code</th>
<th>Logistic Map</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>0</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>0.07</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>0.06</td>
<td>0.04</td>
<td>0.034</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>0.05</td>
<td>0.03</td>
<td>0.03</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8</td>
<td>0.04</td>
<td>0.018</td>
<td>0.015</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td>0.01</td>
<td>0.008</td>
<td>0.007</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>0</td>
<td>0.091</td>
<td>0.085</td>
<td>0.083</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>0.0637</td>
<td>0.0425</td>
<td>0.0415</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>0.0546</td>
<td>0.034</td>
<td>0.02822</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>0.0455</td>
<td>0.0255</td>
<td>0.0249</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8</td>
<td>0.0364</td>
<td>0.0153</td>
<td>0.01245</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td>0.0091</td>
<td>0.0068</td>
<td>0.00581</td>
</tr>
<tr>
<td>3</td>
<td>16</td>
<td>0</td>
<td>0.078</td>
<td>0.0752</td>
<td>0.059926</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>0.0546</td>
<td>0.0376</td>
<td>0.030005</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>0.0468</td>
<td>0.03024</td>
<td>0.020347</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>0.039</td>
<td>0.02259</td>
<td>0.017928</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8</td>
<td>0.0312</td>
<td>0.013536</td>
<td>0.008964</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td>0.0078</td>
<td>0.006016</td>
<td>0.004183</td>
</tr>
<tr>
<td>4</td>
<td>32</td>
<td>0</td>
<td>0.065</td>
<td>0.0623</td>
<td>0.061</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>0.04585</td>
<td>0.0311</td>
<td>0.0305</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>0.03924</td>
<td>0.02492</td>
<td>0.020808</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>0.0326</td>
<td>0.0186</td>
<td>0.01839</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8</td>
<td>0.02608</td>
<td>0.01116</td>
<td>0.0093</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
<td>0.0063</td>
<td>0.00488</td>
<td>0.00434</td>
</tr>
</tbody>
</table>

7.5. Experimental Details and Results:

Extensive experiments have been performed to ascertain the performance of the proposed approaches under different wireless communication conditions. Simulation parameters are summarized in Table 7.3.
For the DS-SSM system, experiments are performed using PN, Gold code and chaos code sequences. Rayleigh channel with Doppler shift due to vehicular movement between 10 to 100 Kmph have been considered. Sequence lengths of 4, 8, 16 and 32 are considered for a range of channel conditions including vehicular and pedestrian. A range of 0 to 10 dB signal to noise ratio (SNR) is considered. The effect of sequence length variation on the QoS of the system is noted down in terms of bit error rate (BER) v/s SNR. An average of ten trials of reading under fading conditions is shown in Table 4. Compared to the PN sequence, for sequence length 4, the chaos code shows a -3dB gain at 10 dB SNR. Similarly, for sequence length 8, the gain at 8dB is 9.3dB, for sequence length 16 the gain is 10.8dB and for 32 the corresponding SNR gain is 8.95dB which is significant. It indicates increase in QoS is established by falling values of BER with rise in SNR due to the use of chaos codes. One noticeable advantage is that for PN and Gold codes, the sequence lengths require registers of specific size unlike that in case of the chaos code. Here, with rise in the number of iterations, the sequence length increases. It leads to the saving of the hardware layout and power.

In case of the coupled logistic map case, the recovery of the data involves iterative processing. It requires a few cycles to establish the synchronization between the received signal and the expected output as described in Section 7.3 C. Figure 7.4 shows a plot between error and iteration obtained during synchronization of the two logistic maps used for transmission and recovery of 1 and 0 bits respectively while the fading is less severe. .
The error plot shown in Figure 7.4 is obtained using a chaos code of length 32 which adds to the computational complexity of the system.

**Table 5:** Average computational time due to varying sequence lengths

<table>
<thead>
<tr>
<th>Sl no</th>
<th>SequenceLength</th>
<th>Time (sec.)</th>
<th>PN Sequence</th>
<th>Gold Code</th>
<th>Logistic Map</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>1.35</td>
<td>1.42</td>
<td>1.4</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>2.13</td>
<td>2.3</td>
<td>3.1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>16</td>
<td>2.52</td>
<td>2.55</td>
<td>3.6</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>32</td>
<td>3.1</td>
<td>3.3</td>
<td>4.4</td>
<td></td>
</tr>
</tbody>
</table>
Table 7.5 shows the average computational complexity of PN, Gold and chaos codes under fading conditions due to varying sequence lengths. It indicates that the chaos code is computationally expansive.

Another set of results are shown in Figure 7.5 which is obtained while establishing synchronization in vehicular and pedestrian conditions of the channel. There are distinct fluctuations which are due to the variations and dynamic behavior of the channel. Increasing the sequence length however reduces the number of iterations required to minimize the error.

7.6. Conclusion:

The nonlinear nature of a frequency selective block fading Rayleigh channel is explored in this chapter and an adaptive NARMA equalizer is proposed for equalizing the channel. Experiments are performed for a lower fading scenario with lower channel taps and a severe fading scenario with higher channel taps to
justify the feasibility of the proposed system. Performance evaluation is done from which it can be concluded that the proposed adaptive NARMA equalizer provides better performance in severe fading scenarios. The proposed model is verified to give good results when compared to the conventional schemes.