カオスを用いたDSP秘話通信システムの試作

西守　克己・久保田裕之・石原　永伯・加藤　義人

電気電子工学科

(1996年9月1日受理)

Digital Signal Processor Application to Secure Communication Using Chaotic Systems

by

Katsumi NISHIMORI, Hiroyuki KUBOTA, Naganori ISHIHARA, Yoshito KATO

Department of Electrical and Electronic Engineering

(Received September 1, 1996)

A circuit implementation of the chaotic system is described. The chaotic behavior of the circuit is simulated by the numerical experiments. The concept of synchronized chaotic systems is applied to the modulation and demodulation techniques for secure communication using the two digital signal processors (DSP). Chaotic synchronization is a nonlinear phenomenon between the two DSP subsystems for the communication with same values at each time as discrete-time nonlinear systems. Each subsystem is tested in the terms of synchronization, modulation and demodulation using the DSP devices. The chaotic digital circuit implementation can be established by the experimental result that an informational signal is transmitted between the simple chaotic systems.

Key words : chaotic systems, circuit implementation, synchronization, secure communication

1．はじめに

カオスといえば，デタラメとか，混沌とかという言葉が連想されるが，ここでは，その工学的応用を目指しているので，少なくとも数量的に表現できうるものでなければならない。そのようなカオス現象とは，決定論的な非線形方程式で表され，近い未来は予測可能であるが，遠い未来は予測不可能な現象である。そのスペクトルにはあらゆる周波数を含んでいるため，カオス的に搬送波を変調すれば通信の当事者以外に対して秘匿性が高くなり，秘話通信に応用できる可能性[1]がある。本報告では，実際に秘話通信システムをＤＳＰ（ディジタル信号処理）ボードを使って構築したので報告する。

2．カオスを用いた秘話通信の原理

2．1．秘話通信とは

秘話通信とは第三者に情報を漏らすことなく通信を行うことである。つまり、第三者が見て簡単に解るようなものは秘話通信ではない。また、正規の受信者にもわからないようなものも意味が無い。よって、ある規則を持ちながら第三者にはなかなか解からないものでなければならない。ここでなかなか解らないというのは、情報の持つ意味が失われるほど長い時間が経過していれば、第三者に解読されてもかまわないということである。

2．2．システムの概要

秘話通信をするにあたって何が必要かを考える。まず、第三者になかなか解らない暗号化をする部分とそれを復元する部分が必要である。また、システムを運用していく上で必要な同期化をする部分や制御をする部分が必要である。また、通信をするための部分が必要である。1つは人間もしくはコンピューター等とのインターフェースであり、通信機となるものに必要なデータを渡したり、送られてきたデータを受信側に伝える部分である。例えば、人間が相手であり、音声を伝達する情報とした場合、先ず、人間の声をひらって電気信号に変換する部分（マイク）が必要である。その次に、この信号を処理するためにディジタル化するA/Dコンバーターが必要である。当然、この逆を行うものも必要になるので、D/Aコンバーターやスピーカーなども必要である。2つめは、当事者双方が通信をするための伝送路の確保が必要である。伝送路が有線である場合は媒体となる線が必要であり、無線の場合は信号を飛ばしたりひらったりするアンテナなどの部分が必要となる。次に、演算を行う部分から出力された信号を伝送路に流す部分が必要である。例えば、シリアルで通信する場合は通信路としてのシリアルポートが必要である。

2．3．実際のシステムの数理

実際のシステムを組むにあたって、前述のことを踏まえた数式を用意する。基本的にはカオスを生成して、これに伝送したい信号に掛け算，足し算するという方法である。以下にその数式を示す。

同期化部

$$X_i(k+1)=1.4-X_i^2(k)+Y_i(k)+U_i(k) \quad (1)$$

$$Y_i(k+1)=0.3X_i(k) \quad \text{ただし} i=1,2 \quad (2)$$

変調部

$$W_i(k+1)=|9Y_i(k)W_i(k)|1-W_i(k)||+|4W_i(k)|1-W_i(k)||+0.1| \times[0.1|Y_i(k)|+S_i(k+1)] \quad (3)$$

復調部

$$T_i(k+1)=[V_i(k)-|9Y_i(k-1)V_i(k-1)|1-V_i(k-1)|||/[|4V_i(k-1)|1-V_i(k-1)||+0.1]-0.1|Y_i(k-1)| \quad (4)$$

情報信号を１（送信側）から２（受信側）へ送るときの制御信号

$$U_1(k)=X_i^2(k)-0.5Y_i(k)-C_{21}(k),\quad U_2(k)=0,\quad V_1(k)=0,\quad V_2(k)=C_{12}(k) \quad (5)$$

１から２および２から１へのそれぞれの伝達信号

$$C_{12}(k)=W_1(k),\quad C_{21}(k)=X_2^2(k)-0.5Y_2(k) \quad (6)$$

上の数式はインターフェースなどは含んでおらず、純粋に発信用変調（スクランブル）をかけて復調（デコード）するということを数式的に計算しているだけであるので、これを利用してシステムを作るには電子回路的なハードウェアの構成が必要である。次に，これについて述べる。

3．カオスを用いた秘話通信システムの構築

3．1．工学的実現のためのハードウェア

作ろうとしている秘話通信システムは、信号を処理する媒体が必要になる。本研究では、一般に信号処理のハードウェアとして広く使用されている、TEXAS INSTRUMENTS社のＤＳＰスタータキット（DSK：TMS320C5x）を使用した。先に述べたように、信号を単に演算によって処理するのみでは秘話通信システムは構成できないので、ＤＳＰ、Ａ／Ｄ，Ｄ／Ａコンバータ

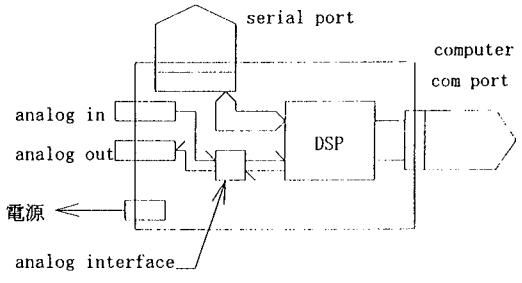

図1　ＤＳＰの概略

ー、Ｉ／Ｏ、電源をワンボードに搭載したＤＳＰボードを２枚使い、秘話通信システムの構築を試みた。

３．２．ハードウェアの概略

図1は、使用するＤＳＰの概略を示す。ＤＳＰボードに与えるプログラムや動作の指令はコンピューターのｃｏｍポートを通じて送られ，外部との入出力はアナログの場合は基板上のANALOG INTERFACEを介して、データをやり取りする。他のＤＳＰと通信をする場合には信号の形態はディジタルなのでそのままシリアルポートを介してデータのやり取りを行う。

３．３．秘話通信システムのハードウェア

秘話通信システムといっても、普通の通信と信号処理の数理が多少異なるだけであるので、ハードウェア的には同じである。具体的には、この場合は二つのＤＳＰをシリアルポートを使用してつなぐだけである。その接続の様子とシリアルポートの結線を図2および3に示す。それぞれの端子名は英文字で表示する。

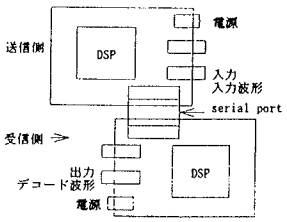

図2　DSP間の接続

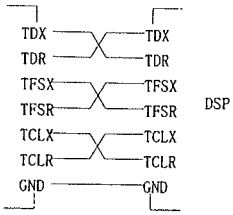

図3　DSPシリアルポート

このシステムでは、ＤＳＰどうしの通信はTDM　Serial Port（時分割）を時分割ポートとして使用せずに普通のシリアルポートとして使用している。それぞれの端子の説明を以下に示す。

・TDX　TDR
　データ用信号線　TDXが送り側　　TDRが受け側
・TFSX　TFSR
　フレームパルス　TFSXが送り側　TFSRが受け側
・TCLX　TCLR
　同期用クロック　TCLXが送り側　TCLRが受け側
・GND
　レベル保証用のグランド線

シリアルポートの入出力は、ＤＳＰの時分割シリアルポートのレジスタへの読み書きによって行われる。ＤＳＰどうしの通信はおおむね以上である。次に、ＤＳＰチップ内のポートを含めたanalog　interfaceを簡単に説明する。ＤＳＰの行う処理操作はＴＤＭポートとほぼ同じであり、analog　interfaceとつながっているシリアルポート用のレジスタに書き込むことにより、アナログ信号に直されて出力される。実際には、書き込まれたレジスタから、専用のシフタに転送されてから転送される。読み込みはこの逆で、analog　interfaceによって、サンプリングされた信号は、デジタイズされて受信レジスタに書き込まれる。このときのサンプリングレートなどはＤＳＰから制御することが可能で、制御は予約されているレジスタに書き込むことによって行う。また、送信レジスタ１６ｂｉｔの内、下位２ｂｉｔはanalog　interfaceの制御用に予約されているので、ここで変換されるデータは、１４ｂｉｔとなる。

３．４．秘話通信システムのソフトウェア

ハードウェアの構築ができたら、秘話通信システムの鍵を握るアルゴリズムなどのプログラミングを行う。アルゴリズムは先に出てきた数式を使用するわけであるが、これだけでは計算しているだけであって、工学的にシステムを構築しているというものにはならない。これに、通信のための設定であるとか、初期値の設定、ＤＳＰそのものの設定などをする必要がある。この場合、プログラミングはアセンブラで行うので、レジスタやメモリの割り当て、Ｉ／Ｏの設定と初期化、２台使用するためのタイミングなどを考慮に入れてプログラミングをしていく。実際の流れを図4の簡単なフローであらわす。

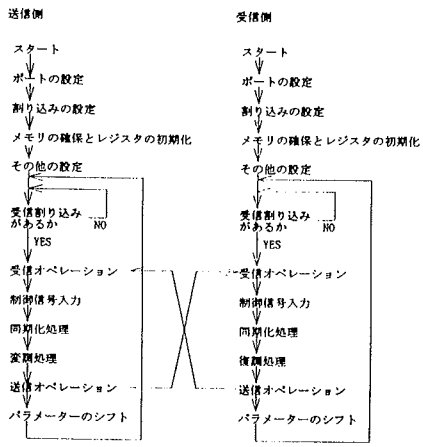

図4　DSP通信システムの流れ

以上が、処理の流れの概略である。演算は、１６ｂｉｔで行い、内訳は上位４ｂｉｔが整数部で下位１２ｂｉｔが小数部の固定小数点である。

４．実験結果

４．１．シミュレーション

実際に秘話通信システムを構築する前に、数理的にどう振る舞うのかを調べるために、計算機によってシミュレーションを行った。設定としては、送信側をシステム１とし、受信側をシステム２とする。伝達したい情報を正弦波の式

$$Si(k)=0.01\sin(0.01t) \qquad (7)$$

として、初期値として０．５を各変数に与えた。

以下に，その実験データを，図5に示す。それぞれは(a)伝達情報信号Si(k)，(b)変調された搬送波C12(k)，(c)通信機２から通信機１への伝達信号C21(k)，および(d)復調された情報Ti(k)である。

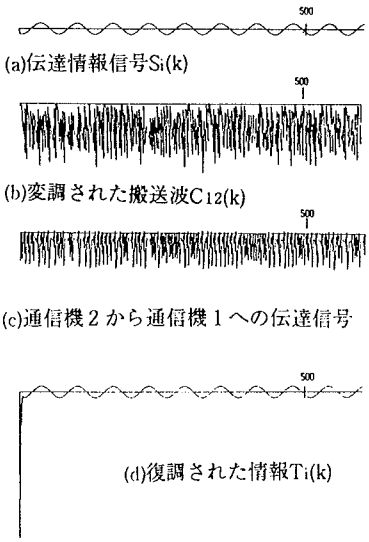

図5　計算機によるシミュレーション

実験結果は上のようになり、数理的には、システムとして成り立つことがわかる。まず、復元できないと通信システムとしては用をなさないので、その点から見てみる。実験結果は、工学的に見て十分満足なものになる。正弦波はきれいに復元されており、ほとんど送信側に入力されたものと重なる。ここで、ほとんどというのは、見た目にはうまく重なっているが、厳密には計算機の丸め誤差などによって、微小な誤差が生じているということである。微小な誤差であっても非線形であるから時間が経てば、大きくずれる可能性もあるが、このモデルではうまく追従しているようなので数理的には全く問題が無い。

次に、秘話通信なので、第三者がデコードすることが困難であることが必要な条件となるわけであるが、この点でも問題は全くないと思われる。通信機間の信号はスクランブルがかかっており、第三者が見ても解読は困難であると言える結果が出ている。また、パラメーターに対しても大変敏感である。例えば、同期化部の数式の内、係数１である$Yi(k)$を$0.01Yi(k)$とした場合、ほかのパラメーターが同じであっても、正しいデコードは不可能であり、結果は通信機どうしのスクランブルのかかった信号のように全く元に戻すことができない。この事は、第三者がパラメーターを探し出して情報を盗もうとしても極めて困難であることを示しており、秘話性の信頼が高いと言える。以上の結果から、数理的には良いシステムであると言え、工学的に実現できるという可能性を示している。

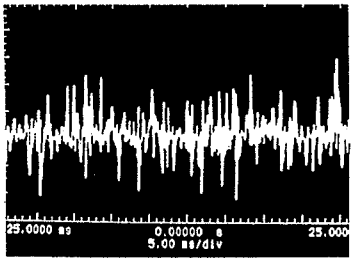

図6　送信信号１

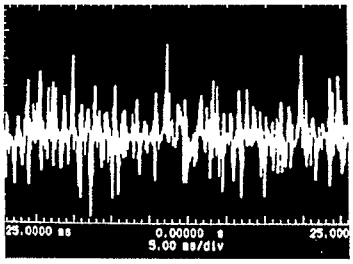

図7　デコード結果１

４．２．実際のシステムの実験結果

次に、上記の複雑なカオス数理式を使って実際のシステムを組み、実際に動作させてみた結果を図6および7にそれらの波形のサンプルを示す。図6は、発信機から送信機へ送られている信号である。理論通りではないものの、変調らしいスクランブルはかかっている。

図7は、受信機側でデコードを試みた結果である。期待した結果は、入力した正弦波が出てくることであったが、実際に出た結果は、通信機間の信号のようにスクランブルがかかったようになっており、復調（デコード）できなかった。また、信号周波数を0.1～６kHzと変更しても、目立った変化は見られなかった。結果としては、変調はかかってはいるが、復調が行われておらず、第三者はもとより受信者も情報を受け取ることができなかった。カオスを発生して変調をかけるという部分は恐らく計算機によるシミュレーションとは異なってはいるが、一応かかってはいる。復調は計算はしてはいるが、入力した正弦波には戻らなかった。

このように，複雑なカオス数理式を用いたDSP通信回路実験ではうまく行かなかった原因を，次に考えてみよう。数理的にはシミュレーションによって、使用したアルゴリズムによる秘話通信システムは可能であることは証明することができたので、工学的に構築する段階で何らかの障害が発生し、それが失敗を招いたと考えられる。考えられる主な原因を以下に示すと，

１．プログラムミスによるバグ

２．ＤＳＰどうしの通信やanalog　interface関連で発生したバグ

3．丸め誤差の累積によるバグ

となる。まず、最初のプログラムミスによるバグの可能性であるが、デバッグを繰り返し、その一つ一つは入力された値についておおよそ正しい結果を返してくるようになっているため、この事による失敗の可能性はかなり低いはずである。次に、ＤＳＰどうしの通信やanalog interface関連で発生したバグの可能性であるが、これらの可能性も低いと思われる。まず、ＤＳＰどうしの通信であるが、予備実験を行った際に、通信ミスは確認することができなかったので、通信中に何らかの原因で値が化けてしまう可能性は少ないはずである。最後に、丸め誤差の累積によるバグの可能性であるが、結論からいえばこれの可能性が最も大きい。カオスの性質のところでも述べたように、このシステムの数式は非線形方程式で成り立っているため、線形では影響のでないような微小な誤差であっても、非線形では時間の経過とともに大きな誤差となってあらわれる可能性がある。よって、丸め誤差によるシステム不安定の可能性が最も高いと考えられる。これによると，複雑なカオス数理は今回使用したDSPの演算性能を超えていると思われる。したがって，次に今のハードウェア環境でのカオス通信を可能にする簡単なモデルによる方法を試みた。

5．簡単なカオス変調モデル

5．1．簡単なカオス変調のアルゴリズム

数式(1)～(6)の複雑なカオス数理に基づくアルゴリズムで今のシステムを組もうというのは、上のような理由からほぼ不可能であることが分かった。ハードウェア環境はそのままで上の条件を満足するものを作るには、演算量を減らすことである。そのためにもっと簡単なカオス数理式に基づくアルゴリズムを考えることである。新しいアルゴリズムを作るにあたって考えなければならないことは、

1．可能な限り、演算回数の少なくてすむような数理にすること，

2．少なくとも送信信号を見られても何を送っているのかは分からないこと，

である。プログラムを単純にしようとして３２bitを演算単位とすることは避け、１６bitのまま行う。これは前の方式とのアルゴリズムの差を出したいからである。そこで、具体的に簡単なモデルを作ってみるわけであるが、まず，上のような条件から単純にカオスを発生して、伝達すべき信号に掛け算するか足し算をするかの二つを行う。カオスを発生する部分と、信号とは掛けられるか足されるかだけであるから、発生されるカオスはこのふたつは同じ物を使用する。使用する簡単なカオス式は，

$$X(k+1)=4X(k)\{1-X(k)\} \qquad (8)$$

である。これを発生するには、１回の掛け算と、１回の引き算、それと１回のビットシフトを行えばよい。具体的には次のようになる。

・１．$X(k)$ の２乗を計算し、アキュムレーターバッファへ格納

つまり，２乗の計算→アキュムレーターバッファへのストア→桁合わせのためのビットシフト（この場合１２ｂｉｔ）

・２．アキュムレーターに$X(k)$を格納

・３．アキュムレーターからアキュムレーターバッファを減算

・４．アキュムレーダーの内容を２ビット左へシフト（４を掛ける）

・５．以上の結果を$X(k+1)$に格納

以上のプロセスを全て実行すると、最短で７ステップで完了する。もちろん、一つ一つの命令が、全て１サイクルで終わるとは限らないので、実際はもう少し時間がかかることになる。しかしながら、失敗した複雑なアルゴリズムを書いたものよりははるかに短いし、演算量も極めて少ないので、誤差は少なくなるはずである。

5．2．簡単なカオス通信システムの実験

後は、こうして作られたカオスに伝達したい信号を掛け算するか、足し算するかであるが、これらについてそれぞれを試した結果を以下に示す。入力した情報は、周波数が６kHzの正弦波である。まずは、簡単なカオス式(8)に情報信号$Si(k+1)$を掛け算変調した後，受信側でそれを割り算するものである。以下にその波形の一例を図8示す。

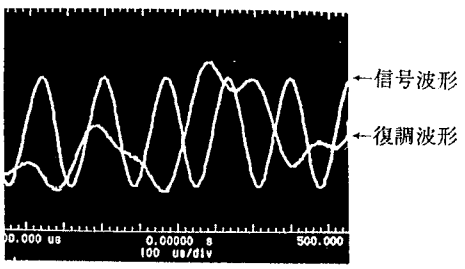

図8　掛け算モデルのデコード波形
(時間軸:×50)

結果として、デコードは正しく行われなかった。規則的な正弦波を情報として入力したにもかかわらず、デコードされて出てきたものは、正弦波にはなっておらずスクランブルのかかったままの状態であった。よって、この結果も失敗であるといわざるをえない。考えられる理由は、前の複雑なモデルと同じく、分解能の不足であると考えられ、このモデルもこのＤＳＰにとっては、負荷が大きすぎるということになる。負荷のかかった部分は掛け算と割り算の部分であると考えられ、掛けてから割るというのは、１６ｂｉｔの分解能では難しいということがわかった。

次に、足し算のモデルであるが、簡単なカオス式(8)に情報信号$Si(k+1)$を加算変調した後，受信側でそれを差引き，復調した。値のフォーマットは同じにしてある。結果からいえば、デコードは正しく行われた。以下にその波形の例を図9，10および11に示す。

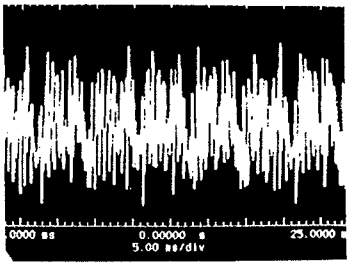

図9　.足し算モデルの送信波形１

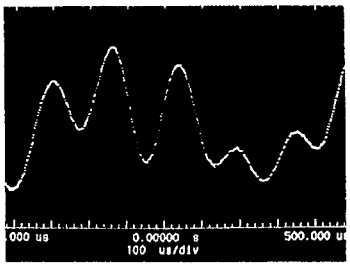

図10　.足し算モデルの拡大した送信波形
（時間軸:×50）

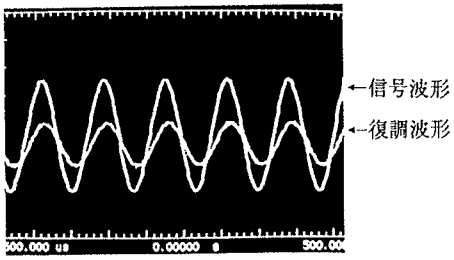

図11　.足し算モデルのデコード波形
（時間軸:×50）

図10はスクランブルのかかった送信波形である。原理的には、生成したカオスに信号を加算しただけであるので、波形そのものはほとんどカオスによるものである。しかし、入力したものが正弦波であるため、入力する正弦波が、カオスによる波より大きくなってしまった場合は、正弦波が浮き出てしまうのでその様な場合には情報が隠れるように小さくする必要がある。図10は図9の拡大である。スクランブルがかかっているためにもとの正弦波はわからない。

図11は入力した波形とデコードされて出てきた波形を重ねたものである。入力された正弦波（振幅の大きいほう）がデコードされて若干がたついてはいるが、正弦波（振幅の小さい方）のようになって出てきている。このがたつきはやはり丸め誤差等のシステム的な誤差であると考えられる。しかしながら、前の掛け算モデルのようにそれが致命的な障害とはなっていない。よって、完璧とまではいかないが、ある程度の誤差を生じても構わないというような情報であれば、この方法でもデータのやり取りは可能である。

５．３．簡単なカオス通信実験結果の検討

搬送波を加算変調する簡単な方法でシステム的な運用が可能となることが分かった。そこでシステム通信速度を検討すると、システムの１周期は図10から読み取ることができ、その周波数は約６kHz である。基本的にこの速度を越えると満足な復元を期待することはできない。これを越えてしまうとそれ以上の成分はサンプリングしきれなくなり、それ以上の高い周波数の成分がなくなってしまい、正弦波の様になってしまう。よって、高周波を多く含んでいる信号にはむかない。また、丸め誤差等により厳密には再生されないので、ある程度の誤差が生じてのも構わないような情報源しか扱うことはできない。例えば、ある数値をディジタル信号で送るといったことには向いておらず、音声や画像（情報の密度が高いものは不可）などには使うことができそうである。

６．まとめ

以上の内容をまとめると，カオスを用いた秘話通信の可能性を探るために、まず、数式を確保してシミュレーションを試みた。結果は良好で数式的には満足のいくシステムであることが解った。数式レベルのシステムは確保できたので、その数理を利用してDSPで実際のシステムを組んだが、演算性能を超えて復調ができなかった。ハードウェア環境をそのまま用いて，システムの数理を簡単なものに変更することで実験を試みた結果，足し算モデルではカオス変調通信が可能であることがわかった。この場合，演算精度が落ちても動くという反面、高い秘話性の保証という点にまだ問題が残る。結論として、複雑なモデルであっても、実現、運用は可能ではあるが、現行のハードウェアでは演算精度に問題がある。現在では費用の問題もあって、満足のいくものを作るのは難しいが、ハードウェアが発達して、より演算精度の高いものがより安く提供されるようになれば、実現、運用は可能であることがわかった。

参考文献

[1]　潮俊光:" カオス同期化制御とその秘匿通信への応用" ，情処学論誌 Vol.36, No.3, pp.525-530 (1995)

[2]　H. Dedieu, M.P. Kennedy and M.Hasler:"Chaos Shift-Keying : Modelation and Demodelation of a Chaotic Carrier Using Self-synchronizing Chua's Circuits", IEEE Trans. Circuits and Systems II, Vol.40, No.10, pp.634-642 (1993)