Uso de Backtracking para Generación de Sucesiones Sonares

Eliseo Gallo Albarracín
D.B.A. Gerencia en Sistemas de Información
Universidad del Turabo, Puerto Rico
MSc. en Computación Científica
Universidad de Puerto Rico, Mayagüez, Puerto Rico
Docente Tiempo Completo, Investigador Grupo SIGMMA,
Universidad Santo Tomás Bucaramanga, Colombia
eliseo12@mail.ustabuca.edu.co
eligallo@hotmail.com

Frank Nicolás Delgado
MSc. Sistemas de Manufactura
Instituto Tecnológico de Estudios Superiores,
Campus Monterrey, México
Docente Tiempo Completo, Investigador Grupo CAyPRO,
Universidad Santo Tomás Bucaramanga, Colombia
franknicolas12@mail.ustabuca.edu.co

Resumen— En un salto de frecuencia de un sistema de radar, la señal se compone de una o más frecuencias elegidas de una posible combinación de m frecuencias disponibles para la transmisión en n intervalos consecutivos de tiempo. Esta señal puede ser representada por una matriz de m x n de 0's y 1's, donde es necesario que en cada columna contenga exactamente un 1. Cuando la señal es reflejada hacia el observador, esta se desplaza en el tiempo y frecuencia. La cantidad de estos movimientos (desplazamientos) se pueden utilizar para determinar la distancia y velocidad. La cantidad de estos saltos, a su vez, se determinan mediante la comparación de todos los turnos de una réplica de la señal transmitida con la señal recibida. Esto es equivalente a contar el número de coincidencias de 1's en una versión desplazada de la matriz de 0's y 1's que representa la señal. El número de coincidencias, como una función de cambios en el tiempo y la frecuencia se llama la función de "auto-correlación". Una matriz sonar es un modelo m x n que tiene a lo más una coincidencia con su función de auto-correlación. En un entorno de múltiples objetos, un patrón se envía para cada objetivo. En este trabajo se presentan algunos métodos que generan secuencias de sonares para el reconocimiento de objetivos múltiples, y también se hace mención a algoritmos de búsqueda para la mismas; como caso particular, se expone el uso de la técnica "backtracking" para hacer una búsqueda exhaustiva para encontrar secuencias de sonares.

Palabras clave— Algoritmo, backtracking, complejidad, Costas, secuencias sonares.

Abstract— In a frequency hopping radar system, the signal consists of one or more frequencies chosen from a set of m available frequencies for transmission at each of a set of n consecutive time intervals. Such a signal can be represented by an m X n matrix of 0's and 1's in which each column contains exactly one 1. When the signal is reflected back to the observer, it is shifted in both time and frequency. The amounts of these shifts can be used to determine both distance and velocity. The amounts of shifts, in turn, are determined by comparing all shifts of a replica of the transmitted signal with the signal received. This is equivalent to counting the number of coincidences of 1s in a shifted version of the matrix of 0's and 1's that represents the signal. The number of such hits as a function of shifts in time and frequency is called the auto-correlation function. A sonar array is a m x n pattern which has at most one hit in its auto-correlation function. In a multiple target environment, one pattern is sent for each target. In this work we study algorithms that generate sonar type sequences for one and multiple target recognition. The use of backtracking technique to find sonar sequences is presented as a particular case.

Keywords— Algorithm, backtracking, complexity, Costas, sonar sequences.

I. INTRODUCCIÓN

Las secuencias sonares son patrones de sincronización bidimensionales, que son usadas en variedad de aplicaciones tales como comunicaciones radares, sonares y ópticas entre otras. Regularmente este tipo de patrones puede ser formulado en términos de poder encontrar modelos de dos dimensiones basados en unos (*puntos*) y ceros (*vacíos*) para los cuales la función de auto correlación no periódica es llamada "función de ambigüedad" de análisis de radar con mínimos valores.

El uso de las sucesiones sonares está estrechamente relacionada al codificar, envío de señales en tiempos y frecuencias determinados; para ello se requiere el disponer del mayor número de secuencias posibles dadas unas condiciones iniciales particulares.

Hasta el momento existen varios tipos de construcciones de sucesiones sonares de tipo algebraico, pero ellas no alcanzan a determinar la totalidad de las sucesiones posibles de las existentes, éstas se limitan a ciertos valores dadas unas condiciones iniciales, pero no la totalidad de las mismas. Como alternativa a estos métodos al-

gebraicos se ha recurrido al uso de implementación de algoritmos computacionales, lo cual en algunas ocasiones implica mucho tiempo de cómputo.

Este trabajo presenta la forma de realizar la búsqueda de sucesiones sonares y secuencias PPM, computacionalmente y para ello se expone el uso de la técnica de *backtracking*, para la implementación, con el fin de utilizar el menor tiempo y alcanzar la mayor eficiencia posible, para ello, este trabajo se basa en algunos de los modelos algebraicos para determinar algunas generalizaciones presentes en las secuencias sonares.

II. NATURALEZA DE LAS SUCESIONES SONARES

Las señales radares y sonares se usan para determinar tanto la distancia (*rango*) de un blanco desde un observador, como la velocidad (llamada *razón de cambio de rango*) en cual el blanco se está acercando o retrocediendo del observador. El *rango* es proporcional al atraso del ciclo completo del tiempo (cambio de tiempo) de una señal, y la velocidad es proporcional al efecto "doppler" (o cambio de frecuencia) de la señal.

En un salto de frecuencia en un sistema sonar o radar, la señal consiste de una o más frecuencias existentes de un conjunto $\{f_1, f_2, ..., f_m\}$ de frecuencias disponibles, para la trasmisión de cada uno de un conjunto $\{t_1, t_2, ..., t_n\}$ de intervalos consecutivos de tiempo. Estos modelos se pueden representar convenientemente con una matriz de permutaciones A de tamaño m x n, donde las m filas corresponden a las m frecuencias, las n columnas corresponden a los n intervalos de tiempo; las entradas de a_{ij} son iguales a 1 si y solo si la frecuencia f_i es trasmitida en un intervalo de tiempo t_j, en caso contrario tendremos que $a_{ij}= 0$.

Cuando las señales son reflejadas desde el objeto y son recibidas nuevamente por el observador, existe una variación tanto en tiempos como en las frecuencias. Según la cantidad de estos cambios, es posible determinar ambos factores, distancias y velocidades. El observador determina la cantidad de estos cambios por comparación (en ambos factores, en tiempo y frecuencia) de una réplica de las señales trasmitidas con la actual señal recibida, y apuntando para cada combinación de cambio de tiempo y cambio de frecuencia la mejor coincidencia. Para llevar la cuenta de estas coincidencias se utiliza una función de correlación.

Dadas A y B dos matrices de permutación, cuando B es cambiado r unidades a la derecha (negativo si el cambio es a la izquierda), y s unidades arriba (negativo si el cambio es hacia abajo), se sobrepone la versión cambiada B^* de B en A y contamos el número de coincidencias entre las primeras en la matriz A y con las primeras en la matriz B^*. El número de estas coincidencias, $C_{AB}(r,s)$, es la función correlación entre A y B^* y claramente satisface las siguientes condiciones:

$$C_{AB}(r,s)=0, \text{ si } |r|\geq n \text{ ó } |s|\geq m, \text{ con } 0\leq C_{AB}(r,s)\leq n$$

Cuando $A\neq B$, la $C_{AB}(r,s)=0$ es llamada función de correlación cruzada. Cuando $A=B$, la $C_{AA}(r,s)$ es llamada función de auto correlación. Además de lo anterior, la función de auto correlación satisface la condición:

$$C_{AA}(0,0)=n$$

La función de auto correlación $C_{AA}(r,s)$, es llamada función ambigüedad en la literatura sonar y radar.

Supongamos que enviamos dos modelos A y B a dos blancos y entonces recibimos una matriz resonancia R (podríamos considerar que algunas señales son perdidas). Al comparar R con todos los cambios de A y B, nosotros necesitamos para determinar que R es la respuesta de algún modelo A ó B, y la cantidad de ambos cambios de tiempos y frecuencias.

Si R contiene al menos tres señales y los modelos A o B satisfacen:

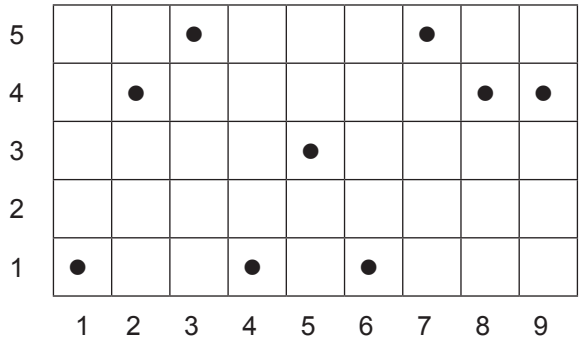
$$\begin{cases} \max C_{AB}(r,s)\leq 2 \\ \max\limits_{(r,s)\neq(0,0)} C_{AA}(r,s)\leq 2 \\ \max\limits_{(r,s)\neq(0,0)} C_{BB}(r,s)\leq 2 \end{cases}$$

También se podría construir un modelo original sin ambigüedades, para ello recurrimos a la secuencias sonares, las cuales fueron sugeridas por Solomon W. Golomb y Herbet Taylor en [4] que son

un modelo m x n, el cual tiene a lo más un *hit*[1] en la función de auto correlación.

En condiciones de reconocimiento de múltiples blancos, un modelo es enviado para cada blanco, las categorías de arreglos de Costas y secuencias sonares para reconocimiento de múltiples objetivos son aportadas por O. Moreno y S. Maric en [9], [10] y [11]. Por el método de construcción dado en [5] se pueden construir series de secuencias que posean características de "buena correlación cruzada", pero no "buena auto correlación cruzada". La figura 1, presenta un modelo con auto correlación cruzada uno.

FIG. 1. REPRESENTACIÓN DE UNA SECUENCIA SONAR DE TAMAÑO 5X9



A. Base de Construcción

El principal problema abierto para las secuencias sonares es encontrar la más grande longitud n para un número de frecuencia dada m tal que ésta sea una secuencia sonar m x n. A continuación vamos a ver algunas de las construcciones de secuencias sonares. Todas estas construcciones de sucesiones producen secuencias sonares modulares que corresponden a cuadradas, o a lo más a arreglos cuadrados. Una serie de transformaciones es aplicada una vez para obtener un número filas vacío en la parte superior del arreglo. Estas filas son borradas, lo que corresponde a reducir m, para así obtener un secuencia sonar ordinaria mejorada. En las construcciones que se presentan a continuación se han realizado cambios en algunas funciones por conveniencia.

(1) **Cuadrática** [2]: Si p es un primo impar; dados a, b y c constantes enteras no congruentes con cero $(\mathrm{mod}\ p)$ Entonces, $f:\{1,2,\ ...,\ p+1\}\rightarrow\{1,2,\ ...,\ p\}$, definida por: $f(i)=ai^2+bi+c\ (\mathrm{mod}\ p)$ es una p x $p+1$ secuencia sonar modular.

(2) **Secuencia Alternada** [3]: Si p es un primo, α un elemento primo de $GP(p^{2r})$, y β un elemento primo en $GP(p^r)$. Para $p=2$, dado $f:\{1,2,\ ...,\ p^r\}\rightarrow\ \{1,2,\ ...,\ p^r-1\}$ definida por $f(i)=\log_\beta(\alpha^i)$ $^{p^r}+\alpha^i)$. Para p impar, se define f similarmente, excepto por cambiar el dominio a $\{i:(-p^r-1)/2\le i\le(p^r-1)/2\}$. Entonces, f es una (p^r-1)xp secuencia sonar modular.

Las siguientes construcciones inicialmente se originaron en el contexto de construcción de arreglos de Costas[2] y han sido adaptados a secuencias sonares.

(3) **Exponencial Welch** [5], [6], [7]: Si α es un elemento primitivo módulo el primo p. Entonces $f:\{1,2,\ ...,p-1\}\rightarrow\{1,2,\ ...,p\}$ definida por: $f(i)=\alpha^i$ es una px$(p-1)$ secuencia sonar modular.

(4) **Logarítmica Welch** [7]: Si α es un elemento primitivo módulo el primo p. Entonces $f:\{1,2,\ ...,\ p-1\}\rightarrow\{1,2,\ ...,\ p-1\}$ definida por: $f(i)=\log_\alpha i$ es una $(p-1)$x$(p-1)$ secuencia sonar modular.

(5) **Lempel** [3], [5], [7]: Si $q>2$ es una potencia prima, y cuando α y β son elementos primos de $GF(q)$. Entonces $f:\{1,2,\ ...,\ q-2\}\rightarrow\{1,2,\ ...,\ q-1\}$ definida por: $f(i)=j$ si y solo si $\alpha^i+\alpha^j=1$ es una $(q-1)$x$(q-2)$ secuencia sonar modular.

(6) **Golomb** [5], [7]: Si $q>2$ es una potencia prima, y cuando α y β son elementos primos de $GF(q)$. Entonces $f:\{1,2,\ ...,\ q-2\}\rightarrow\{1,2,\ ...,\ q-1\}$ definida por: $f(i)=j$ si y solo si $\alpha^i+\beta^j=1$ es una $(q-1)$x$(q-2)$ secuencia sonar modular.

B. Transformaciones en Sucesiones Sonares

Una sucesión sonar modular m x n puede ser sujeta a tres transformaciones [10] en un intento para así producir una mejor secuencia sonar. Estas transformaciones son:

(1) Adición por a módulo m:

$$y_{+\alpha}(k) = y(k)+a \qquad (\mathrm{mod}\ m)$$

Esta corresponde a una rotación cíclica de filas en la secuencia sonar a unidades. Las filas vacías contiguas son rotadas a la parte superior de la sucesión y son reubicadas para así obtener una mejor sucesión sonar. Esta corresponde a la

[1] Posee diferentes valores en triángulo de diferencias para dicha sucesión.

[2] Los arreglos de Costas son un caso particular de secuencias sonares, cuando $n = m$.

restricción del tamaño de *m*. Una vez *m* ha sido restringido, la secuencia sonar resultante no aumentada satisface las distintas propiedades de diferencia modular

(2) Multiplicación por una unidad *u* módulo *m*:

$$y_{xu}(k) = u \cdot y(k) \qquad (\mathrm{mod}\ m)$$

Esta corresponde a una permutación de filas en la secuencia sonar. Para cada iteración se aplican permutaciones, con el fin de aumentar el número de filas vacías contiguas en la sucesión sonar.

(3) Recortar con s módulo *m*:

$$y_{shear(s)}(k) = y(k) + s \cdot i \qquad (\mathrm{mod}\ m)$$

Esta corresponde a recortar las columnas de la sucesión sonar por *s* unidades y recortarlas módulo *m*. El recorte es aplicado en un intento, para aumentar el número de filas vacías en una secuencia sonar.

La mayoría de los métodos existentes para construir arreglos de Costas son construcciones algebraicas, pero ello excluye algunos arreglos posibles que no se obtienen por estos métodos algebraicos, por esta razón se han tratado de encontrar todos los posibles arreglos de Costas existentes utilizando métodos computacionales, una muestra de este tipo de trabajos es el dado por O. Moreno, J. Ramírez, E. Orozco y D. Bollman [11] en el cual emplean un algoritmo utilizando *backtracking*[3] para generar un cierto número de permutaciones. Luego de trabajar este tipo de generación de permutaciones se estima que con la ayuda de un algoritmo con *backtracking* se pueden generar arreglos de Costas pero que ello requiere cierto costo computacional, E. Orozco [12] implementa un algoritmo para la generación de arreglos de Costas en paralelo con el fin de disminuir dichos costos.

III. USO DE *BACKTRACKING* PARA LA BÚSQUEDA DE SECUENCIAS SONARES

"*Backtracking*" (también conocido como "vuelta atrás") es una técnica que de forma sistemática y organizada, genera y recorre un espacio que contiene todas las posibles secuencias de decisiones. A este espacio se le denomina "espacio de búsqueda del problema".

[3] Técnica de búsqueda en secuencias numéricas

Los algoritmos de *backtracking* hacen una búsqueda sistemática de todas las posibilidades, sin dejar ninguna por considerar. Cuando intenta una solución que no lleva a ningún sitio, retrocede deshaciendo el último paso, e intentando una nueva variante desde esa posición (es normalmente de naturaleza recursiva).

La técnica de *backtracking* ofrece un método para resolver problemas tratando de completar una o varias soluciones por etapas [4]. En cada paso se intenta extender una solución parcial de todos los modos posibles, y si ninguno resulta satisfactorio se produce un retroceso (o vuelta atrás) hasta el último punto donde quedaban alternativas por explorar. Es una mejora de la búsqueda completa de soluciones y una alternativa a los algoritmos voraces.

En muchas aplicaciones del método de *backtracking*, la solución deseada se puede expresar como una n -tupla $(x_1, x_2, \ldots, x_n)$, donde cada x_i hace parte de la selección de un conjunto finito S_i. Frecuentemente para resolver el problema (encontrar un vector máximo, o mínimo o que satisfaga las condiciones deseadas) se utiliza una *"función criterio"* $P(x_1, x_2, \ldots, x_n)$.

En nuestro caso particular queremos encontrar todas las posibles soluciones factibles (secuencias sonares dados m y n), por eso es conveniente utilizar la técnica mencionada. Para una sucesión $s=\{s_1, s_2, \ldots, s_n\}$, tendremos m^n soluciones posibles. La función "prueba", va a hacer una selección de las sucesiones las cuales satisfacen la condición del "triángulo de diferencias". La restricción de este problema va a ser entonces dada en poder verificar la prueba del triángulo de diferencias.

A. El algoritmo

Este algoritmo está diseñado en dos funciones principales, la primera de estas, la *función backtracking* (Fig. 2), y la segunda la "*función criterio*", llamada función prueba (Fig. 3).

La estructura general del algoritmo se basa en utilizar una matriz denominada "matriz de diferencias" A, la cual es global a los llamados de la función, y en la cual permanentemente se están registrando los valores de las diferencias correspondientes que se calculan. Si una "rama" va hacia adelante, se conservan las diferencias ya registradas para evitar calcularlas

nuevamente cada vez, si la rama va hacia atrás, se deben borrar las diferencias calculadas por la rama subyacente.

FIG. 2. ESQUEMA GENERAL DE LA FUNCIÓN *BACKTRACKING*

```
void backtracking (int * sol, int n)
{
  int k, i;

  sol[0] = 0;          /* inicializa la
primera componente */
  k = 0;               /*  k representa la
componente actual */
  while (k >= 0)
    {
     sol[k]++; /* Prueba el siguiente valor
para la componente actual */

   if (prueba(sol[k],n)==true)       /* Si la
        solución actual es factible */
        {
         if (k == n-1)  /* Si ya tiene una
solución completa, la imprime */
          {
            for (i = 0; i<n; i++)
              printf ("%d ",sol[i]);
            printf ("\n");
          }
        else
          {
            k++;      /* En caso contrario,
pasa a la siguiente componente */
            sol[k] = 0;
          }
       }
     else
        k--; /* Si el vector solución actual
no es factible, va hacia atrás */
    }/* vuelve a la componente anterior */
}
```

Cuando una solución es "factible" se imprime, se registra el número de secuencias factibles en cada caso y posteriormente se continúa el bactracking. Si al finalizar la búsqueda no ha encontrado ninguna sucesión, retorna cero (0), en caso contrario retorna el valor de las sucesiones encontradas.

B. Complejidad del Algoritmo

Sea $s=\{s_1,s_2, ..., s_k\}$ una subsucesión sonar, con k -términos, $1\leq k<n$, y sea $\hat{s}=\{s_1,s_2, ..., s_{k+1}\}$, la sucesión resultante al insertar un nuevo término s_{k+1} en la sucesión.

Haciendo un seguimiento del algoritmo se debe realizar los siguientes pasos:

i) Agregar un término nuevo. Esto es una operación

ii) Computar s_{k+1}-s_{k+1-i}, i=1,2, ..., k.

iii) Compara si cada s_{k+1}-s_{k+1-i} ya ha sido determinado como una diferencia en un paso anterior.

FIG. 3. ESQUEMA GE DE LA FUNCIÓN PRUEBA

```
int prueba(int * v, int n , int M)
{
int i;
int R[50];
if (n <= 1)            /* Si la sucesión posee
un elemento */
        return(1);
else
    {
     for(i=1; i<n; i++)
          {                                  /*
Verifica si las diferencias  */
           if (A[i][v[n]-v[n-i]+M]<1)  /* no
se han registrado */
                R[i]=v[n]-v[n-i]+M;
           else
                 return(0);           /*
devuelve falso */
                                    /* si ya se
ha registrado devuelve falso */
          }
        for(i=1; i<n; i++)
                A[i][R[i]]++;  /* Actualiza
la matriz de diferencias */
        return(1);      /* devuelve verdadero
*/
        }
}
```

Así podemos notar que para el primer paso efectúa dos operaciones (una diferencia y una comparación), para el segundo efectúa dos operaciones, y así sucesivamente. Para el k-ésimo término realiza dos operaciones, esto es:

$$\underbrace{2+2+\ldots+2}_{(k-1)\ veces}=\sum_{i=1}^{k-1}2=2(k-1)$$

De esto es posible deducir que la complejidad para determinar si $\hat{s}$ es sonar es de orden $O(k)$.

Al considerar los casos k varía k=1,2,...,n-1, que sería el resultado de generar desde una sucesión de un solo término, $s=(s_1)$, hasta generar una sucesión sonar completa de tamaño n. Así, para obtener una sucesión de tamaño n, tendremos, cuando $k=n$, 2(n-1) operaciones; si $k=n$-1, 2(n-2) operaciones; si k=1, 0 operaciones.

Al hacer un cómputo del total de operaciones se tiene:

$$2(k-1)+2(k-2)+\ldots+2+0=$$

$$=\sum_{k=1}^{n-1}2(k-1)=n^2-3n+2$$

Es decir, que para poder generar cada sucesión sonar el costo es de orden $O(n^2)$.

Ahora el conjunto en el cual debemos buscar todas las sucesiones sonares es de cardinalidad m^n, puesto que para cada posición i de la sucesión $1 \leq i \leq n$, puede tomar cualquier valor posible j, $1 \leq j \leq m$. Así pues la complejidad de generar todas las todas las sucesiones sonares es de orden $O(m^n)$, que es el tiempo necesario para generar todas las sucesiones sonares de tamaño m x n, un crecimiento exponencial con n.

IV. CONCLUSIONES

La búsqueda de sucesiones sonares mediante el uso de medios computacionales ofrece la posibilidad de encontrar todas las sucesiones existentes para las condiciones iniciales dadas y con esta ayuda se pueden favorecer otros métodos de construcciones de sucesiones algebraicos.

Dadas las condiciones del problema propuesto, definitivamente la técnica de "Backtracking" es la que más favorece en la búsqueda exhaustiva de sucesiones, puesto que además de hacer una búsqueda ordenada y dirigida, controla los campos de búsqueda, a la vez que descarta las posibilidades donde no va a encontrar soluciones factibles.

El algoritmo de búsqueda de secuencias sonares arroja soluciones que corroboran los resultados algebraicos y coinciden con las soluciones exactas arrojadas por el programa en ejecución. El tiempo que le ocupa al computador ejecutar este algoritmo es insignificante para instancias de prueba con $2 \leq m \leq 8$ y $4 \leq n \leq 12$ (por debajo de los 2 segundos), mientras que para valores superiores $m \geq 9$ y $n \geq 13$ se requiere de un tiempo relativamente considerable.

El tiempo de ejecución es bastante bajo para la configuración del hardware donde se realizaron las pruebas. Esto permite inferir que, con una configuración más potente sí se podría pasar esta barrera. El tiempo de CPU empleado no pasaba de los 2 segundos para las instancias más grandes

REFERENCIAS

[1] G. Costas, "On PPM Sequences whit Good Autocorrelation Properties," IEEE Trans. Inform. Theory, vol. 35, NO. 1, pp. 146-149, Mayo 1988

[2] R. Gagliardi, J. Robbins, y H. Taylor, "Acquisition sequences in PPM communications," IEEE Trans. Inform. Theory, vol. IT-33, pp. 738-744, 1987

[3] R. A Games, "An algebraic construction of sonar sequences using M-sequences," SIAM J. Algebraic Discrete Methods, vol. 8, pp. 753 – 761, Octubre 1987

[4] S. W. Golomb,L. Baumert, "Backtrack Programming," Journal of the ACM (JACM), vol. 12, pp. 516-524, 1965

[5] S. W. Golomb, "Algebraic constructions for Costas arrays," J. Combinatorial Theory, Ser. A, vol. 37, pp. 13-21, 1984

[6] S. W. Golomb y H. Taylor, "Two-dimensional synchronization patterns for minimum ambiguity," IEEE Trans. Inform. Theory, vol. IT-28, pp. 263-272, Julio 1982

[7] S. W. Golomb y H. Taylor, "Constructions and properties of Costas arrays," Proc. IEEE, vol. 72, pp. 1143-1163, Septiembre 1984

[8] S.V. Maric y E. L. Titlebaum, "A class of Frequency Hop Codes with Nearly Ideal Characteristics for Use in Multiple Access Spread Spectrum Communications and Radar and Sonar System," en IEEE Transactions on Communications, Septiembre 1992

[9] O. Moreno, S Maric "Classes of Costas and Sonar Sequences For Multi-target Recognition," en IEEE ISIT 1997

[10] O. Moreno, S Maric and LiYuchun "Best Know Sonar Sequences for Multi-target Recognition. In IEEE ISIT 1997

[11] O. Moreno y R. A. Games, "Sonar Sequences from Costas Arrays and the Best Known Sonar Sequences with up to 100 Symbols," IEEE Trans. Inform. Theory, vol. 39, NO. 6, pp. 1985-1987, Noviembre 1993

[12] O. Moreno. S.V. Maric, "A Class of Frequency Hop Codes with Nearly Ideal Characteristics for Multiple-target Recognition. In thirty-third Annual Allerton Conference on Communication, Control, and Computing, Octubre 1995

[13] O. Moreno, P. Pei and J. Ramirez, A Parallel Algorith for Enumeration of Costas, In proceedings of the 7th SIAM Conference on Paralllel Conference Processing for Scientific Computing pp. 255-260, 1995

[14] O.Moreno, J. Ramirez, D. Bollman, y E. Orozco, "Faster algorithms for the generation of Symmetry-Invariant Permutations", Marzo de 2002

[15] E. Orozco, "On the parallel generation of coastal arrays," Tesis, University of Puerto Rico,1998