Introducción a los Algoritmos Genéticos con Matlab

Juan Carlos Seck Tuoh Mora
Joselito Medina Marín
Norberto Hernández Romero

Área Académica de Ingeniería-ICBI-UAEH
Cuerpo Académico: Tecnologías Avanzadas en Ingeniería

Mayo 2016

Universidad Autónoma
del Estado de Hidalgo

Contenido

Universidad Autónoma
del Estado de Hidalgo

Contenido

- Cruce
- Mutación
- Ejemplos (Sintonización de un control PID, identificación de parámetros en una máquina de inducción)

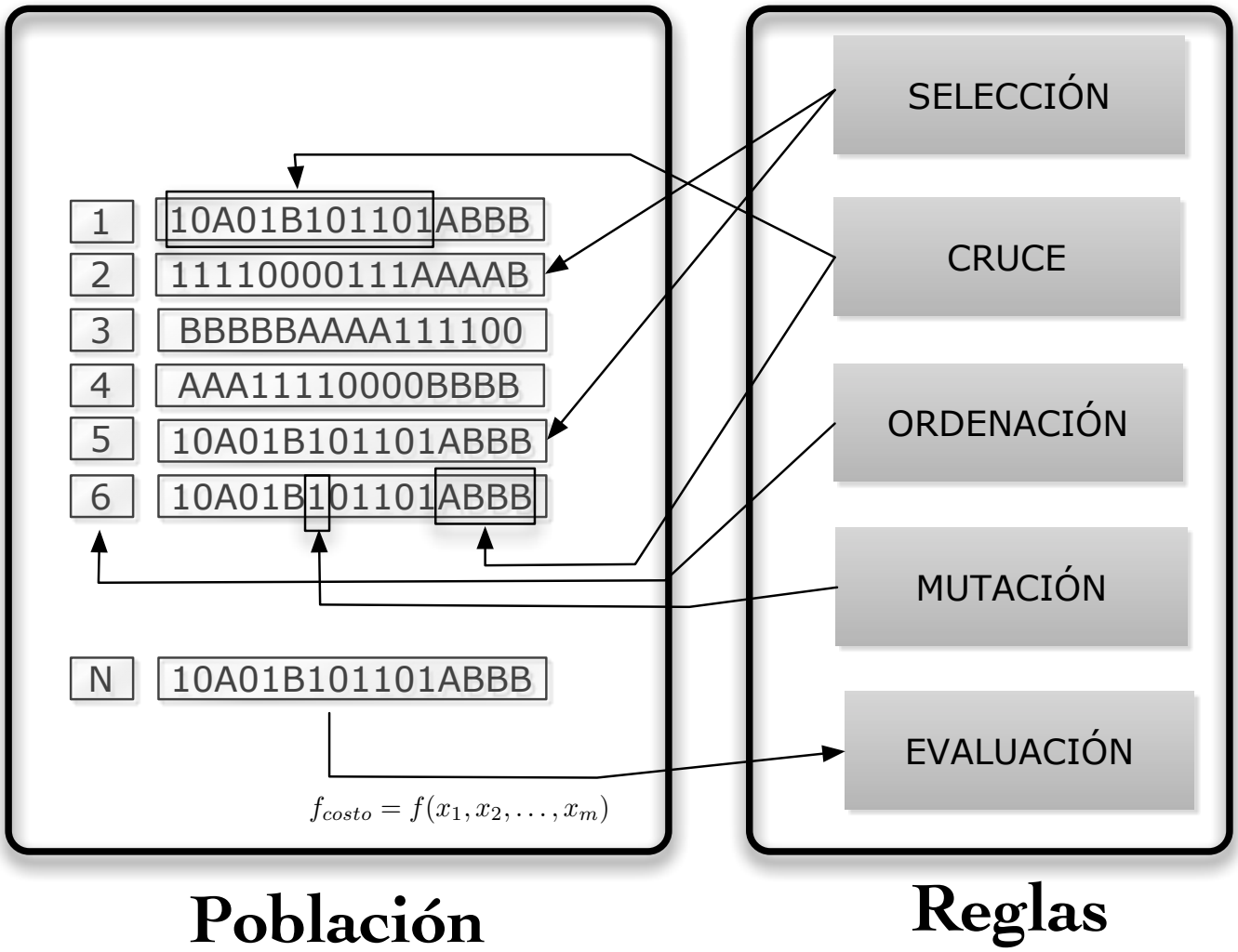
Definición de Algoritmos Genéticos

- Los AG son técnicas de búsqueda y optimización basados en los principios de la genética y selección natural.

- Los AG están integrados por un conjunto de individuos y diferentes tipos de reglas que aplican directamente sobre la población.

Universidad Autónoma del Estado de Hidalgo

Algoritmo Genético (Población y Reglas)

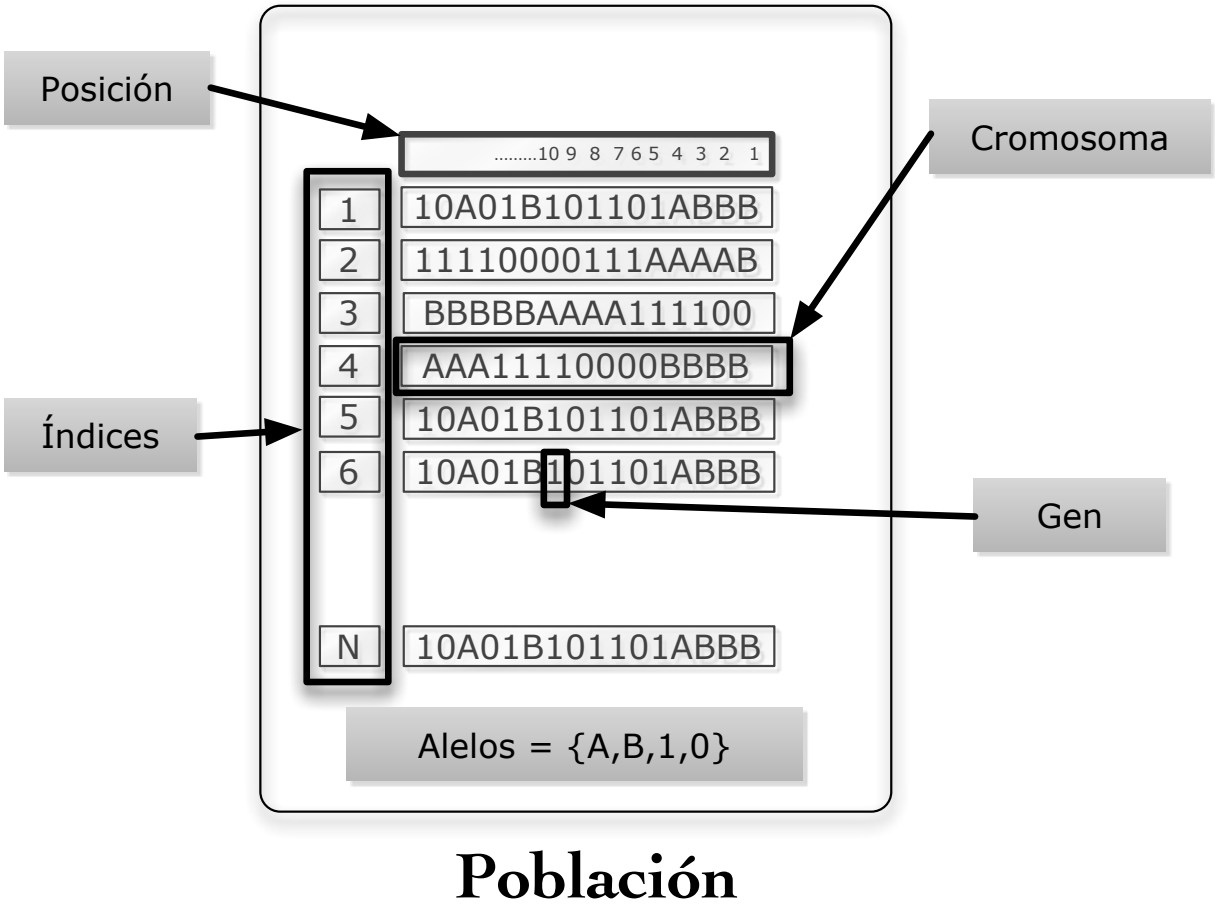


Definiciones

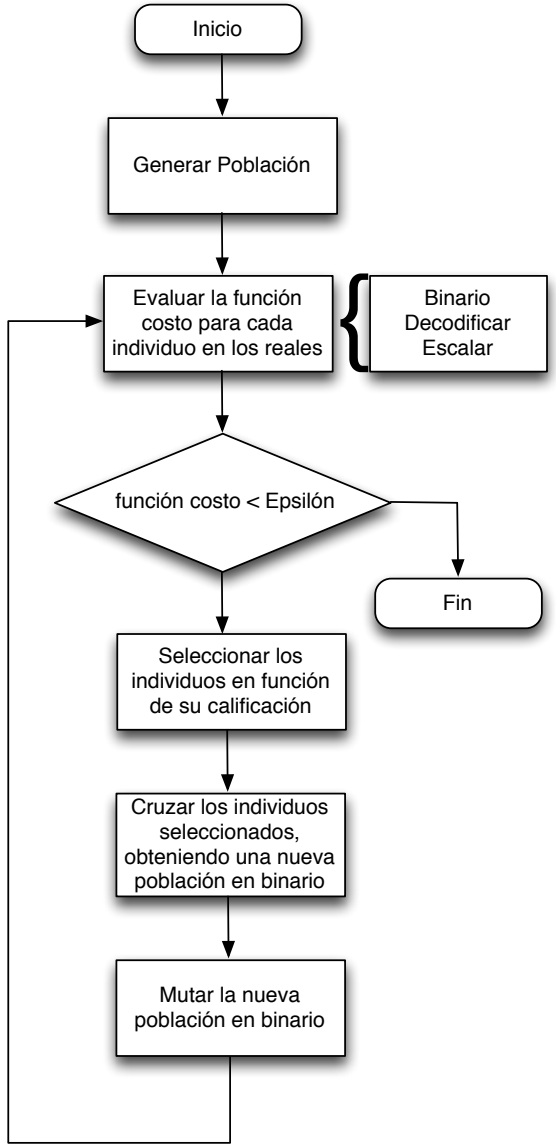
- **Alelo.** Son los distintos valores con los cuales se puede representar un gen.
- **Gen**. Es el valor de un alelo dentro de un arreglo.
- **Cromosoma**. Es una colección de genes en forma de arreglo.
- **Posición**. Es el lugar que ocupa un gen dentro del cromosoma.
- **Índice**. Es la posición que tiene el individuo dentro de la población

Universidad Autónoma
del Estado de Hidalgo

Definiciones en la población



Secuencia de ejecución



Universidad Autónoma del Estado de Hidalgo

Ventajas

- **Implementación computacional**. Las operaciones a realizar en un AG es a través de operaciones aritméticas, lógicas y de ordenamiento.
- **Información del Sistema**. No necesitan información a priori. El AG genera multiples soluciones de forma aleatoria y si algunas de ellas mejoran, entonces, son soluciones que se tomarán en cuenta para evolucionar la población.
- **Singularidades**. El AG realiza un amplia exploración en el intervalo de búsqueda factible sin importar que el sistema sea de funciones discontinuas.

Universidad Autónoma del Estado de Hidalgo

Desventajas

- **Funcion costo**. La única forma para evaluar el desempeño de los individuos en las evoluciones del AG es a través de una función de evaluación o una colección de datos.

- **Reglas**. No existen reglas para determinar el número de individuos en una población, que tipo de selección aplicar o como realizar la mutación.

Desventajas

- **Programación serie**. Cuando los AG se programan en plataformas de procesamiento serie, no son tan rápidos en su tiempo de ejecución y por lo tanto, es difícil implementarlos en aplicaciones donde es necesario realizar ajustes en tiempo real o en línea.

- **Aplicaciones**. Los AG genéticos tienen una amplia gama de aplicaciones donde han logrado optimizar sistemas con éxito, pero esto no quiere decir que se puedan utilizar en cualquier aplicación y que logren un mejor desempeño que los métodos analíticos matemáticos.

Universidad Autónoma del Estado de Hidalgo

1. Población aleatoria

- La población aleatoria de un AG binario esta definida básicamente por N = número de individuos y l = longitud de cromosomas. Ejemplo: N individuos con una longitud de cromosomas de 10 bits

$$\begin{array}{llll} 1 & 0110111010 & \rightarrow & 442 \\ 2 & 1111000111 & \rightarrow & 967 \\ 3 & 1110101011 & \rightarrow & 939 \\ \vdots & \vdots & \vdots & \vdots \\ N & 1111000011 & \rightarrow & 963 \end{array} \quad (1)$$

Universidad Autónoma del Estado de Hidalgo

1. Población aleatoria

- Ejemplo: Una población de N individuos, dos variables y una longitud de 6 bits.

	var 1	var 2	var 1	var 2
1	101010	110011	42	51
2	111111	000001	63	1
3	011011	101110	27	46
⋮	⋮	⋮		
N	101111	100111	47	39

(2)

Universidad Autónoma del Estado de Hidalgo

1. Función en Matlab para crear una población

```
1 function [P] = Poblacion(N,L,V)
2 %N = Numero de individuos
3 %L = Longitud del cromosoma
4 %V = Numero de variables
5     P = round(rand(N,L*V));
6 end
```

Universidad Autónoma del Estado de Hidalgo

1. Función en Matlab para decodificar una población binaria

```
1  function [ P_10 ] = PobDec(P,L,V)
2  %P = Poblacion en binario
3  %L = Longitud cromosoma
4  %V = Numero de variables
5  [R C]=size(P);
6  aux = 1;
7   for j =1:V
8      P_10(:,j) =bin2dec(num2str(P(:,aux:L*j)));
9      aux = aux + L;
10  end
11
12 end
```

2. Escalar la población en un intervalo de búsqueda

- Escalar la población en decimal P_{10} en un intervalo de búsqueda $[I_{j,min}, I_{j,max}]$

$$P_{i,j} = (I_{j,max} - I_{j,min})\frac{P_{i,j}}{2^L - 1} + I_{j,min} \tag{3}$$

Donde $i = 1, 2, \ldots, N$ es i-ésimo individuo de la población, j es la dimensión de la función a optimizar y N es el número máximo de individuos que contiene la población. $P_{i,j}$ es un número decodificado en decimal a partir de los cromosomas de la población.

Universidad Autónoma del Estado de Hidalgo

```
1  function [Pob_float] = Escalamiento(Pob_10,I,V,L)
2  %I es el vector con los intervalos de b_squeda
3  %I = [I(min,1) I(max,2);
4  %      I(min,1) I(max,2);
5  %      ...        ...
6  %      I(min,1) I(max,2)];
7  %Donde el  numero de renglones equivale al
8  %numero de variables
9
10 for i = 1:V
11  Pob_float(:,i) = ...
        Pob_10(:,i)/(2^L-1)*(I(i,2)-I(i,1))+I(i,1);
12 end
13 end
```

2. Escalar una población en un intervalo de búsqueda

- Ejemplo: Generar una población aleatoria en R^2 de 12 individuos en un intervalo de búsqueda de $[-5, 5; -1, 1]$. Suponga una longitud del cromosoma de 12 bits.

Universidad Autónoma del Estado de Hidalgo

3. Evaluar el desempeño de los individuos

- Medir de forma cuantitativa el desempeño de cada individuo con la finalidad de optimizar una función.
- Existen funciones benchmarking para evaluar el desempeño de los algoritmos evolutivos
- La forma más convecional es una función matemática.
- Cualquier forma que nos permita evaluar el individuo puede considerarse como función costo.

Universidad Autónoma del Estado de Hidalgo

3. Evaluar el desempeño de los individuos

- Evaluar la población en la función objetivo, también se conoce como obtener la función costo o fitness fuction

$$ff_i = Costo_i(P_i) \rightarrow \begin{bmatrix} f_1(P_{1,1}, P_{1,2}, P_{1,j}) \\ f_2(P_{2,1}, P_{2,2}, P_{2,j}) \\ \vdots \\ f_N(P_{N,1}, P_{N,2}, P_{N,j}) \end{bmatrix} i = 1, 2, \ldots, N \qquad (4)$$

- El vector costo se ordena de menor a mayor si se desea encontrar el mínimo óptimo
- El vector costo se ordena de mayor a menor si se realiza una maximización

Universidad Autónoma del Estado de Hidalgo

3. Evaluar el desempeño de los individuos

- Existe un conjunto de funciones benchmarking para verificar el desempeño de los AG

- El artículo "GSA: A Gravitational Search Algorithm" muestra algunas funciones bechmarking. http://www.sciencedirect.com/science/article/pii/S0020025509001200

Universidad Autónoma del Estado de Hidalgo

4. Seleccionar los mejores individuos

- La función del operador seleccción es transmitir a generaciones futuras los cromosomas que tienen un buen desempeño, gran parte del éxito de los AG se debe a este operador, existen diferentes métodos.
- Se selecciona el 50% de la población con el mejor desempeño.
- Se realiza un torneo entre los individuos.
- Se implementa un tipo ruleta entre la población.

Universidad Autónoma del Estado de Hidalgo

4. Función selección en Matlab

```
1  function [ PobSel ] = Seleccion(Pob_2,ff)
2  %Obtener tamano de la poblacion
3    [R C]= size(Pob_2);
4  %Ordenar de menor a mayor
5    [B,idx]= sort(ff);
6    Pob_2 = Pob_2(idx,:);
7  %Seleccionar la mitad de
8  %la mejor poblaciÛn
9    PobSel = Pob_2(1:R/2,:);
10 end
```

4. Función torneo en Matlab

```
1  function [PobSel ] = Torneo(Pob_2,ff)
2
3  [R, C] = size(Pob_2);
4  %Vector que contiene los individuos seleccionados
5  idx = zeros(1,R/2);
6
7  t = randperm(R);
8  fc = reshape(t,R/2,2);
9
10 for i = 1: R/2
11    if (ff(fc(i,1)) < ff(fc(i,2)))
12       idx(i) = fc(i,1);
13    else
14       idx(i) = fc(i,2);
15    end
16
17 end
18 PobSel = Pob_2(idx,:);
19
20 end
```

Universidad Autónoma del Estado de Hidalgo

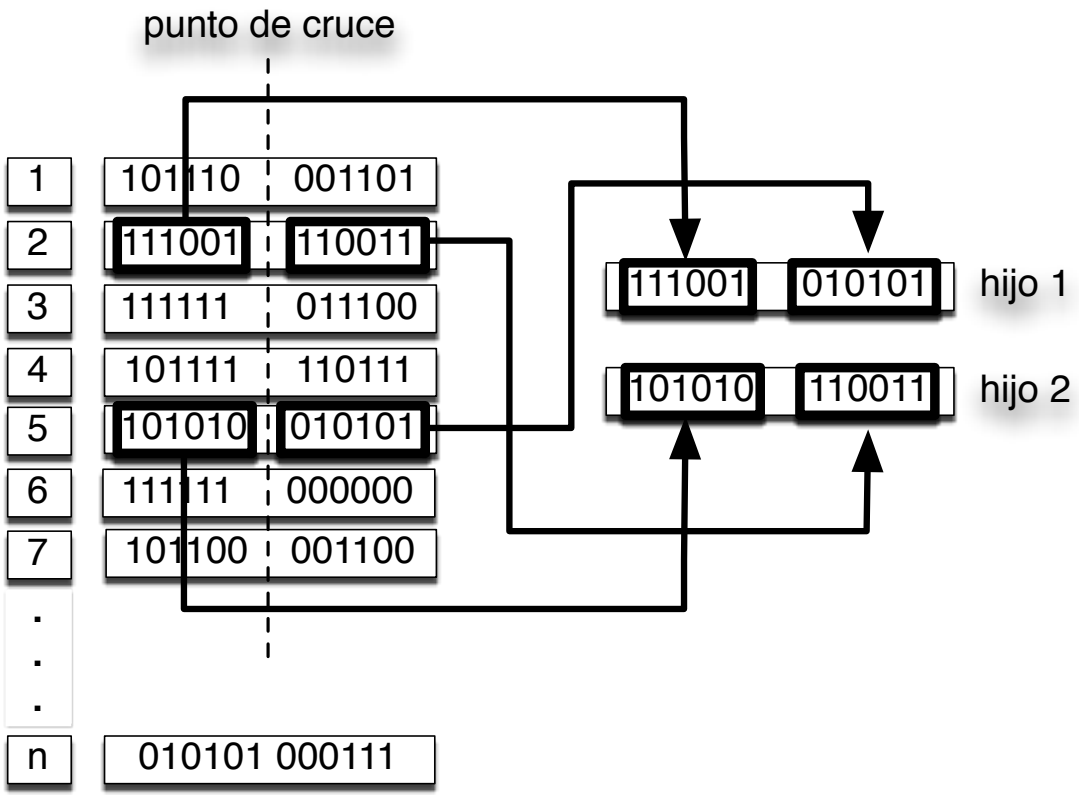
5. Operador cruce

- El operador cruce es la acción de seleccionar dos individuos llamados padres y aparearlos para obtener nuevos individuos llamados hijos.

- Los hijos son el resultado de una combinación de cromosomas de los padres y así cada hijo contiene información genética de los papás.

Universidad Autónoma del Estado de Hidalgo

5. Apareamiento de dos individuos



Universidad Autónoma del Estado de Hidalgo

5. Función cruce en Matlab

```
1  function [ Pob_C ] = Cruce(Pob,L,V)
2  [R C] = size(Pob);     Pob_C = zeros(R,C);
3    ix = 1;
4    for i = 1: V
5     Pob_Aux1 = Pob(:,ix:L*i); Pob_Aux2 = zeros(R,L);
6      t1 = randperm(R);
7      t2 = reshape(t1,R/2,2); i1 = 1;
8      for j=1:R/2
9          pcruce = randi([1,L-1]);
10         aux1 = Pob_Aux1(t2(j,1),1:pcruce);
11         aux2 = Pob_Aux1(t2(j,1),pcruce+1:L);
12         aux3 = Pob_Aux1(t2(j,2),1:pcruce);
13         aux4 = Pob_Aux1(t2(j,2),pcruce+1:L);
14         Pob_Aux2(i1,1:L) = [aux1 aux4];
15         Pob_Aux2(i1+1,1:L) = [aux3 aux2];
16         i1 = i1+2;
17     end
18     Pob_C(:,ix:L*i) = Pob_Aux2(:,:);
19     ix = ix+L;
20   end
```

Universidad Autónoma del Estado de Hidalgo

5. Operador mutación

- La mutación es una forma de explorar nuevas alternativas con la finalidad de lograr individuos que generen nuevas soluciones.
- la mutación se realiza modificando de forma aleatoria los genes de una población con un cierto porcentaje.

Universidad Autónoma del Estado de Hidalgo

función mutación en Matlab

```
1  function [ Pob ] = Mutacion(Pob,L,prob)
2  % Pob = Poblacion que se desea mutar.
3  % L   = Longitud del cromosoma
4  % Prob= Porcentaje de mutacion
5
6  %Se determina la cantidad de genes de la poblacion
7  [R C] = size(Pob);
8  %Numero de  genes que contiene la poblaciÛn
9  Tgenes = R*C;
10 %Numero de genes a mutar
11 GenesMut = round(Tgenes*prob/100);
12
13 for i = 1:GenesMut
14  fila = randi([1 R]);  colum= randi([1 C]);
15  %Se realiza el complemento del gen seleccionado
16  Pob(fila, colum) = 1 - Pob(fila,colum);
17 end
18 end
```

Universidad Autónoma del Estado de Hidalgo

Ejemplo 1

Mediante AG optimizar la función:

$$\begin{aligned} &\min f(x) = x^2 \\ &\text{Sujeta a:} \\ &-5 < x < 5 \end{aligned} \tag{5}$$

Usar una configuración de 20 individuos, con una longitud del cromosoma de 10 bits y una probabilidad de mutación del 2%

Universidad Autónoma del Estado de Hidalgo

Ejemplo 1, Programa AG1.m

```
1  % ALGORITMO GENETICO BINARIO
2  % Optimizacion de una funcion en R^2, min f(x) = x^2
3  % Intervalo de busqueda -5 ≤ x ≤ 5
4  clear all
5  clc
6  % N = Numero de individuos, L= Longitud del cromosoma.
7  N = 20; L = 12;
8  V = 3;            % Numero de variables independientes.
9  Epsilon = 1e-8; % Maximo error de la optimizacion
10 prob = 2;       % Porcentaje de mutacion
11 ITER_MAX = 100;
12 %Intervalos de busqueda
13 I = [-5 5; -5 5; -5 5];
14 %Generemos la poblacion de individuos
15 Pob_2 = Poblacion(N,L,V);
16 %Poblacion en base decimal
17 Pob_10= PobDec(Pob_2,L,V);
18 %Poblaciones en R
19 Pob_real = Escalamiento(Pob_10,I,V,L);
20 cont = 1;
21 ff_aux = 1000;
```

Universidad Autónoma del Estado de Hidalgo

Ejemplo 1, Continuación ... AG1.m

```
1  %Continuacion de AG1
2  while ((cont ≤ ITER_MAX) && (ff_aux > Epsilon))
3      %Determinamos la funcion costo
4      ff = (Pob_real).^2;
5      %Graficar los individuos
6      graficar(Pob_real,ff);
7      %Seleccionamos la poblacion
8      PobSel_2 = Seleccion(Pob_2,ff);
9      %Cruzamos la Pob seleccionada
10     Pob_C = Cruce(PobSel_2,L,V);
11     %Concatenamos PobSel + Pob_C
12     Pob_2 = [PobSel_2;Pob_C];
13     %Mutamos la nueva poblacion
14     Pob_2 = Mutacion(Pob_2,L,prob);
15     %Poblacion en base decimal
16     Pob_10= PobDec(Pob_2,L,V);
17     %Poblacion en R
18     Pob_real = Escalamiento(Pob_10,I,V,L);
19     [ff_aux, idy] = min(ff)
20     cont = cont+1
21 end
```

Pruebas

- Realizar las evoluciones del AG con N = 20, L= 10, V=1, ϵ = 0.00001, prob = 2, $ITER_{MAX} = 100$
- Determine cuál es la opción por la que se detiene el AG
- ¿Cuales son los valores alcanzados en la población cuando termina el programa?
- ¿Cuales serán los valores en la población después de la ejecución del AG, para L = 12, 14,16 y 18 bits, use ϵ = 1e-12
- Mencione las limitaciones del AG binario.

Universidad Autónoma del Estado de Hidalgo

Pruebas

- Cambie el método de selección por el método del torneo
- Realice algunas pruebas con diferentes longitudes de bits en el cromosoma.
- Mencione si encuentra algunas ventajas el usar este tipo de método. Sugerencia: Use tic-toc

Ejemplo 2 Función Ackley

Optimizar mediante AG la función Ackley para para dos variables independientes.

$$f(x_0 \cdots x_n) = -20exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - exp(\frac{1}{n}\sum_{i=1}^{n} cos(2\pi x_i)) + 20 + e$$

$$-5 \leq x_i \leq 5$$

El mínimo está en $f(0, \cdots, 0) = 0$

Universidad Autónoma del Estado de Hidalgo

Ejemplo 2 Programa AG2

- Usar una población de 60 individuos, cromosomas con longitud de 12 bits, 2 variables independientes, probabilidad de mutación del 1 % e intervalo de búsqueda de $[-5, 5]$.

Universidad Autónoma
del Estado de Hidalgo

Ejemplo 2 Programa AG2

```
% ALGORITMO GENETICO BINARIO
% Optimizacion de la funcion Ackley
% Intervalo de busqueda -5 ≤ x ≤ 5
clear all
clc
% N = Numero de individuos, L= Longitud del cromosoma.
N = 60; L = 19;
V = 2;            % Numero de variables independientes.
Epsilon = 1e-6; % Maximo error de la optimizacion
prob = 1;       % Porcentaje de mutacion
ITER_MAX = 50;
%Intervalos de busqueda
I = [-5 5;-5 5];
%Generemos la poblacion de individuos
Pob_2 = Poblacion(N,L,V);
%Poblacion en base decimal
Pob_10= PobDec(Pob_2,L,V);
%Poblaciones en R
Pob_real = Escalamiento(Pob_10,I,V,L);
cont = 1;
ff_aux = 1000;
```

Universidad Autónoma del Estado de Hidalgo

Limitaciones de los AG Binarios

- La magnitud del error de la función costo están en función de la longitud del cromosoma.
- La longitud del cromosoma no puede ser mayor a 18 bits, restricción del comando bin2dec().
- Difícil alcanzar una épsilon menor a 1e-6
- Tratar de optimizar una función de dimensión 30 haría ocupar bastante memoria en el arreglo para la población y consumir tiempo computacional.
- Los operadores de conversión y decodificación no hay forma de evitarlos.

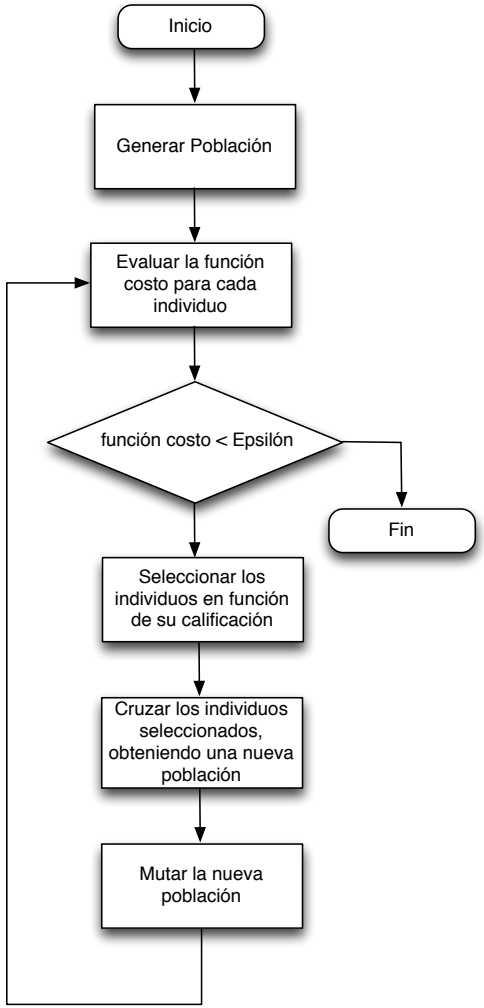
Universidad Autónoma del Estado de Hidalgo

Justificación para usar AG con números reales

- Las limitaciones anteriores quedan remediadas con los Algoritmos Genéticos Reales AGR
- La población se pone directamente en el conjunto de los reales
- La selección se aplica de forma igual que en los binarios
- Existe una variante para realizar el cruce de individuos usando números reales
- Existe una variante para realizar la mutación sobre una población usando números reales

Universidad Autónoma del Estado de Hidalgo

Diagrama de flujo de un AGR



Universidad Autónoma del Estado de Hidalgo

Población Aleatoria Real

- Se genera una matriz de elementos aleatorios entre cero y uno
- Se escala la población en el intervalo de búsqueda.
- En esta población se trabaja directamente con los operadores de cruce y mutación.

Población Aleatoria en Matlab

```
1  function [P] = Poblacion(N,L,V)
2  %N = Numero de individuos
3  %L = Longitud del cromosoma
4  %V = Numero de variables
5     P = round(rand(N,L*V));
6  end
```

Evaluar el desempeño de los individuos (función Ackley)

```
1  function [ PobSel ] = Seleccion(Pob_2,ff)
2  %Obtener tamano de la poblacion
3    [R C]= size(Pob_2);
4  %Ordenar de menor a mayor
5    [B,idx]= sort(ff);
6    Pob_2 = Pob_2(idx,:);
7  %Seleccionar la mitad de
8  %la mejor poblaciÛn
9    PobSel = Pob_2(1:R/2,:);
10 end
```

Función selección en Matlab

```
1  function [ PobSel ] = Seleccion(Pob_2,ff)
2  %Obtener tamano de la poblacion
3    [R C]= size(Pob_2);
4  %Ordenar de menor a mayor
5    [B,idx]= sort(ff);
6    Pob_2 = Pob_2(idx,:);
7  %Seleccionar la mitad de
8  %la mejor poblaciÛn
9    PobSel = Pob_2(1:R/2,:);
10 end
```

Operador cruce para el AGR

- Suponga un intervalo representado por dos números reales $[a, b]$, donde $a < b$.
- El cruce consiste en generar dos individuos entre el intervalo $[a, b]$, los cuales se pueden obtener de la siguiente forma;

$$hijo_1 = a + rand()[b - a] \tag{6}$$

-

$$hijo_2 = b - rand()[b - a] \tag{7}$$

Función cruce en Matlab

```
1  function [ Pob_C ] = CruceR(Pob,V )
2  [R C]=size(Pob);     Pob_C = zeros(R,C);
3  for i1 = 1:V
4      t = randperm(R);
5      f = reshape(t,R/2,2);
6      idx = 1;
7   for i2 = 1:R/2
8          beta1 = rand();
9          beta2 = rand();
10         aux1 = Pob(f(i2,1),i1) + ...
               beta1*(Pob(f(i2,2),i1)-Pob(f(i2,1),i1));
11         aux2 = Pob(f(i2,2),i1) - ...
               beta2*(Pob(f(i2,2),i1)-Pob(f(i2,1),i1));
12         Pob_C(idx,i1) = aux1;
13         Pob_C(idx+1,i1) = aux2;
14         idx = idx+2;
15 end
16 end
17 end
```

Universidad Autónoma del Estado de Hidalgo

Operador mutación para AGR

Sea $I = [I_{min}, I_{max}]$ un intervalo de búsqueda, donde se generará un individuo de forma aleatoria en este intervalo.

$$X_{mutado} = rand[I_{max} - I_{min}] + I_{min} \tag{8}$$

Universidad Autónoma
del Estado de Hidalgo

Función mutacion en Matlab

```
1  function [ Pob ] = Mutacion( Pob,I,mut)
2  [R C] = size(Pob);
3
4  NumMut = round(mut*R*C/100.0);
5  for i = 1: NumMut
6      ix = ceil(C*rand);
7      iy = ceil(R*rand);
8      Pob(iy,ix) = I(ix,1) + rand()*(I(ix,2)-I(ix,1));
9  end
10 end
```

Ejemplo 3

Optimizar la función Ackley mediante un AGR para una dimensión dos en un intervalo de búsqueda de $[-5, 5]$. Usar N $= 100$, $\epsilon = 1e - 3$, $ITER_{MAX} = 100$, porcentaje de mutación del 5%,

Código AGR1

```
%ALGORITMO GENETICO REAL
%JULIO DE 2015
clear all;  clc;   format long
N = 52;       %Numero de individuos
V = 30;       %Numero de variables independientes
Epsilon = 1e-20; % Maximo error de la optimizacion
ITER_MAX = 500; %Maximo numero de iteraciones
%Intervalo de busqueda
mut = 1; %Porcentaje de mutacion
%I =  [-5 5; -5 5]%Generamos la poblacion en el
%intervalo de busqueda
F_index = 1;     %Numero de funciÛn
I = [-100 100;-100 100;-100 100;-100 100; -100 100;-100 ...
    100;-100 100;-100 100;-100 100;-100 100; ...
     -100 100;-100 100;-100 100;-100 100; -100 100;-100 ...
        100;-100 100;-100 100;-100 100;-100 100; ...
     -100 100;-100 100;-100 100;-100 100; -100 100;-100
        100;-100 100;-100 100;-100 100;-100 100];

PobReal = Poblacion(N,V,I);
```

Universidad Autónoma del Estado de Hidalgo

Código AGR1

```
1   %Seleccionamos la poblacion
2       PobSel = Seleccion(PobReal,ff);
3       %Cruzamos la Pob seleccionada
4       Pob_C = CruceR(PobSel,V);
5       %Concatenamos PobSel + Pob_C
6       PobReal = [PobSel;Pob_C];
7       %Mutamos la nueva poblacion
8       PobReal = Mutacion(PobReal,I,mut);
9       cont = cont +1
10      pause;
11  end
12  %Se imprime la poblacion
13  PobReal
14  %Se imprime la funciÛn costo
15  ff = Ackley(PobReal)
```

Pruebas

- Adicione al programa anterior AGR1.m la graficación de la función costo para el intervalo de evolución.
- Adicione al programa anterior AGR1.m una nueva variable para controlar repetir el AGR, ponga esta variable en 50, para obtener 50 desempeños del AGR con la función Ackley.
- Diseñe una función selección tipo torneo e intercambie por la función selección().
- Compare el desempeño del AGR usando un promedio y una desviación estándar de la función costo.

Aplicación (Sintonización de un control PID)

- Sea un sistema dinámico representado por una ecuación diferencial ordinaria lineal de segundo orden cuya función de transferencia está definida por:

$$\frac{Y(s)}{R(s)} = \frac{w_n^2}{s^2 + 2\xi w_n s + w_n^2} \tag{9}$$

Donde $Y(s)$ es la salida de la planta y $R(s)$ es la función entrada o excitatriz, ξ es el amortiguamiento del sistema y w_n es la frecuencia natural no amortiguada.

- El sistema puede tener tres tipos de comportamiento, críticamente amortiguado, subamortiguado y sobreamortiguado, estas respuestas dependen del valor de ξ.

Universidad Autónoma del Estado de Hidalgo

Aplicación (Sintonización de un control PID)

- Se desea controlar el sistema de segundo orden mediante un PID

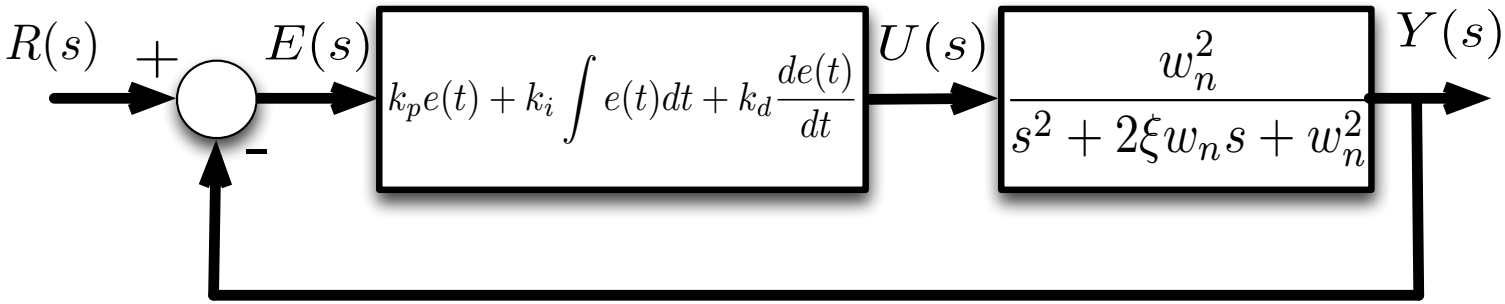


Figura: Función de transferencia en lazo cerrado del sistema de segundo orden con un control PID

Universidad Autónoma del Estado de Hidalgo

Aplicación (Sintonización de un control PID)

- La ecuación en el tiempo del control PID

$$u(t) = kpe(t) + ki \int e(t)dt + kd\frac{de(t)}{dt} \tag{10}$$

Donde $u(t)$ es la acción de control y $e(t)$ es la desviación de la salida del sistema dinámico respecto a una referencia definida.

- Aplicando Laplace obtenemos la función de trasferencia

$$\frac{U(s)}{E(s)} = \frac{kds^2 + kps + ki}{s} \tag{11}$$

Aplicación (Sintonización de un control PID)

- Diagrama del sistema dinámico en lazo cerrado con un control PID

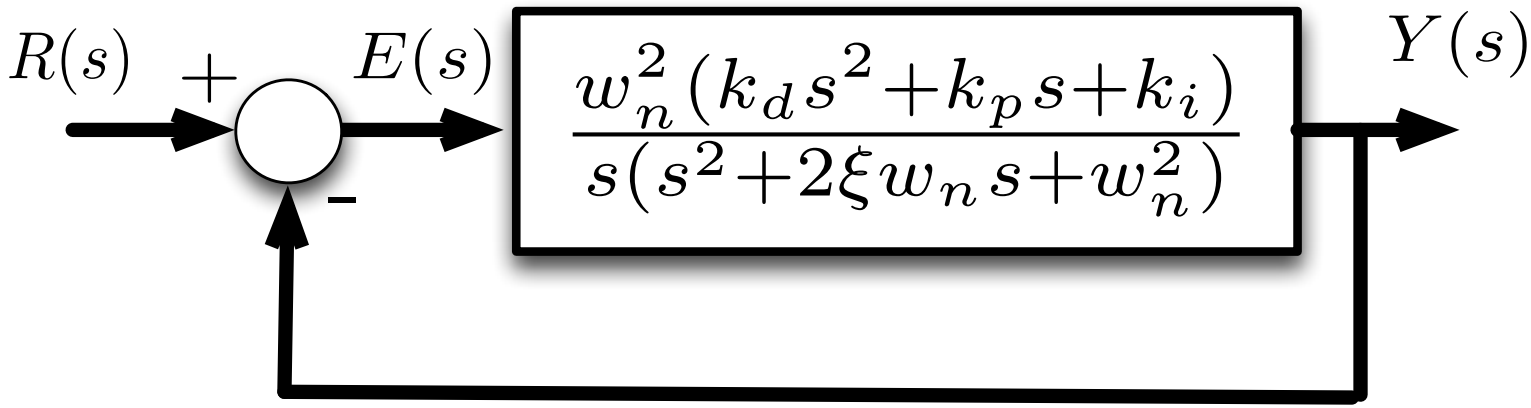


Figura: Función de transferencia en lazo cerrado del sistema de segundo orden con un control PID

Universidad Autónoma del Estado de Hidalgo

Aplicación (Sintonización de un control PID)

- La función de transferencia en lazo cerrado esta determinada por

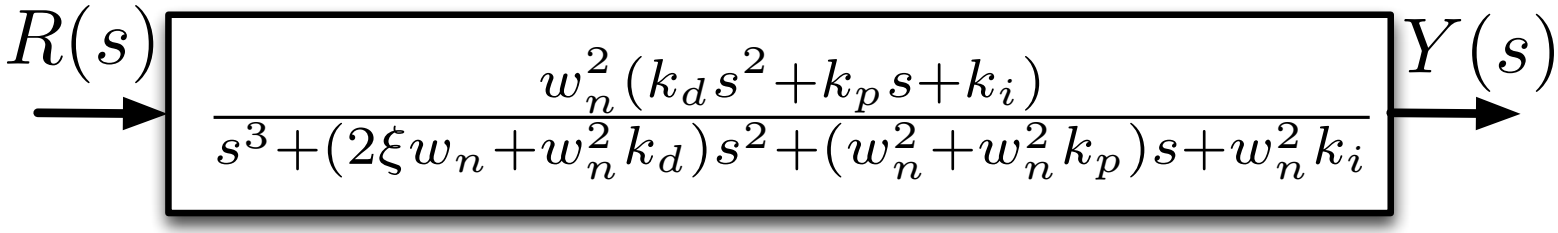
$$\frac{Y(s)}{R(s)} = \frac{w_n^2(k_d s^2 + k_p s + k_i)}{s^3 + (2\xi w_n + w_n^2 k_d)s^2 + (w_n^2 + w_n^2 k_p)s + w_n^2 k_i} \tag{12}$$

El comportamiento dinámico de la función de transferencia anterior esta determinado por la ubicación de polos de la ecuación característica.
Se puede observar en la Ec. (19) que dichos polos se pueden elegir mediante una asignación correcta de las ganancias del controlador.

Universidad Autónoma del Estado de Hidalgo

Aplicación (Sintonización de un control PID)

- Se define el comportamiento del sistema dinámico mediante una función $Y_d(s)$



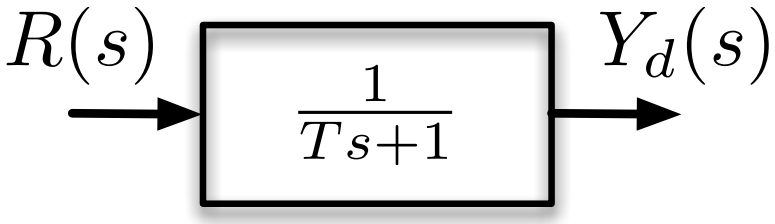


Figura: $Y_d(s)$ es la respuesta de un sistema de primer orden, $Y(s)$ es la respuesta del sistema de segundo orden en lazo cerrado con un PID

Universidad Autónoma del Estado de Hidalgo

Aplicación (Sintonización de un control PID)

- Se pone un intervalo de simulación del sistema en lazo cerrado t = [0,2]
- Se define la función costo por:

$$fcosto = min \sum_{t=0}^{2} (Y_d(t) - Y(t, k_p, k_i, k_d))^2 \tag{13}$$

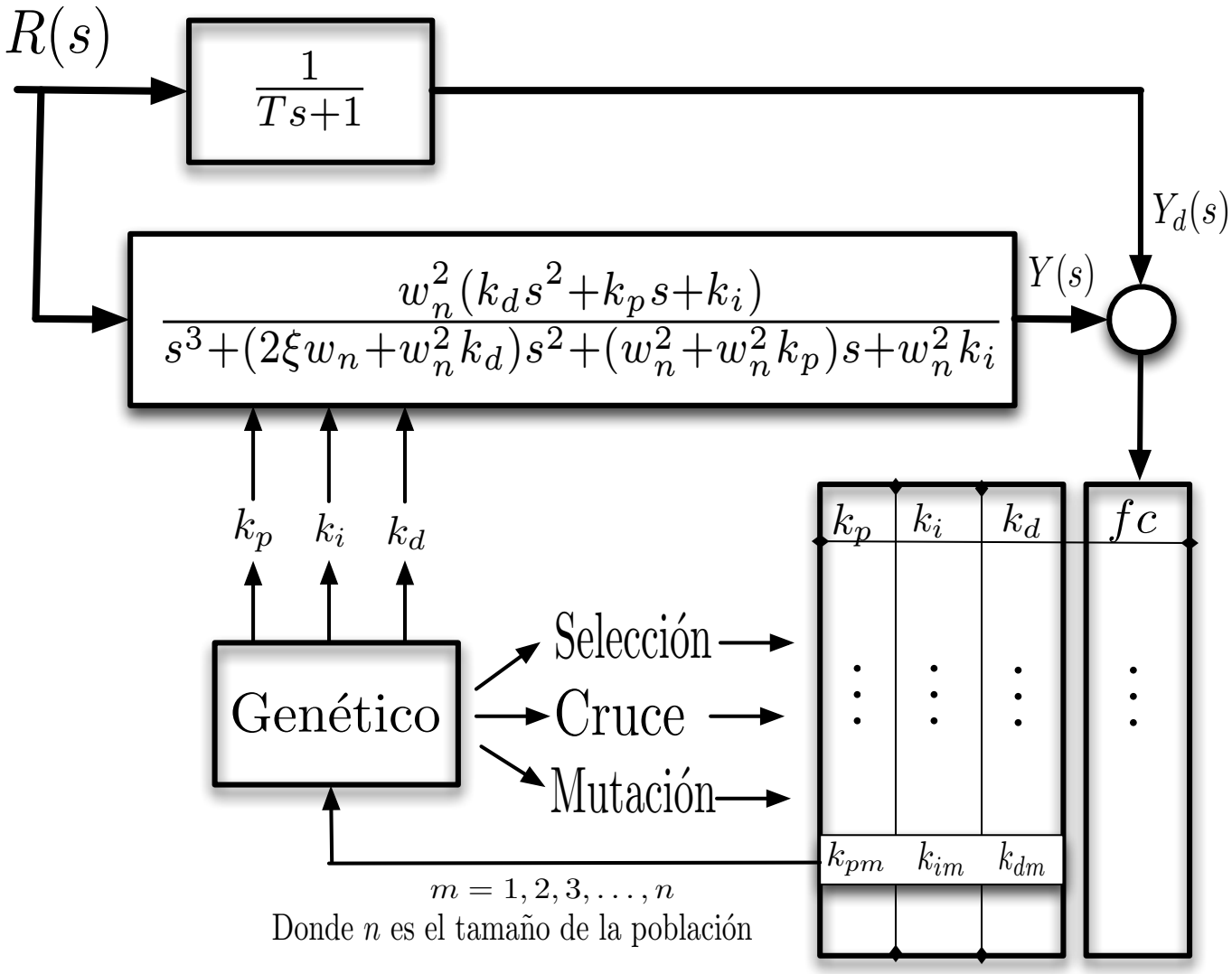
- Sujeta a las restricciones

$$k_p, K_i, K_d > 0 \tag{14}$$

Universidad Autónoma del Estado de Hidalgo

Aplicación de los AG al diseño de controles PID

- Sintonización de un PID con un AG



Aplicación (mainPID.m)

```
%SintonizaciÛn de un control PID
%mediante un algoritmo genÈtico
clear all;  clc;   format long
tic
N = 200;       %Numero de individuos
V = 3;       %Numero de variables independientes
ITER_MAX = 20; %Maximo numero de iteraciones
mut = 5; %Porcentaje de mutacion
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Parametros del sistema din·mico
tspan = 0:0.01:4; %Tiempo de simulaciÛn
w = 10; z = 0.5; %parametros del sistema de
                  %segundo orden
T1 = 0.2;         % Tiempo de retardo del sistema
                  % primer orden
%Intervalo de busqueda
%I =  [0.4 0.55; 4 6;0 0.1];
I =  [0.4 0.6; 4 6;0 0.08];
w1 = 1;
z1 = 0.5;
```

Universidad Autónoma del Estado de Hidalgo

Pruebas

- Sintonice el controlado PID para los siguientes modelos. Determine el tiempo de cómputo y la función costo. Valide con Simulink

$$\frac{Y_d(s)}{R(s)} = \frac{1}{s+1} \tag{15}$$

$$\frac{Y_d(s)}{R(s)} = \frac{2}{s+2} \tag{16}$$

$$\frac{Y_d(s)}{R(s)} = \frac{4}{s+4} \tag{17}$$

$$\frac{Y_d(s)}{R(s)} = \frac{1}{s^2+s+1} \tag{18}$$

Universidad Autónoma del Estado de Hidalgo

Aplicación (Identificación de parámetros de la máquina de inducción con AG)

La placa de datos y el catálogo del fabricante de motores de inducción especifican lo siguiente:

- Potencia mecánica nominal (P_m)
- Voltaje de línea nominal (V_l)
- Frecuencia (f)
- Factor de potencia (fp)
- Eficiencia (η)
- Torque máximo (T_m)
- Torque de arranque (T_a)
- Torque nominal (T_n)

Universidad Autónoma del Estado de Hidalgo

Aplicación (Identificación de parámetros de la máquina de inducción con AG)

Con los datos del fabricante hay que estimar los parámetros del circuito equivalente que describen al motor de inducción.

- Resistencia del estator R_s
- Reactancia de dispersión del estator X_s
- Resistancia de rotor R_r
- Reactancia de dispersión del rotor Xr
- Reactancia de magnetización X_m

Universidad Autónoma del Estado de Hidalgo

Aplicación (Identificación de parámetros de la máquina de inducción con AG)

- La relación entre datos del fabricante y parámetros del circuito equivalente está determinada por medio de las ecuaciones de torque y factor de potencia.

- De acuerdo a [Chapman:2012] la ecuación general de torque en la máquina de inducción esta definida de la siguiente forma:

$$T = \frac{3V_{Th}^2 * R_2/s}{w_s[(R_{Th} + R_2/s)^2 + (X_{Th} + X_2)^2]} \tag{19}$$

Universidad Autónoma del Estado de Hidalgo

Aplicación (Identificación de parámetros de la máquina de inducción con AG)

La función objetivo es:

$$Min\ F(X_i) = \left(\frac{\hat{T}_{max} - T_{max}}{T_{max}}\right)^2 + \left(\frac{\hat{T}_{nom} - T_{nom}}{T_{nom}}\right)^2 + \left(\frac{\hat{T}_{arr} - T_{arr}}{T_{arr}}\right)^2 + \left(\frac{\hat{fp} - fp}{fp}\right)^2 \quad (20)$$

Sujeta a las restricciones

$$X_{i,min} < X_i < X_{i,max} \quad (21)$$

$$R_1, R_2, X_1, X_2, X_m > 0 \quad (22)$$

$$X_1 = X_2 \quad (23)$$

UNIVERSIDAD AUTÓNOMA DEL ESTADO DE HIDALGO

Gracias por su atención

Juan Carlos Seck Tuoh Mora (jseck@uaeh.edu.mx)
Joselito Medina Marín (jmedina@uaeh.edu.mx)
Norberto Hernández Romero (nhromero@uaeh.edu.mx)

Universidad Autónoma
del Estado de Hidalgo