



UNIVERSIDAD DE CANTABRIA

FACULTAD DE CIENCIAS

**DEPARTAMENTO DE MATEMÁTICA APLICADA Y
CIENCIAS DE LA COMPUTACIÓN**

**MASTER OFICIAL UNIVERSITARIO EN MATEMÁTICAS Y
COMPUTACIÓN**

TRABAJO DE FIN DE MASTER:

**RECONSTRUCCIÓN DE CURVAS B-SPLINE Y DE
BÉZIER MEDIANTE SELECCIÓN CLONAL**

**PRESENTADO POR:
Andreina Avila Torres**

**DIRIGIDO POR:
Andrés Iglesias Prieto
Akemi Galvez Tomida**

Santander, 6 de Julio de 2012

Índice

1. Introducción	3
1.1. El Problema de la Reconstrucción.....	3
1.2. Reconstrucción de Curvas de Forma Libre.....	5
1.3. Sistemas Inmunológicos Artificiales.....	6
1.4. Organización y Estructura de esta Memoria.....	8
2. El Sistema Inmunológico	9
2.1. Cronología de la Inmunología.....	11
2.2. Propiedades de los Sistemas Inmunológicos Artificiales.....	13
3. La Teoría de la Selección Clonal	16
3.1. El algoritmo de Selección Clonal (ClonalG).....	17
3.2. Medición de la Afinidad Ab y Ag.....	18
3.3. Descripción del Modelo.....	19
3.4. El Pseudocódigo.....	20
4. Las Curvas B-spline	23
4.1 Representación General de las Curvas B-spline.....	23
4.2 El Vector de Nodos.....	24
4.3. El Algoritmo ClonalG y las Curvas B-spline.....	25
4.4 Complejidad Computacional del ClonalG.....	27
5. Las Curvas de Bézier	28
5.1. El algoritmo ClonalG y las Curvas de Bézier.....	30
6. Resultados	32
7. Detalles de Implementación	45
8. Conclusiones	46
8.1. Publicaciones de este Proyecto.....	47
9. Referencias	48

1. Introducción

1.1. El Problema de la Reconstrucción

Uno de los efectos más visibles del actual proceso de globalización es la creciente competencia a nivel mundial entre las empresas y agrupaciones industriales por ofrecer productos a menor coste y mayor calidad. Como consecuencia, se busca optimizar los procesos y reducir el ciclo de vida de desarrollo de un producto. Una herramienta básica en este proceso es el manejo de datos digitales, los cuales permiten una mejor manipulación, almacenamiento y transmisión. Y es aquí donde la *Ingeniería Inversa*, entendida como la transformación de objetos reales en sus modelos digitales, juega un papel realmente crucial. Las ventajas de la ingeniería inversa son obvias: el modelo digital puede ser usado incluso cuando el objeto físico que representa ya no existe o no se encuentra disponible o utilizable. Además, los modelos digitales son más fáciles de manipular y analizar que los objetos reales.

La tecnología más habitual en ingeniería inversa es el escaneado por láser, el cual genera conjuntos masivos de puntos de datos ruidosos que son posteriormente usados para reconstruir un modelo matemático del objeto real en términos de curvas y superficies de forma libre (generalmente *B-spline* y *NURBS*). Este proceso, que recibe el nombre de *reconstrucción de curvas y superficies*, presenta enormes ventajas competitivas, pues el modelo digital permite representar una forma muy compleja compuesta de cientos de miles de puntos de forma altamente económica, con sólo unas pocas decenas de parámetros (aquellos de las curvas y superficies que describen el objeto). Este proceso presenta una gran relevancia en muchos procesos de producción, manufactura y CAD/CAM (en campos como el sector de la automoción, el diseño y construcción de barcos, la industria aeroespacial, la industria del calzado, aplicaciones en medicina como la tomografía computarizada o la resonancia magnética, en la reconstrucción de restos arqueológicos, etc.).

Las Figuras 1 y 2 ilustran dos ejemplos de casos reales de procesos de escaneado: en el primer caso, aplicado a la industria del automóvil, y en el segundo caso a la restauración de objetos arqueológicos.



Figura 1: Ejemplo de escáner 3D aplicado a la industria del automóvil (*Fuente:* Nvision Inc., <http://www.nvision3d.com>).



Figura 2: Ejemplo de escáner 3D aplicado a la restauración arqueológica: (izquierda) proceso de escaneado; (derecha) figura obtenida por computador. (*Fuente:* Cyberware Inc., <http://www.cyberware.com>).

Hasta la fecha, se han alcanzado muy buenos resultados en la reconstrucción de piezas usando modelos poligonales (mallas). Ésta es una elección muy popular, que da lugar a modelos flexibles, rápidos de manipular y fáciles de dibujar, pues aprovechan muy bien las ventajas de las potentes tarjetas gráficas actuales. Pero las mallas poligonales son

meras aproximaciones planas, nunca una representación fiel de una superficie suave. Si buscamos la máxima calidad, debemos recurrir necesariamente a los modelos paramétricos de forma libre. Existe, sin embargo, una limitación clave en este proceso: la reconstrucción automática de curvas y superficies paramétricas es un problema muy difícil y aún sin resolver de forma general.

Algunos trabajos previos han mostrado que el problema puede abordarse de forma más exitosa mediante la aplicación de algunas técnicas de inteligencia artificial y soft computing (Gu & Yan, 1995), (Hoffmann, 2005), (Iglesias & Gálvez, 2001), (Iglesias et al., 2004), (Knopf & Kofman, 1999), (Kumar et al., 2004), (Gálvez & Iglesias, 2011), (Gálvez & Iglesias, 2012a), (Gálvez & Iglesias, 2012b). Por ello, hemos adoptado dicha aproximación en el trabajo reportado en la presente memoria.

1.2. Reconstrucción de Curvas de Forma Libre

En este contexto, nos proponemos en este trabajo la resolución del problema de la reconstrucción de curvas a partir de un conjunto de puntos dato, obtenidos (posiblemente aunque no necesariamente) mediante un proceso de escaneado. La reconstrucción de curvas es, en principio, más sencilla que la de superficies. Por ello, constituye un banco de pruebas muy útil para probar nuestros métodos. Pero este problema presenta además un interés intrínseco, que no se limita a ser una mera primera fase de la reconstrucción de superficies, sino que posee su propia identidad, pues las curvas son primitivas básicas de diseño, con amplias aplicaciones también en el campo de la producción.

Un aspecto interesante es que las curvas permiten abordar de forma más adecuada los distintos sub-problemas asociados al problema general de reconstrucción, como es la reconstrucción del vector de nodos (ver sección 4 para más detalles). El problema a resolver consiste en la determinación de vectores de nodos óptimos para el ajuste de los datos. Es conocido que una buena elección de los nodos es esencial para obtener una buena aproximación: una elección inadecuada de los nodos puede llevar a generar

formas impredecibles o inaceptables para diseño (Burchard, 1974), (Dierckx, 1993). Por ello, se han descrito varias estrategias en las últimas décadas para determinar buenos vectores de nodos. Dichos métodos pueden agruparse en dos tipos: nodos fijos y nodos libres. En el primer caso, los nodos son calculados de acuerdo a algún criterio pre-establecido (Piegl & Tiller, 1997), (Jing & Sun, 2005), (Park, 2004), (Park & Lee, 2007), (Wang et al., 2006), (Yang et al., 2004), de forma que el problema de ajuste por mínimos cuadrados se convierte en un problema lineal resoluble por diversas técnicas de ajuste paramétrico, como *mínimos cuadrados*.

En el segundo caso, los nodos son variables libres del problema, el cual se transforma ahora en un problema fuertemente no-lineal, pues las funciones básicas son funciones no lineales de los nodos (de Boor & Rice, 1968a), (de Boor & Rice, 1968b), (Jupp, 1978), (Dierckx, 1993), (Lyche & Morken, 1987), (Lyche & Morken, 1988), (Powell, 1970), (Alhanaty & Bercovier, 2001), (Goldenthal & Bercovier, 2004). En este último caso, nos enfrentamos además con la posible existencia de muchos puntos estacionarios, haciendo muy difícil determinar cuándo se ha llegado a la elección óptima (Molinari et al., 2004), (Rice, 1969), (Lindstrom, 1999).

Además, muchos métodos actuales generan nodos múltiples redundantes, los cuales causan un comportamiento no suave de la forma resultante. Por ello, se han definido varios métodos que evitan ese problema, impidiendo nodos repetidos. Estos métodos tampoco son suficientemente precisos y resultan muy difíciles de aplicar en el caso general, pues no siempre ocurre que los nodos repetidos sean innecesarios y redundantes. Por ejemplo, las formas angulosas o discontinuas requieren necesariamente nodos múltiples. Por ello, muchos métodos actuales no son capaces de trabajar con funciones discontinuas o no diferenciables, lo cual limita sustancialmente el grado de aplicabilidad de dichas propuestas.

1.3. Sistemas Inmunológicos Artificiales

La propuesta innovadora (y principal aportación) de este Trabajo Fin de Master

consiste en aplicar un reciente paradigma de *Soft Computing* llamado *Sistemas Inmunológicos Artificiales* al problema de la elección de nodos para la reconstrucción de curvas *B-spline* a partir de nubes de puntos dato.

Los *Sistemas Inmunológicos Artificiales (SIA)* constituyen una serie de modelos del *Sistema Inmunológico* de los vertebrados que tratan de resolver problemas en los campos de la ciencia y tecnología de la información mediante algoritmos computacionales. Dichos modelos computacionales se centran en *Teorías Inmunológicas* como los algoritmos de *Selección Negativa*, *Redes Inmunológicas*, *Teorías de Peligro* y algoritmos de *Selección Clonal*.

Los algoritmos de *Selección Clonal* toman en cuenta la *Hipermutación* y mecanismos de expansión clonal (Cutello & Nicosia, 2005), donde la primera respuesta inmune sería tomada como la fase de entrenamiento y la respuesta secundaria ya mejorada sería la fase de prueba. De Castro y Von Zuben propusieron el *Algoritmo de Selección Clonal (ASC)* como una aplicación del principio de *Selección Clonal* para resolver problemas como el reconocimiento de patrones y optimización. Los mismos investigadores renombraron a dicho algoritmo como *ClonalG* (De Castro & Von Zuben, 1999), (De Castro, 2001).

Los principios biológicos de la *Teoría de la Selección Clonal*, como la *Maduración de Afinidad* de la respuesta inmunológica, reunidos en esta versión del algoritmo *ClonalG*, pueden resolver problemas comunes en el diseño de ingeniería, como los surgidos al tratar de ajustar una curva arbitraria a una serie de valores experimentales. El algoritmo implementado en este estudio busca solucionar problemas de optimización multimodal, específicamente aquellos relacionados con la reconstrucción de curvas *B-spline*, así como el ajuste mediante curvas de Bézier, según se describe en los capítulos 4 y 5 de esta memoria.

1.4. Organización y Estructura de esta Memoria

Esta memoria consta de 9 capítulos y está organizada de la forma siguiente: en el capítulo 2 se describe el funcionamiento del *Sistema Inmunológico* de los vertebrados, se relata una reseña sobre la cronología de la *Inmunología* y se definen los *Sistemas Inmunológicos Artificiales*. Seguidamente, en el capítulo 3 se enuncian los principios de la *Teoría de la Selección Clonal*, se analiza el algoritmo *ClonalG* derivado de los mismos, se describe el método empleado para la medición de la afinidad de los anticuerpos producidos y se explica el modelo computacional propuesto con su pseudocódigo, mientras que en el capítulo 4 se presenta una breve descripción de los principales conceptos matemáticos relacionados con las curvas *B-spline*, incluida la importante cuestión de la elección del vector de nodos, aspecto éste abordado en la sección 4.2 de esta memoria. También se explica cómo es aplicado el algoritmo *ClonalG* en la reconstrucción de las curvas *B-spline*. Por último en este capítulo se analiza también la complejidad computacional que presenta el algoritmo implementado.

En el capítulo 5 se presentan las definiciones matemáticas referidas a las curvas de *Bézier* y se narra cómo es aplicado el algoritmo *ClonalG* en la reconstrucción de las curvas de *Bézier*.

El capítulo 6 describe los resultados obtenidos después de la implementación del algoritmo *ClonalG* en la reconstrucción de curvas *B-spline* y de *Bézier*.

El capítulo 7 se ofrecen los detalles de implementación del método, seguido del capítulo 8, que recoge las principales conclusiones de este estudio. La memoria concluye con una selección de las referencias bibliográficas utilizadas en el mismo.

2. El Sistema Inmunológico

El *Sistema Inmunológico* de los vertebrados está compuesto por células y moléculas que trabajan en colaboración con otros sistemas, como el *Neurológico* y el *Endocrino*, para proteger al individuo de células propias del organismo que funcionen mal y de agentes infecciosos externos (como virus, hongos, bacterias, parásitos, etc.) llamados *patógenos* (De Castro & Von Zuben, 1999).

Existen dos tipos de inmunidad en el individuo y que trabajan en perfecta coordinación: la *innata* y la *adaptativa*. El individuo nace con la *Inmunidad Innata*, ésta no cambia durante la vida del mismo y es la primera barrera de protección del organismo (piel, tos, ácido estomacal, lágrimas, inflamación, etc.). Posee mecanismos efectores, como *fagocitos*, *enzimas de complemento* y *péptidos antimicrobianos*, que se activan para combatir cualquier *patógeno* que entra en contacto con el cuerpo. Dichas células contienen *Receptores de Reconocimiento de Patrones (RRP)* que buscan *Patrones Moleculares Asociados a Patógenos (PMAP)* presentes en la superficie de unas estructuras llamadas *antígenos* y éstos, una vez detectados, son neutralizados o eliminados (Floreano & Mattiusi, 2008).

La *Inmunidad Adaptativa o Adquirida* puede cambiar con el transcurso del tiempo y está formada por *Células Presentadoras de Antígenos (CPA)* como los *macrófagos*, que digieren a los *antígenos* y los fragmentan en péptidos antigénicos que son atajados por las *Moléculas de Histocompatibilidad Compleja (MHC)*. Los *linfocitos* llamados *células T* tienen receptores capaces de reconocer diferentes combinaciones de péptido-MHC que provocan la activación de dichas *células T*, las cuales seguidamente se dividen y secretan señales químicas (*linfoquinas*) que activan a las *células B*, otros *linfocitos* que además pueden detectar a los *antígenos* que no están unidos a las moléculas *MHC*. Cuando las *células B* son activadas, se diferencian en células de plasma que secretan proteínas de *anticuerpo* que se enlazan con los *antígenos* para desencadenar la destrucción del *antígeno* a través de *enzimas de complemento* (De Castro & Von

Zuben, 1999). El funcionamiento de la *Inmunidad Adaptativa* descrito anteriormente se muestra en la figura 3.

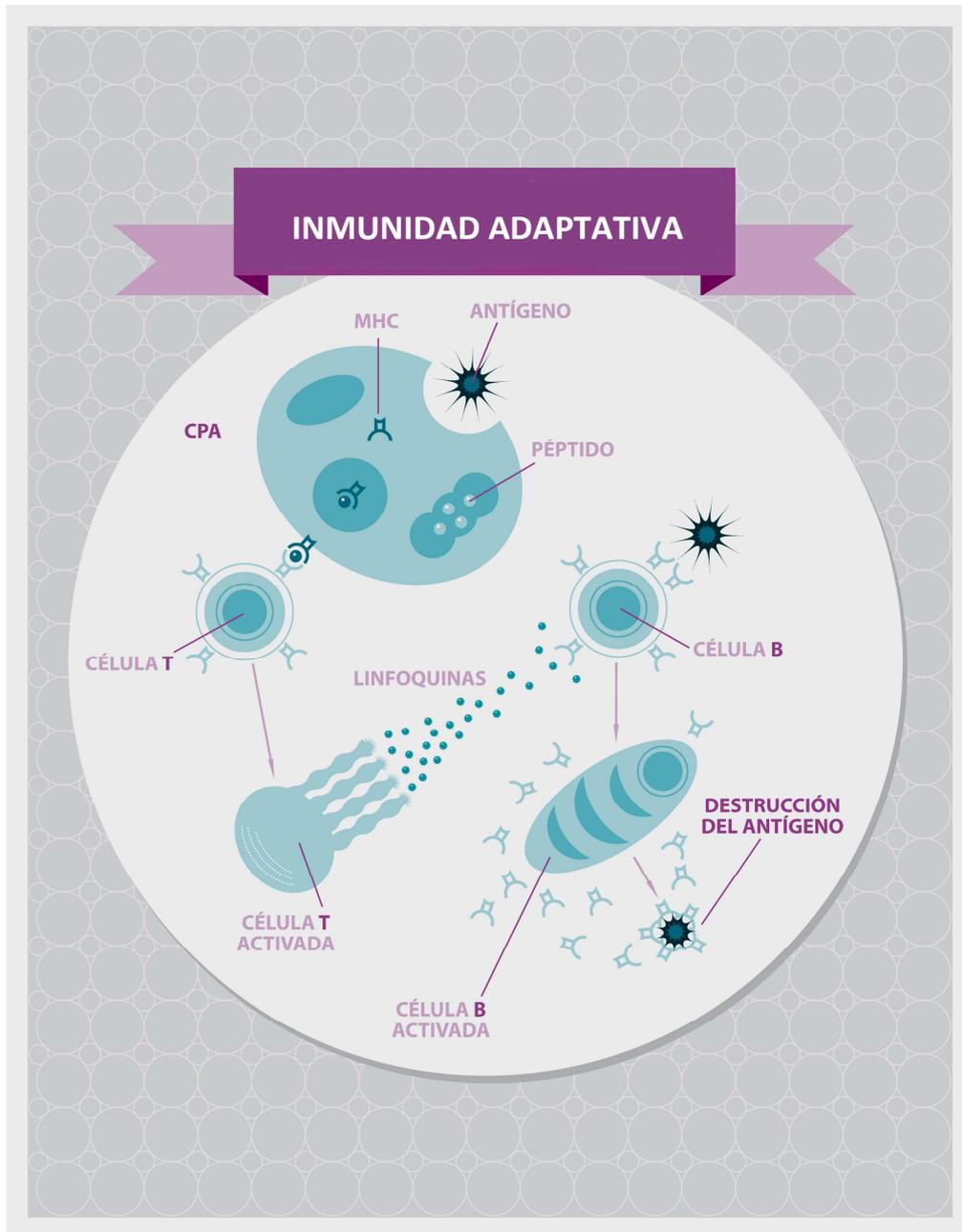


Figura 3: Funcionamiento de la *Inmunidad Adaptativa* (adaptada de De Castro & Von Zuben, 1999)

La *Inmunidad Adaptativa* presenta varias características que la diferencian de la *Inmunidad Innata*, por ejemplo:

- a) Las *células T* y *B* pueden cambiar sus receptores de reconocimiento al sufrir una *Hipermutación Somática*.
- b) También ocurre un proceso llamado *Maduración de Afinidad* donde se seleccionan las células que mejor atacaron al *patógeno*.
- c) Las *células T* y *B* más efectivas se convierten en células de memoria que perduran en la circulación y preparan al *Sistema Inmunológico* para futuras exposiciones al mismo *antígeno*. (Floreano & Mattiusi, 2008).

2.1. Cronología de la Inmunología

La *Inmunología* nació en 1796 gracias al descubrimiento de la *Vacuna Antivariólica* por el médico Edward Jenner. Dicho proceso de protección contra la viruela fue llamado *Vacunación*, que describe la inoculación de individuos sanos con ejemplares débiles o atenuados de los agentes que causan enfermedades. En el siglo XIX, Robert Koch probó que las enfermedades infecciosas eran causadas por microorganismos patógenos, y cada uno era responsable de cierta patología.

En la década de 1880, Louis Pasteur desarrolló la vacuna contra la *Cólera Aviar* y la *Vacuna Antirrábica*. Al mismo tiempo, el biólogo ucraniano Elie Metchnikoff descubrió la *Fagocitosis* como el proceso en el cual algunas células (*neutrófilos* y *macrófagos*) rodean con su membrana citoplasmática a un *antígeno* para degradarlo.

En 1890, Emil von Behring y Shibasaburo Kitasato encontraron que el suero sanguíneo de individuos inoculados contenía sustancias llamadas *anticuerpos*, que se enlazaban específicamente con los agentes infecciosos. Paul Ehrlich bacteriólogo alemán, ganador del Premio Nóbel de Medicina en 1908, junto a Elie Metchnikoff, estaba intrigado por el incremento excesivo de la producción de *anticuerpos* después de la exposición al *antígeno* y formuló la *Teoría de la Inmunidad de Cadena Lateral* que explica cómo los receptores de la parte externa de las células se combinan con toxinas para producir cuerpos inmunes capaces de combatir la enfermedad.

En 1900, Jules Bordet y Karl Landsteiner crearon el concepto de *Especificidad Inmunológica*, según la cual el *Sistema Inmunológico* es capaz de producir *anticuerpos* específicos contra químicos sintetizados artificialmente. Todas las teorías originadas durante el período 1930-1950 eran subcelulares, enfocadas en la biosíntesis de moléculas de *anticuerpo*. Dichos estudios llegaron a la conclusión de que el *antígeno* proporcionaba información celular sobre la estructura complementaria de la molécula de *anticuerpo*, dando lugar a la *Teoría Instruccionista de la Producción de Anticuerpos*, formulada por Breinl y Haurowitz, que luego fue desarrollada por el ganador del premio Nóbel Linus Pauling.

En los 20 años siguientes, estas teorías instruccionistas tuvieron menos seguidores debido a la aparición de las *Teorías de Selección*, como la *Teoría de la Selección Clonal*, propuesta por McFarlane Burnet. Otros ganadores del premio Nóbel hicieron grandes aportes entre 1970 y 1990, como Niels K. Jerne, con su *Teoría Reticular*, que describe al *Sistema Inmunológico* como una red autorregulada de *anticuerpos* y *linfocitos*, y Susumu Tonegawa, con el estudio de la estructura y diversidad de los receptores.

En los últimos años, el trabajo en *Inmunología* se centró en temas como la *Apoptosis*, la presentación de *antígeno*, las *citoquinas*, la regulación inmunológica, la memoria, las enfermedades autoinmunes, las vacunas de ADN, la señalización intracelular e intercelular y la maduración de la respuesta inmunológica (De Castro & Von Zuben, 1999). La tabla 1 resume la cronología de los principales conceptos formulados por los investigadores en el área de *Inmunología*.

Objetivo	Período	Investigadores	Conceptos
Aplicación	1796-1870	Jenner	Inmunización.
		Koch	Patología.
	1870-1890	Pasteur	Inmunización.
		Metchnikoff	Fagocitosis. Anticuerpos.
Descripción	1890-1910	von Behring & Kitasato	Teoría Inmunidad Cadena Lateral.
		Ehrlich	
	1910-1930	Bordet & Landsteiner	Especificidad.
Mecanismos	1930-1950	Breinin & Haurowitz	Síntesis de anticuerpos.
		Pauling	Teoría Instruccionista de Producción de Anticuerpos.
	1950-1980	Burnet	Teoría Selección Clonal.
		Jerne	Teoría Reticular.
Molecular	1980-1990	Tonegawa	Estructura y diversidad de receptores.

Tabla 1: Cronología de la *Inmunología* (adaptada de De Castro & Von Zuben, 1999)

2.2. Propiedades de los Sistemas Inmunológicos Artificiales

Los modelos de los sistemas humanos surgieron una vez que los científicos trataron de resolver problemas complejos inspirándose en la naturaleza. Al comienzo, dichos modelos fueron desarrollados utilizando redes neuronales artificiales, luego se emplearon algoritmos evolutivos basados en la *Teoría de la Evolución de Darwin* y

finalmente se crearon los *Sistemas Inmunológicos Artificiales (SIA)* generados a partir del modelado de los *Sistemas Inmunológicos Humanos*.

Los *SIA* presentan las mismas propiedades que la *Inmunidad Adaptativa*, ya que tratan de simular el comportamiento del *Sistema Inmunológico Humano* con la finalidad de resolver problemas computacionales en las áreas de matemática, ingeniería y tecnología de la información. Los *SIA* se caracterizan por poseer estructuras híbridas y algoritmos que copian mecanismos inmunológicos. Dichos algoritmos computacionales están basados en principios inmunológicos como la *Teoría de la Selección Clonal*, *Teorías de Redes Inmunológicas*, optimización basada en la inmunidad, aprendizaje, auto-organización, modelos cognitivos, reconocimiento de patrones y detección de anomalías (De Castro & Von Zuben, 1999).

Existen otras propiedades del *Sistema Inmunológico* que últimamente han despertado el interés de más científicos, ingenieros, matemáticos, filósofos y otros investigadores. A consecuencia de ello, el número de áreas de la investigación que implementan los principios inmunológicos humanos está en aumento. Según De Castro & Von Zuben dichas propiedades son:

- *Unicidad*: Cada individuo posee su propio *Sistema Inmunológico*, con sus características y vulnerabilidades particulares.
- *Reconocimiento de cuerpos extraños*: Las moléculas no naturales al cuerpo son reconocidas y eliminadas por el *Sistema Inmunológico*.
- *Detección de anomalías*: El *Sistema Inmunológico* puede detectar y reaccionar contra patógenos que nunca antes hayan entrado en contacto con el cuerpo.
- *Detección distribuida*: Las células del sistema están distribuidas por todo el cuerpo sin estar sujetas a ningún control centralizado.
- *Detección imperfecta*: No se requiere un reconocimiento absoluto de los patógenos, de ahí que el sistema sea flexible.

- *Refuerzo del aprendizaje y la memoria*: El sistema puede “aprender” las estructuras de los patógenos; así, las futuras respuestas a los mismos patógenos son más rápidas y fuertes.

Los *Algoritmos Genéticos (AG)* y los *Sistemas Inmunológicos Artificiales (SIA)* poseen componentes similares, sin embargo se ha comprobado que los *SIA* presentan grandes ventajas sobre los *AG* (O. Engin & A. Doyen, 2004). La propiedad más importante de los *SIA* es la velocidad de convergencia hacia la función objetivo que los convierte en una mejor alternativa frente a los *AG*.

3. La Teoría de la Selección Clonal

La *Selección Clonal* es un proceso empleado por el sistema inmunológico que describe las respuestas inmunológicas básicas ante un estímulo antigénico, basándose en la selección y proliferación de las células que reconocen a los *antígenos*. Este principio suele ocurrir en las *células T* y *B* (De Castro & Von Zuben, 1999).

La *Teoría de la Selección Clonal* (Burnet, 1978) contiene las siguientes premisas:

- Las nuevas células son copias (clones) de sus padres y están sometidas a mecanismos de mutación (*Hipermutación Somática*).
- La eliminación de los nuevos *linfocitos* diferenciados que portan receptores auto-reactivos (*Apoptosis*).
- La proliferación y diferenciación de las células maduras al entrar en contacto con el *antígeno* (*Maduración de Afinidad*).
- La persistencia de clones prohibidos, resistentes a la eliminación temprana por *antígenos* (*Memoria Inmunológica*).

En la *Hipermutación Somática* se sostiene que la proporción de la mutación es mayor para aquellos linfocitos cuyos *RRP* poseen una menor afinidad para los *antígenos* presentes, lo que significa que las células menos eficientes contra el *patógeno* son las que van a mutar. Debido a la naturaleza aleatoria de la mutación, algunas células se convierten en auto-reactivas, por lo tanto se recurre al proceso de *Apoptosis* (muerte celular) para eliminarlas eficazmente, como también a aquellas que presenten afinidad baja, ya que de esta manera no podrán formar parte del repertorio de células de memoria (Bereck & Ziegner, 1993).

La *Maduración de Afinidad* es un proceso donde se seleccionan las células mutadas más efectivas contra los *patógenos* para la futura activación de los detectores y efectores inmunológicos. Cuando la proporción de eliminación del *patógeno* sobrepasa

la proporción de replicación del mismo, el huésped es liberado de dicha invasión. Una vez que los *patógenos* han sido destruidos, las señales de peligro decaen, las células inmunológicas dejan de ser reactivadas y la mayoría son eliminadas por falta de presencia de células con *PMAP*, a excepción de una pequeña población de células de larga duración, formada por los detectores y efectores que mejor combatieron al *patógeno*. Dichos elementos constituyen la *Memoria Inmunológica*, permitiendo dar una rápida respuesta inmunológica contra el mismo *patógeno* si éste es encontrado nuevamente (Floreano & Mattiusi, 2008).

3.1. El Algoritmo de Selección Clonal (CLONALG)

El *Algoritmo de Selección Clonal (ClonalG)* es una implementación computacional basada en la *Teoría de la Selección Clonal*, englobando el proceso de *Maduración de Afinidad*. Es capaz de resolver problemas que implican aprendizaje automático, reconocimiento de patrones y optimización multi-modal (De Castro & Von Zuben, 1999).

En dicho algoritmo, los *anticuerpos* *Ab* y los *antígenos* *Ag* son representados como un conjunto de moléculas *m* de valores reales $m = \langle m_1, m_2, \dots, m_L \rangle$ vistos como un punto en el espacio real de dimensión *L*, ($Ab \in S_L \subseteq R_L$) donde *S* representa el *Espacio de Búsqueda*. El concepto de *Espacio de Búsqueda* *S* fue introducido por Perelson y Oster (Perelson & Oster, 1979) y representa el lugar ocupado por el repertorio completo de células inmunológicas donde ocurre la interacción entre éstas y los *antígenos*.

El grado de enlace entre un receptor y una molécula (*ligando*) envuelve un corto rango no covalente de interacciones basadas en cargas electrostáticas, enlaces de hidrógeno, *Interacciones de van der Waal*, etc. Las moléculas suelen aproximarse entre ellas sobre una apreciable porción de sus superficies y para ello debe haber extensas regiones de complementariedad. Estas distribuciones de forma, carga y grupos químicos en posiciones apropiadas de complementariedad son propiedades de los *antígenos* y *anticuerpos*, denominadas *Formas Generalizadas de las Moléculas* (Perelson, 1989). La

forma generalizada de un *anticuerpo* se llama *paratopo* y la del *antígeno*, *epitopo*. Si las formas de los *paratopos* y *epitopos* no son tan complementarias, entonces las dos moléculas se enlazarán pero con una afinidad menor. Se asume que cada *paratopo* interactúa específicamente con todos los *epitopos* situados en una pequeña región vecina dentro de S , llamada *Región de Reconocimiento*. Cada *anticuerpo* puede reconocer todos los *epitopos* dentro de una región de reconocimiento y un *antígeno* debe presentar varios tipos de *epitopos* diferentes, es por ello que un número finito de *anticuerpos* es capaz de reconocer un número infinito de *antígenos*.

Usualmente, los problemas a ser resueltos mediante el *Algoritmo de Selección Clonal* están representados a través de una codificación $Ag-Ab$ (binaria o de valores reales) y una medida de distancia (llamada *Afinidad*), usada para calcular el grado de interacción entre estas moléculas.

3.2. Medición de la Afinidad Ag y Ab

Para calcular matemáticamente la *Afinidad* entre Ag y Ab , se debe tomar en cuenta la distancia entre dos vectores utilizando la *Distancia Euclídea*. Si las coordenadas de un *anticuerpo* son dadas por $\langle Ab_1, Ab_2, \dots, Ab_L \rangle$ y las coordenadas de un *antígeno* son dadas por $\langle Ag_1, Ag_2, \dots, Ag_L \rangle$, entonces la distancia D entre éstos viene dada por la siguiente ecuación:

$$D = \sqrt{\sum_{i=1}^L (Ab_i - Ag_i)^2} \quad (1)$$

Los *Espacios de Búsqueda* que utilizan coordenadas de valor real y que miden la distancia D por la ecuación (1), se llaman *Espacios de búsqueda Euclídeos* (Segel & Perelson, 1988). Obviamente, un valor grande de D representa una menor similitud entre los dos vectores, mientras que el valor nulo simboliza una afinidad perfecta (Floreano & Mattiussi, 2008).

3.3. Descripción del Modelo

El *Algoritmo de Selección Clonal (ClonalG)* fue propuesto por De Castro y Von Zuben (De Castro & Von Zuben, 1999) y aplicado para resolver problemas de reconocimiento de patrones y optimización de funciones. Este algoritmo contempla la *Maduración de Afinidad*, que es un proceso iterativo de producción de clones, mutación y selección de mejores *anticuerpos*. Los mejores *anticuerpos* resultantes de este proceso de *Selección Clonal* son preservados como células de memoria.

Los principios inmunológicos tomados en cuenta en este algoritmo son:

- El mantenimiento de células de memoria.
- Selección y clonación de los individuos más estimulados.
- Muerte de las células no estimuladas (*Apoptosis*).
- *Maduración de Afinidad* y selección de los clones con mayor afinidad.
- Generación y mantenimiento de la diversidad.
- *Hipermutación* proporcional a la afinidad celular.

El algoritmo *ClonalG* considera dos colecciones de vectores, integrados por los *antígenos* Ag y los *anticuerpos* Ab . Se asume que Ab puede ser descompuesto en dos subconjuntos con r y m elementos llamados de *memoria* y *residual*, respectivamente, con $n = m + r$.

El algoritmo puede ser descrito de la forma siguiente:

- 1) Seleccionar aleatoriamente un *antígeno* Ag_i , $i \in \{1 \dots n\}$.
- 2) Calcular la afinidad $(Ag_i, Ab_j) = \Delta_{ij}$, $j \neq i$, $j \in \{1 \dots n\}$.
- 3) Seleccionar los p *anticuerpos* Ab_k con mayor afinidad Δ_{ik} .
- 4) Clonar los Ab_k , ($k = 1 \dots p$) proporcionalmente a Δ_{ik} para generar un conjunto K de

clones, con cardinalidad $N_c = \sum_{i=1}^p \left(\frac{\beta \times n}{i} \right)$, siendo β un factor de multiplicidad.

- 5) Someter a los elementos de K a un proceso de maduración inversamente proporcional a la afinidad.
- 6) Determinar la afinidad de los clones madurados con respecto del *antígeno* Ag_j .
- 7) Seleccionar los *anticuerpos* Ab_j^* con mayor afinidad como candidatos a entrar en la lista de los m Ab de *memoria*. Si $\Delta_{ij}^* > \Delta_{ik}$ ($k= 1, \dots, m$), entonces Ab_j^* reemplaza a Ab_k .
- 8) Finalmente reemplazar los d *anticuerpos* con menor afinidad en el subconjunto *residual* por nuevos anticuerpos Ab_l , ($l= 1, \dots, d$).

Cada ejecución de los pasos anteriores para todos los *antígenos* Ag dados representa una generación. El algoritmo es repetido para un cierto número de generaciones $Ngen$, parámetro que depende del problema específico analizado.

3.4. El Pseudocódigo

En el caso de optimización de funciones, el objetivo es producir una población de *anticuerpos* que constituye un conjunto de soluciones candidatas al problema. Para conseguir este fin, el algoritmo anterior se modifica de acuerdo a los pasos siguientes:

- 1) Generar aleatoriamente una población L de *anticuerpos* Ab .
- 2) Calcular la afinidad f de cada *anticuerpo* de Ab . La afinidad está descrita en términos de una función de fitness f .
- 3) Seleccionar los n *anticuerpos* Ab_j , ($j= 1, \dots, n$) con la afinidad más alta.
- 4) Clonar esos *anticuerpos* C en proporción a la afinidad.
- 5) Someter a los clones a un proceso de maduración de afinidad para generar una nueva población C^* que sufrirá una *Hipermutación Somática*.
- 6) Calcular la afinidad f^* de los clones mutados C^* .
- 7) Seleccionar los n mejores *anticuerpos* Ab_k $k= (1, \dots, n)$ e insertarlos en la población de *anticuerpos* Ab .
- 8) Sustituir los d *anticuerpos* de baja afinidad por nuevos *anticuerpos* e insertarlos en el repertorio de Ab .

La figura 4 muestra los pasos del pseudocódigo del *Algoritmo ClonalG* para resolver problemas de optimización según se explica a continuación:

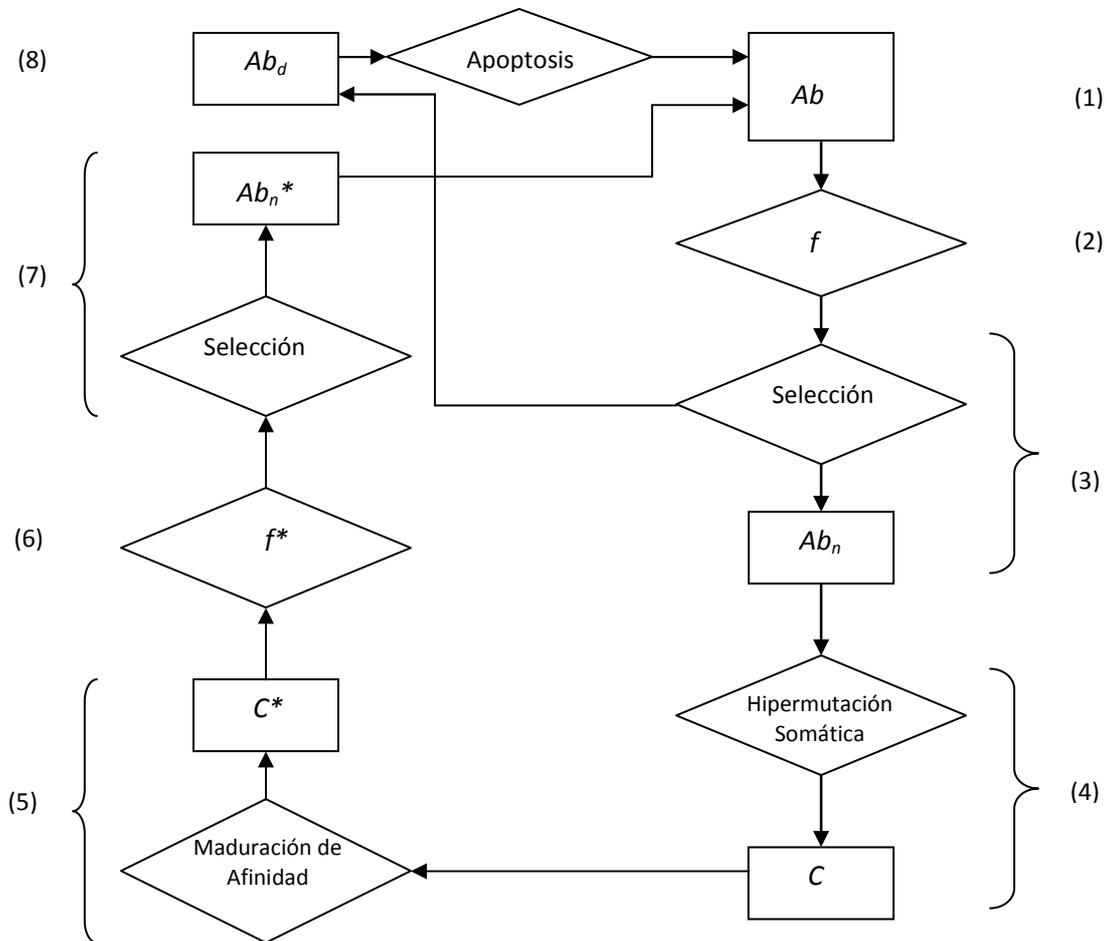


Figura 4: Diagrama de flujo del *Algoritmo ClonalG*

El paso 1 consiste en la generación del repertorio inicial de *anticuerpos* en forma aleatoria para propiciar la diversidad de la población. El paso 2 conlleva la medición de la afinidad de cada *anticuerpo* a través de una función de fitness. El paso 3 es importante para el buen desempeño del algoritmo, ya que los *anticuerpos* con mejor afinidad son seleccionados para la posterior clonación en el paso 4, donde el mejor *anticuerpo* dentro del ranking de afinidad tendrá una mayor cantidad de réplicas con respecto a los demás.

En el paso 5 los clones seleccionados son sometidos a una *Maduración de Afinidad* y dicho conjunto de mejores *anticuerpos* sufrirá una *Hipermutación Somática*, la cual se

traduce en provocar una perturbación en alguno de los componentes de cada clon. En el paso 6 se vuelven a calcular las afinidades de los clones mutados para luego re-seleccionar los mejores (paso 7) y formar un nuevo conjunto de individuos para ser insertados en el repertorio inicial de *anticuerpos*. En el paso 8 los *anticuerpos* de baja afinidad son eliminados y sustituidos por nuevos *anticuerpos* generados al azar.

Los parámetros de entrada y salida del pseudocódigo del *ClonalG* antes mencionado son los siguientes:

Entrada:

- L : número de *anticuerpos*.
- Ab : población inicial de *anticuerpos*.
- $iter$: número de iteraciones (generaciones) a realizar.
- n : número de *anticuerpos* seleccionados para clonación ($n > L$).
- d : número de *anticuerpos* seleccionados para sustitución ($d > L$).
- f : función de fitness.

Salida:

- Ab_{max} = *anticuerpo* óptimo.

Pseudocódigo:

```

for j=1 to iter do
  for i=1 to L do
     $f_i \leftarrow f(Ab_i)$  /* Paso 1*/
  end for
   $F \leftarrow \text{sort}(\{f_i, i=1, \dots, L\})$  /* Paso 2*/
   $Ab_{[n]} \leftarrow \{Ab_k / k=1, \dots, n\}$  /* Paso 3*/
   $C \leftarrow \text{clonación}(Ab_{[n]})$  /* Paso 4*/
   $C^* \leftarrow \text{hipermutación}(C)$  /* Paso 5*/
  for i= 1 to L do
     $f_i \leftarrow f(Ab_i^*)$  /* Paso 6*/
  end for
   $F^* \leftarrow \text{sort}(\{f_i^*, i=1, \dots, L\})$ 
   $Ab_{[n]}^* \leftarrow \{Ab_k^* / k=1, \dots, n\}$  /* Paso 7*/
   $Ab \leftarrow \text{insertar}(Ab, Ab_{[n]}^*)$ 
   $Ab_{[d]} \leftarrow \text{sustituir}(Ab, Ab_{[d]}, f)$  /* Paso 8*/
end for
    
```

4. Las Curvas B-spline

El término *spline* en computación gráfica y modelado geométrico se refiere a la representación paramétrica a trozos de una geometría con un nivel específico de continuidad paramétrica. La letra *B* antepuesta a la palabra *spline* hace alusión a las funciones básicas. A diferencia de las funciones básicas de las curvas de *Bézier*, que usan *Polinomios de Bernstein*, las funciones básicas de las curvas *B-spline* generan una curva paramétrica polinómica a trozos a través de un número de puntos de control. El grado de la curva polinómica puede ser seleccionado por el diseñador independientemente del número de los puntos de control. El grado de las funciones básicas controla el grado de la curva *B-spline* resultante. Dichas curvas exhiben control local; esto significa que si movemos un punto de control, sólo algunos de los segmentos de la curva se verán afectados y el resto de la curva no variará (Anand, 1993).

4.1. Representación General de las Curvas B-spline

Las curvas *B-spline* pueden ser representadas en términos de sus funciones básicas $N_{i,k}(t)$ definidas mediante la expresión de Cox-de Boor:

$$N_{i,k}(t) = \begin{cases} 1, & t_i \leq t \leq t_{i+1} \\ 0, & \text{otro caso} \end{cases} \quad (2)$$

$$N_{i,k}(t) = \frac{t - t_i}{t_{i+k-1} - t_i} N_{i,k-1}(t) + \frac{t_{i+k} - t}{t_{i+k} - t_{i+1}} N_{i+1,k-1}(t)$$

donde $T = \{t_0, t_1, \dots, t_m\}$ es una secuencia de números reales no decrecientes conocidos como nodos.

Habiendo definido con anterioridad a las funciones básicas, la curva *B-spline* viene dada por:

$$C(t) = \sum_{i=0}^n N_{i,k}(t)P_i \quad (3)$$

donde P_i es el conjunto de $n+1$ puntos de control P_0, P_1, \dots, P_n y $N_{i,k}(t)$ son las funciones básicas *B-spline* de grado $k-1$ (Anand, 1993).

Los parámetros n , m y k deben satisfacer la ecuación $m+1 = n+k+1$ (Piegl & Tiller, 1997). Por otro lado, dados el vector de nodos T de longitud $m+1$ y $n+1$ puntos de control, el grado de la curva *B-spline* es $k = m-n-1$. El punto en la curva que corresponde al nodo t_i , $C(t_i)$, es un punto nodo. Los puntos nodo dividen una curva *B-spline* en varios segmentos de curva.

4.2. El Vector de Nodos

Dado que el vector de nodos influye en la forma de la curva *B-spline*, su elección determina el tipo de curva *B-spline* considerada. En general, se pueden considerar los siguientes tipos de vectores de nodos:

- Uniforme y Periódico: Cuando el vector de nodos posee valores igualmente espaciados, donde $[(t_i - t_{i-1}) = a]$ y a es un número real. Ej.: $T = [0 \ 1 \ 2 \ 3 \ 4]$ con $a = 1$.
- No Periódico: Cuando un vector de nodos tiene valores repetidos de nodos al principio y al final con una multiplicidad igual al orden k y el resto de nodos están igualmente espaciados. Ej.: $k = 2$, $T = [0 \ 0 \ 1 \ 2 \ 3 \ 3]$.
- No Uniforme: Cuando las dos condiciones anteriores no se cumplen, es decir, que el vector de nodos no presenta multiplicidad al principio y al final, ni sus nodos están igualmente espaciados. Ej.: $T = [0 \ 1 \ 2 \ 3 \ 3 \ 4]$, donde se introduce un nodo interno múltiple. O cuando los nodos no están

igualmente espaciados, por ejemplo, $T = [0.0 \ 0.20 \ 0.55 \ 0.75 \ 1.0]$ (Anand, 1993).

4.3. El Algoritmo ClonalG y las Curvas B-spline

El algoritmo *ClonalG* propuesto en este estudio representa una implementación computacional de la *Teoría de la Selección Clonal*. Dicho algoritmo es aplicado para solucionar un problema de optimización multimodal donde se realiza una búsqueda local alrededor de cada individuo de la población.

Supongamos un conjunto de M puntos dato $\{C_i\}$, $i = 1, \dots, M$. El objetivo es obtener una curva *B-spline* $Q(t)$ que aproxime los datos en el sentido de *mínimos cuadrados*, de forma que $Q(t_i) = C_i^* \approx C_i$, $i = 1, \dots, M$.

En la reconstrucción de curvas *B-spline*, la función a ser optimizada es la función de *fitness* $g(\cdot)$ dada por:

$$g(Ab) = \min \left(\sum_{j=1}^M (C_j - C_j^*)^2 \right) = \min \left(\sum_{j=1}^M \left(C_j - \sum_{i=0}^n P_i^* N_{i,k}^*(t_j) \right)^2 \right) \quad (4)$$

donde $N_{i,k}^*(t)$ son las funciones básicas para el vector de nodos Ab dado por el *anticuerpo* óptimo.

De esta forma la afinidad de un *anticuerpo* corresponde al valor de la función objetivo $g(\cdot)$ evaluada para cada *anticuerpo*, es decir, cada *anticuerpo* Ab_i representa un valor codificado de un argumento de la función $g(\cdot)$. Al final del proceso, como no existe una población específica de *antígenos* a ser reconocidos, toda la población Ab corresponderá a *anticuerpos de memoria*.

Como parámetros de entrada del algoritmo implementado se tienen:

- El número n de puntos de control.
- El orden k de la curva.
- M puntos dato de la curva *B-spline* $C(t)$.
- El vector de nodos inicial T .

La finalidad del algoritmo es la generación de los anticuerpos Ab con mejor afinidad y éstos representan los vectores nodo. A fin de comprobar el buen funcionamiento del método, en nuestros primeros ejemplos hemos considerado como datos el conjunto de puntos de control P_i y el orden de la curva k dados para calcular un vector de nodos T no periódico y obtener así, mediante la ecuación (3), los puntos dato de una curva *B-spline* $C(t)$ original.

En otros casos, donde sólo se tiene como parámetros de entrada los puntos dato de la curva *B-spline* $C(t)$, el número n de puntos de control y el orden k , se hace necesario calcular el conjunto de puntos de control P_i de la curva *B-spline* original aplicando la fórmula de aproximación de *mínimos cuadrados*.

En nuestras simulaciones consideramos una población L de 100 *anticuerpos*, generada aleatoriamente, para mantener la diversidad en el repertorio inicial de individuos. Con cada *anticuerpo* producido se procede a calcular los puntos dato de las nuevas curvas *B-spline* $Q(t)$, buscando los puntos de control correspondientes.

Seguidamente se evalúa la afinidad de cada uno de los *anticuerpos* nuevos, aplicando la fórmula (4) entre los puntos dato de la curva *B-spline* inicial $C(t)$ y los puntos dato de las 100 curvas nuevas $Q(t)$. Luego se crea una lista de los $n=10$ mejores *anticuerpos* que fueron seleccionados para ser sometidos a una *Hipermutación Somática*, que consiste en modificar al azar los puntos nodo internos que componen cada uno de estos vectores. Una vez mutados dichos anticuerpos, pasan por un proceso de clonación que depende de la proporción de su respectiva afinidad, ya que cuanto mayor sea la afinidad, mayor es el número de clones producidos. En total se crean 200 clones que serán evaluados en relación a su afinidad.

En cada iteración del algoritmo, los *anticuerpos* de baja afinidad son sustituidos en la población inicial L de 100 individuos por los $n=10$ mejores *anticuerpos* o padres que concibieron a la generación siguiente. Al final de cada iteración, la población L se compone de los $n=10$ mejores *anticuerpos* mas los mejores clones $C^*=90$ que se van generando. En cada iteración es importante que los *anticuerpos* padres sean conservados para evitar la pérdida de las soluciones óptimas en la población L después de cada mutación. En la tabla 2 se muestra la cantidad de *anticuerpos* generados durante las iteraciones del *Algoritmo ClonalG*.

Población inicial	Mejores n	Clones C	Clones C^*	Población final
100	10	200	90	100

Tabla 2: Cantidad de anticuerpos generados en las iteraciones del *ClonalG*

4.4. Complejidad Computacional del ClonalG

El *Algoritmo de Selección Clonal* presenta tres procesos principales:

- 1) El cálculo de la afinidad f de los anticuerpos de la población.
- 2) La selección de los n mejores individuos.
- 3) La *Hipermutación* de los clones C .

El coste computacional para la medición de la afinidad de la población en el problema de la reconstrucción de las curvas *B-spline* viene dado por los procesos de selección e *Hipermutación*, donde el tiempo ejecutado en el primer proceso de selección es de $O(N)$ y en los procesos subsiguientes de selección es de $O(N_c)$, donde N_c representa la cantidad de clones generados a partir de Ab_n . En el proceso de *Hipermutación* de los N_c clones, es necesario un tiempo de procesamiento del orden $O(N_c L)$ donde L es el conjunto de atributos utilizados para representar a los Ab . Sumando la complejidad de cada etapa, la complejidad computacional del algoritmo es de orden $O(N_c L)$.

5. Las Curvas de Bézier

Las curvas de *Bézier* fueron extensamente divulgadas en 1962 por el ingeniero francés Pierre Bézier, quien las usaba para diseñar carrocerías de automóviles. Pero el estudio de estas curvas fue desarrollado por primera vez en 1959 por el matemático Paul de Casteljaou usando el *Algoritmo de de Casteljaou*, un método numéricamente estable para evaluar curvas de *Bézier*.

Una curva paramétrica de forma libre o curva de *Bézier* es definida como (Farin, G., 2002), (Hoschek & Lasser, 1993):

$$C(t) = \sum_{j=0}^N P_j B_j(t) \quad (5)$$

donde $P_j(t)$ son coeficientes vectoriales (usualmente referidos como *puntos de control*), $B_j(t)$ son las funciones básicas de la curva paramétrica $C(t)$ y t es el parámetro, comúnmente definido en un intervalo $[\alpha, \beta]$.

Las funciones básicas $B_j(t)$ son los *polinomios de Bernstein* $B_j^N(t)$, de índice j y grado N , dados por:

$$B_j^N(t) = \binom{N}{j} t^j (1-t)^{N-j} \quad (6)$$

donde

$$\binom{N}{j} = \frac{N!}{j!(N-j)!}, \quad (7)$$

los *polinomios de Bernstein* son evaluados en el intervalo $[0,1]$ así como también lo es la Curva de *Bézier* $C(t)$. Nótese que por convención $0! = 1$.

Supóngase un conjunto de puntos dato $\{Q_i\}_{i=1,\dots,M}$ dados en un espacio n -dimensional (usualmente $n=2$ o $n=3$). El objetivo es la obtención de la curva paramétrica de forma

libre $C(t)$ que mejor ajuste los puntos dato en el sentido de *mínimos cuadrados*. Para lograr eso, se deben calcular los puntos de control $P_j (j= 0, \dots, N)$ de la curva aproximada $C(t)$ minimizando el error mínimo cuadrático, E , definido como la suma de los cuadrados de los residuos:

$$E = \sum_{i=1}^M \left(Q_i - \sum_{j=0}^N P_j B_j^N(t_i) \right)^2 \quad (8)$$

donde se necesita un valor del parámetro t_i para asociarlo a cada punto dato $\{Q_i\}$, $i=1, \dots, M$. Considerando a los vectores columna $B_j = (B_j^N(t_1), \dots, B_j^N(t_M))^T$, $j = 0, \dots, N$, donde $(.)^T$ significa transposición, y $\bar{Q} = (Q_1, \dots, Q_M)$, Ec. (8) se convierte en el siguiente sistema de ecuaciones (llamado *ecuación normal*):

$$\begin{pmatrix} B_0^T \cdot B_0 & \cdots & B_N^T \cdot B_0 \\ \vdots & \vdots & \vdots \\ B_0^T \cdot B_N & \cdots & B_N^T \cdot B_N \end{pmatrix} \begin{pmatrix} P_0 \\ \vdots \\ P_N \end{pmatrix} = \begin{pmatrix} \bar{Q} \cdot B_0 \\ \vdots \\ \bar{Q} \cdot B_N \end{pmatrix} \quad (9)$$

la cual puede compactarse en:

$$M \cdot P = R \quad (10)$$

donde $M = \left[\sum_{j=1}^M B_l(t_j) B_l(t_j) \right]$ y $R = \left[\sum_{j=1}^M Q_j B_l(t_j) \right]$ para $l= 0, \dots, N$.

La solución algebraica de (10) está dada por: $P = B^+ \cdot Q$ donde B^+ denota la pseudoinversa Moore-Penrose de $B = (B_0, B_1, \dots, B_N)^T$. Si la matriz B tiene un rango de $N+1$, M es no-singular (o lo que equivale a decir que $B^T \cdot B$ es invertible) y (10) define unívocamente a P .

Además, disponemos de una formula explícita para hallar B^+ como $B^+ = (B^T \cdot B)^{-1} \cdot N^T = M^{-1} \cdot N^T$. Nótese que B^+ es entonces una inversa izquierda de

B. Por otra parte, B^+ está definido como el P que minimiza $\|P\|^2$ entre las soluciones de (9).

Debido al hecho de que las funciones básicas $B_j^N(t)$ son no lineales en t , la minimización de los errores mínimos cuadráticos es un problema fuertemente no lineal, con un alto número de conjuntos de puntos dato desconocidos, un caso que sucede muy a menudo en la práctica. Por otro lado, si los valores son asignados al t_i , el problema se convierte en una minimización lineal clásica de mínimos cuadrados, con los coeficientes $\{Q_i\}$ como incógnitas.

Nuestra estrategia para resolver el problema en el caso general consiste en aplicar la metodología de los SIA para determinar valores paramétricos adecuados t_i para los puntos dato, y luego calcular los mejores coeficientes ajustados con *mínimos cuadrados* Q_i . El proceso es realizado iterativamente mientras la evolución de los parámetros no estabilice la minimización del error.

5.1. El Algoritmo ClonalG y las Curvas de Bézier

El algoritmo *ClonalG* también ha sido aplicado para reconstruir curvas de *Bézier* dados unos conjuntos de puntos dato. El objetivo es obtener los puntos de control de la curva aproximada que ajuste los mejores puntos dato en el sentido de los *mínimos cuadrados*. Como se ha resaltado en (Barhak & Fischer, 2001) la selección de una parametrización adecuada es esencial para obtener un buen ajuste de la curva paramétrica.

Para aplicar este algoritmo al problema, se han hecho algunas modificaciones al algoritmo *ClonalG* original:

- La medición de la afinidad corresponde a la evaluación de la función de *mínimos cuadrados*, donde cada Ab es una solución potencial al problema.

- No es necesario mantener el subconjunto de Ab de memoria ya que no hay un *antígeno* específico a ser reconocido (aprendizaje no supervisado).
- Varios anticuerpos con alta afinidad son seleccionados en el paso 7 en vez de uno solo, para incrementar la proporción de convergencia.
- Se aplica el *elitismo* al algoritmo, para permitir que algunos de los mejores Ab de la actual generación puedan ser llevados a la próxima, sin ser alterados. Esta característica proporciona una mayor velocidad de convergencia con respecto a las variantes no elitistas y mejora la capacidad de la memoria.

Los parámetros de control del *ClonalG* implementados son:

- El número de puntos de control n_p
- El número de *anticuerpos*, N_{ab}
- El número de *anticuerpos* d a ser reemplazados en $Ab_{\{d\}}$
- El número de Ab a ser clonados, n_{best}
- El número de clones de Ab seleccionados, N_c
- La proporción de *Hipermutación Somática*
- El número de iteraciones (generaciones) N_{iter}

Después de la selección de estos parámetros, el *ClonalG* es ejecutado el número de iteraciones indicadas. El anticuerpo con el mejor (mínimo) valor de afinidad es seleccionado como la mejor solución al problema.

6. Resultados

En esta memoria, hemos aplicado el *Algoritmo de Selección Clonal* descrito en el capítulo 3 a las curvas *B-spline* y de *Bézier*, según se ha descrito en los capítulos 4 y 5, respectivamente. En este capítulo presentamos los principales resultados obtenidos como consecuencia de este trabajo.

En los gráficos que se muestran a continuación se pueden visualizar dos curvas *B-spline*, la original $C(t)$ y la reconstruida $Q(t)$, esta última representa al mejor clon generado. Las curvas clones son obtenidas mediante la optimización de la función afinidad de sus vectores nodo. El mejor *anticuerpo* (vector de nodos) también está representado en los gráficos junto a las curvas.

Para obtener las curvas *B-spline* $C(t)$ y $Q(t)$ en forma de s invertida representadas en la figura 5, se emplearon los siguientes parámetros de entrada: $n=6$ puntos de control, orden $k=3$ y unos 100 puntos dato de la curva. La curva $C(t)$ se dibujó en color celeste y la curva $Q(t)$ en magenta. El vector de nodos no periódico generado después de 3 iteraciones del algoritmo *ClonalG* presenta una afinidad (error de ajuste *RMSE*) de 9.938587×10^{-3} .

Cabe resaltar que cuanto mayor es el número de generaciones *iter*, la afinidad del vector de nodos aumenta. Esto se puede apreciar en la figura 6 donde se realizaron 15 iteraciones empleando los mismos parámetros de entrada para conseguir una afinidad de 3.497251×10^{-4} . Las curvas *B-spline* $C(t)$ y $Q(t)$ son casi idénticas.

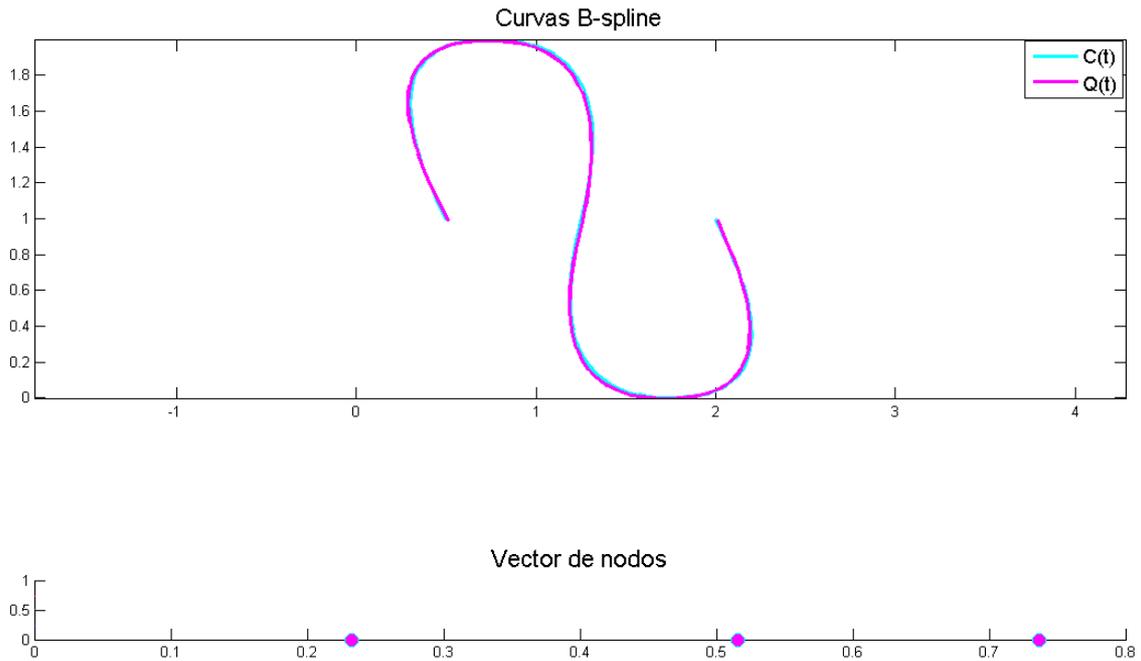


Figura 5: Curvas *B-spline* en forma de S invertida con 3 iteraciones

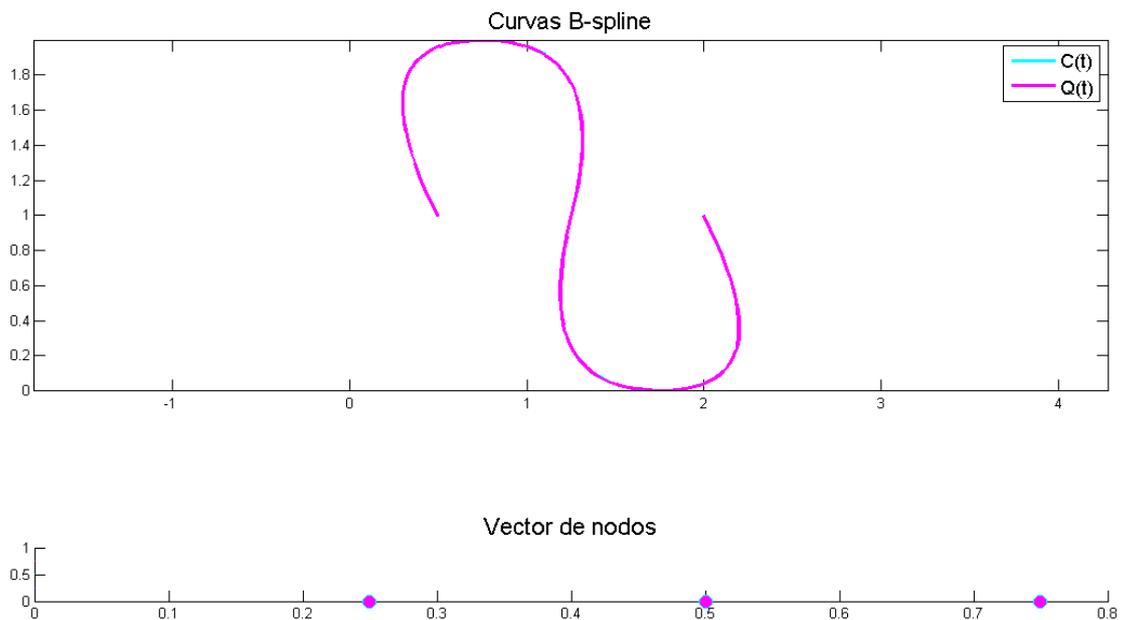


Figura 6: Curvas *B-spline* en forma de S invertida con 15 iteraciones

Las curvas *B-spline* $C(t)$ en color celeste y $Q(t)$ en magenta que forman la letra W en la figura 7 se calcularon con los parámetros de entrada: $n=6$ puntos de control, orden $k=3$ y unos 100 puntos dato de la curva. Después de 15 generaciones se obtuvo un vector de nodos no periódico con una afinidad de 7.687855×10^{-5} . El mejor *anticuerpo*

generado también presenta una afinidad alta y la curvas original $C(t)$ y la reconstruida $Q(t)$ son similares.

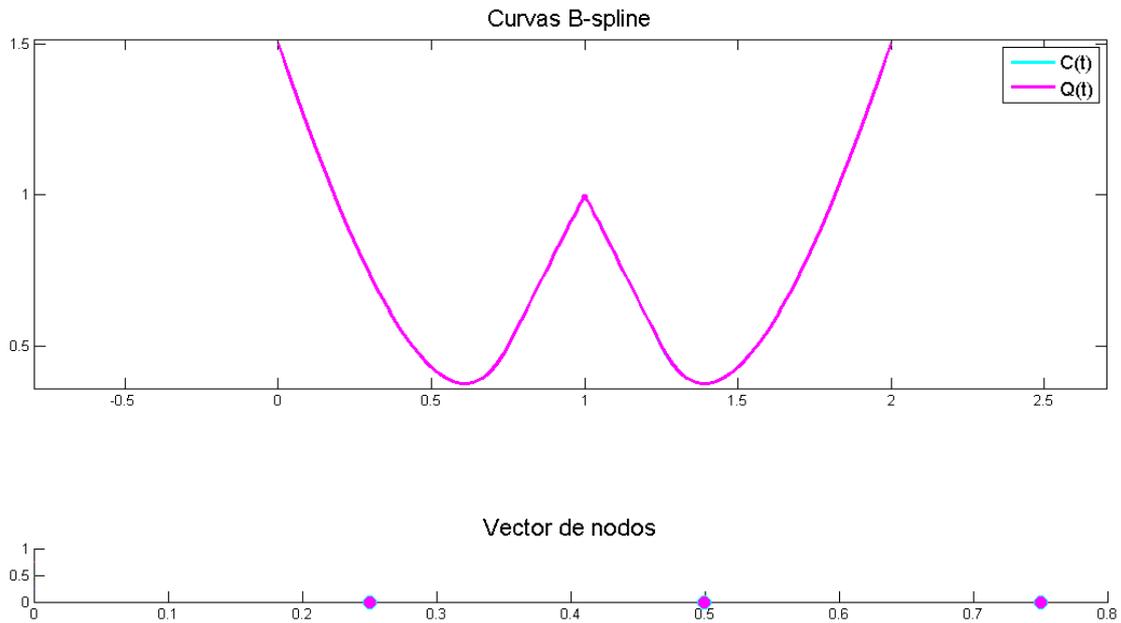


Figura 7: Curvas *B-spline* en forma de W con líneas continuas

En la figura 8 están representadas las curvas *B-spline* en forma de w con los mismos parámetros de entrada e igual número de generaciones, pero en vez de ser dibujadas con líneas continuas como en el gráfico anterior, se representa cada punto dato de la curva $C(t)$ con un círculo magenta mientras que los puntos dato de la curva $Q(t)$ son asteriscos azules, que siendo similares a los puntos dato de $C(t)$, se dibujan dentro de los círculos magenta. La afinidad del vector de nodos obtenida es de 7.687855×10^{-5} . Esta figura muestra claramente el excelente ajuste entre los puntos dato y los puntos reconstruidos, obtenidos mediante nuestro método.

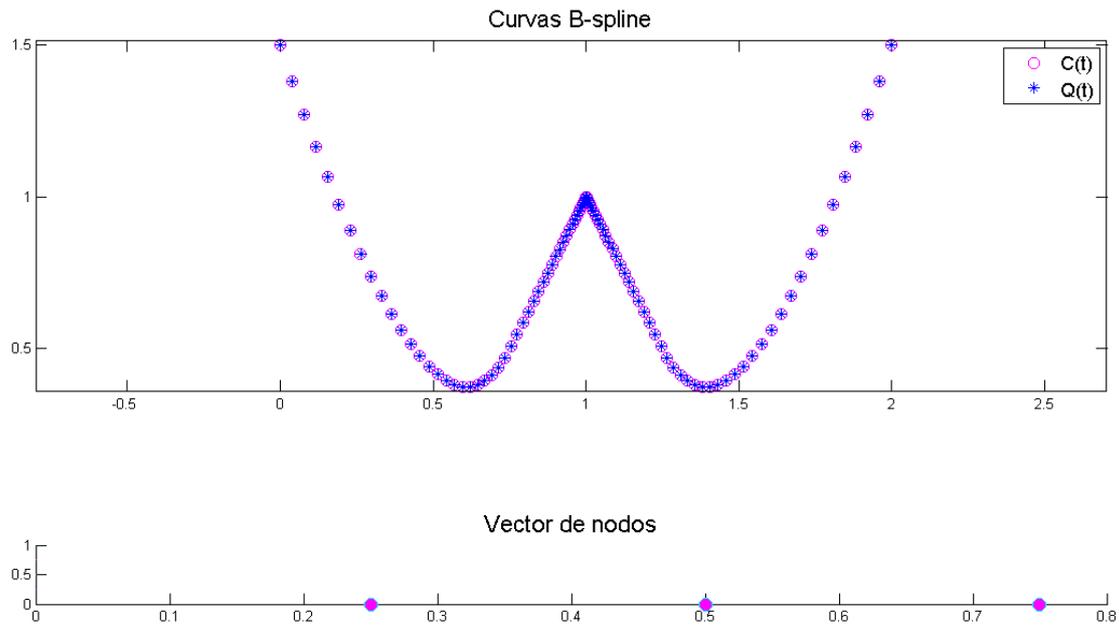


Figura 8: Curvas *B-spline* en forma de W con círculos y asteriscos

Las curvas *B-spline* $C(t)$ en color celeste y $Q(t)$ en magenta dibujadas en forma de gancho en la figura 9 se obtuvieron con los siguientes parámetros de entrada: $n=7$ puntos de control, orden $k=3$ y unos 100 puntos dato de la curva. También se ejecutaron 15 iteraciones para generar un vector de nodos no periódico con una afinidad de 1.811548×10^{-4} . Se puede apreciar que el mejor *anticuerpo* generado también posee una afinidad alta y las curvas $C(t)$ y $Q(t)$ se asemejan.

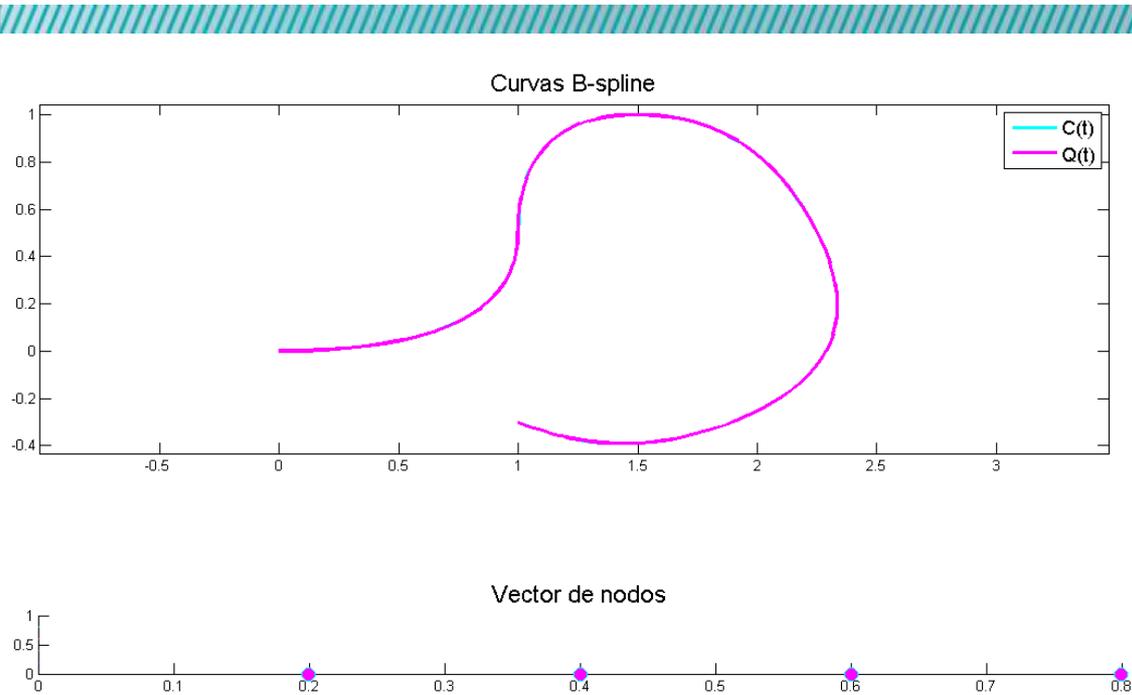


Figura 9: Curvas *B-spline* en forma de gancho

En la figura 10 se trata de representar unas curvas *B-spline* cerradas en forma de paloma. La curva $C(t)$ se dibujó en color celeste mientras que la $Q(t)$ en magenta. Los parámetros de entrada fueron: un número de puntos de control mayor $n= 30$, el orden $k= 4$ y 100 puntos dato de la curva. Después de realizar 50 iteraciones, se obtuvo el mejor vector de nodos con afinidad de 4.147187×10^{-1} . Es de hacer notar que cuando las curvas *B-spline* poseen numerosos puntos de control es recomendable realizar una cantidad mayor de iteraciones para poder alcanzar un valor más óptimo de la afinidad.

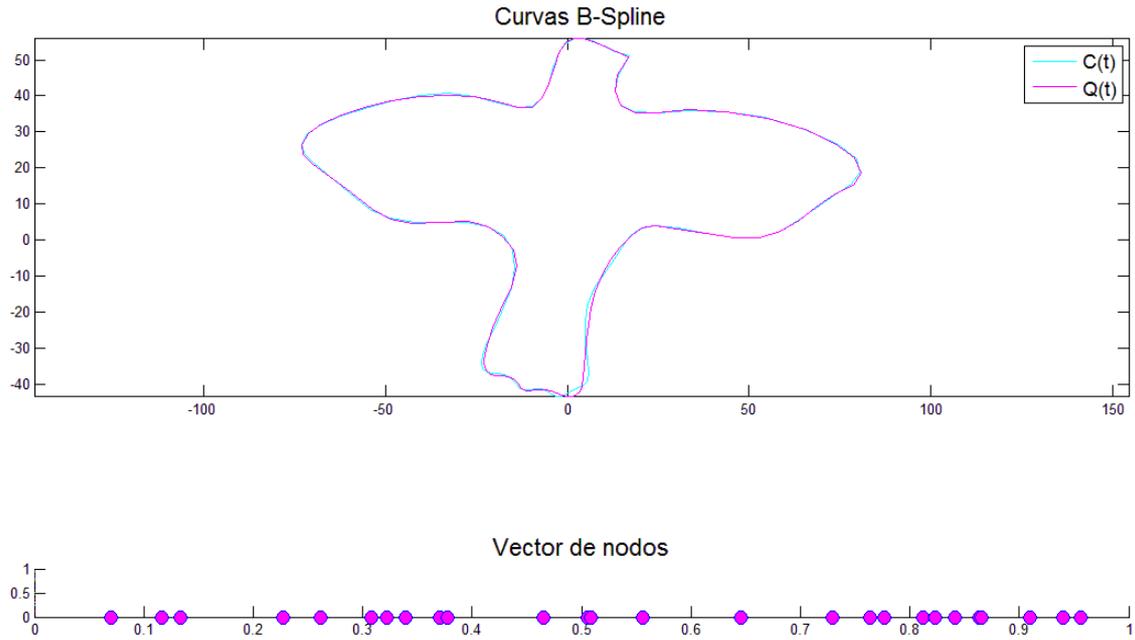


Figura 10: Curvas *B-spline* en forma de paloma

En la figura 11 se muestra una mano representada por dos curvas *B-spline* cerradas en con mayor número de puntos de control $n=43$, el orden $k=3$ y 100 puntos dato de la curva. La curva $C(t)$ se dibujó en color verde y la curva $Q(t)$ en color magenta. Al cabo de 40 iteraciones, se obtuvo el mejor vector de nodos con 2.407730×10^{-1} de afinidad.

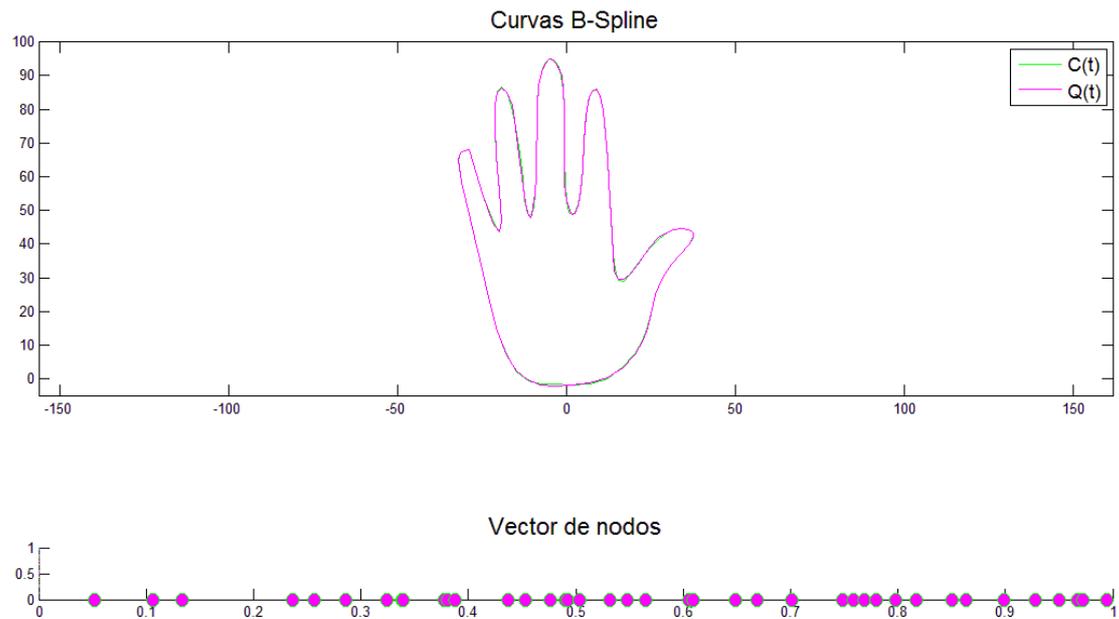


Figura 11: Curvas *B-spline* en forma de mano

Reconstrucción de curvas B-spline y de Bézier mediante Selección Clonal

En la figura 12 se muestran dos curvas *B-spline* cerradas en forma de moto con número de puntos de control $n = 46$, el orden $k = 4$ y 100 puntos dato de la curva. Al finalizar las 40 iteraciones, se obtuvo el mejor vector de nodos con afinidad de 1.238544×10^{-1} .

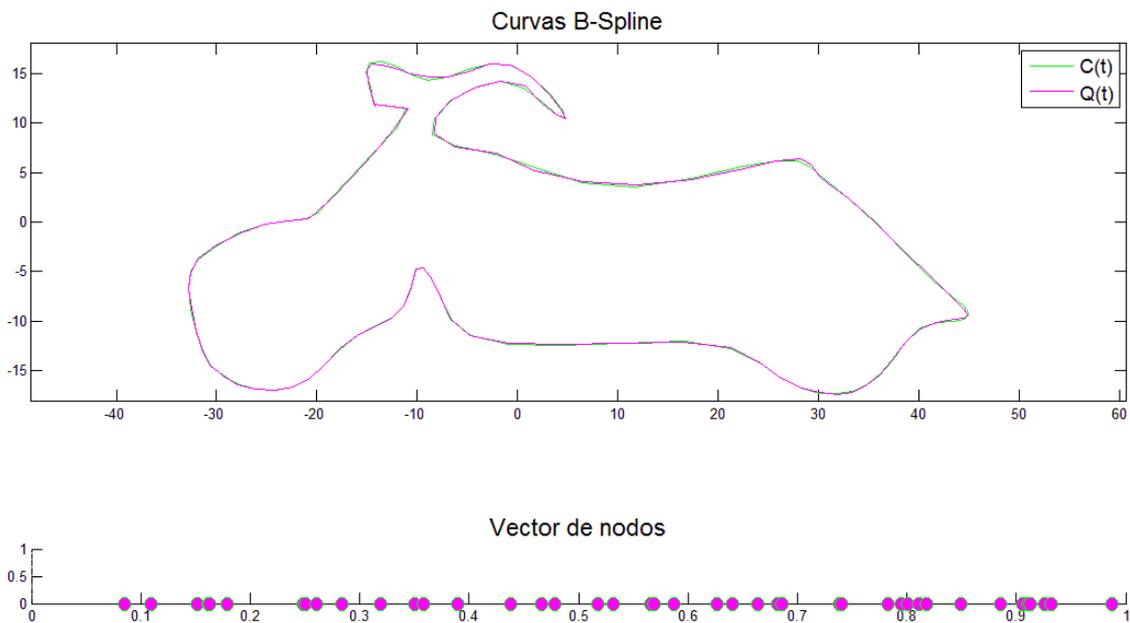


Figura 12: Curvas *B-spline* en forma de moto

En la tabla 3 están representados los parámetros de entrada y salida del algoritmo *ClonalG* aplicado a la reconstrucción de curvas *B-spline*.

Figura	Ptos. Control	Orden	Ptos. dato	Iteraciones	Afinidad
5	6	3	100	3	9.938587×10^{-3}
6	6	3	100	15	3.497251×10^{-4}
7	6	3	100	15	7.687855×10^{-5}
8	6	3	100	15	7.687855×10^{-5}
9	7	3	100	15	1.811548×10^{-4}
10	30	4	100	50	4.147187×10^{-1}
11	43	3	100	40	2.407730×10^{-1}
12	46	4	100	40	1.238544×10^{-1}

Tabla 3: Parámetros del algoritmo *ClonalG* en la aproximación de curvas *B-spline*

En relación a la implementación del algoritmo *ClonalG* para la reconstrucción de curvas de Bézier, se han realizado 3 ejemplos ilustrativos, que han sido escogidos con sumo

cuidado para reflejar la variedad de las situaciones en las que nuestro algoritmo puede ser aplicado.

La figura 13 corresponde a un conjunto de 100 puntos dato obtenidos de una muestra no uniforme de la curva *Agnesi*, dados por $x^2y = 4(2-y)$. Estos puntos dato son ajustados usando una curva de *Bézier* con $n_p = 30$ puntos de control. Los demás parámetros son los siguientes: $N_{ab} = 100$, $d = 20$, $n_{best} = 10$ y $N_{iter} = 3$. El número de clones ha sido tomado proporcionalmente a la medida de la afinidad, donde los *anticuerpos* con mejor afinidad tendrán más clones. A continuación, se seleccionan 60 clones para los *anticuerpos* con la mejor afinidad, 40 clones para los segundos mejores, 20 clones para los terceros mejores y 10 clones para el resto. El número total de clones es $N_c = 190$, los cuales sufren una *Hipermutación Somática* de valor real inversamente proporcional a la afinidad.

En la figura 13, los puntos originales se muestran como círculos rojos vacíos y los puntos dato reconstruidos como símbolos de suma azules. Es de hacer notar la buena correspondencia entre los puntos dato originales y los reconstruidos. De hecho, se obtuvo una afinidad de 5.1×10^{-5} para estos puntos dato, indicando así que la curva de *Bézier* reconstruida aproxima los puntos dato con gran exactitud. También se calculó el valor absoluto medio de la diferencia entre los puntos dato originales y los reconstruidos, y se obtuvieron muy buenos resultados: $(2.59 \times 10^{-5}, 5.32 \times 10^{-5})$. En la figura 14 también se representan los puntos dato originales y los reconstruidos de la curva de *Bézier* aproximada, dibujados en negro y azul, respectivamente.

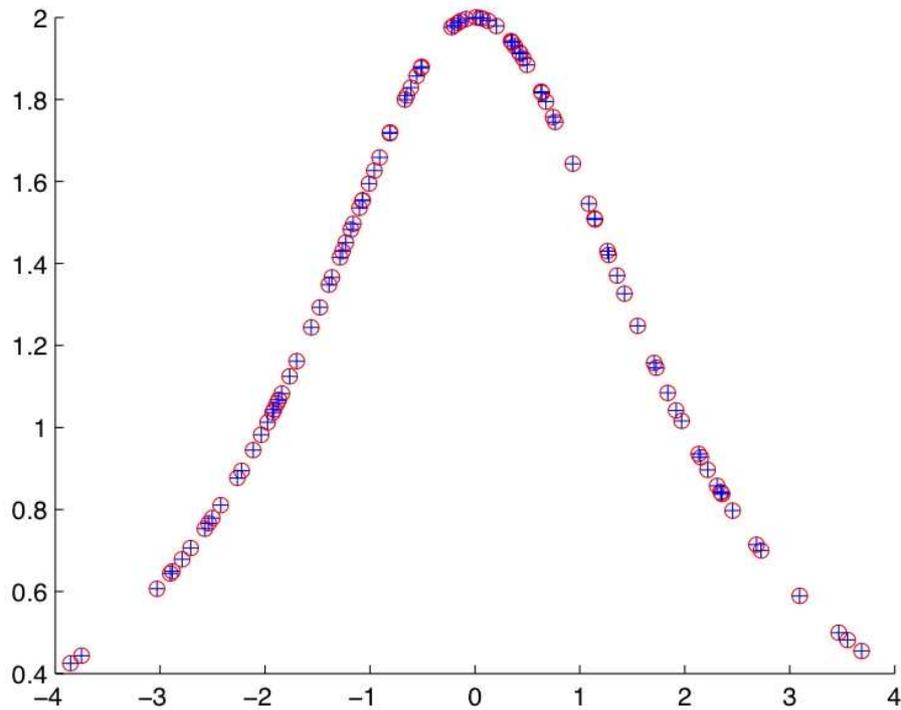


Figura 13: Aplicación del *ClonalG* para reconstrucción de curvas de *Bézier*: Puntos dato originales (círculos rojos) y reconstruidos (símbolos suma azules) para la curva *Agnesi*.

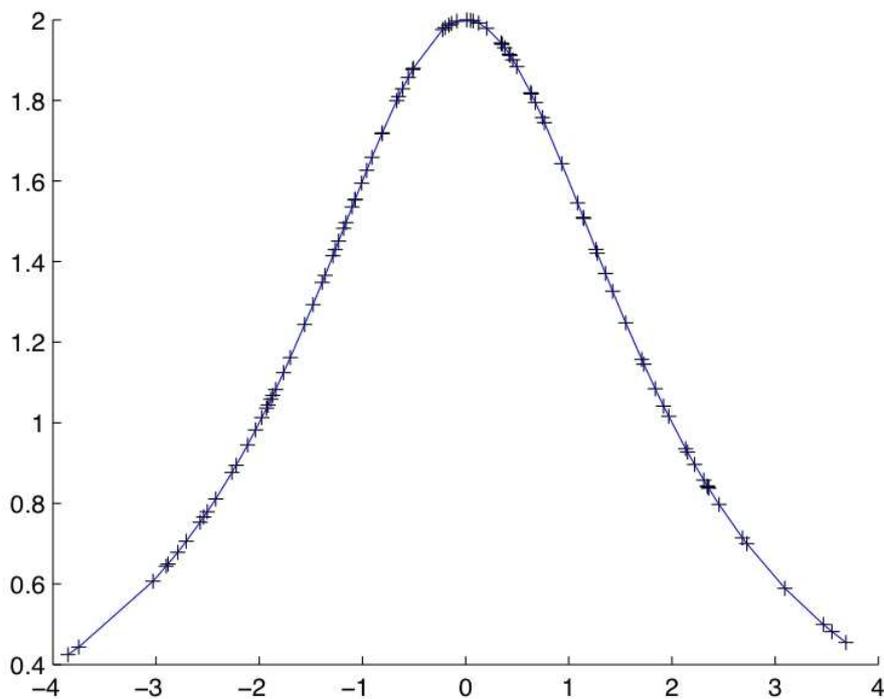


Figura 14: Aplicación del *ClonalG* para el ajuste de datos: puntos dato originales (en negro) y curva de *Bézier* reconstruida (en azul) para la curva *Agnesi*.

La figura 15 corresponde a la curva espiral de *Arquímedes* (conocida como la curva espiral aritmética). Dicha curva es el lugar geométrico correspondiente a las posiciones en el tiempo de un punto que se mueve alejándose de un punto fijo con una velocidad constante a lo largo de una línea que rota con una velocidad constante angular. En este ejemplo, se considera un conjunto de 100 puntos dato provenientes de esta curva, que son subsecuentemente aproximados usando una curva de *Bézier* con $n_p=16$ puntos de control. Los demás parámetros son similares a los usados en el ejemplo anterior.

La figura 15 (arriba) muestra los puntos dato originales dibujados como círculos rojos y los reconstruidos como símbolos de suma azules. Es de hacer notar la excelente correspondencia entre los puntos dato originales y los reconstruidos. En este caso, los resultados son mucho mejores, con una afinidad de 8.77×10^{-10} para esos puntos dato, mientras que el valor absoluto medio de la diferencia entre los puntos dato originales y los reconstruidos es de $(2.27 \times 10^{-10}, 1.06 \times 10^{-9})$, es decir que la curva de *Bézier* reconstruida aproxima muy bien los puntos dato. En la figura 15 (abajo) se muestran los puntos dato originales, la curva de *Bézier* reconstruida, sus puntos de control, y el polígono de control dibujados en negro, azul y rojo respectivamente.

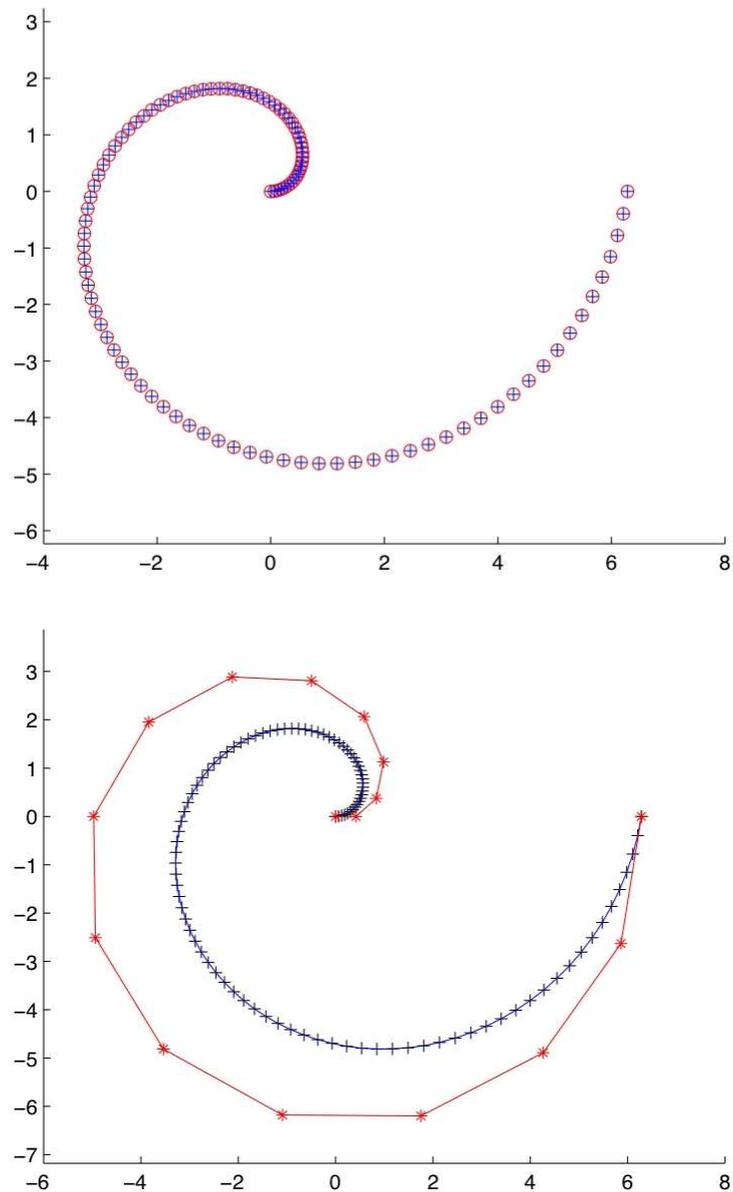


Figura 15: Aplicación del *ClonalG* para la aproximación de los puntos dato de la curva espiral de Arquímedes

En las figuras 16 y 17 se representa una curva *Hipotrochoide*, la cual es una ruleta trazada por un punto vinculado a un círculo de radio r rodando alrededor del interior de un círculo fijo de radio R (donde $R > r$), donde el punto está a una distancia d desde el centro del círculo interior. Por ejemplo, se considera una curva de *Bézier* ajustada con $n_p = 40$ puntos de control. El resto de parámetros son iguales a los empleados en los ejemplos anteriores.

La figura 16 muestra los resultados obtenidos: Los puntos dato originales y los reconstruidos se representan como círculos rojos vacíos y símbolos de suma azules, respectivamente. En la figura 17, los puntos dato originales y la curva de *Bézier* reconstruida se dibujan en negro y azul, respectivamente. Se obtuvo una correspondencia excelente entre los puntos dato originales y los reconstruidos, aunque la curva en ambas figuras exhibe varios puntos de auto-intersección, con una afinidad de 2.17×10^{-9} . El valor absoluto medio de la diferencia entre los puntos datos originales y los reconstruidos para cada coordenada es $(7.19 \times 10^{-10}, 2.59 \times 10^{-9})$.

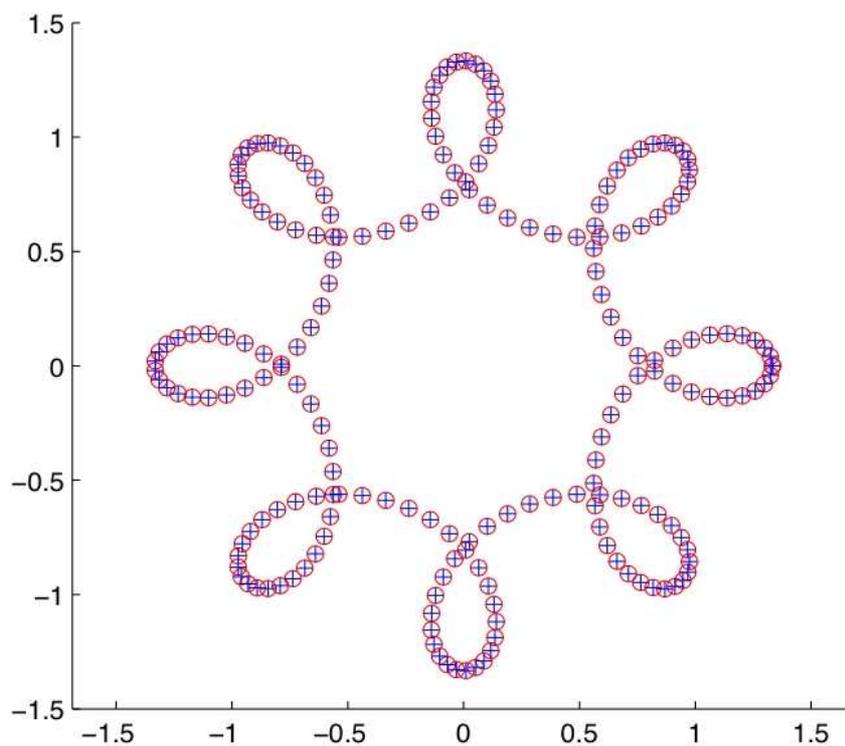


Figura 16: Aplicación del *ClonalG* para el ajuste de datos: Puntos dato originales (círculos rojos) y curva Bézier reconstruida (símbolos suma azules) para la curva *Hipetrochoide*.

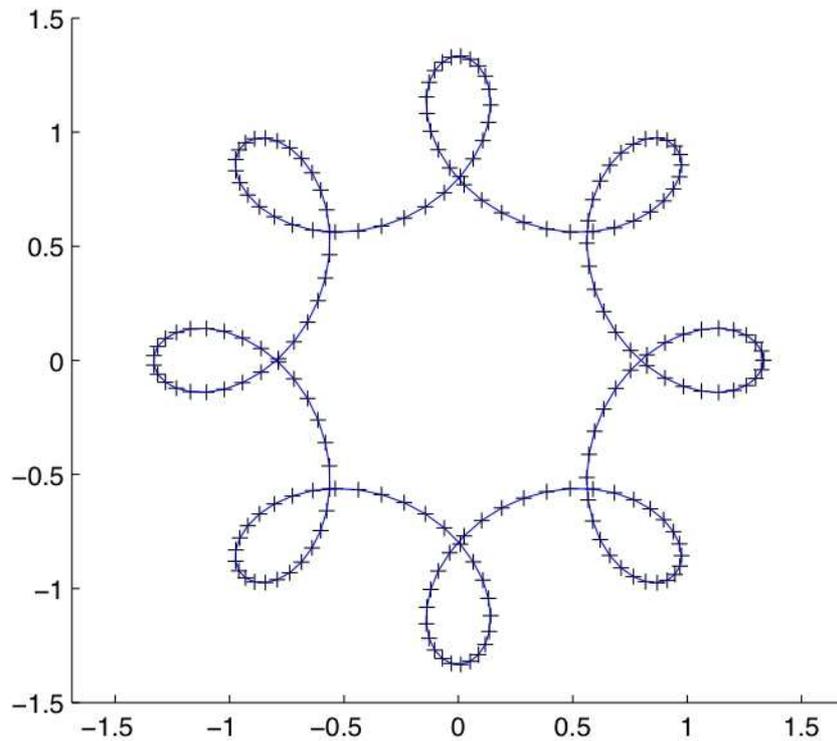


Figura 17: Aplicación del *ClonalG* para el ajuste de datos: puntos dato originales (en negro) y curva Bézier reconstruida (en azul) para la curva *Hipotrochoide*.

En la tabla 4 están representados los parámetros de entrada y salida del algoritmo *ClonalG* aplicado a la reconstrucción de curvas de Bézier.

Figura	Ptos. Control	N_{ab}	N_{iter}	Afinidad	Diferencia
13	30	100	3	5.1×10^{-5}	$(2.59 \times 10^{-5}, 5.32 \times 10^{-5})$
14	30	100	3	5.1×10^{-5}	$(2.59 \times 10^{-5}, 5.32 \times 10^{-5})$
15	16	100	3	8.77×10^{-10}	$(2.27 \times 10^{-10}, 1.06 \times 10^9)$
16	40	100	3	2.17×10^{-9}	$(7.19 \times 10^{-10}, 2.59 \times 10^9)$
17	40	100	3	2.17×10^{-9}	$(7.19 \times 10^{-10}, 2.59 \times 10^9)$

Tabla 4: Parámetros del algoritmo *ClonalG* en la aproximación de curvas de Bézier

7. Detalles de Implementación

Todos los cálculos realizados en esta memoria han sido ejecutados en un procesador Intel Core 2 Duo, de 2.4 GHz con 4 GB de RAM. El código fuente ha sido implementado en el lenguaje nativo de programación del programa científico Matlab, versión 2010b. Matlab es la herramienta apropiada para esta tarea, por ser rápida y porque proporciona rutinas fiables y bien probadas para la manipulación eficiente de matrices. También contiene un variado conjunto de recursos y opciones para resolver sistemas de ecuaciones.

En este proyecto hemos aplicado la descomposición de matrices en valores singulares, ya que nos proporciona la mejor solución numérica en el sentido de mínimos cuadrados para aquellos casos donde no se puede hallar la solución exacta de los sistemas de ecuaciones. Con este propósito, la matriz M es descompuesta como el producto $M = U \cdot W \cdot V^T$ donde U es una matriz columna ortogonal, W es una matriz diagonal con elementos positivos o ceros w_k llamados valores singulares y V es una matriz ortogonal cuadrada. Como M es cuadrada, su inversa puede ser obtenida así:

$$M^{-1} = V \cdot \left[\text{diag} \left(\frac{1}{w_k} \right) \right] \cdot U^T \quad (11)$$

Matlab también proporciona el comando `mldivide` para resolver la ecuación (9) para sistemas cuadráticos y no cuadráticos. Además, Matlab posee unas excelentes opciones gráficas y código optimizado para las interacciones de entrada/salida y la computación de alto rendimiento.

Una de las características más atractivas de este método se refiere a los tiempos de cómputo, ya que son bastante pequeños comparados con otros métodos bioinspirados, como los algoritmos genéticos o la optimización de enjambre de partículas. Esto puede ser explicado por el bajo número de iteraciones requeridas para obtener un buen ajuste de los puntos dato.

8. Conclusiones

Los *Algoritmos de Selección Clonal* se basan en dos principios: la *Hipermutación* y los mecanismos de expansión clonal. La *Hipermutación* es un proceso de búsqueda local que conlleva a la maduración rápida y la etapa de expansión clonal provoca el crecimiento de la nueva población de *células B* útiles enfocándose en el *antígeno* que desencadenó este mecanismo de defensa en el organismo.

El algoritmo *ClonalG* aplicado al caso de optimización de la función multimodal consta de 3 pasos fundamentales, que son el asignar una función que transforme el valor de la función de optimización en un valor de afinidad, asignar una función que relacione la afinidad al porcentaje de mutación a ser aplicado y asignar una función que conecte la afinidad al número de clones a ser producidos.

Los resultados obtenidos por la implementación del algoritmo *ClonalG* en la reconstrucción de curvas *B-spline* tanto abiertas como cerradas son bastante alentadores ya que dicho método determina una apropiada localización de los vectores nodo de manera automática. Al mismo tiempo, no ha sido necesario establecer ningún parámetro de entrada subjetivo o generar una buena población de vectores nodo inicial para que se llevase a cabo una buena búsqueda local iterativa.

Este algoritmo también ha sido implementado para calcular la curva de *Bézier* que mejor aproxima los puntos dato dados en el sentido de *mínimos cuadrados*. El método ha sido probado con tres ejemplos ilustrativos exhibiendo propiedades interesantes, como el ejemplo con la muestra de puntos dato no uniformes y el de las curvas multivaluadas o que se intersecan a ellas mismas.

El algoritmo *ClonalG* se diferencia de los *Algoritmos Genéticos (AG)* en que presentan el mecanismo de afinidad relacionado con la mutación. Los *AG* presentan una convergencia lenta y su coste computacional es mayor para las generaciones cercanas a la solución óptima, mientras que el algoritmo *ClonalG* presenta un coste

computacional bajo, debido a los procesos que caracterizan a los *SIA*, como son la capacidad de generar mejores *anticuerpos* de buenos *anticuerpos*, producir mayor número de clones para los buenos *anticuerpos* y preservar los *anticuerpos de memoria* en la población.

En el futuro este método puede ser usado con puntos dato que estén muy afectados por medidas de ruido de alta intensidad. También puede utilizarse para la determinación de los valores óptimos de otros parámetros y, por último, puede ser extendido a la aproximación de superficies *B-spline* y de *Bézier*.

8.1. Publicaciones de este Proyecto

Este Trabajo de Fin de Master ha dado lugar a la publicación siguiente:

A. Iglesias, A. Gálvez, A. Avila: "Discrete Bézier Curve Fitting with Artificial Immune Systems". *Intelligent Computer Graphics 2012*. Springer-Verlag, Series: *Studies in Computational Intelligence*, Berlin Heidelberg (en prensa).

9. Referencias

(Alhanaty & Bercovier, 2001) Alhanaty M, Bercovier M. "Curve and surface fitting and design by optimal control methods". *Computer-Aided Design*; 33:167–82.

(Anand, 1993) Anand, Vera B. "Computer Graphics and Geometric Modeling for Engineers". John Wiley & Sons, Inc.

(Barhak & Fischer, 2001) Barhak, J., Fischer, A. "Parameterization and reconstruction from 3D scattered points based on neural network and PDE techniques". *IEEE Trans. on Visualization and Computer Graphics*, 7(1) 1-16.

(Bereck & Ziegner, 1993) Bereck, C. y Ziegner, M. "The maturation of the immune response". *Immunology Today*, 14(8):400-404.

(Burchard, 1974) Burchard HG. "Splines (with optimal knots) are better". *Applicable Analysis*; 3: 309–19.

(Burnet, 1978) Burnet, F. M. "Clonal Selection and After". En: *Theoretical Immunology*, (Eds.) G. I. Bell, A.S. Perelson & G. H. Pimbley Jr., Marcel Dekker Inc., pp. 63-85.

(Cutello & Nicosia, 2005) Cutello, V. y Nicosia, G. "Chapter VI. The Clonal Selection Principle for In Silico and In Vivo Computing". En: *Recent Developments in Biologically Inspired Computing*, eds. Leandro Nunes de Castro and Fernando J. Von Zuben. Hershey, London, Melbourne, Singapore: Idea Group Publishing, pp. 104-146.

(Dierckx, 1993) Dierckx, P. "Curve and surface fitting with splines". Oxford: Oxford University Press.

(de Boor & Rice, 1968a) de Boor, CA, Rice, JR. "Least squares cubic spline approximation. I: fixed knots". CSD TR 20. Lafayette (IN): Purdue University.

(de Boor & Rice, 1968b) de Boor, CA, Rice, JR. "Least squares cubic spline approximation. II: variable knots". CSD TR 21. Lafayette (IN): Purdue University; 1968.

(De Castro & Von Zuben, 1999) De Castro, Leandro y Von Zuben, Fernando. "Artificial Immune Systems: Part I – Basic Theory and Applications". *Technical Report TR – DCA 01/99*.

(De Castro & Von Zuben, 2001) De Castro, Leandro y Von Zuben, Fernando. "Immune and neural network models: theoretical and empirical comparisons". *International Journal of Computational Intelligence and Applications* 1 (3) 239–257.

(De Castro, 2001) De Castro, Leandro. "Tesis de Doctorado. Ingeniería Inmunológica: Desarrollo y Aplicación de Herramientas Computacionales Inspiradas en Sistemas Inmunológicos Artificiales". Universidade Estadual de Campinas, Brasil.

(O. Engin & A. Doyen, 2004) Engin, O., Doyen, A. "A new approach to solve hybrid flow shop scheduling problems by artificial immune system". *Future Generation Computer Systems* 20, 2004, 1083–1095.

(Farin, G., 2002) Farin, Gerald. "Curves and surfaces for CAGD". 5th ed. Morgan Kaufmann, San Francisco.

(Floreano & Mattiussi, 2008) Floreano, D. y Mattiussi, C. "Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies". *The MIT Press*, Cambridge.

(Gálvez & Iglesias, 2011) A. Gálvez, A. Iglesias. "Efficient Particle Swarm Optimization Approach for Data Fitting with Free Knot B-splines". *Computer-Aided Design - CAD*, Vol. 43, Issue 12, pp. 1683-1692

(Galvez et al., 2012a) Galvez A, Iglesias A, Puig-Pey J. "Iterative two-step genetic-algorithm method for efficient polynomial B-spline surface reconstruction". *Information Sciences*, Vol. 182, Issue 1, pp. 56-76.

(Gálvez & Iglesias, 2012b) Gálvez A, Iglesias A. "Particle swarm optimization for non-uniform rational B-spline surface reconstruction from clouds of 3D data points". *Information Sciences*, Vol. 192, Issue 1, pp. 174-192.

(Goldenthal & Bercovier, 2004) Goldenthal R, Bercovier M. "Spline curve approximation and design by optimal control over the knots". *Computing*; 72:53–64.

(Gu & Yan, 1995) P. Gu, X. Yan. "Neural network approach to the reconstruction of free-form surfaces for reverse engineering". *Computer Aided Design* 27, (1), 59–64.

(Hoffmann, 2005) M. Hoffmann. "Numerical control of Kohonen neural network for scattered data approximation". *Numerical Algorithms* 39, 175–186.

(Hoschek & Lasser, 1993) Hoschek, J., Lasser, D. "Fundamentals of Computer Aided Geometric Design". A.K. Peters.

(Iglesias & Gálvez, 2001) A. Iglesias, A. Gálvez. "A new artificial intelligence paradigm for computer aided geometric design". *Lectures Notes in Artificial Intelligence* 1930, 200–213.

(Iglesias et al., 2004) A. Iglesias, G. Echevarría, A. Gálvez. "Functional networks for B-spline surface reconstruction", *Future Generation Computer Systems* 20 (8) 1337–1353.

(Jing & Sun, 2005) Jing L, Sun L. "Fitting B-spline curves by least squares support vector machines". In: *Proc. of the 2nd. int. conf. on neural networks & brain*. Beijing (China): *IEEE Press*; p. 905–9.

(Jupp, 1978) Jupp DLB. "Approximation to data by splines with free knots". *SIAM Journal on Numerical Analysis*; 15:328–43.

(Knopf & Kofman, 1999) G.K. Knopf, J. Kofman. "Free-form surface reconstruction using Bernstein basis function networks". In: C.H. Dagli et al. (Eds.), *Intelligent Engineering Systems Through Artificial Neural Networks*, vol. 9, ASME Press, pp. 797–802.

(Kumar et al., 2004) S.G. Kumar, P.K. Kalra, S.G. Dhande. "Curve and surface reconstruction from points: an approach based on self-organizing maps", *Applied Soft Computing* 5 (5) 55–66.

(Lindstrom, 1999) Lindstrom MJ. "Penalized estimation of free-knot splines". *Journal of Computational and Graphical Statistics*; 8(2):333–52.

(Lyche & Morken, 1987) Lyche T, Morken K. "Knot removal for parametric B-spline curves and surfaces". *Computer-Aided Design*; 4:217–30.

(Lyche & Morken, 1988) Lyche T, Morken K. "A data-reduction strategy for splines with applications to the approximation of functions and data". *IMA Journal of Numerical Analysis*; 8:185–208.

(Molinari et al., 2004) Molinari N, Durand JF, Sabatier R. "Bounded optimal knots for regression splines". *Computational Statistics & Data Analysis*; 45(2):159–78.

(Park, 2004) Park, H. "An error-bounded approximate method for representing planar curves in B-splines". *Computer-Aided Geometric Design*; 21: 479–97.

(Park & Lee, 2007) Park, H, Lee, JH. "B-spline curve fitting based on adaptive curve refinement using dominant points". *Computer-Aided Design*; 39:439–51.

(Perelson & Oster, 1979) Perelson, A. S., Oster, G. F. "Theoretical Studies of Clonal Selection: Minimal Antibody Repertoire Size and Reliability of Self-Nonself Discrimination". *J. theor. Biol.*, 81, pp. 645-670.

(Perelson, 1989) Perelson, A. S. "Immune Network Theory". *Immunological Review*, 110, pp. 5-36.

(Piegl & Tiller, 1997) Piegl L, Tiller W. "The NURBS book". Berlin (Heidelberg): *Springer Verlag*.

(Powell, 1970) Powell, MJD. "Curve fitting by splines in one variable". In: Hayes JG, editor. *Numerical approximation to functions and data*. London: Athlone Press.

(Rice, 1969) Rice JR. "The approximation of functions, vol. 2". Reading, MA: Addison-Wesley.

(Segel & Perelson, 1988) Segel, L. y Perelson, A. S. "Computations in Shape Space: A New Approach to Immune Network Theory". In *Theoretical Immunology, Part Two*, (Ed.) A. S. Perelson, pp. 321-343.

(Wang et al., 2006) Wang, W.P., Pottmann, H., Liu Y. "Fitting B-spline curves to point clouds by curvature-based squared distance minimization". *ACM Transactions on Graphics*; 25(2):214–38.

(Yang et al., 2004) Yang HP, Wang WP, Sun JG. Control point adjustment for B-spline curve approximation. *Computer-Aided Design*; 36:639–52.