

**ALGORITMO ESTOCÁSTICO PARA LA OPTIMIZACIÓN DE LA  
TRAYECTORIA DE UN ROBOT**



**Nidia Pilar Carrero Poveda  
Carolina Quijano Chacón**

**UNIVERSIDAD AUTONOMA DE BUCARAMANGA  
ESCUELA DE CIENCIAS NATURALES E INGENIERIAS  
FACULTAD DE INGENIERIA DE SISTEMAS  
LINEA DE OPTIMIZACIÓN  
BUCARAMANGA  
2005**

**ALGORITMO ESTOCÁSTICO PARA LA OPTIMIZACIÓN DE LA  
TRAYECTORIA DE UN ROBOT**

**NIDIA PILAR CARRERO POVEDA  
CAROLINA QUIJANO CHACÓN**

**Trabajo de grado para optar al título de  
Ingeniero de Sistemas**

**Directora:  
ALINA FEDOSSOVA  
Ph.D. en Ciencias Físico-Matemáticas**

**UNIVERSIDAD AUTONOMA DE BUCARAMANGA  
ESCUELA DE CIENCIAS NATURALES E INGENIERIAS  
FACULTAD DE INGENIERIA DE SISTEMAS  
LINEA DE OPTIMIZACIÓN  
BUCARAMANGA**

**2005**

Nota de Aceptación

---

---

---

---

---

---

Firma del presidente del jurado

---

Firma del jurado

---

Firma del jurado

Bucaramanga, 2005

A Dios a quien todo le debo y a María Auxiliadora quien nunca me abandona. Ellos me han dado la fuerza en los momentos más difíciles de mi vida.

A mi padre Bercelí Carrero (q.e.p.d), a quien no voy a tener la fortuna de abrazar el día de mi graduación, pero sé que siempre ha estado y estará a mi lado, apoyándome y dándome fuerzas para que su “chata fea” sea la profesional que él siempre quiso que fuera..

A mi madre Ofelia Poveda, quien se ha desvelado por mí desde que era una niña, siempre contando con su apoyo, amor y comprensión. Gracias por todo su sacrificio y esmero, que hicieron posible alcanzar esta meta.

A todas las personas que de una u otra forma me motivaron y apoyaron para lograr uno de los muchos sueños que tengo por cumplir.

Con mucho cariño

Pilar Carrero

A mis padres por apoyarme, por brindarme su amor y por los esfuerzos dedicados para que mi hermana y yo salgamos adelante; a mis profesores por su constante enseñanza y valiosos aportes durante mi camino de aprendizaje; a mis amigos que me permitieron ser parte de sus vidas y formaron parte de la mía y a todas las personas que me ofrecieron una palabra de aliento para seguir adelante.

Con cariño

Carolina Quijano

## **AGRADECIMIENTOS**

Los autores expresan su agradecimiento:

A nuestra directora de proyecto Alina Fedosova, por toda su colaboración en el desarrollo de este trabajo.

A nuestros compañeros, por su amistad y sus palabras de ánimo.

A nuestros padres y familiares quienes siempre estuvieron dándonos la fortaleza y su ejemplo para cumplir a cabalidad esta etapa de nuestro proceso de formación.

## GLOSARIO

**ESTOCÁSTICO:** Proceso aleatorio, que se contrasta normalmente a un proceso determinístico. Las variables que determinan el proceso estocástico son aleatorias.

**GRADOS DE LIBERTAD:** Son los parámetros que se precisan para determinar la posición y la orientación del elemento terminal del manipulador. También se pueden definir los grados de libertad, como los posibles movimientos básicos (giratorios y de desplazamiento) independientes.

**MANIPULADOR:** Mecanismo formado generalmente por elementos en serie, articulados entre sí, destinado al agarre y desplazamiento de objetos. Es multifuncional y puede ser gobernado directamente por un operador humano o mediante dispositivo lógico.

**PROGRAMACIÓN SEMI-INFINITA:** Problemas de programación matemática con un número finito e infinito de restricciones (por ejemplo, puntos de una región o instantes en un período de tiempo). Son restricciones infinitas porque están en función del tiempo.

**RESTRICCIONES ACTIVAS:** Una restricción realmente "restringe" cuando su holgura es cero, lo cual significa que en el punto óptimo se utiliza totalmente el recurso para el cual se escribió la restricción. Al agotarse el recurso en la fabricación de la mezcla de productos indicada por la solución óptima, no es

posible que las variables tomen un valor mayor; por eso la restricción restringe efectivamente los valores que pueden tomar las variables de decisión<sup>1</sup>.

ROBOT: Manipulador automático servocontrolado, reprogramable, polivalente, capaz de posicionar y orientar piezas, útiles o dispositivos especiales, siguiendo trayectorias variables reprogramables, para la ejecución de tareas variadas. Normalmente tiene la forma de uno o varios brazos terminados en su muñeca. Su unidad de control incluye un dispositivo de memoria y ocasionalmente la percepción del entorno. Normalmente su uso es el de realizar una tarea de manera cíclica, pudiéndose adaptar a otra sin cambios permanentes en su material<sup>2</sup>.

ROBOT INDUSTRIAL: Según la Organización Internacional de Estándares (ISO) es un manipulador reprogramable con varios grados de libertad, capaz de mover materias, piezas, herramientas o dispositivos especiales, según trayectorias variables, programadas para realizar tareas diversas.

---

<sup>1</sup> UNIVERSIDAD DE ANTIOQUIA, Restricciones Activas y Pasivas, 2003. Disponible en Internet <URL:<http://docencia.udea.edu.co/ingenieria/plineal/solucionGrafica04.html>>

<sup>2</sup> Robot Institute of America/ [Schlussel, 85].

## RESUMEN

Este trabajo de grado presenta el desarrollo de un algoritmo estocástico de aproximaciones externas y su proceso de simulación correspondiente, implementado en una herramienta computacional prototipo basada en Matlab; a fin de optimizar el tiempo total de desplazamiento de un brazo robot, de tal forma que las trayectorias de sus articulaciones cumplan con restricciones mecánicas tales como velocidad, aceleración, tirón (jerk) entre otras. Este problema de optimización en donde el conjunto factible plantea además de un número finito de restricciones un número infinito de ellas (debido a que están en función del tiempo), se ha abordado mediante el concepto de la programación semi-infinita (Semi-Infinite Programming - SIP).

Debido al alto nivel de complejidad y dificultad que plantea la solución de estos problemas SIP, se han desarrollado dentro del estado del arte diferentes métodos que buscan una mejor solución con menor esfuerzo computacional. Dentro de este trabajo de grado se desarrolló una versión mejorada del método estocástico de aproximaciones externas, siendo el único aplicado en el campo de la robótica para la solución de este tipo de problemas de posicionamiento y optimización de trayectorias de un brazo robot. El algoritmo desarrollado lleva el problema original con el conjunto infinito de restricciones, a una sucesión de problemas clásicos de programación lineal o no lineal sobre un conjunto finito de restricciones, mediante el empleo de la función "fmincon" de MATLAB. El trabajo presenta además de los resultados obtenidos de la aplicación del método de aproximaciones externas, el modelo matemático empleado y la comparación de los resultados obtenidos con uno de los métodos más usados: el método de discretización.

## CONTENIDO

	Pág
<b>INTRODUCCION</b>	18
<b>1. FUNDAMENTOS DE ROBÓTICA</b>	21
<b>1.1 ROBOTS MÓVILES</b>	22
<b>1.2 ROBOTS MANIPULADORES</b>	23
<b>1.3 ESTRUCTURA MECÁNICA DE UN ROBOT</b>	29
<b>1.4 POSICIONAMIENTO Y ORIENTACIÓN DEL BRAZO ROBOT</b>	30
<b>1.4.1 Coordenadas polares y cilíndricas</b>	32
<b>1.4.2 Coordenadas esféricas</b>	33
<b>1.5 CINEMÁTICA DEL BRAZO ROBOT</b>	33
<b>1.5.1 Posición</b>	34
<b>1.5.2 Velocidad</b>	34
<b>1.5.3 Aceleración</b>	34
<b>1.5.4 Problema Cinemático Directo</b>	35
<b>1.5.5 Problema Cinemático Inverso</b>	35
<b>1.5.6 Matriz Jacobiana Directa</b>	36

<b>1.5.7 Matriz Jacobiana Inversa</b>	<b>36</b>
<b>1.5.8 Control Cinemática</b>	<b>37</b>
<b>1.5.9 Tipos de trayectorias</b>	<b>37</b>
<b>1.5.10 Interpolación de trayectorias</b>	<b>37</b>
<b>1.6 DINÁMICA DEL BRAZO ROBOT</b>	<b>38</b>
<b>1.6.1 Modelo Dinámico Directo</b>	<b>39</b>
<b>1.6.2 Modelo Dinámico Inverso</b>	<b>40</b>
<b>1.6.3 Control dinámico</b>	<b>40</b>
<b>1.6.4 Programación de robots</b>	<b>40</b>
<b>1.6.5 Control del movimiento del robot</b>	<b>41</b>
<b>2. EL PROBLEMA DE OPTIMIZACIÓN</b>	<b>42</b>
<b>2.1 PROCESO DE OPTIMIZACIÓN</b>	<b>45</b>
<b>2.1.1 Definición Del Problema</b>	<b>45</b>
<b>2.1.2 Creación Del Modelo</b>	<b>46</b>
<b>2.2 PROGRAMACIÓN SEMI-INFINITA</b>	<b>49</b>
<b>2.2.1 Método Discretización</b>	<b>50</b>
<b>2.2.2 Método de Intercambio (Exchange)</b>	<b>51</b>
<b>2.2.3 Métodos basados en reducción local</b>	<b>52</b>
<b>2.2.4 Método Estocástico</b>	<b>52</b>

<b>3. DESARROLLO DE LA HERRAMIENTA COMPUTACIONAL</b>	<b>54</b>
<b>3.1 ESPECIFICACIONES</b>	<b>55</b>
<b>3.2 ANÁLISIS</b>	<b>56</b>
<b>3.3 DISEÑO</b>	<b>56</b>
<b>3.4 CODIFICACIÓN</b>	<b>57</b>
<b>3.5 PRUEBAS E INTEGRACIÓN</b>	<b>57</b>
<b>3.6 MANTENIMIENTO</b>	<b>58</b>
<b>3.7 DOCUMENTACIÓN</b>	<b>58</b>
<b>4. DISEÑO DEL PROTOTIPO</b>	<b>59</b>
<b>4.1 MATLAB COMO HERRAMIENTA DE SOPORTE</b>	<b>59</b>
<b>4.2 PROPUESTA ALGORITMICA</b>	<b>60</b>
<b>4.2.1 Algoritmo Estocástico de Aproximaciones Externas</b>	<b>61</b>
<b>4.2.2 Procedimiento Sproc.Activ.Robot</b>	<b>64</b>
<b>4.2.3 Algoritmo Smeth Activ</b>	<b>65</b>
<b>4.2.4 Método Proyección de Gradiente</b>	<b>66</b>
<b>4.2.5 Modelo Matemático</b>	<b>66</b>
<b>4.2.6 Método de Optimización</b>	<b>71</b>

<b>4.3 DESARROLLO DEL ALGORITMO</b>	<b>72</b>
<b>4.3.1 Primera Versión</b>	<b>72</b>
<b>4.3.2 Segunda Versión</b>	<b>74</b>
<b>4.3.3 Tercera Versión</b>	<b>76</b>
<b>4.3.4 Cuarta Versión</b>	<b>76</b>
<b>5. PRUEBAS Y RESULTADOS</b>	<b>80</b>
<b>5.1 PRUEBAS PRELIMINARES</b>	<b>80</b>
<b>5.2 PRUEBAS DURANTE EL DESARROLLO DEL PROTOTIPO</b>	<b>82</b>
<b>5.2.1 Prueba 1</b>	<b>82</b>
<b>5.2.2 Prueba 2</b>	<b>82</b>
<b>5.2.3 Prueba 3</b>	<b>83</b>
<b>5.3 PRUEBAS CON EL PROTOTIPO FINAL</b>	<b>84</b>
<b>5.3.1 Prueba 1</b>	<b>85</b>
<b>5.3.2 Prueba 2</b>	<b>85</b>
<b>5.3.3 Prueba 3</b>	<b>86</b>
<b>5.3.4 Prueba 4</b>	<b>86</b>
<b>5.3.5 Prueba 5</b>	<b>87</b>
<b>5.4 COMPARACIÓN DE RESULTADOS</b>	<b>88</b>

<b>5.5 ANÁLISIS DE RESULTADOS</b>	89
<b>CONCLUSIONES</b>	91
<b>BIBLIOGRAFÍA</b>	94
<b>BIBLIOGRAFÍA COMPLEMENTARIA</b>	95

## LISTA DE FIGURAS

	<b>Pág</b>
Figura 1. Brazo robot típico de 6 grados de libertad	26
Figura 2. Zona de Trabajo de un Robot Industrial	27
Figura 3. Configuraciones más frecuentes en robots industriales.	28
Figura 4. Distintos tipos de articulaciones para robots.	30
Figura 5. Descripción de la Posición y la Orientación.	31
Figura 6. Representación de a) coordenadas polares y b) cilíndricas	32
Figura 7. Representación de un vector en coordenadas esféricas.	33
Figura 8. Diagrama de Relación entre Cinemática Directa e Inversa	35
Figura 9. Matriz Jacobiana Directa e Inversa	36
Figura 10. Funcionamiento del control cinemática (en sombreado).	38
Figura 11. Grados de libertad	42
Figura 12. Ciclo de vida en Cascada	55
Figura 13. Seudocódigo del Algoritmo Estocástico de Aproximaciones Externas	62

Figura 14. Diagrama de Flujo del Algoritmo Estocástico de Aproximaciones Externas	63
Figura 15. Primera Versión.	73
Figura 16. Ingreso de Restricciones al Modelo(Versión 1).	73
Figura 17. Presentación del Prototipo.	77
Figura 18. Interfaz del Algoritmo Estocástico para la Optimización de la Trayectoria de un Robot.	77
Figura 19. Carga de Datos.	78
Figura 20. Restricciones Semi-Infinitas.	81

## LISTA DE TABLAS

	Pág
Tabla 1. Pruebas del GPM.	82
Tabla 2. Pruebas con la función Fsemif.	83
Tabla 3. Pruebas con el Modelo del Robot.	84
Tabla 4. Datos de Entrada del Modelo del Robot.	84
Tabla 5. Prueba Número uno con el modelo del Robot.	85
Tabla 6. Prueba Número dos con el modelo del Robot.	85
Tabla 7. Prueba Número tres con el modelo del Robot.	86
Tabla 8. Prueba Número cuatro con el modelo del Robot.	87
Tabla 9. Prueba Número uno con puntos fuera del intervalo.	87
Tabla 10. Prueba Número dos con puntos fuera del intervalo.	88
Tabla 11. Comparación de Resultados con el Método de Discretización.	88

## INTRODUCCIÓN

En menos de 30 años la robótica ha pasado de ser un mito, propio de la imaginación de algunos autores literarios, a una realidad imprescindible en el actual mercado productivo. Desde sus inicios, entre las décadas de los setenta y ochenta, el acelerado desarrollo de la investigación y la tecnología en el tema, abonaron el terreno para lo que en los noventa se empezara a considerar como la mayoría de edad de la robótica, caracterizada por una estabilización de la demanda y una aceptación y reconocimiento pleno en la industria.

Hoy en día se ve a la robótica como un campo de trabajo mucho más amplio que el que tenía hace algunos años, abarcando investigación y desarrollo en una serie de áreas interdisciplinarias, ya que en ella participan diferentes disciplinas y tecnologías, tales como la teoría de control, la mecánica, la electrónica, la matemática aplicada y la informática, entre otras. Actualmente, los robots están integrados plenamente en la industria participando en tareas que por ser monótonas, repetitivas o de riesgo para el hombre, son realizadas de forma automática con un alto grado de precisión.

Uno de los problemas de mayor estudio dentro del campo de la robótica es sin duda la determinación de la manera óptima como el robot debe realizar sus movimientos en el trabajo para el cual fue diseñado, de manera tal, que sujeto a unas restricciones de carácter locativas, mecánicas, eléctricas entre otras, pueda ubicarse adecuadamente desde un punto inicial a un punto final donde este actuará. En otras palabras, optimizar la trayectoria o el movimiento del robot.

El principal objetivo de esta optimización es reducir al mínimo el esfuerzo requerido o maximizar la ventaja mecánica en el movimiento deseado, buscando además el mínimo tiempo en el trayecto entre un punto inicial y un punto final. Para esto, se han desarrollado dentro del estado del arte de la

programación matemática, diferentes algoritmos que permiten encontrar un óptimo a partir de una función objetivo y un conjunto de restricciones.

Actualmente los problemas de optimización de trayectorias en el campo de la robótica se han abordado especialmente con el método de discretización planteado por Elken Haaren-Retagne (Universität Trier – Alemania 1992) y retomado por Ismael Vaz (Universidade do Minho – Portugal 2003). En el trabajo de grado presentado se implementa por primera vez, el método estocástico de aproximaciones externas al problema del desplazamiento de un brazo robot. A fin de validar los resultados obtenidos se realiza una comparación entre el método de discretización encontrado comúnmente dentro del estado del arte y el método estocástico de aproximaciones externas presentado en este trabajo. Para fines de comparación este trabajo se basa en modelo matemático propuesto por Haaren-Retagne para de la trayectoria de un brazo robot específico tomado como ejemplo.

Como ocurre en el caso del establecimiento de la trayectoria óptima de un robot, en casi todas las aplicaciones prácticas de la programación matemática, el número de restricciones es enorme, existiendo algoritmos especialmente diseñados para resolver problemas con estructuras particulares. En los problemas de optimización restringida donde el conjunto factible se describe mediante un número “infinito” de restricciones (en este caso debido a que la variable a optimizar es el tiempo), se pueden emplear como estrategia de optimización conceptos como el de la “Programación Semi-infinita” (Semi-Infinite Programming).

Este documento presenta además del marco conceptual en torno al problema de optimización de la trayectoria de un brazo robot, el proceso de desarrollo de un algoritmo estocástico de aproximaciones externas basado en Programación Semi-infinita, el cual generó una herramienta computacional prototipo basada en Matlab, que permitió optimizar el tiempo total de desplazamientos de un brazo robot, de tal manera que las trayectorias de las articulaciones cumplan

con las restricciones de velocidad, aceleración, esfuerzo mecánico, entre otras. Se presentan también los resultados obtenidos del proceso de simulación de la herramienta prototipo y su comparación con los resultados del estado del arte, los cuales se convierten en un aporte al mismo, dado su carácter innovador en la manera como se aborda el problema.

## 1. FUNDAMENTOS DE ROBÓTICA

Una de las definiciones de robot más completas y más comúnmente utilizados es la propuesta por la organización Internacional para la Estandarización (ISO):<sup>3</sup>

"El robot industrial es un manipulador multifuncional, reprogramable, de posiciones o movimientos automáticamente controlados, con varios ejes, capaz de manejar materiales, partes, herramientas o instrumentos especializados a través de movimientos variables programados para la ejecución de varias tareas. Con frecuencia tienen la apariencia de uno o varios brazos que terminan en una muñeca; su unidad de control utiliza un sistema de memoria y algunas veces puede valerse de instrumentos sensores y adaptadores que responden a estímulos del medio ambiente y sus circunstancias, así como las adaptaciones realizadas. Estas máquinas multifuncionales son generalmente diseñadas para realizar funciones repetitivas y pueden ser adaptados a otras funciones sin alteraciones permanentes en el equipo".

Los robots poseen tres características que le son propias: planificación, captación de la información sensorial y aprendizaje. La captación de la información sensorial es fundamental sobre todo en el reconocimiento de formas u objetos, lo que ha dado un gran auge a las investigaciones sobre todo en el campo de la visión artificial (Visiónica). Muchas de las tareas que realizan conllevan un alto nivel de complejidad y toma de decisiones, actividades que no puede llevar a cabo un autómatas, dado que suponen principios de acción considerados "inteligentes" por lo que este ámbito se ha constituido en uno de los más importantes de la inteligencia artificial.

---

<sup>3</sup> UPM-DISAM, Robótica Industrial, Norma ISO 8373:1998, vocabulario, Disponible en Internet  
<URL:<http://www.disam.upm.es/~barrientos/Doctorado/Trasparencias/ISO%209283.pdf>>

Por otra parte, si se compara a los robots con los humanos podemos distinguir las siguientes características<sup>4</sup>:

Los robots pueden ser más fuertes, lo que les permite levantar pesos considerables y aplicar mayores fuerzas.

No se cansan y pueden trabajar fácilmente las 24 hs. del día y los siete días de la semana. No necesitan descansos y rara vez se enferman.

Son consistentes. Una vez que se han instruido para realizar un trabajo pueden repetirlo, prácticamente de forma indefinida, con un alto grado de precisión. El desempeño humano tiende a deteriorarse con el paso del tiempo.

Son casi completamente inmunes a su ambiente. Pueden trabajar en entornos extremadamente fríos o calientes, o en áreas donde existe el peligro de gases tóxicos o radiación. Manipulan objetos con temperaturas muy elevadas. Son capaces de trabajar en la oscuridad.

Se puede definir la robótica como el diseño, fabricación y utilización de máquinas automáticas programables con el fin de realizar tareas repetitivas como el ensamble de automóviles, aparatos, etc. y otras actividades. Básicamente, la robótica se ocupa de todo lo concerniente a los robots, lo cual incluye el control de motores, mecanismos automáticos neumáticos, sensores, sistemas de cómputos, etc. La robótica es una tecnología multidisciplinar, ya que ésta hace uso de los recursos que le proporcionan otras ciencias afines, solamente hay que pensar que en el proceso de diseño y construcción de un robot intervienen muchos campos pertenecientes a otras ramas de la ciencia, como pueden ser: la mecánica, la electrónica, la informática, la automática, la matemática, entre otras. Los dos tipos de robot principales son: los robots móviles y los robots manipuladores.

---

<sup>4</sup>MENDOZA, Carlos, Robótica en Mendoza, Disponible en Internet  
<URL:[http://www.roboticajoven.mendoza.edu.ar/rob\\_car3.htm](http://www.roboticajoven.mendoza.edu.ar/rob_car3.htm)>

## 1.1 ROBOTS MÓVILES

Son robots con gran capacidad de desplazamiento, basada en carros o plataformas y dotados de un sistema locomotor de tipo rodante.<sup>5</sup> Se caracterizan por un entorno fuertemente estructurado para facilitar la automatización y están provistos de patas, ruedas u orugas que los capacitan para desplazarse de acuerdo a su programación. Procesan la información que reciben a través de sus propios sistemas de sensores y se emplean en determinado tipo de instalaciones industriales, sobre todo para el transporte de mercancías en cadenas de producción y almacenes. También se utilizan robots de este tipo para la investigación en lugares de difícil acceso o muy distantes, como es el caso de la exploración espacial y las investigaciones o rescates submarinos.

## 1.2 ROBOTS MANIPULADORES

Estos robots son los más usados en la industria. Esencialmente podrían considerarse como “brazos articulados” o como comúnmente se les denomina “Brazo Robot”. Según la definición del “Robot Institute of America”, un robot industrial es un manipulador programable multifuncional diseñado para mover materiales, piezas, herramientas o dispositivos especiales mediante movimientos variados programados para la ejecución de distintas tareas<sup>6</sup>. Su sistema mecánico se compone de una serie de articulaciones, donde se distingue el brazo del robot y el efector final u órgano terminal (pinzas, garras, etc.), que es un elemento que se coloca en el extremo del último enlace del manipulador y que suministra la capacidad de agarre del objeto. Por ser este tipo de robot el elegido para el desarrollo de este proyecto, se dedicará mayor atención a su estructura y fundamentación física.

---

<sup>5</sup> AMERICARUM UNIVERSITARS, Trabajo de investigación, Disponible en Internet <URL:[http://www.pue.udlap.mx/~tesis/lis/contreras\\_o\\_l/capitulo1.pdf](http://www.pue.udlap.mx/~tesis/lis/contreras_o_l/capitulo1.pdf)>

<sup>6</sup> OLLERO, Aníbal. Robótica, Manipuladores y robots móviles. Editorial Planeta. 2001. p. 5

Un robot que no tenga un brazo o algún elemento de agarre solamente se limitaría a moverse ya sea en forma autónoma o programada y su utilidad no sería muy grande. Todo tipo de robots, desde los más simples hasta los más sofisticados ya sean para aplicaciones en ciencia, industria, investigación y desarrollo deben tener por lo menos un brazo con el propósito de agarrar, mover o manipular objetos. Es por eso que además del movimiento de la base o cuerpo, una buena parte de la ciencia de la robótica se dedica al estudio, diseño y fabricación de brazos y manos artificiales cada vez más sofisticadas y eficientes en sus movimientos desde el punto de vista de los esfuerzos mecánicos y el consumo de energía para ejecutarlos.

Al comparar un brazo robot altamente desarrollado con el brazo humano, se puede ver que el avance tecnológico aún no ha alcanzado a emular completamente a la naturaleza. Si se analiza desde el concepto de la robótica a un brazo humano, este sería un mecanismo verdaderamente sorprendente desde el punto de vista del control, la computación, la electrónica e incluso la mecánica, ya que labores cotidianas como por ejemplo agarrar un objeto, escribir palabras, manejar el teclado de un computador, tocar un instrumento musical, pintar o realizar cualquier obra de arte, son de alto modelamiento matemático para lograr artificialmente "los mismos" resultados de la naturaleza.

En robótica, el término "Grados de libertad" es uno de los más importantes de comprender y se refiere a la posibilidad de realizar diferentes tipos de movimiento; en otras palabras, el número de grados de libertad es el número de ejes de movimiento independientes. Cada grado de libertad puede ser lineal, como en el caso de una extensión telescópica o giratorio como en las articulaciones pivotantes. Un mayor número de grados de libertad conlleva un aumento de la flexibilidad en el posicionamiento del elemento terminal. Aunque la mayoría de las aplicaciones industriales requieren 6 grados de libertad, como en el caso de la industria automotriz, donde se emplean brazos robot en labores de soldadura y mecanizado, otras más complejas reciben un número

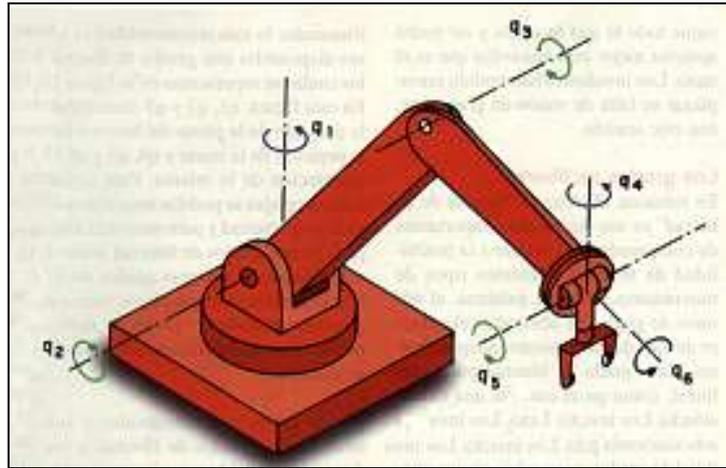
mayor de grados de libertad; tal es el caso en las labores de montaje donde se requiere acceder a lugares difíciles. Tareas más sencillas y con movimientos mas limitados, como el caso de la pintura, normalmente exigen 4 o 5 grados de libertad.

En el caso del brazo humano por ejemplo, desde el punto de vista mecánico, éste posee dos palancas: el brazo y el antebrazo, así como dos uniones o articulaciones: el hombro y el codo. En el hombro hay dos tipos de movimiento: hacia arriba y abajo, y hacia adentro y afuera. En el codo también hay dos posibilidades de movimiento: hacia adelante y atrás (flexión), y a la izquierda o a la derecha (rotación). Se puede decir entonces que el brazo humano tiene cuatro posibilidades diferentes de movimiento o cuatro grados de libertad. Esto considerando la muñeca como parte de la mano y ésta como un elemento aparte del brazo con sus propios grados de libertad. La mano y la muñeca tienen tres grados de libertad así: uno le permite girar, otro hacia atrás y adelante, y el otro cuando se abren y cierran los dedos para agarrar algo. Todo el conjunto tendrá, entonces, siete grados de libertad.

Al desempeñarse en un mundo tridimensional, un brazo robot debe tener por lo menos tres grados de libertad. Sin embargo, con estos solamente se logra llegar a un objeto pero no agarrarlo y manipularlo. Es por eso que para los robots industriales y experimentales lo más recomendable es tener disponibles seis grados de libertad los cuales se representan en la figura 1.

En figura 1,  $q_1$ ,  $q_2$  y  $q_3$  determinan la posición de la punta del brazo o sea la posición de la mano y  $q_4$ ,  $q_5$  y  $q_6$  la orientación de la misma. Para tareas más complejas se podrían necesitar más grados de libertad y para otras más simples, menos grados de libertad como 4 o 5. Cuando se agregan grados de libertad generalmente se pierde fuerza y precisión debido al "juego" o movimiento indeseado que aparece en cada articulación.

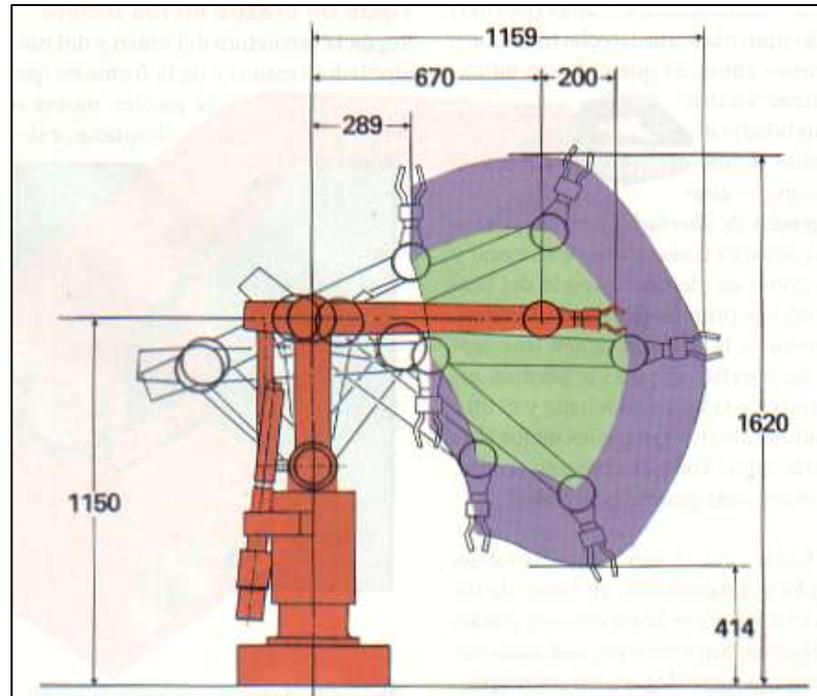
Figura 1 Brazo robot típico de 6 grados de libertad



Fuente: GONZALES Felipe, Electronica y computadores Editorial Cedit S.A edición 43. 1998 P. 21

En los robots industriales y sus modelos, los grados de libertad y las dimensiones del brazo y la mano establecen la zona de trabajo (ver Figura 2), una de las características fundamentales que se deben tener en cuenta para la selección o su diseño y fabricación. Nótese la diferencia si el brazo está recto o tiene un determinado ángulo de inclinación. Así mismo, el área o zona de trabajo se restringe de acuerdo a las limitaciones de giro en las articulaciones. Para simplificar este análisis, solamente se tendrá en cuenta la mecánica del brazo y no del cuerpo móvil del robot, si lo tiene.

Figura 2 . Zona de Trabajo de un Robot Industrial

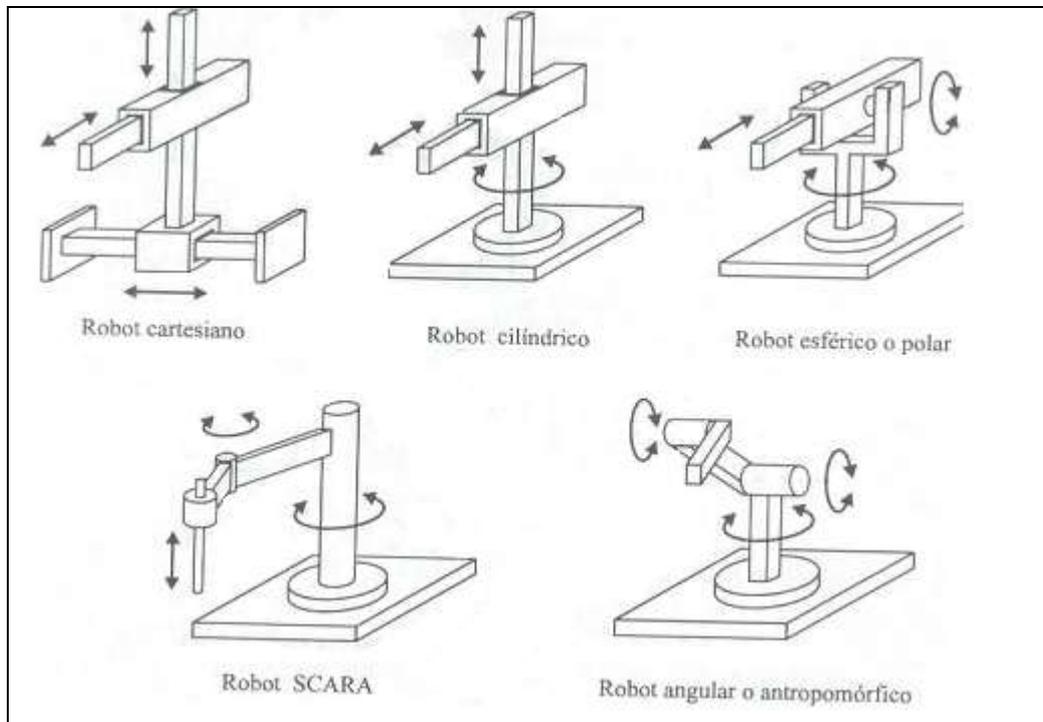


Fuente: GONZALES Felipe, Electronica y computadores Editorial Cedit S.A edición 43. 1998 P. 21

Según la estructura del brazo y del manipulador, así como la forma en que se pueden mover o desplazar existen cuatro estructuras clásicas que derivan su nombre de los correspondientes modelos de coordenadas en el espacio a las cuales pueden acceder. Cada una de estas estructuras define zonas o áreas de trabajo diferentes y son: la cartesiana, la cilíndrica, la polar y la angular o de brazo articulado.

Como se observa en la figura 3, el robot cartesiano, también llamado XYZ, realiza tres movimientos lineales en los cuales cada coordenada o posición está definida por una distancia a lo largo, otra a lo ancho y otra hacia arriba, a partir de un punto de origen.

Figura 3 Configuraciones más frecuentes en robots industriales.



Fuente: BARRIENTOS, Antonio; PEÑIN, Luis Felipe; BALAGUER, Carlos y ARACIL, Rafael. Fundamentos de Robótica. Editorial McGraw Hill, 1997. pag 16

Los robots de tipo cilíndrico, tienen en cambio del movimiento en el eje x, un movimiento rotacional en su base con lo que se obtiene un área de trabajo en forma de cilindro y de ahí su nombre. En este caso el brazo también se mueve hacia arriba y hacia abajo, y hacia adelante y atrás, permitiendo ubicar su extremo en cualquier posición dentro del cilindro de la zona de trabajo. En la mayoría de los casos, la base no gira los 360° de la circunferencia, limitando un poco el área de este cilindro.

Los robots de coordenadas polares, establecen una zona de trabajo con una forma de media esfera lo que les permite una mayor flexibilidad en cuanto a la posibilidad de llegar a más puntos que los modelos anteriores. Esto se logra

con una rotación en la base, movimiento circular de elevación del brazo y movimiento lineal de extensión del mismo. Como su nombre lo indica, para establecer una determinada posición de su extremo, se deben utilizar coordenadas polares en donde tendremos el ángulo de giro, el ángulo de elevación y el radio del brazo.

Por último, los brazos articulados o angulares, que son los más comunes, utilizan solamente articulaciones (unión de dos partes) de tipo rotacional para conseguir cualquier posición siendo el más versátil de todos los tipos y el que más se acerca al modelo humano. Sus movimientos se logran rotando la base, haciendo girar el "hombro" y estirando o encogiendo el "codo".

### **1.3 ESTRUCTURA MECÁNICA DE UN ROBOT**

“Dadas las ecuaciones del movimiento dinámico de un manipulador, el objetivo del control del robot es mantener la respuesta dinámica del mismo de acuerdo con algún criterio de funcionamiento pre-especificado.”<sup>7</sup> La solución de este tipo de problemas se hace más complejo debido a las fuerzas inerciales, fuerzas de reacción en las articulaciones y la carga de la gravedad sobre los elementos, es decir el problema del control consiste en obtener modelos dinámicos del manipulador y utilizando estos modelos determinar leyes o estrategias de control para conseguir la respuesta y funcionamiento deseado del sistema. Para minimizar el tiempo de ejecución de una determinada tarea lo ideal es mover el manipulador a la velocidad máxima permitida conduciendo el efector del manipulador de una posición inicial a una posición final.

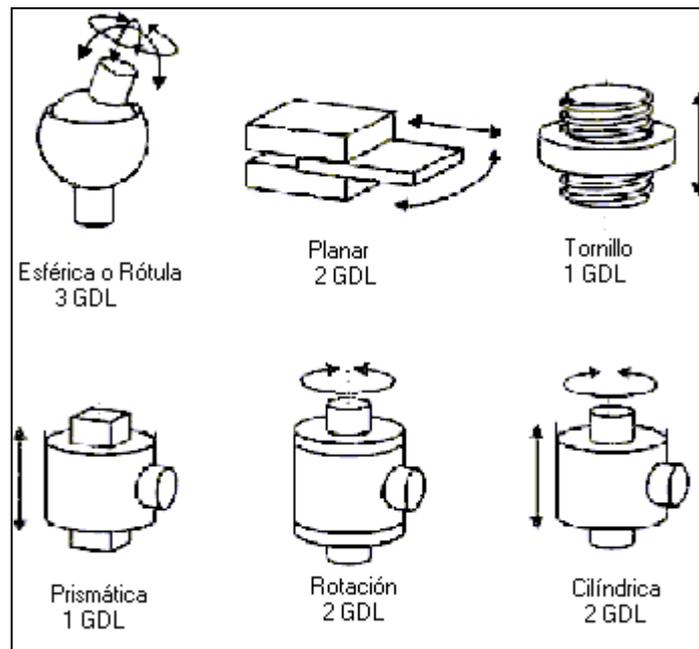
Mecánicamente un robot está formado por una serie de elementos o eslabones unidos mediante articulaciones que permiten un movimiento relativo entre cada dos eslabones consecutivos. El movimiento de cada articulación puede ser de

---

<sup>7</sup> FU. K.S; GONZALEZ, R.C; y LEE, C.S.G. Robótica: Control, Detección, Visión e Inteligencia. Editorial: McGraw Hill, Julio de 1988..p. 206

desplazamiento, de giro, o de una combinación de ambos. <sup>8</sup> Por esta razón son posibles los seis tipos diferentes de articulaciones que se muestran a continuación en la figura 4.

Figura 4 Distintos tipos de articulaciones para robots.



Fuente: BARRIENTOS, Antonio; PEÑIN, Luis Felipe; BALAGUER, Carlos y ARACIL, Rafael. Fundamentos de Robótica. Editorial McGraw Hill, 1997. pag 16

#### 1.4 POSICIONAMIENTO Y ORIENTACIÓN DEL BRAZO ROBOT

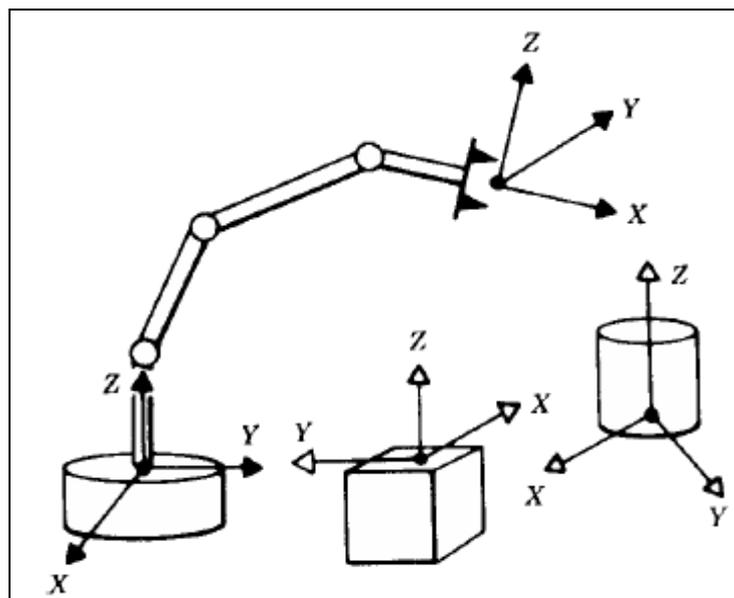
La estructura del manipulador y la relación entre sus elementos proporcionan una configuración mecánica, que da origen al establecimiento de los parámetros que hay que conocer para definir la posición y orientación del

---

<sup>8</sup> BARRIENTOS, Antoni; PEÑIN, Luis Felipe; BALAGUER, Carlos y ARACIL, Rafael. Fundamentos de Robótica. Madrid España: McGraw Hill, 1997. pp 16

elemento terminal. “En el estudio de la robótica es importante prestar una atención constante a la localización de los objetos en el espacio. Para describir la posición y la orientación de un cuerpo en el espacio se asocia siempre a un sistema de coordenadas o marco al objeto. Describimos entonces la posición y la orientación de este objeto con respecto a un sistema de coordenadas de referencia”<sup>9</sup>. Considerando el brazo robot en estudio para este proyecto, se definirán a continuación los sistemas de coordenadas mas aplicables al modelo de la trayectoria del sistema en estudio como se muestra en la figura 5.

Figura 5 Descripción de la Posición y la Orientación.



Fuente: MUÑOZ, Antonio J. Introducción a la Robótica. Disponible en: <http://www.isa.uma.es/personal/antonio/robotica/tema1%20-%20introduccion.pdf>

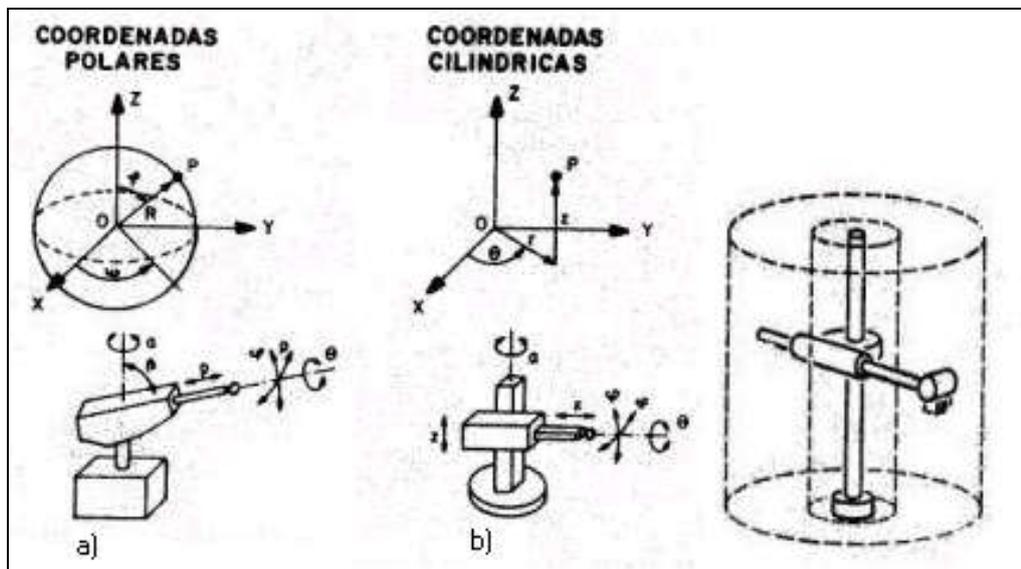
---

<sup>9</sup> MUÑOZ, Antonio J. Introducción a la Robótica. Disponible en: <http://www.isa.uma.es/personal/antonio/robotica/tema1%20-%20introduccion.pdf>

### 1.4.1 Coordenadas polares y cilíndricas

Para localizar un punto o vector  $p$  respecto a un sistema de ejes cartesianos de referencia  $OXY$  se utiliza las coordenadas polares  $p(r, \theta)$  como se puede ver en la figura 5(a) donde  $r$ , representa la distancia desde el origen  $O$  del sistema hasta el extremo del vector  $p$ , mientras que  $\theta$ , es el ángulo que forma el vector  $p$ , con el eje  $OX$ .

Figura 6 . Representación de a) coordenadas polares y b) cilíndricas



Fuente: OLLERO, Aníbal. Robótica, Manipuladores y robots móviles. Editorial Planeta. 2001. p. 5

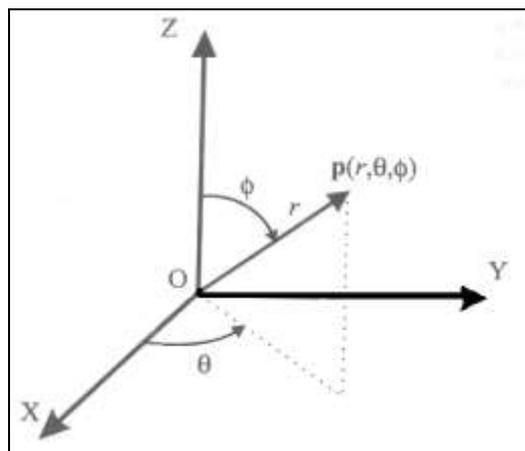
Cuando se trabajan con tres dimensiones, el vector  $p$  se puede expresar con respecto a un sistema de referencia  $OXYZ$ , mediante las coordenadas cilíndricas  $p(r, \theta, z)$  como se puede ver en la figura 5(b). Las componentes  $r$  y  $\theta$  tienen el mismo significado que en el caso de las coordenadas polares,

aplicando el razonamiento sobre el plano OXY, mientras que la componente z, expresa la proyección sobre el eje OZ del vector  $p$ .

### 1.4.2 Coordenadas esféricas

Se utiliza el sistema de referencia OXYZ, el vector  $p$  tendrá como coordenadas esféricas  $(r, \theta, \phi)$ , donde la componente  $r$  es la distancia desde el origen  $O$  hasta el extremo del vector  $p$ ; la componente  $\theta$  es el ángulo formado por la proyección del vector  $p$  sobre el plano OXYZ con el eje  $OX$ ; y la componente  $\phi$  es el ángulo formado por el vector  $p$  con el eje  $OZ$ , como se puede ver en la siguiente figura:

Figura 7 Representación de un vector en coordenadas esféricas.



Fuente: HAAREN-RETAGNE, Elke. A Semi-Infinite Programming Algorithm for Robot Trajectory Planning. Tesis Doctoral, Universität Trier, Alemania, Junio 1992.

## 1.5 CINEMÁTICA DEL BRAZO ROBOT

Cinemática es la parte de la física que estudia el movimiento de los cuerpos, sin considerar las fuerzas que lo producen, además estudia las propiedades geométricas como las temporales del movimiento.

En los modelos geométricos y cinemáticos se involucra esencialmente el estudio de las relaciones existentes entre el espacio de las variables articulares y el espacio de trabajo (espacio operacional) que suele ser un espacio cartesiano<sup>10</sup>.

Las magnitudes que define la cinemática son principalmente tres, la posición, la velocidad y la aceleración.<sup>11</sup>

### 1.5.1 Posición

Es el lugar en que se encuentra el móvil en un cierto instante de tiempo  $t$ . Suele representarse con el vector de posición  $\vec{r}$ . Dada la dependencia de este vector con el tiempo, es decir, si nos dan  $\vec{r}(t)$ , tenemos toda la información necesaria para los cálculos cinemáticos.

### 1.5.2 Velocidad

Es la variación de la posición con el tiempo. Se define como la razón entre el espacio recorrido (desde la posición  $\mathbf{x}_1$  hasta la posición  $\mathbf{x}_2$ ) y el tiempo transcurrido. La velocidad en física se corresponde al concepto intuitivo y cotidiano de velocidad.

### 1.5.3 Aceleración

Indica cuánto varía la velocidad al ir pasando el tiempo. La aceleración se define como la razón entre el cambio de velocidad y el intervalo en el cual ésta

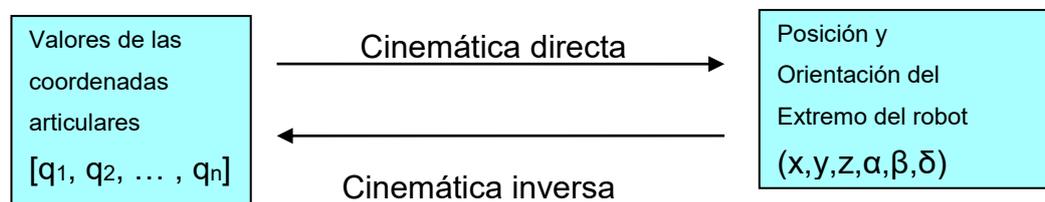
---

<sup>10</sup> Ollero, Op.cit., p 24

<sup>11</sup> MOORE, Ross, Cinemática, 1999 Disponible en Internet  
<URL:<http://bellota.ele.uva.es/~imartin/libro/node9.html>>

ocurre; es una magnitud vectorial, es decir, tiene un módulo y una dirección. El módulo define el "tamaño" que tiene la aceleración, mientras que la dirección define hacia donde apunta esa aceleración. La cinemática del robot estudia el movimiento del mismo con respecto a un sistema de referencia.<sup>12</sup> Existen dos problemas fundamentales a resolver en la cinemática del robot como se observa en la figura 8.

Figura 8 Diagrama de Relación entre Cinemática Directa e Inversa



#### 1.5.4 Problema Cinemático Directo

Consiste en determinar la posición y orientación del extremo del robot, con respecto a un sistema de coordenadas de referencia, conocidos los valores de las articulaciones y los parámetros geométricos de los elementos del robot.

#### 1.5.5 Problema Cinemático Inverso

Consiste en determinar la configuración que debe adoptar el robot para alcanzar una posición y orientación conocidas.

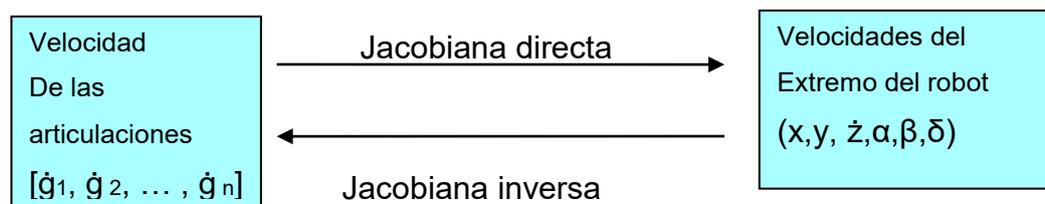
Matemáticamente las ecuaciones cinemáticas directas definen una función entre el espacio cartesiano de posiciones y orientaciones y el espacio de las articulaciones. Las relaciones de velocidad se determinan por el Jacobiano de esta función.

<sup>12</sup> Barrientos, Op.cit, p. 31

El Jacobiano es una matriz que se puede ver como la versión vectorial de la derivada de una función escalar. El Jacobiano es importante en el análisis y control del movimiento de un robot (planificación y ejecución de trayectorias suaves, determinación de configuraciones singulares, ejecución de movimientos coordinados, derivación de ecuaciones dinámicas).

El problema cinemático se resume en establecer un conjunto de ecuaciones del movimiento del brazo robot en sus articulaciones de tal modo que su extremo (actuador) alcance una posición dada. A fin de relacionar las ecuaciones del movimiento de las articulaciones con las del movimiento del actuador se emplea el Modelo Diferencial el cual se basa en un matriz Jacobiana que relaciona las velocidades del movimiento de las articulaciones y las del extremo del robot. Hay dos tipos de matriz Jacobiana, la Directa que permite conocer las velocidades del extremo del robot a partir de los valores de las velocidades de cada articulación; y la Matriz Jacobiana Inversa que permite conocer las velocidades articulares necesarias para obtener velocidades determinadas en el extremo del robot como se muestra en la figura 9.

Figura 9 . Matriz Jacobiana Directa e Inversa



### 1.5.6 Matriz Jacobiana Directa

Cuando ya se conocen las ecuaciones que resuelven el problema cinemático Directo de un robot con n grados de libertad, el método más directo para obtener la relación entre velocidades articulares y el extremo del robot, consiste en diferenciar las ecuaciones correspondientes al modelo cinemático directo como se muestra en la siguiente expresión matricial:

$$\begin{bmatrix} \dot{c}_1 \\ \vdots \\ \dot{c}_n \end{bmatrix} = J \begin{bmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_n \end{bmatrix} \quad \text{con} \quad J = \begin{bmatrix} \frac{\partial f_x}{\partial q_1} & \cdots & \frac{\partial f_x}{\partial q_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_y}{\partial q_1} & \cdots & \frac{\partial f_y}{\partial q_n} \end{bmatrix}$$

### 1.5.7 Matriz Jacobiana Inversa

Para obtener la matriz Jacobiana Inversa se invierte simbólicamente la matriz Jacobiana Directa, partiendo de las ecuaciones obtenidas del modelo Cinemático Inverso se deriva con respecto al tiempo cada elemento de la igualdad para obtener la siguiente ecuación diferencial:

$$\begin{bmatrix} \dot{c}_1 \\ \vdots \\ \dot{c}_n \end{bmatrix} = J \begin{bmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_n \end{bmatrix} \quad \text{con} \quad J^{-1} = \begin{bmatrix} \frac{\partial f_1}{\partial q_x} & \cdots & \frac{\partial f_1}{\partial q_y} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial q_x} & \cdots & \frac{\partial f_n}{\partial q_y} \end{bmatrix}$$

### 1.5.8 Control Cinemática

Establece cuales son las trayectorias que debe seguir cada articulación del robot a lo largo del tiempo para lograr los objetivos fijados por el usuario. Estas trayectorias se seleccionarán atendiendo a las restricciones físicas propias de los accionamientos y a ciertos criterios de calidad de trayectoria, como suavidad o precisión de la misma.

### 1.5.9 Tipos de trayectorias

Para realizar una tarea determinada del robot debe moverse desde un punto inicial a un punto final. Este movimiento puede ser realizado según infinitas trayectorias espaciales. Puede encontrarse que los robots dispongan de trayectorias punto a punto, coordinadas y continuas.

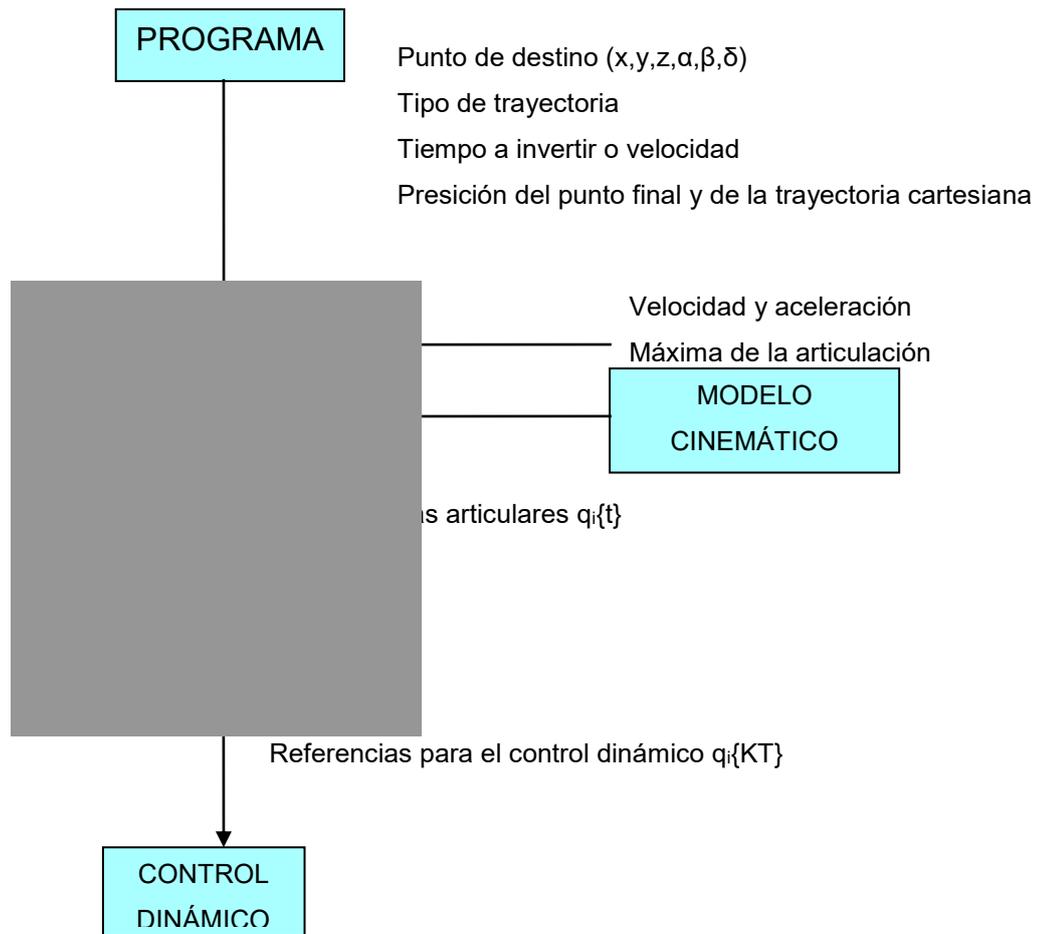
### **1.5.10 Interpolación de trayectorias**

Una de las funciones del control cinemática es la de unir una sucesión de puntos en el espacio articular por los que se quiere que pasen las articulaciones del robot en un instante determinado. Además, junto con las condiciones de posición-tiempo, es conveniente añadir restricciones en la velocidad y aceleración de paso por los puntos, de manera que se asegure la suavidad de la trayectoria y se limiten las velocidades y aceleraciones máximas. Estas restricciones garantizan que los actuadores están capacitados para implementar la trayectoria final.<sup>13</sup>

Figura 10 Funcionamiento del control cinemática (en sombreado).

---

<sup>13</sup> BARRIENTOS, Op.cit., Cap 6



## 1.6 DINÁMICA DEL BRAZO ROBOT

La dinámica se ocupa de la relación entre las fuerzas que actúan sobre un cuerpo y el movimiento que en él se origina<sup>14</sup>. El modelo dinámico de un robot tiene por objetivo conocer la relación entre el movimiento del robot y las fuerzas implicada en el mismo. Esta relación se obtiene mediante el denominado modelo dinámico, que relaciona matemáticamente:

<sup>14</sup> BARRIENTOS, Op.cit., cap 7

La localización del robot definida por sus variables articulares o por las coordenadas de localización de su extremo, y sus derivadas: velocidad y aceleración.

Las fuerzas y pares aplicados en las articulaciones (o en el extremo del robot).  
Los parámetros dimensionales del robot, como longitud, masas e inercias de sus elementos.

Debido a que la complejidad del planteamiento y obtención del modelo dinámico aumenta con el número de Grados de libertad (GDL), el modelo dinámico debe ser resuelto de manera iterativa mediante la utilización de un procedimiento numérico. El problema de la obtención del modelo dinámico de un robot es uno de los aspectos más complejos de la robótica, lo que ha llevado a ser obviado en numerosas ocasiones. Sin embargo, el modelo dinámico es imprescindible para conseguir los siguientes fines:

- Simulación del movimiento del robot.
- Diseño y evaluación de la estructura mecánica del robot.
- Dimensionamiento de los actuadores.
- Diseño y evaluación del control dinámico del robot.

El modelo dinámico completo de un robot debe incluir no sólo la dinámica de sus elementos (barras o eslabones) sino también la propia de sus sistemas de transmisión, de los actuadores y sus equipos electrónicos de mando.

A partir del planteamiento del equilibrio de fuerzas y pares que intervienen sobre el robot se obtienen los denominados modelos dinámicos directo e inverso:

### **1.6.1 Modelo Dinámico Directo**

Expresa la evolución temporal de las coordenadas articulares del robot en función de las fuerzas y pares que intervienen.

### **1.6.2 Modelo Dinámico Inverso**

Expresa las fuerzas y pares que intervienen en función de la evolución de las coordenadas articulares y sus derivadas.

En el planteamiento del equilibrio de fuerzas en un robot real de 5 o 6 grados de libertad, se deben tener en cuenta que junto con las fuerzas de inercia y gravedad, aparecen fuerzas de Coriolis debidas al movimiento relativo existente entre los diversos elementos, así como de fuerzas centrípetas que dependen de la configuración instantánea del manipulador. Como planteamiento alternativo para la obtención del modelo se puede usar la formulación Lagrangiana, basada en consideraciones energéticas, para más información hacemos referencia.<sup>15</sup>

### **1.6.3 Control dinámico**

Su misión es procurar que las trayectorias realmente seguidas por el robot sean lo más parecidas posibles a las propuestas por el control cinemática. Para ello hace uso del análisis y diseño aportadas por la teoría del servocontrol (representación interna, estado, estabilidad, control adaptativo, etc.).

Normalmente el control dinámico se realiza en el espacio articular controlando las trayectorias articulares del robot.<sup>16</sup>

### **1.6.4 Programación de robots**

Un robot industrial es básicamente un manipulador multifuncional reprogramable, permitiendo así su adaptación de una manera rápida y económica a diferentes aplicaciones. La programación de un robot se puede definir como el proceso mediante el cual se le indica a este la secuencia de acciones que deberá llevar a cabo durante la realización de su tarea.

---

<sup>15</sup> BARRIENTOS, Op.cit., p.31

<sup>16</sup> BARRIENTOS, Op.cit., p.31

### **1.6.5 Control del movimiento del robot**

Un método de programación de robots debe incluir la posibilidad de especificar el movimiento del robot. Además del punto de destino, puede ser necesario especificar el tipo de trayectoria espacial que debe ser realizada, la velocidad media del recorrido o la precisión con la que se debe alcanzar el punto destino.

17

---

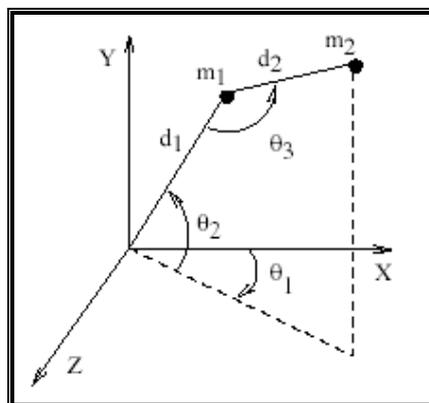
<sup>17</sup> BARRIENTOS, Op.cit., p.31

## 2. EL PROBLEMA DE OPTIMIZACIÓN

La programación matemática se ocupa de la optimización de funciones (que bien pueden representar costos, beneficios, tiempos de espera, etc) bajo todo tipo de restricciones. Esta disciplina se ha convertido en una herramienta imprescindible en la toma de decisiones (planificación de actividades económicas, diseño industrial, etc) y en la resolución de problemas de optimización que surgen en otras disciplinas matemáticas (estadística, aproximación funcional, cálculo variacional, etc). En casi todas las aplicaciones prácticas de la programación matemática, el número de restricciones es enorme, existiendo algoritmos especialmente diseñados para resolver estos problemas con estructura particular, como es el caso del problema del posicionamiento de un brazo robot.

Un robot se puede representar esquemáticamente como una conexión de puntos que representan las articulaciones. El robot representado en la figura 10, tiene 2 puntos, siendo cada punto asociado a una longitud y una masa ( $d_1$ ,  $d_2$ ,  $m_1$ ,  $m_2$ , para la longitud y masa de los 2 puntos respectivamente). En el problema del posicionamiento no se tienen en cuenta los dispositivos mecánicos que proporcionan el movimiento a los puntos de una cierta manera, tales como servomotores, motores paso a paso o cualquier tipo de actuador.

Figura 11 Grados de libertad



Fuente: HAAREN-RETAGNE, Elke. A Semi-Infinite Programming Algorithm for Robot Trajectory Planning. Tesis Doctoral, Universität Trier, Alemania, Junio 1992.

Los movimientos de un robot están relacionados con los grados de libertad; en el caso representado en la figura 9 tiene tres GDL: uno en la base del punto que proporciona rotación sobre el eje y ( $\theta_1$ ), otro que controla la abertura del primer punto ( $\theta_2$ ), y un tercero que controla la abertura en el punto dos ( $\theta_3$ ). Debido a que la posición del robot (valores de GDL) varía con el tiempo, se puede definir la trayectoria de un robot como una curva paramétrica.

$$\theta(\tau) = (\theta_1(\tau), \theta_2(\tau), \dots, \theta_l(\tau))^T \quad \tau \in [0, \mathcal{T}]$$

donde  $l$  es el número de GDL y  $\mathcal{T}$  es el tiempo total transcurrido.

Las restricciones aplicadas a la curva paramétrica son:

$$\sum_{i=1}^l \left( \frac{d\theta_i}{d\tau} \right)^2 > 0, \tau \in (0, \mathcal{T})$$

y

$$\frac{d\theta}{d\tau}(0) = \frac{d\theta}{d\tau}(\mathcal{T}) = 0$$

$$\frac{d^2\theta}{d\tau^2}(0) = \frac{d^2\theta}{d\tau^2}(\mathcal{T}) \neq 0.$$

La optimización consiste en encontrar los desplazamientos totales óptimos que encajen con la trayectoria de las articulaciones de un robot usando restricciones para la velocidad, la aceleración, entre otros.

El problema planteado ha sido resuelto por otros medios, para obtener la solución óptima que deseada. El trabajo de grado aquí presentado plantea como método de solución el tratamiento del problema como de programación semi-infinita (SIP).

Este problema se plantea como un problema de programación semi-infinita, SIP y se formula matemáticamente de la siguiente manera:

$$P(Y^0) : \quad \min_{x \in R^n} f(x)$$

$$g_i(x, y) \leq 0 \quad i = 1, \dots, k$$

$$\forall y \in Y^0,$$

donde  $Y^0$  es un conjunto infinito, por ejemplo, un producto cartesiano de intervalos de límites finitos y es un subconjunto de  $R^p$ .

El problema SIP puede formularse de manera matemática, de la siguiente manera:

$$\min \sum_{j=1}^{n-1} d_j$$

$$\text{s.t. } |Q^{i,j}(t)| \leq C_{i,1}$$

$$\text{s.t. } |Q^{i,j}(t)| \leq C_{i,2}$$

$$\text{s.t. } |Q^{i,j}(t)| \leq C_{i,3} \quad i = 1, \dots, l$$

$$d_j > 0 \quad j = 1, \dots, n-1;$$

donde  $C_{i,1}$ ,  $C_{i,2}$  y  $C_{i,3}$  son los límites de velocidad, aceleración y giro, respectivamente, en la articulación  $i$ .

$t_1 < t_2 < t_3 < \dots < t_n$  es la sucesión de tiempo donde  $t_i$  es el tiempo en la posición del robot en cuanto a la articulación  $[\theta_1(T_i), \theta_2(T_i), \dots, \theta_l(T_i)]$ .

Permitiendo que  $d_1 = t_2 - t_1$ ,  $d_2 = t_3 - t_2$ , ...,  $d_{n-1} = t_n - t_{n-1}$ ; sea el tiempo de los desplazamientos. Se plantea un robot con 3 puntos, tres grados de libertad.

Siendo  $Q_{ij}(t)$  el spline cúbico para la articulación  $i$ , se aproxima  $\theta_i(t)$  en  $[t_j, t_{j+1}]$ . Definiendo así funciones  $Q_{i,j}(t)$ , un sistema lineal más fácil de resolver.

El problema a resolver es un problema de optimización restringida donde el conjunto factible se describe con el número infinito de restricciones, se destaca por su importancia teórica y grandes aplicaciones prácticas. Estos problemas se llaman problemas de optimización (programación) semi-infinita (Semi-Infinite Programming) y tienen el proceso de desarrollo presentado a continuación.

## 2.1 PROCESO DE OPTIMIZACIÓN

Todo proceso de optimización requiere el planteamiento de cuatro etapas: <sup>18</sup>

Definición del problema.

Creación del modelo que describe o formula el sistema.

Elección de una medida de eficacia que abarque y defina el objeto que debe optimizarse (superficie de mínimos cuadrados).

Elección de un algoritmo que permita obtener el valor óptimo al aplicarlo a la resolución del modelo.

### 2.1.1 Definición Del Problema

Para la construcción de un modelo se necesita tener un conocimiento detallado tanto de cada una de las partes individuales que forman el sistema como de las interacciones existentes entre ellas, aproximándose tanto más el modelo a la realidad cuanto más detallado sea dicho conocimiento. Sin embargo, en muchos casos, para que un modelo sea manejable es necesario renunciar a algunos elementos que componen el sistema y/o a algunas de las

---

<sup>18</sup> BAEZA, Juan J., Optimización de Modelos, septiembre de 1996. Disponible en Internet <URL:<http://www.uv.es/~baeza/optimi.html> >

interrelaciones entre ellos. Si los elementos que se han conservado y las interrelaciones que se han establecido son correctas y abarcan los aspectos más relevantes del sistema, el modelo será útil. De lo contrario, si se han identificado incorrectamente los elementos, o han sido mal definidos, o si se han olvidado algunas de las relaciones importantes, el modelo estará deformado y será inútil. Podría decirse que la clave para la construcción de un modelo, radica esencialmente, en identificar de manera adecuada y sin ambigüedades los elementos cruciales, definirlos de forma precisa y operativa y establecer las principales relaciones entre ellos.

### **2.1.2 Creación Del Modelo**

La construcción del modelo tiene dos fines básicos: En primer lugar reproducir el comportamiento histórico del sistema y, en segundo, poder simular el comportamiento que el sistema tendría frente a diversas hipótesis que pueden establecerse (cambios de los parámetros que lo definen). Los elementos estructurales básicos de un modelo son las variables o magnitudes cuyo valor depende de una variable independiente, como por ejemplo el tiempo, los parámetros o constantes y las funciones o relaciones funcionales de las variables entre sí y de éstas con los parámetros.

Aparte de problemas relacionados con la robótica, existen muchos problemas los cuales se llevan a optimización semi-infinita, como el problema de aproximación de funciones, problema de los momentos, problemas de ingeniería, entre otros.

También, en práctica, muchas veces conviene trabajar con un conjunto factible en el modelo donde existe alguna dependencia de parámetro  $y$  (tiempo, espacio, etc.) que está dado con una sola desigualdad  $g(x_1, x_2, \dots, x_k; y) \leq r(y)$ , y no con una cantidad de desigualdades no relacionadas como

$g(x_1, x_2, \dots, x_k) \leq r_i$ . Es mucho mejor tanto para guardar la información, como para analizar el problema en general. En el caso cuando desde principio la descripción esta dada como en el segundo caso, los datos pueden ser llevados a la primera descripción con ayuda de una aproximación.

Uno de los métodos avanzados y muy actuales para solucionar problemas de optimización semi-infinita es el método estocástico de aproximaciones externas, construido a partir del mecanismo de las funciones, conjuntos cuasi-óptimos y del esquema de la búsqueda estocástica activa de las restricciones críticas. Este método está bien estudiado en la parte de su convergencia y aplicabilidad, por ejemplo, para el problema famoso de aproximación de Chebyshev.<sup>19</sup>

Este proyecto va dirigido a la construcción de los algoritmos del método estocástico de aproximaciones externas para la solución del problema de optimización del tiempo total de desplazamientos de un robot de tal forma que sus trayectorias de articulaciones cumplan todas las restricciones de velocidad, aceleración, entre otras.

Una de las mejores herramientas para la solución de esta clase de problemas es el método estocástico de las aproximaciones externas. Su idea principal consiste en el reemplazo del problema original  $P(Y^0)$  con la secuencia de los problemas aproximados más sencillos  $P(Y_n)$ ,  $n=1,2,\dots$ , donde cada problema  $P(Y)$  depende solo del conjunto finito  $Y$  de parámetros:

$$P(Y_n): \quad \text{encontrar } x \in \mathcal{X}_{opt}^n [Y],$$

$$\mathcal{X}_{opt}^n = \left\{ x \in \mathcal{X}_n \mid f(x) = \min_{x' \in \mathcal{X}_n} f(x') \right\},$$

---

<sup>19</sup> Volkov Y.V., Zavriev S.K., 1997. "A general Stochastic Outer Approximations Method". SIAM J. Control Optim., V. 35 (No. 4): p. 1387-1421

$$X_n = \{x \in X^0 \mid g(x, y) \leq 0, \forall y \in Y_n\}$$

$$|Y_n| < +\infty$$

Los subproblemas aproximados de las restricciones activas crecen considerablemente de manera continua, o sea, el número de restricciones de estos subproblemas  $P(Y)$  crecen muy rápido y por eso los problemas aproximados  $P(Y)$  se tornan tan complejos como el problema original  $P(Y^0)$ . Eso sucede por utilizar solo las técnicas estocásticas, las cuales empeoran la velocidad de la convergencia de los algoritmos aumentando rápidamente la cantidad de restricciones relevantes.

Por tal razón es muy importante controlar el crecimiento del número de los elementos en los conjuntos  $Y_n, n=1,2,\dots$ . Durante el desarrollo de los algoritmos estocásticos de aproximaciones externas, se ha demostrado que la combinación de las estrategias estocásticas junto con las búsquedas activas puede dar mejores resultados que la utilización de las mallas predefinidas (*a priori*) como por ejemplo la usada en los métodos de discretización.

El método estocástico de aproximaciones externas usa los mecanismos de búsqueda activa de las restricciones relevantes y la eliminación de las restricciones irrelevantes, restringiendo el número de las restricciones en los subproblemas  $P(Y)$ <sup>20</sup>. Por eso la investigación descrita, en contraste con investigaciones realizadas anteriormente con otros métodos, favorecerá, con probabilidades de éxito, a la solución del problema de optimización del tiempo total de desplazamientos de un robot.

El criterio de optimización utiliza funciones cuasi-óptimas  $\Theta(x, y): X^0 \times M_c(Y^0) \rightarrow \mathfrak{R}_+^1$ , donde  $M_c$  es el conjunto de todos los

---

<sup>20</sup> FEDOSSOVA Alina. El método estocástico de aproximaciones externas para minimización de los costos de control de contaminación del aire.2001.

subconjuntos compactos del conjunto  $Y^0$ ; teniendo como criterio de optimización  $\Theta(x, Y) \leq \varepsilon$  para evaluar qué tan cerca se encuentra el punto  $x$  de la solución del problema aproximado  $P(Y_n)$ .<sup>21</sup>

Pero las funciones cuasi-óptimas que se han venido desarrollando, se encuentran relacionadas con algún método específico de la programación matemática, como, por ejemplo, con el método de linearización de Pschenichny. Estas hendeduras que existen también enredan los algoritmos porque a parte de solucionar en cada iteración los problemas aproximados hay que designar un método definido para evaluar el punto donde estamos, o sea, hallar el valor de la función cuasi-óptima. En el proyecto propuesto intentamos construir un criterio de optimización que no dependa de ningún método específico, si no una función cuasi-óptima universal.

## 2.2 PROGRAMACIÓN SEMI-INFINITA

Un problema de optimización donde aparece un número finito de variables sometidas a restricciones infinitas se entiende como un problema de programación semi-infinita. Esta representación resulta evidentemente en las ciencias básicas relacionado con la atención a las restricciones en las variables de estado o de control del sistema durante un período de tiempo o en cualquier punto de una región geométrica.

En los problemas de aplicación los datos del modelo son generalmente estimaciones de parámetros reales (por ejemplo precios o stock) o bien aparecen errores de cálculo por aproximaciones, de tal forma que el problema que se analiza y se resuelve es, en realidad, un problema con datos perturbados<sup>22</sup>. Surge así la necesidad de estudiar la estabilidad de estos

---

<sup>21</sup> FEDOSSOVA, Op.cit., p.49

<sup>22</sup> CARRILLO Esneider y MORA Gleimer. El método Estocástico de Aproximaciones Externas para la Minimización de los Costos de Control de Contaminación del Aire. Bucaramanga, 2003, p.138-139.

problemas. La programación matemática se ocupa de la optimización de funciones (que bien pueden representar costos, beneficios, tiempos de espera, etc.) bajo todo tipo de restricciones.

En casi todas las aplicaciones prácticas de la programación matemática, el número de restricciones es enorme, existiendo algoritmos especialmente diseñados para resolver problemas de estructura particular, como los de transporte, los de redes o los de planificación de empresas multiplantas. Cuando el problema carece de una estructura que pueda ser aprovechada, su dificultad es semejante a la de los problemas de programación semi-infinita (PSI), en los que el número de restricciones es infinito (por ejemplo, puntos de una región o instantes en un período de tiempo). Este tipo de problemas surge de modo natural en problemas de decisión en ambiente de incertidumbre, en el diseño de filtros (telecomunicaciones) o en el diseño de robots, además del control de la contaminación.

Para solucionar problemas de optimización semi-infinita, se hallan diversos métodos, como: el método de discretización, el método de intercambio, el método de reducción local y el método estocástico.

### **2.2.1 Método Discretización**

En el método de discretización se emplea una malla. La malla se crea numéricamente presentándola como el conjunto coordinado de puntos constituido por las intersecciones de las líneas de un sistema de coordenadas. La característica primordial de un sistema de esta clase, es que cualquier coordenada bidimensional (o en tres dimensiones) sea

compatible con cada fragmento del límite de la zona física. El manejo de intersecciones de líneas coordenadas para definir los puntos de la malla proveen un esqueleto organizacional que permite que todos los cálculos sean efectuados en una malla cuadrada precisa.

- Trabaja con subconjuntos finitos pertenecientes al conjunto principal.
- Es muy costoso, computacional y numéricamente.
- El tiempo, necesario para comprobar la viabilidad del problema y resolverlo, crece considerablemente de acuerdo a la cardinalidad del conjunto de restricciones.
- Los nodos de la malla se eligen a priori y por eso la mayoría no son puntos activos.
- Se utiliza solo para resolver la primera etapa y generar una solución aproximada del problema.

### **2.2.2 Método de Intercambio (Exchange)**

Este Método es iterativo ya que sustituye en cada iteración restricciones infinitas por un conjunto de restricciones finitas. En este método se remueven e insertan nuevas restricciones en un conjunto de restricciones finitas de una iteración a otra. Realiza Búsqueda de puntos máximos y para ello usa el Método Simplex.

Dentro de este método en cada paso se agrega un número de nuevas restricciones y se pueden suprimir algunas restricciones anteriores. Tiene como beneficio la reducción del tiempo de cálculo por el intercambio múltiple y es empleado para problemas grandes de SIP. No obstante, presenta el inconveniente de incluir sólo restricciones de igualdades y variables positivas, además de tener un funcionamiento inestable, numéricamente hablando.

### 2.2.3 Métodos basados en reducción local

Este método usa la teoría de reducción local que consiste en la sustitución de restricciones infinitas en un problema SIP por un número finito de restricciones considerando los puntos máximos globales del problema planteado, como funciones dependientes de  $X^{23}$ .

El problema principal de este método es que se convierte en un método local, solo sirve en la parte final de la solución del problema por que necesita un estado en el cual ya tenga una solución aproximada, o por lo menos que este localizada la red de puntos mas factible y La selección de los puntos máximos globales se hace con la construcción de una red o malla uniforme de puntos.

### 2.2.4 Método Estocástico

En nuestro entorno, varias cosas se operan con sistemas, como por ejemplo los procesos bancarios, los pasajeros que tienen la posibilidad de usar un medio de transporte determinado, etc., estos sistemas no se pueden representar a través de modelos determinísticos ya que estos modelos manipulan variables aleatorias, mientras que si se utiliza un modelo estocástico o probabilístico hay al menos una variable cuyo valor se calcula en forma aleatoria.

---

<sup>23</sup> VAZ A, Ismael F. Robot trajectory Planning with Semi-infinite Programming [online].Confenecia, Paris, Sep. 2001 Citado el 20 de Sep 2004] Disponible en Internet <URL:<http://www.norg.uminho.pt/aivaz/publications.html>.>

Una de las principales herramientas para manejar el procedimiento de este tipo de problemas es el método estocástico de las aproximaciones externas. Su principal característica radica en el reemplazar el problema original  $P(Y^0)$  con una secuencia de problemas aproximados mas simples  $P(Y_n)$ ,  $n=1,2,\dots$ , donde cada problema  $P(Y)$  se sujeta solo al conjunto finito  $Y$ . Así, el problema original con el conjunto infinito de restricciones en cada iteración, se lleva a un problema clásico de programación lineal o no lineal sobre un conjunto finito de restricciones.

La descripción de este tipo de sistemas, se hace generalmente, definiendo una familia de variables aleatorias,  $\{X(t)\}$  o  $\{Xt\}$ ; donde  $X(t)$  mide, en el instante del tiempo  $t$ , el aspecto del sistema bajo consideración. Un proceso estocástico  $X = \{X(t), t \in T\}$  es una colección de variables al azar<sup>24</sup>. Es decir, para cada  $t$  en el índice  $T$  determinado,  $X(t)$  es una variable al azar. A menudo interpretamos  $t$  como tiempo y llamamos  $X(t)$  el estado del proceso en el tiempo  $t$ .

---

<sup>24</sup> CARRILLO, Op.cit., p.50



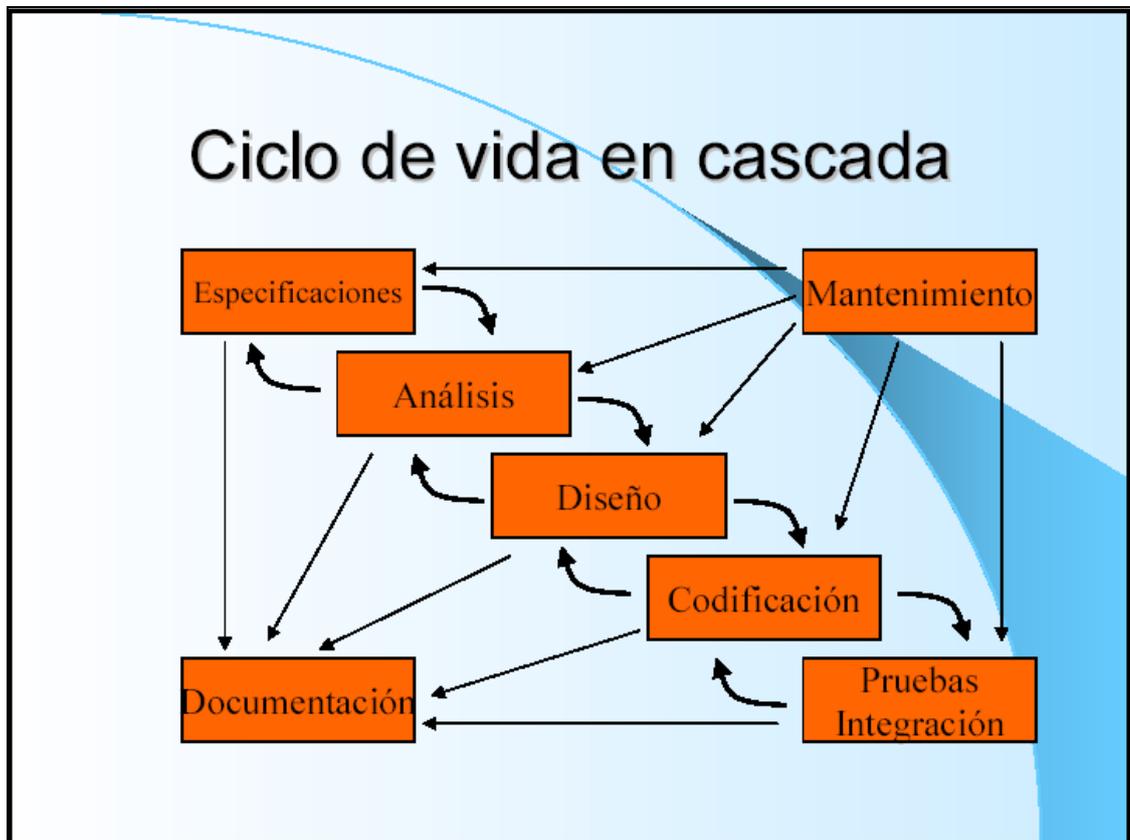
### **3. DESARROLLO DE LA HERRAMIENTA COMPUTACIONAL PROTOTIPO**

Para llevar a cabo el desarrollo del producto software de este trabajo de grado, se realizó un análisis siguiendo la metodología del ciclo de vida en cascada. Para ello, se recopiló toda la información necesaria dentro del estado del arte y se analizaron los posibles lenguajes de programación (C++, Java, Visual Basic, Delphi, entre otros), así como la herramienta computacional MATLAB, a fin de establecer las posibilidades de cada una de ellas como plataforma para la herramienta computacional prototipo, sobre la cual se demostrarían los resultados obtenidos.

Resultado de este análisis fue la conclusión de emplear MATLAB como plataforma para el prototipo, ya que a pesar de las posibilidades de creación de interfaz amigable, rapidez de cálculo y facilidades de programación de lenguajes comunes, MATLAB ofrece la ventaja de sus "toolboxes" en las cuales los algoritmos de optimización, así como otras herramientas de cálculo matemático y graficación ya están programadas. Esto permite ahorrar tiempo de programación y elimina de posibles errores matemáticos ya que se usan algoritmos probados. A partir de este razonamiento, se inicio el proceso de desarrollo del prototipo partiendo de un análisis de los requerimientos del mismo.

Debido a que nuestro proyecto requería una metodología secuencial, optamos por el modelo de desarrollo en cascada (ver figura 11) que se caracteriza porque sus fases se distribuyen en forma secuencial, una después de otra. Primero se realiza la fase de análisis. Una vez completada dicha fase, se desarrolla la fase de diseño. Luego, la fase de producción, que incluye el diseño detallado, programación y aplicación del producto. El modelo de desarrollo se llama cascada ya que su dibujo parece una cascada, en donde el producto de una fase va cayendo a la siguiente para su procesamiento.

Figura 12 Ciclo de vida en Cascada



### 3.1 ESPECIFICACIONES

Para establecer las especificaciones del prototipo, se estudió en el estado del arte dentro del tema de la optimización de la trayectoria en el desplazamiento de un robot, los aspectos deseables de una herramienta que realice el proceso de optimización. Dentro de las consideraciones tenidas en cuenta para este análisis es conveniente anotar que en el trabajo de grado presentado, no se determinarán las características mecánicas (cinemáticas y dinámicas) del movimiento, ya que la finalidad de la herramienta, es realizar la optimización de una función objetivo a partir de un modelo matemático que ya está considerando un análisis cinemático y dinámico previo. Este modelo matemático del movimiento del brazo robot es específico para un manipulador

de dimensiones y accionamientos fijos, en el cual se tiene planteada una trayectoria y el modelamiento mecánico desarrollado en el trabajo de Haaren-Retagne<sup>25</sup>. El modelo en cuestión ha sido empleado sin modificaciones para poder validar los resultados obtenidos por el algoritmo presentado en este trabajo, con respecto a trabajos anteriores, que utilizaron este mismo modelo matemático para plantear su función objetivo.

Dentro de las especificaciones de la herramienta computacional prototipo se tienen:

Facilidad de programación de los procesos matemáticos ejecutados mediante el algoritmo de optimización presentado

Rapidez de cálculo y precisión en los resultados obtenidos en el proceso de optimización

Interfaz de usuario amigable que permita realizar cambios en el modelo matemático a fin de realizar pruebas con otros tipos de brazos robot

Visualización gráfica de los resultados obtenidos en el proceso

### **3.2 ANÁLISIS**

En esta etapa se fueron presentando y aclarando los diferentes conceptos a utilizar en el proyecto. Así mismo se realizaron los diagramas de flujo para una mejor comprensión del algoritmo estocástico de aproximaciones externas, además de los métodos que este algoritmo presenta, tales como `sproc.active`, de proyección de gradiente y `smet.active`.

---

<sup>25</sup> HAAREN-RETAGNE, Elke. A Semi-Infinite Programming Algorithm for Robot Trajectory Planning. Tesis Doctoral, Universität Trier, Alemania, Junio 1992.

### **3.3 DISEÑO**

En esta fase se realiza el análisis de los métodos y la construcción de los algoritmos a utilizar en la solución del problema planteado. También se buscaron y analizaron alternativas de solución de problemas finitos y búsqueda de puntos relevantes para formar el conjunto factible de restricciones.

Para realizar el diseño de la aplicación en MATLAB se presentaron alternativas de interfaces las cuales fueron analizadas en cuanto a datos de entrada, datos de salida, información requerida para obtener los resultados esperados y las gráficas correspondientes. Así mismo, se probaron preliminarmente diferentes opciones en el código fuente para cada uno de los métodos descritos anteriormente.

### **3.4 CODIFICACIÓN**

Para implementar el algoritmo propuesto, se recurrió a las diferentes funciones y toolboxes que maneja MATLAB, lo cual hace que el proceso de programación sea menos complicado en cuanto al modelamiento matemático. No obstante, esta herramienta presenta como desventaja que su utilización requiere un aprendizaje y habilidad previa en torno a sus comandos.

### **3.5 PRUEBAS E INTEGRACIÓN**

Del proceso de codificación se generaron diferentes archivos con extensión ".m" en MATLAB los cuales fueron integrados a través de Interfaces gráficas de Usuario (GUI), a fin de conectar cada una de las etapas del proceso de optimización planteadas en el diagrama de flujo diseñado para el algoritmo estocástico de aproximaciones externas.

Para determinar si el algoritmo propuesto para la optimización de la trayectoria de un robot es eficiente y cumple con los objetivos planteados, se realizaron diferentes pruebas para verificar si la herramienta arroja resultados que además de óptimos, se acerquen e incluso mejoren los resultados vistos en trabajos similares, los cuales emplearon el método de proyección de gradiente.

### **3.6 MANTENIMIENTO**

Esta etapa se encarga de la retroalimentación para el mejoramiento o cambios que se le puedan realizar a la aplicación en un futuro, también actualizaciones de esta. Para el prototipo presentado en este trabajo, se realizó mantenimiento a partir de su primera versión, en cada una de las cuales aplicando cambios con el propósito de probar nuevas alternativas y corregir errores a fin de lograr los resultados esperados.

### **3.7 DOCUMENTACIÓN**

Al igual que el mantenimiento, esta etapa es en realidad un proceso de ejecución continua durante todo el desarrollo del prototipo. Inicia con la recopilación y análisis de la información obtenida de la investigación del estado del arte y continúa con el proceso de registro del historial del desarrollo del prototipo, a fin de consignar en él, los cambios dados del proceso de mantenimiento.

## 4. DISEÑO DEL PROTOTIPO

La optimización de la trayectoria de un robot a través de un algoritmo estocástico de aproximaciones externas, tiene grandes ventajas frente a otros métodos de optimización, destacando como principal el que realiza búsquedas activas para hallar un conjunto de restricciones factible. La implementación de este concepto de programación matemática requiere de herramientas fuertes en cuanto al dominio de las estructuras y funciones matemáticas, ventaja que puede ofrecer MATLAB.

### 4.1 MATLAB COMO HERRAMIENTA DE SOPORTE

*MATLAB* es un lenguaje de programación interpretado en el que las variables son matrices y, por tanto, las operaciones básicas aritméticas y lógicas son operaciones matriciales.

*MATLAB* es un lenguaje de programación de alto nivel, con un enfoque directo hacia la computación científica (funciones matemáticas y números complejos). Más concretamente, es un programa de cálculo numérico con gran número de instrucciones dirigidas a la resolución de problemas científicos. Desde este punto de vista puede ser considerado entonces como una gran calculadora científica programable y muy potente.<sup>26</sup>

El nombre de MATLAB proviene de las palabras inglesas *Matrix Laboratory* (laboratorio de matrices). Algunas de las aplicaciones del programa MATLAB pueden ser:

- Computación y matemáticas.
- Desarrollo de algoritmos.
- Modelado y simulación.

---

<sup>26</sup> INFANTE, Juan-Antonio, REY José María, Introducción a Matlab, 1998. Disponible en Internet <URL:<http://www.mat.ucm.es/~infante/matlab/notas.htm>>

- Exploración, visualización y análisis de datos.
- Creación de gráficas científicas.

MATLAB integra análisis numérico, matrices, procesamiento de señales y gráficas, todo esto en un ambiente donde los problemas y soluciones son expresados tal como se escriben matemáticamente. Matlab fue escrito inicialmente en Fortran y como auxiliar en la programación de cálculo con matrices, actualmente está escrito en lenguaje C.

MATLAB es un lenguaje de programación amigable al usuario con características más avanzadas y mucho más fáciles de usar que los lenguajes de programación como Basic, Pascal o C. Para ciertas operaciones es muy rápido, cuando puede ejecutar sus funciones en código nativo con los tamaños más adecuados para aprovechar sus capacidades de vectorización. Como ventajas a la utilización de Matlab se pueden considerar las siguientes:<sup>27</sup>

Su programación requiere menos tiempo que otros lenguajes como FORTRAN, C, Pascal, etc.

Utiliza un lenguaje más cercano a la matemática.

Permite definir fácil y rápidamente nuevas funciones que se incorporan a Matlab (mediante *toolboxes*)

Tiene grandes capacidades gráficas.

## 4.2 PROPUESTA ALGORITMICA

Para la definición del modelo matemático, a fin de solucionar el problema de la planeación de la trayectoria de un brazo robot, se realizó una serie de pasos que permitieran alcanzar los objetivos trazados para el trabajo de grado. Teniendo en cuenta lo anterior se desarrolló una versión del método

---

<sup>27</sup> INFANTE, Op.cit., p.61

estocástico de aproximaciones externas, que permitiera convertir el problema original que es un conjunto infinito de restricciones a un problema tradicional de programación lineal o no lineal con un conjunto finito de restricciones.

#### **4.2.1 Algoritmo Estocástico de Aproximaciones Externas**

En la figura 13 se presenta el pseudocódigo que resuelve el problema de optimización de la trayectoria de un brazo robot y en la figura 14 se presenta su diagrama de flujo.

Figura 13 Seudocódigo del Algoritmo Estocástico de Aproximaciones Externas

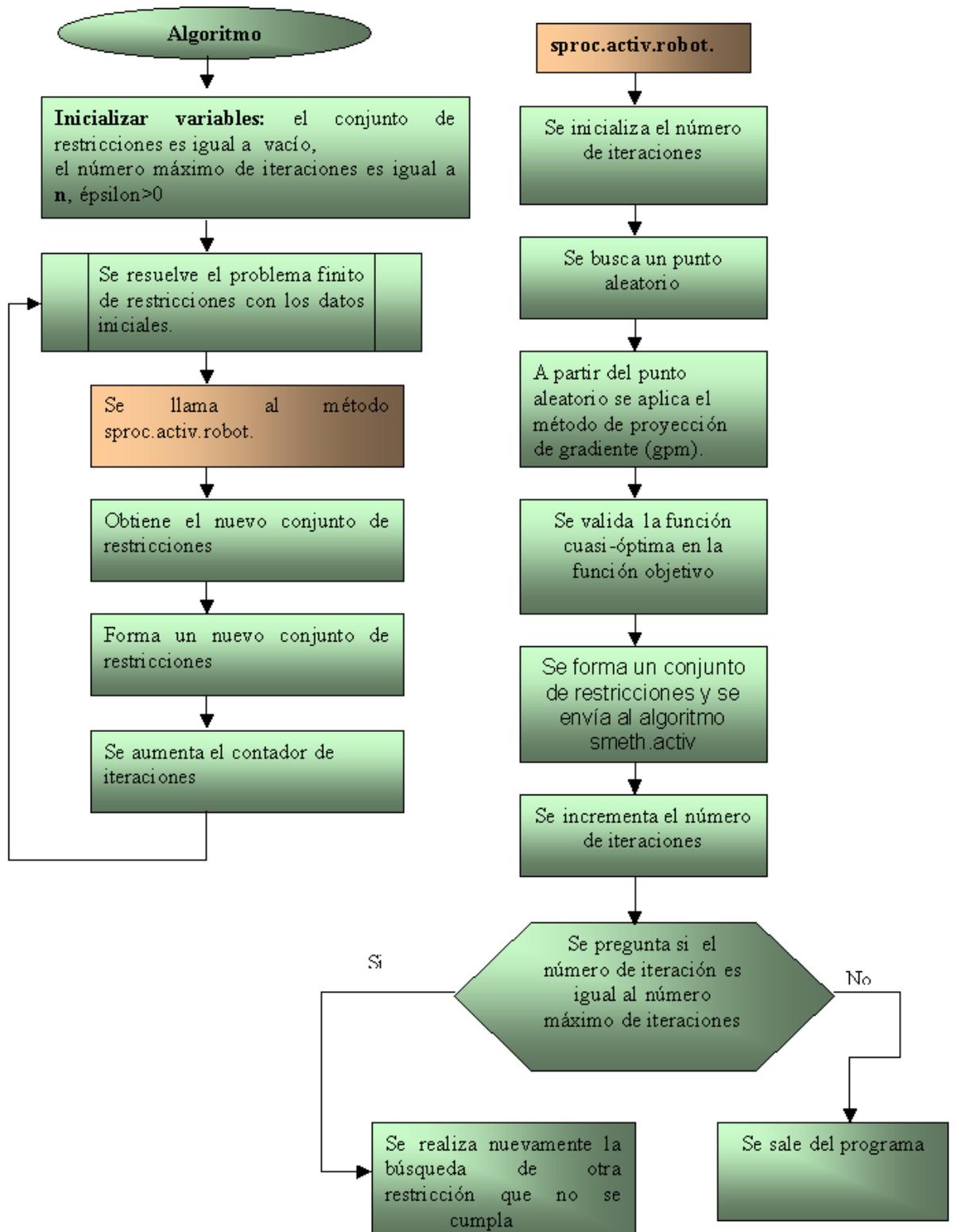
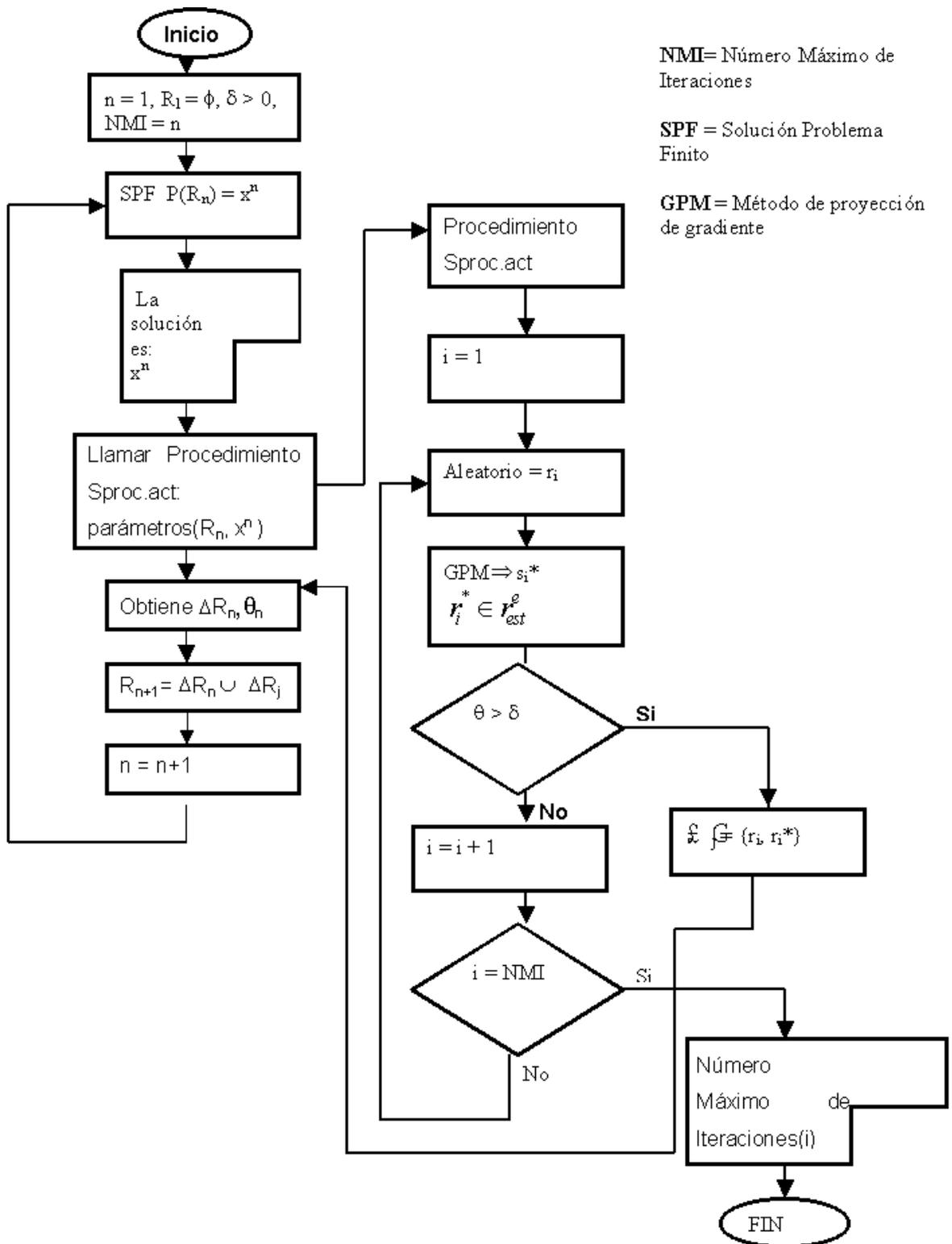


Figura 14 Diagrama de Flujo del Algoritmo Estocástico de Aproximaciones Externas.



#### 4.2.2 Procedimiento Sproc.Activ.Robot

Este procedimiento es el que se encarga de generar un punto aleatorio, averigua si este punto cumple con la restricción, si no la cumple, lo guarda en el conjunto de puntos críticos. Dentro de las características del procedimiento se tienen:

- Utiliza una búsqueda aleatoria de una restricción.
- Aplica un algoritmo de búsqueda local para encontrar las restricciones que no se cumplen resolviendo un problema de máximo global partiendo de la restricción aleatoria.
- Comprueba el cumplimiento del criterio de optimalidad del algoritmo. Si no se cumple regresa al programa principal llevando las restricciones nuevas relevantes.
- Si se cumple sigue buscando las restricciones no cumplidas.

A continuación se describe el método Sproc.Activ que se emplea en el algoritmo propuesto en el presente trabajo de grado<sup>28</sup>.

Parámetros de entrada:  $x \in X^0, Y \in M_f(Y^0)$

Parámetros de salida:  $\theta \in \mathfrak{R}_+^1, \Delta Y \in M_f(Y^0)$

Parámetro de procedimiento:  $\sigma > 0$

A continuación se describe este proceso:

**Paso 0** se inicializa  $i=0$

---

<sup>28</sup> FEDOSOVA, Op.cit., p.49

**Paso 1** Aplicar el algoritmo de la búsqueda local para la solución del problema  $\max_{y \in Y^0} g(x, y)$  empezando desde un punto aleatorio para obtener el

punto  $y_i^* \in Y^0$  tal que:  $g(x, y_i) \leq g(x, y_i^*)$ .

**Paso 2**  $i=1, \theta_i = \max(g(x, y_1^*), \dots, g(x, y_i^*))$

**Paso 3** (paso de control). Si  $i \cdot \theta_i \leq \delta$  (función cuasi-óptima) entonces  $i=i+1$  y regresa al paso 1.

**Paso 4**  $\theta := \theta_i$

$$\Delta Y := \{y_i, y_i^*\}.$$

Salida.

### 4.2.3 Algoritmo Smeth Activ

Este algoritmo resuelve un problema de optimización finita y llama el procedimiento SPROC.ACTIV.robot enviándole como parámetro la solución obtenida; así mismo, recibe del SPROC.ACTIV.robot las nuevas restricciones relevantes y forma el nuevo conjunto de restricciones.

Los siguientes pasos describen explícitamente el algoritmo Smeth.Activ.<sup>29</sup>

**Paso 0**  $n = 1,$

$Y_1 = \Phi$  conjunto inicial  $\rightarrow$  vacío

**Paso 1** encontrar  $x^n$  – solución del problema  $P(Y_n)$

**Paso 2** llamar el procedimiento esproc.ative.robot con los parámetros  $x^n$  y

$Y_n$

obtener  $\Delta Y_n$  y  $\theta_n$  (función cuasióptima)

**Paso 3** formar el conjunto nuevo de restricciones

---

<sup>29</sup> FEDOSOVA, Op.cit., p.49

$$Y_{n+1} := \Delta Y_n \cup \bigcup_{\substack{j: \theta_j > \delta/n \\ 1 \leq j \leq n-1}} \cdot$$

**Paso 4**       $n = n+1$

Regresar al paso 1

#### 4.2.4 Método Proyección de Gradiente

Este algoritmo encuentra el máximo local, mediante el siguiente procedimiento<sup>30</sup>

$$\sum_{j=1}^n (1-x_j)q_j(r) + q_0(r) \leq \varphi(r) \quad \forall r \in R,$$

**Paso 1**      Hallar el vector gradiente  $\nabla f'(x)$

**Paso 2**      Hallar punto nuevo  $X^* = (X_0 - \text{paso} F'(X_0))$

**Paso 3**      Proyectar el punto  $X^*$

**Paso 4**      Si  $F(X^*) > F(X_0)$  pasa al paso 5;

Si no disminuimos el paso y nos olvidamos de  $X^*$  y vuelve al paso 1

**Paso 5**      Si  $|F'(X^*)| \leq 0.01$  obtenemos el punto y va al paso 1;

Si no, se sigue al paso 6

**Paso 6**      Si distancia  $(X_0, X^*) \leq 0.01$  se obtiene el punto y se va al paso 1;

Si no, se sigue al paso 7

**Paso 7**      Se hace  $X_0 = X^*$  y ancho de paso = 0.7, se sigue al paso 1

---

30 CARRILLO Esneider y Mora Gleimer, 2003. "El método Estocástico de Aproximaciones Externas para la Minimización de los Costos de Control de Contaminación del Aire". Universidad Autónoma de Bucaramanga.

#### 4.2.5 Modelo Matemático

$$\begin{aligned}
 P(Y^0) &= \min \sum_{j=1}^{n-1} d_j & d &= [d_1, \dots, d_{n-1}]^T \text{ vector de intervalo entre los instantes} \\
 & & & \text{de tiempo.} \\
 \text{s.t. } |Q_{ij}(t)| &\leq C_{i,1} & C_{i,1}, C_{i,2}, C_{i,3} &= \text{limites de velocidad, aceleración y} \\
 \text{s.t. } |Q'_{ij}(t)| &\leq C_{i,2} & & \text{velocidad de aceleración en el punto i.} \\
 \text{s.t. } |Q''_{ij}(t)| &\leq C_{i,3} \quad i=1, \dots, l & Q_{ij}(t) &= \text{spline cúbico en el punto i.} \\
 d_j > 0, \quad j &= 1, \dots & &
 \end{aligned}$$

31

El modelo matemático utiliza la función B-Spline que permite aproximar una curva dados unos nodos por los cuales se debe generar la trayectoria. Una función B-Splines esta formada por una combinación de funciones lineales base representadas de la siguiente manera:

$$B_{k,\xi}(t) = \sum_{i=1}^n x_i B_{i,k,\xi}(t)$$

$$H_0 = g - \varepsilon \geq 0$$

$$H_{j,1} = 1 - \frac{(-1)^p \Theta_j'}{C_{j,1} g} \geq 0$$

$$H_{j,2} = 1 - \frac{(-1)^p \Theta_j'' - \Theta_j' \frac{g'}{g}}{C_{j,2} g^2} \geq 0$$

32  $g(x,t) = \text{Restricciones}$

$$H_{j,3} = 1 - \frac{(-1)^p \Theta_j''' - 3\Theta_j'' \frac{g'}{g} + \Theta_j' \frac{3(g')^2 - gg''}{g^2}}{C_{j,3} g^3} \geq 0$$

activas

A partir de estas funciones se desarrolló el modelo matemático:

<sup>31</sup> Tomado de: <http://bellota.ele.uva.es/~imartin/libro/node9.html> , Cinemática

<sup>32</sup> HAAREN RETAGNE, Elke. A Semi-Infinite Programming Algorithm for Robot Trajectory Planning. Junio 1992.

$$\frac{1}{K} \sum_{i=1}^n X_i (\xi_{i+k} - \xi_i) \quad n = 7, k = 4$$

$$\frac{1}{4} \left( \begin{array}{l} X_1 (\xi_5 - \xi_1) + X_2 (\xi_6 - \xi_2) + X_3 (\xi_7 - \xi_3) + X_4 (\xi_8 - \xi_4) + X_5 (\xi_9 - \xi_5) + \\ X_6 (\xi_{10} - \xi_6) + X_7 (\xi_{11} - \xi_7) \end{array} \right)$$

$$\xi_1 = 0$$

$$\xi_2 = 0$$

$$\xi_3 = 0$$

$$\xi_4 = 0$$

$$\xi_5 = 0.25$$

$$\xi_6 = 0.5$$

$$\xi_7 = 0.75$$

$$\xi_8 = 1$$

$$\xi_9 = 1$$

$$\xi_{10} = 1$$

$$\xi_{11} = 1$$

$n$  = número de coeficientes

$k$  = grado de B-spline

$nk$  = número de nodos

$t$  = parámetro de B-Spline

$\{\xi_1, \xi_2, \xi_3, \dots, \xi_{nk}\}$  = vector de nodos

$X\{1\dots n\}$  = coeficientes

$g(x,t)$  = función bspline

$$\frac{1}{4} \left( \begin{array}{l} X_1 (0.25 - 0) + X_2 (0.5 - 0) + X_3 (0.75 - 0) + X_4 (1 - 0) + X_5 (1 - 0.25) + \\ X_6 (1 - 0.5) + X_7 (1 - 0.75) \end{array} \right)$$

$$\mathbf{F.O} \rightarrow \frac{1}{4} (X_1 (0.25) + X_2 (0.5) + X_3 (0.75) + X_4 (1) + X_5 (0.75) + X_6 (0.5) + X_7 (0.25))$$

$$\left( \begin{array}{l} X_1 (0.0625) + X_2 (0.125) + X_3 (0.1875) + X_4 (0.25) + X_5 (0.1875) + \\ X_6 (0.125) + X_7 (0.0625) \end{array} \right)$$

Las Restricciones son:

$$\theta_1(T) = 1.5C(T)$$

$$\theta_2(T) = -0.5 \text{sen}(4.7C(T))$$

$$\theta_3(T) = -1.3C(T) + 2.6$$

$$C(T) = T^3 (6T^2 - 15T + 10)$$

$$\theta_1 = 1.5(T^3 (6T^2 - 15T + 10)) = 1.5(6T^5 - 15T^4 + 10T^3) = 9T^5 - 22.5T^4 + 15T^3$$

$$\theta_1' = 45T^4 - 90T^3 + 45T^2 = 45(T^4 - 2T^3 + T^2)$$

$$\theta_1'' = 180T^3 - 270T^2 + 90T = 90(2T^3 - 3T^2 + T)$$

$$\theta_1''' = 540T^2 - 540T + 90 = 90(6T^2 - 6T + 1)$$

$$\theta_1 = \frac{45(T^4 - 2T^3 + T^2)}{g(T)}$$

$$\theta_1'' = \frac{90(2T^3 - 3T^2 + T) - \frac{45(T^4 - 2T^3 + T^2)}{g(T)}}{g^2(T)}$$

$$\theta_1''' = \frac{90(6T^2 - 6T + 1) - 3 \frac{90(2T^3 - 3T^2 + T)g'}{g} + \frac{45(T^4 - 2T^3 + T^2)}{g^2} 3(g')^2 - gg''}{g^3}$$

$$\theta_2(T) = -0.5 \text{sen}(4.7(6T^5 - 15T^4 + 10T^3)) = -0.5 \text{sen}(28.2T^5 - 70.5T^4 + 47T^3)$$

$$\theta_2' = -0.5 \cos(28.2T^5 - 70.5T^4 + 47T^3) * (141T^4 - 282T^3 + 141T^2)$$

$$\theta_2' = -(70.5T^4 - 35.25T^3 + 23.5T^2) * \cos(28.2T^5 - 70.5T^4 + 47T^3)$$

$$\theta_2'' = \left( \frac{\text{sen}(28.2T^5 - 70.5T^4 + 47T^3) * (141T^4 - 282T^3 + 141T^2) *}{(70.5T^5 - 35.25T^4 + 23.5T^3)} \right) +$$

$$\left( (352.5T^4 - 141T^3 + 70.5T^2) * (-\cos(28.2T^5 - 70.5T^4 + 47T^3)) \right)$$

$$\theta_2'' = (3313.5)(3T^9 - 7.5T^8 + 7T^7 - 2.15T^6 + T^5) \text{sen}(28.2T^5 - 70.5T^4 + 47T^3) +$$

$$70.5(5T^4 - 2T^3 + T^2)(-\cos(28.2T^5 - 70.5T^4 + 47T^3))$$

$$\begin{aligned}\theta_2''' &= 313.5(27T^8 - 60T^7 + 49T^6 - 12.9T^5 + 5T^4)sen(28.2T^5 - 70.5T^4 + 47T^3) + \\ & 3313.5(3T^9 - 7.5T^8 + 7T^7 - 2.15T^6 + T^5)cos(28.2T^5 - 70.5T^4 + 47T^3) * \\ & (141T^4 - 282T^3 + 141T^2) + \\ & 70.5(20T^3 - 6T^2 + 2T)(-\cos(28.2T^5 - 70.5T^4 + 47T^3)) + 75.5(5T^4 - 2T^3 + T^2) \\ & (141T^4 - 282T^3 + 141T^2)sen(28.2T^5 - 70.5T^4 + 47T^3)\end{aligned}$$

$$\begin{aligned}\theta_2'' &= 47(27T^8 - 60T^7 + 49T^6 - 12.9T^5 + 5T^4)sen(28.2T^5 - 70.5T^4 + 47T^3) + \\ & cos(28.2T^5 - 70.5T^4 + 47T^3)47(3T^{13} - 13.5T^{12} + 25T^{11} - 23.6T^{10} + 12.3T^9 - 4.15T^8 + T^7) \\ & + cos(28.2T^5 - 70.5T^4 + 47T^3) \\ & (-20T^3 + 6T^2 - 2T) + (10T^8 - 24T^7 + 20T^6 - 8T^5 + 2T^4)sen(28.2T^5 - 70.5T^4 + 47T^3)\end{aligned}$$

$$\theta_2 = \frac{-\cos(28.2T^5 - 70.5T^4 + 47T^3)(70.5T^5 - 35.25T^4 + 23.5T^3)}{g(T)}$$

$$\theta_2' = \left[ \begin{aligned} & 3313.5(3T^9 - 7.5T^8 + 7T^7 - 2.15T^6 + T^5)sen(28.2T^5 - 70.5T^4 + 47T^3) + \\ & 70.5(5T^4 - 2T^3 + T^2) \\ & -\cos(28.2T^5 - 70.5T^4 + 47T^3) + \cos(28.2T^5 - 70.5T^4 + 47T^3) \end{aligned} \right] \frac{g'}{g^2}$$

$$\begin{aligned}\theta_2'' &= \left[ \begin{aligned} & 47(497T^8 - 1188T^7 + 989T^6 - 388.9T^5 - 99T^4)sen(28.2T^5 - 70.5T^4 + 47T^3) + \\ & 47(3T^{13} - 13.5T^{12} + 25T^{11} - 23.6T^{10} + 12.3T^9 - 4.15T^8 + T^7 - 940T^3 + 282T^2 - 94T) \\ & cos(28.2T^5 - 70.5T^4 + 47T^3) - 3((3313.5)(3T^9 - 7.5T^8 - 7T^7 - 2.15T^6 + T^5) \\ & sen(28.2T^5 - 70.5T^4 + 47T^3) + 70.5(5T^4 - 2T^3 + T^2)(-\cos(28.2T^5 - 70.5T^4 + 47T^3))) \end{aligned} \right] \frac{g'}{g} + \\ & -\left( (70.5T^4 - 35.25T^3 + 23.5T^2) * \cos(28.2T^5 - 70.5T^4 + 47T^3) \frac{3(g')^2 - gg''}{g^2} \right) \frac{g'}{g^3}\end{aligned}$$

$$\theta_3 = -1.3(6T^5 - 15T^4 + 10T^3) + 2.6 = -7.8T^5 + 19.5T^4 - 13T^3 + 2.6$$

$$\theta_3' = -39T^4 + 78T^3 - 39T^2 = 39(-T^4 + 2T^3 - T^2)$$

$$\theta_3'' = -1.3(120T^3 - 180T^2 + 60T) = 78(-2T^3 + 3T^2 - T)$$

$$\theta_3''' = -1.3(360T^2 - 360T + 60T) = 78(-6T^2 + 6T - 1)$$

$$\theta_3' = \frac{39(-T^4 + 2T^3 - T^2)}{g}$$

$$\theta_3'' = \frac{78(-2T^3 + 3T^2 - T) + 39(T^4 - 2T^3 + T^2) \frac{g'}{g}}{g^2}$$

$$\theta_3''' = \frac{78(-6T^2 + 6T - 1) + \frac{78(6T^3 - 9T^2 + 3T)g'}{g} + 39(-T^4 + 2T^3 - T^2) \frac{3(g')^2 - gg''}{g^2}}{g^3}$$

#### 4.2.6 Método de Optimización

El método de optimización utilizado es el incluido en la función `fmincon` de Matlab, el cual encuentra el valor mínimo de una función no lineal multivariable, sujeta a restricciones. La función es la siguiente:

$$\min_x f(x)$$

sujeto a:

$$c(x) \leq 0$$

$$ceq(x) = 0$$

$$A \cdot x \leq b$$

$$Aeq \cdot x = beq$$

$$lb \leq x \leq ub$$

Donde  $x$ ,  $b$ ,  $beq$ ,  $lb$  y  $ub$  son vectores;  $A$  y  $Aeq$  son matrices;  $c(x)$  y  $ceq(x)$  son funciones que retornan vectores y  $f(x)$  es la función que retorna un escalar;  $f(x)$ ,  $c(x)$  y  $ceq(x)$  pueden ser funciones no lineales.

Sintaxis

```

x = fmincon(fun,x0,A,b)
x = fmincon(fun,x0,A,b,Aeq,beq)
x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub)
x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon)
x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon,options)
[x,fval] = fmincon(...)
[x,fval,exitflag] = fmincon(...)
[x,fval,exitflag,output] = fmincon(...)
[x,fval,exitflag,output,lambda] = fmincon(...)
[x,fval,exitflag,output,lambda,grad] = fmincon(...)
[x,fval,exitflag,output,lambda,grad,hessian] = fmincon(...)

```

La función **Fmincon** resuelve problemas de la forma:

Min  $F(x)$  sujeto a:

$A * X \leq B, Aeq * X = Beq$  (restricciones lineales)

$C(X) \leq 0, Ceq(X) = 0$  (restricciones no lineales)

$LB \leq X \leq UB$  (limites)

$X = \text{fmincon}(\text{FUN}, X_0, A, B)$  comienza en  $X_0$ ; encontrar un mínimo  $X$  de la función  $\text{FUN}$ , sujeta a las desigualdades lineales  $A * X \leq B$  ( $X$  es un vector)

$X = \text{FMINCON}(\text{FUN}, X_0, A, B, Aeq, Beq, LB, UB, \text{NONLCON})$

Sujeta a minimizar las restricciones definidas en la función  $\text{NONLCON}$ ; esta función acepta  $X$  el retorno de vectores  $C$  y  $Ceq$  que representan desigualdades e igualdades no lineales, respectivamente. En el caso en el que no exista ninguna restricción se toma como " $[]$ ".

### 4.3 DESARROLLO DEL ALGORITMO

Para el desarrollo del prototipo se fueron realizando diferentes versiones las cuales se iban adaptando de acuerdo a los requisitos y necesidades del trabajo de grado, teniendo en cuenta la facilidad de manejo del ingreso de los datos para el usuario final.

A continuación se presenta el desarrollo de las diferentes versiones desarrolladas a lo largo del trabajo de grado.

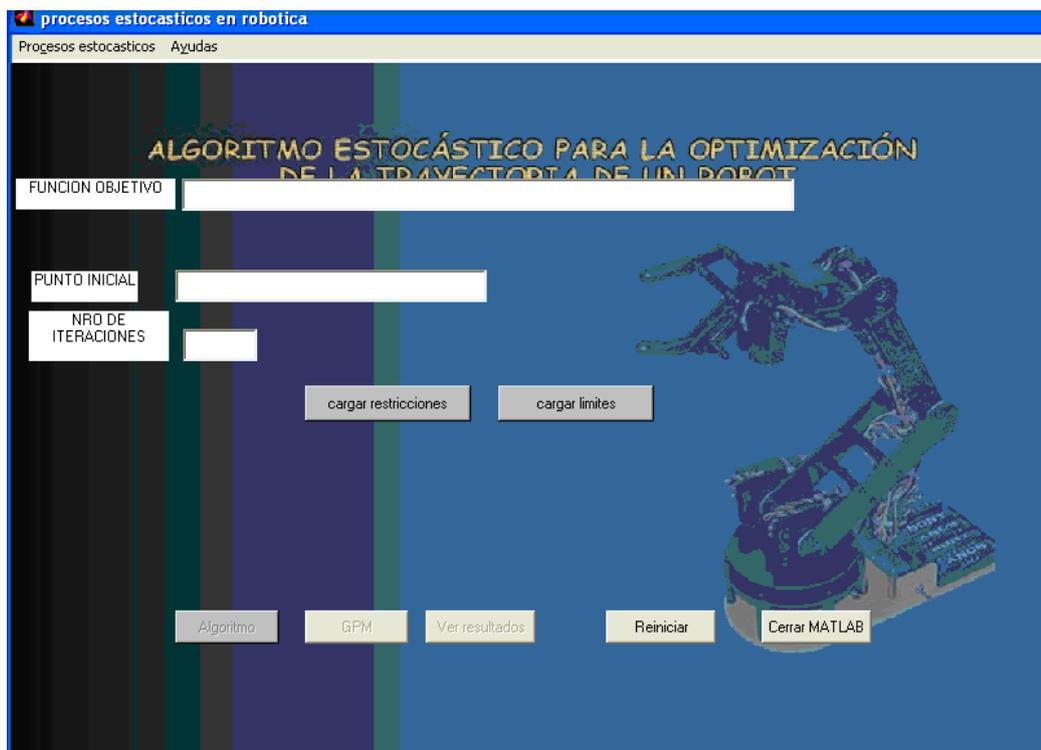
### 4.3.1 Primera Versión

Como punto de partida se revisa el mecanismo de proyección del gradiente el cual inicialmente se trabaja de manera lineal.

El punto que no se halla en el área solución se proyecta en ella mediante la aproximación al valor superior o inferior de acuerdo al límite que incumpla mediante la proyección por medio de líneas. Los resultados son aceptables para modelos lineales “simples” sujetos a restricciones de igual tipo, pero presenta fallas a medida que se incrementa el grado en las restricciones y/o función objetivo.

Inicialmente la interfaz del prototipo tenía la siguiente presentación:

Figura 15 Primera Versión.



Esta presentación incluye el ingreso de la función objetivo de manera manual por parte del usuario, así como el punto inicial y el número de restricciones;

estos datos debían ingresarse a través de otra interfaz como se muestra en la siguiente figura:

Figura 16 Ingreso de Restricciones al Modelo(Versión 1).

The screenshot shows a software window titled 'restricciones' with a blue header bar. The main content area is titled 'RESTRICCIONES DEL MODELO'. It contains a list of input fields for constraints: 'Restricción base', 'Restricción 1', 'Restricción 2', 'Restricción 3', 'Restricción 4', 'Restricción 5', 'Restricción 6', 'Restricción 7', and 'Restricción 8'. Below these is an 'INTERVALO (para las variables diferentes a la básica)' field with the value '100'. At the bottom center is an 'Aceptar' button.

En el campo “Restricción base” se ingresa la función común de las restricciones del modelo y para el campo “Intervalo” se da la opción de cambiar el valor del intervalo de las variables. Los demás campos (Restricciones) se debía ingresar las restricciones con la sintaxis de MATLAB, lo cual era muy tedioso y requería un tiempo considerable. Debido a estos inconvenientes se implementó que los datos se cargaran a través de un archivo .txt.

#### 4.3.2 Segunda Versión

Se reinicia la revisión bibliográfica en búsqueda de un estudio relacionado con las posibles soluciones a las fallas ocurridas durante la proyección del gradiente. Se implanta en el algoritmo la metodología sugerida por S. Rao en Optimized Engineering. En este algoritmo la proyección se basa en la

evaluación de los puntos iniciales en el área y en las restricciones. Los pasos que sigue este método son:

1. Se inician las iteraciones en 1

2. Si el punto esta en el área se define una dirección de búsqueda definida por

$$S_i = \frac{-\nabla f(X_i)}{\|\nabla f(X_i)\|}$$

se pasa al paso 4. Si una de las restricciones evaluadas en el punto es cero se pasa al paso 3.

3. Calculo de la matriz de proyección

$$P = 1 - Np(Np^T Np)^{-1} Np^T$$

donde:

$$Np = [\nabla g_{j1}(X_i) \quad \nabla g_{j2}(X_i) \quad \dots \quad \nabla g_{jn}(X_i)]$$

4. Adicionalmente se calcula la dirección de búsqueda como

$$S_i = \frac{-P_i \nabla f(X_i)}{\|P_i \nabla f(X_i)\|}$$

5. Se prueba el valor de la dirección de búsqueda en caso de ser cero se calcula

$$\text{Lambda} = -(Np^T Np)^{-1} Np^T \nabla f(X_i)$$

En caso contrario se pasa a 6. Si las componentes de Lambda son mayores a cero se obtuvo el punto optimo. Si alguna de las componentes es negativa se busca la más negativa y se forma la nueva Np

$$Np = [\nabla g_{j1}(X_i) \quad \nabla g_{j2}(X_i) \quad \dots \quad \nabla g_{jq-1}(X_i) \quad \nabla g_{jq+1}(X_i) \quad \nabla g_{jp}(X_i)]$$

Y se regresa a 3

Si es cero se calcula la máxima longitud permisible sin violar las restricciones, encontrando el mínimo  $\lambda$  y la derivada de la  $f_0$  respecto a  $\lambda$  definiéndola como:

$= S_i^T \nabla f(X_i + \lambda_M S_i)$  si dicha función es negativa la longitud del paso será el valor obtenido, en caso contrario se minimizara el paso mediante interpolación

La siguiente aproximación será

$X_{i+1} = X_i + \lambda_i S_i$  De manera similar a los otros desarrollos se obtienen resultados satisfactorios en caso de funciones objetivos de orden menor a tres, en caso contrario luego de la tercera iteración comienza a divergir. Hasta este punto se prueba de manera independiente los diversos módulos enfocando la mayor parte de los esfuerzos al correspondiente a la proyección del gradiente. La entrada de datos se hace mediante una línea de comandos embebida en una plantilla grafica.

#### **4.3.3 Tercera Versión**

Como solución se recurre a las fuentes iniciales del trabajo desarrollado por Rosen respecto a las restricciones no lineales, implementándose su versión de proyección para dichos casos. La idea básicamente se plantea como una linealización de las restricciones mediante su derivación, ponderación y posterior evaluación, como mecanismo de parada se plantean los criterios KT. Como producto de dicha linealización en los problemas con restricciones senoidales se presentan grandes variaciones respecto a las diferentes pruebas. Posteriormente se remueven los criterios de KT y se implementan los criterios de minimización de la función, similitud entre las diversas salidas, los resultados en caso de las restricciones senoidales dejan de oscilar. En esta versión se realizaron pruebas con la interfaz del prototipo.

#### **4.3.4 Cuarta Versión**

En esta versión se incorporó el algoritmo final. Este algoritmo se basa en la siguiente fórmula:  $X(i+1)=X(i) \pm \alpha \nabla f(X(i))$  El caso de la maximización contempla el uso del signo más en caso contrario se usa el menos.

La proyección de los puntos externos se realiza mediante

$$X(i) = \frac{X(i)}{\|X(i)\|}$$

El  $\alpha$  se divide en dos si el punto hallado no se halla dentro del área de la solución. Los criterios de parada se conservan como en el caso anterior. En esta versión se diseñó la interfaz definitiva de carga de datos con el fin de facilitar el trabajo de un usuario con el prototipo. Como se muestra en las siguientes figuras:

Figura 17 . Presentación del Prototipo.



Figura 18 Interfaz del Algoritmo Estocástico para la Optimización de la Trayectoria de un Robot.



Figura 19 Carga de Datos.



**Versión 4.1** en esta versión se realizaron pruebas independientes del smeth.active, carga de archivos y derivadas de las restricciones, adicionalmente se realizó la inclusión del método de interpolación.

**Versión 4.2** en esta versión se creó el archivo de soluciones para el prototipo y se realizaron varias pruebas para verificar el correcto funcionamiento del algoritmo.

## 5. PRUEBAS Y RESULTADOS

Se realizaron diferentes pruebas para poder verificar si el algoritmo implementado genera resultados que además de ser reales se espera que logren un mayor grado de minimización con respecto al método de proyección de gradiente.

Inicialmente se realizaron unas pruebas preliminares con el objeto de obtener resultados ya conocidos para que fueran objeto de comparación con el algoritmo planteado.

### 5.1 PRUEBAS PRELIMINARES

Estas pruebas fueron realizadas para ver el funcionamiento en la Herramienta a utilizar, MATLAB, en relación a problemas de gradiente. A continuación se presenta la prueba realizada con un problema de optimización que utiliza el método de gradiente, donde encuentra el mínimo de una función no lineal de infinitas restricciones:

Planteamiento:

Encontrar los valores de  $x$  que minimice

$$\min_{x \in R^3} f(x) = (x_1 - 0.5)^2 + (x_2 - 0.5)^2 + (x_3 - 0.5)^2$$

donde

$$K_1(x, w_1) = \sin(w_1 x_1) \cos(w_1 x_2) - \frac{1}{1000} (w_1 - 50)^2 - \sin(w_1 x_3) - x_3 - 1 \leq 0$$

$$K_2(x, w_2) = \sin(w_2 x_2) \cos(w_2 x_1) - \frac{1}{1000} (w_2 - 50)^2 - \sin(w_2 x_3) - x_3 - 1 \leq 0$$

$\forall w \in [1, 100], [1, 100]$   $w$ , de dos coordenadas

( $K_1, K_2$ : La primera y segunda restricción semi-infinita en  $x$  y  $w$  ;  $w$ : conjunto de datos).

Resultados:

Resolviendo este problema en Matlab se obtiene la siguiente gráfica (ver Figura 16).

Digitándole a  $x = [0.5; 0.2; 0.3]$ , se obtiene después de 8 iteraciones:

$x =$

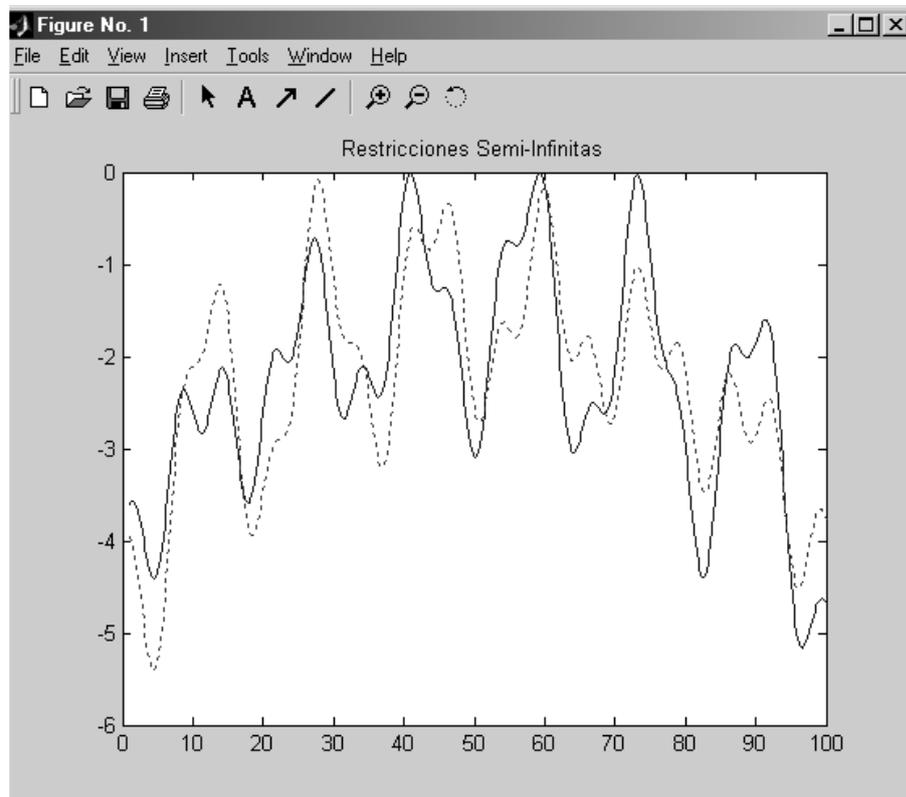
0.6673

0.3013

0.4023

$fval = 0.0770$

Figura 20 . Restricciones Semi-Infinitas.



## 5.2 PRUEBAS DURANTE EL DESARROLLO DEL PROTOTIPO

Las pruebas realizadas en el proceso de desarrollo del prototipo desarrollado arroja los siguientes resultados:

### 5.2.1 Prueba 1

Se usó un modelo identificado con el nombre funcion3.txt realizando la prueba de la opción del GPM sin generar un primer punto randómicamente para comprobar el funcionamiento externo del algoritmo.

Planteamiento:

Función objetivo =  $\exp(x(1)) \cdot (4 \cdot x(1)^2 + 2 \cdot x(2)^2 + 4 \cdot x(1) \cdot x(2) + 2 \cdot x(2) + 1)$

Punto inicial = -1;1

Numero máximo de iteraciones = 100

Cantidad de restricciones = 2

Restricciones =  $1.5 + x(1) \cdot x(2) - x(1) - x(2)$   
 $-x(1) \cdot x(2) - 10$

X = números pertenecientes a las variables de decisión de la función objetivo

Tabla 1 Pruebas del GPM.

	X(1)	X(2)
Prueba 1	0.50000349990676	-1.00000872178081
Prueba 2	-1.25751560882001	0.48496878235998
Prueba 3	-1.07896979596795	0.59347061541585
Prueba 4	-1.00334764119993	0.59001997299872
Prueba 5	-0.96704479548601	0.57889753019355
Prueba 6	-0.94889205882058	0.57145531134384
Prueba 7	-0.93977316193403;	0.56730319278781

### 5.2.2 Prueba 2

Pruebas con el modelo llamado fseminf.txt

Planteamiento:

$$\text{Función objetivo} = (x(1)-.5)^2+(x(2)-.5)^2+(x(3)-.5)^2$$

$$\text{Función objetivo} = \exp(x(1))*(4*x(1)^2+2*x(2)^2+4*x(1)*x(2)+2*x(2)+1)$$

Punto inicial = .5;.2;.3

Numero máximo de iteraciones = 100

Cantidad de restricciones = 2

Restricciones =

$$\sin(w1*x(1))*\cos(w1*x(2))-(1/1000*(w1-50)^2)-\sin(w1*x(3))-x(3)-1$$

$$\sin(w2*x(2))*\cos(w2*x(1))-(1/1000*(w2-50)^2)-\sin(w2*x(3))-x(3)-1$$

Tabla 2 Pruebas con la función Fseminf.

	X(1)	X(2)	X(3)
Prueba 1	0.50000000444888	0.50000000122985	0.50000000035730
Prueba 2	0.50000000000000	0.62000000000000	0.58000000000000
Prueba 3	0.50000000000000	0.53600000000000	0.52400000000000
Prueba 4	0.50000000000000	0.52340000000000	0.51560000000000
Prueba 5	0.50000000000000	0.51930500000000	0.51287000000000

### 5.2.3 Prueba 3

Pruebas con el modelo del Robot (caso de estudio)

Planteamiento:

Función objetivo

$$=.0625*x(1)+.125*x(2)+0.1875*x(3)+.25*x(4)+.1875*x(5)+.125*x(6)+.0625*x(7)$$

Punto inicial = 0;0;1;1;0;1;1

Numero máximo de iteraciones = 100

Cantidad de restricciones = 3

$$\text{Restricciones} = 1.5*t^3*(6*t^2-15*t+10)$$

$$-0.5 \sin(4.7t^3(6t^2-15t+10))$$

$$-1.3t^3(6t^2-15t+10)$$

Los valores arrojados por el prototipo son los valores que representan la trayectoria del movimiento

Tabla 3 Pruebas con el Modelo del Robot.

	Prueba 1	Prueba 2	Prueba 3
X(1)	0.03439440374295	0.00305689237926	0.01684745814935
X(2)	0.00602048414490	0.00265395034003	0.01601653604124
X(3)	.04902076584568	0.00376556898866	0.03676613507072
X(4)	0.00031414415515	0.00032237851808	0.01435579252913
X(5)	0.04020820779641	0.00205356817339	0.04436620966064
X(6)	0.03125764435870	0.00023051267809	0.00000000000000
X(7)	0.39743192925678	0.21652240745158	0.46977859934039
F.o.	0.04845788047211	0.01525594696410	0.05121745836768

### 5.3 PRUEBAS CON EL PROTOTIPO FINAL

Teniendo en cuenta los valores planteados por investigadores anteriores (usando el método de discretización) se usaron los siguientes datos del modelo:

Tabla 4 Datos de Entrada del Modelo del Robot.

	$\theta_1$			$\theta_2$			$\theta_3$		
	C <sub>11</sub>	C <sub>12</sub>	C <sub>13</sub>	C <sub>21</sub>	C <sub>22</sub>	C <sub>23</sub>	C <sub>31</sub>	C <sub>32</sub>	C <sub>33</sub>
A	2	8	250	3	18	650	4	50	1000
B	1	3	100	1	3	100	1	3	100

C	2	8	25	3	18	65	4	50	100
---	---	---	----	---	----	----	---	----	-----

Mod → modelo trabajado (datos de la tabla)

$\varepsilon$  → valor del epsilon

F.o. → valor de la función objetivo en el punto óptimo

Con este modelo se realizaron varias pruebas que se muestran a continuación:

### 5.3.1 Prueba 1

Punto inicial: 0, 0, 0.3, 1, 0, 0.8, 1

Tabla 5 Prueba Número uno con el modelo del Robot.

Mod	$\varepsilon$	Punto optimo				F.o.
A	0.01	0.25477	0.8656	0.23235	0.80487	0.58318
		0.9084	0.23189	0.23931		
A	0.005	0.90953	0.27026	0.085751	0.05992	0.20209
		0.11078	0.22371	0.50676		
B	0.01	0.054979	0.65469	0.18407	0.64423	0.45441
		0.48881	0.52326	0.26408		
B	0.005	0.0081469	0.61115	0.52706	0.85994	0.52088
		0.42963	0.31804	0.15777		
C	0.01	0.83793	0.68066	0.12166	0.37471	0.40007
		0.48518	0.029831	0.82278		
C	0.005	0.3837	0.61091	0.67447	0.37257	0.52458
		0.38902	0.63881	0.82938		

### 5.3.2 Prueba 2

Punto inicial: 0.1, 0.2, 0.25, 0.1, 0.8, 0.3, 0.4

Tabla 6 Prueba Número dos con el modelo del Robot.

Mod	$\varepsilon$	Punto optimo				F.o.
A	0.01	0.28958	0.26119	0.34857	0.31964	0.37103
		0.47045	0.34725	0.69437		
A	0.005	0.40458	0.016373	0.44634	0.72841	0.48846
		0.55234	0.53404	0.4002		
B	0.01	0.010997	0.49449	0.58601	0.84744	0.61096
		0.63567	0.66343	0.39359		
B	0.005	0.047946	0.18928	0.28603	0.35099	0.39767
		0.78594	0.51422	0.28794		
C	0.01	0.40588	0.54406	0.83685	0.52517	0.45028
		0.13248	0.27614	0.14958		
C	0.005	0.93412	0.32535	0.37719	0.28308	0.37091
		0.38696	0.26953	0.38592		

### 5.3.3 Prueba 3

Punto inicial: 0, 0, 0.3, 1, 0, 0.8, 1

(Contando las restricciones activas → R.Act.)

Tabla 7 Prueba Número tres con el modelo del Robot.

Mod	$\varepsilon$	Punto optimo			R.Act.	F.o.
A	0.01	0.44488	0.69457	0.62131	4	0.72957
		0.79482	0.95684	0.52259		
		0.88014				
A	0.005	0.17296	0.97975	0.27145	2	0.51215
		0.25233	0.87574	0.73731		
		0.13652				
B	0.01	.064781	0.98833	0.58279	3	0.5082
		0.4235	0.51551	0.33395		
		0.43291				
B	0.005	0.050269	0.41537	0.305	4	0.49033

		0.87437 0.97084	0.015009	0.76795		
C	0.01	0.01576 0.58692 0.63145	0.016355 0.057581	0.19007 0.36757	7	0.28161
C	0.005	0.71588 0.25477 0.80487	0.89284 0.8656	0.2731 0.23235	4	0.5129

### 5.3.4 Prueba 4

Punto inicial: 0.1, 0.2, 0.25, 0.1, 0.8, 0.3, 0.4

(Contando las restricciones activas → R.Act.)

Tabla 8. Prueba Número cuatro con el modelo del Robot.

Mod	$\varepsilon$	Punto óptimo			R.Act.	F.o.
A	0.01	0.66493 0.13701 0.89032	0.87038 0.81876	0.0099273 0.43017	6	0.4494
A	0.005	0.41986 0.91996 0.6208	0.75367 0.84472	0.79387 0.36775	6	0.74244
B	0.01	0.56677 0.99945 0.36031	0.82301 0.96164	0.67395 0.058862	7	0.72471
B	0.005	0.053444 0.43444 0.11334	0.35666 0.56246	0.4983 0.61662	6	0.43959
C	0.01	0.62623 0.088962 0.47404	0.53685 0.27131	0.059504 0.40907	3	0.27128
C	0.005	0.95612	0.59555	0.028748	3	0.55047

		0.81212	0.61011	0.70149		
		0.092196				

### 5.3.5 Prueba 5

Prueba con valores del punto inicial fuera del intervalo [0,1]

Esta prueba se realizo para comprobar el funcionamiento de la restricción del intervalo, para comprobar que desde cualquier punto inicial que se ingrese lo va a proyectar dentro del intervalo válido.

Punto inicial 1 → 2.5;4;1.8;0.5;3.2;0.9;1.2

Tabla 9. Prueba Número uno con puntos fuera del intervalo.

Mo d	Punto optimo:	R.Act.	F.o.	$\varepsilon$
A	0.018504 0.82141 0.4447 0.61543 0.79194 0.92181 0.73821	3	0.65093	0.01
B	0.35287 0.81317 0.0098613 0.13889 0.20277 0.19872 0.60379	3	0.26087	0.01
C	0.37837 0.86001 0.85366 0.59356 0.49655 0.89977 0.82163	6	0.69653	0.01

Punto inicial 2 → 10;20;18.2;13.5;9.6;8.4;6.9

Tabla 10. Prueba Número dos con puntos fuera del intervalo.

Mo d	Punto óptimo:	R.Act.	F.o.	$\varepsilon$
A	0.050269 0.41537 0.305 0.87437	4	0.49033	0.01

	0.015009	0.76795	0.97084			
	0.01576	0.016355	0.19007	0.58692		
B	0.057581	0.36757	0.63145		7	0.28161
	0.71588	0.89284	0.2731	0.25477		
C	0.8656	0.23235	0.80487		4	0.5129
						0.01

## 5.4 COMPARACIÓN DE RESULTADOS

Valor de la Función Objetivo con el Método del Gradiente.

Tabla 11. Comparación de Resultados con el Método de Discretización.

Haren Retagne	Hettich y Reemtsen	Carrero y Quijano
1.08351	1.08	0.74244
2.55783	---	0.49033
1.52472	1.52	0.55047
2.48325	2.50	
2.39669	2.40	0.72957
4.52595	4.49	0.5082
2.91143	2.91	0.28161

## 5.5 ANÁLISIS DE RESULTADOS

Con el objeto de analizar los resultados del prototipo se plantearon las pruebas del numeral anterior. Los resultados que arrojaron permitieron establecer un punto de comparación entre el algoritmo propuesto en este trabajo y otros resultados encontrados en el estado del arte. Estas pruebas se clasificaron en varias etapas:

Las primeras pruebas (preliminares) se realizaron al principio del desarrollo del proyecto para analizar el funcionamiento de la herramienta a utilizar. Las

siguientes permitieron comprobar el desarrollo por partes del prototipo, evaluar cada procedimiento que se maneja en el algoritmo (proyección de gradiente, método sproc.active, el algoritmo en general) y las finales se enfocaron en modelo planteado como caso de estudio, haciendo pruebas y experimentos con diferentes datos:

Diferente punto inicial (dentro y fuera del intervalo valido)

Diferente criterio de optimización (epsilon, 0.01 o 0.005)

Manejo de diferentes modelos de restricciones propuestos (A,B,C)

Los datos de las pruebas realizadas con los modelos A y C son aproximadamente similares ya que las restricciones propuestas tienen los datos equivalentes, con esto comparamos que el procedimiento si esta trabajando acorde con los datos que se le dan de los ejemplos que plantean los otros investigadores

Los puntos óptimos que dan como resultado, todos pertenecen al intervalo  $[0,1]$  por lo tanto se demuestra que la restricción se esta cumpliendo con el método de proyección de gradiente utilizado

Según los resultados obtenidos en las pruebas realizadas en el transcurso del proceso de construcción de la herramienta y las pruebas realizadas con el prototipo final, se deduce que el método de Algoritmos Estocásticos junto con el manejo de proyección de gradiente arroja resultados apropiados para el objetivo principal del proyecto, optimización de la trayectoria de un robot definida como la reducción del tiempo en que esta se desarrolla.



## CONCLUSIONES

El diseño de un algoritmo de optimización para dar solución a un determinado problema, implica además del logro de la eficiencia en términos de tiempo de ejecución computacional, el análisis y selección de las técnicas más apropiadas para lograr un nivel de generalización que permita aplicarlo y extenderlo para solución de otros problemas estrechamente relacionados. En el caso de este proyecto, se comprobó la efectividad del algoritmo estocástico frente a otra técnica de optimización encontrada en el estado del arte del problema de la optimización de trayectorias de un brazo robot (algoritmo de discretización).

Uno de los principales logros ha sido la utilización de conceptos de programación semi-infinita y en especial el algoritmo estocástico para la solución de problemas relacionados con la robótica, temas que por su nivel de especialización no son profundizados a lo largo de la carrera, a pesar de las oportunidades de estudio e investigación que estos conceptos ofrecen. No obstante, se puede decir que el trabajo de grado es un aporte al estado del arte en el tema de la optimización de la trayectoria de un brazo robot, ya que es la primera vez que este algoritmo se emplea en la solución de este tipo de problemas.

El uso de MATLAB como herramienta de programación, optimización, cálculo y visualización gráfica e interfaz, facilita notablemente el desarrollo y solución de problemas matemáticos de alta complejidad como el abordado en este proyecto, ya que los esquemas de programación habituales tales como C++, Visual Basic o Java, entre otros, requieren de la programación de las herramientas matemáticas que MATLAB ya tiene implementadas en sus toolboxes, lo cual permite ahorrar gran cantidad de esfuerzo de programación y líneas de código.

Los resultados obtenidos con el método estocástico de aproximaciones externas mediante la herramienta computacional prototipo basada en MATLAB, presentada en este trabajo, son apropiados para el objetivo principal del proyecto y mejoran notablemente los resultados obtenidos en los trabajos presentados por Haaren-Retagne y Vaz. No obstante se recomienda para estudios posteriores, emplear exactamente los mismos puntos de inicio y fin de la trayectoria a fin de estimar más acertadamente, el nivel de mejoramiento obtenido, ya que estas comparaciones se realizaron partiendo del mismo modelo de trayectoria y teniendo como objetivo principal el minimizar el tiempo de recorrido del manipulador, pero no consideraron iguales puntos de entrada al modelo, por falta de información de los investigadores anteriores.

Dentro de las pruebas realizadas se tomaron tres variantes que modifican las restricciones de la función objetivo (denominados dentro del documento como: a, b y c), estas variantes modifican parámetros como velocidad, aceleración y tirón (jerk); las variantes a y c tienen restricciones similares y los resultados obtenidos reflejan esta similitud verificando que el algoritmo estocástico de aproximaciones externas se aplica de manera correcta.

Se puede considerar este proyecto como un aporte al fortalecimiento de una nueva línea de investigación en la UNAB en el campo de la optimización aplicada a la robótica, la cual promete un desarrollo interdisciplinario a partir de proyectos conjuntos entre las Facultades de Ingeniería de Sistemas e Ingeniería Mecatrónica, fundamentado en el planteamiento de nuevos algoritmos numéricos y su implementación en la solución de problemas relacionados con el modelamiento matemático del movimiento robótico y la optimización de sus trayectorias.

El trabajo presentado abre las puertas ante un amplio desarrollo de esta temática en el futuro, fundamentado principalmente en la generalización de la solución al problema aquí planteado, en el posicionamiento para diferentes tipos de robots, además de su implementación física en un robot real, donde es necesario tener en cuenta un redimensionamiento de todos los parámetros mecánicos y de control, aplicados en una solución específica dentro de una función objetivo.

## BIBLIOGRAFÍA

BARRIENTOS, Antoni; PEÑIN, Luis Felipe; BALAGUER, Carlos y ARACIL, Rafael. Fundamentos de Robótica. Madrid España: McGraw Hill, 1997. cap.2,3,4,5,6.

CARRILLO Esneider y MORA Gleimer. El método Estocástico de Aproximaciones Externas para la Minimización de los Costos de Control de Contaminación del Aire. Bucaramanga, 2003, p.138-139. Trabajo de Grado (Ingeniero de Sistemas). Universidad Autónoma de Bucaramanga. Facultad de Ingeniería de Sistemas. Área de Ciencias Naturales.

FU. K.S; GONZALEZ, R.C; y LEE, C.S.G. Robótica: Control, Detección, Visión e Inteligencia. Editorial: McGraw Hill, Julio de 1988. Cap. 2,3,4,5.

HEIM, Alexander and Von Stryk,Oskar. Germany Trajectory Optimization of Industrial Robots whit Aplication to Computer-Aided Robotics and Robot Controllers. 2000.

OLLERO, Aníbal. Robótica, Manipuladores y robots móviles. Editorial Planeta. 2001. p. 5

R. HETTICH and K.O. Kortanek, Semi-infinite Programing: Theory, Methods and Aplications. 1993.

RÜCKMANN, Jan-J., Rembert Remtsen, Semi-Infinite Programming, Dordrech, The Netherlands: Kluwer Academia Publishers. 1998. Cap. 4,5,7,12.

## BIBLIOGRAFÍA COMPLEMENTARIA

AMERICARUM UNIVERSITARS, Trabajo de investigación 2003 [citado en 2004-02-25] Disponible en Internet <URL:[http://www.pue.udlap.mx/~tesis/lis/contreras\\_o\\_l/capitulo1.pdf](http://www.pue.udlap.mx/~tesis/lis/contreras_o_l/capitulo1.pdf)>

BAEZA, Juan J., Optimización de Modelos, septiembre de 1996 [citado en 2004-02-23]. Disponible en Internet <URL:<http://www.uv.es/~baeza/optimi.html>>

CASILLAS DE LA TORRE, Sergio, Características Morfológicas , Junio del 2000. [citado en 2004-02-21] Disponible en Internet <URL:<http://proton.ucting.udg.mx/materias/robotica/r166/r72/r72.htm>>

FEDOSSOVA Alina. El método estocástico de aproximaciones externas para minimización de los costos de control de contaminación del aire.2001.

FEDOSSOVA Alina.V., 1999. Los algoritmos estocásticos de las aproximaciones externas para la solución de los problemas convexos de programación semi-infinita. Resumen ejecutivo de la tesis doctoral. DIALOG – MGU, Rusia.

FEDOSSOVA, Alina; KAFAROV, V; MAHECHA, D.P. Solución Numérica del Problema de Control de Contaminación del Aire. [citado en 2004-02-15] Disponible en Internet <URL:[http://www.unab.edu.co/editorialunab/revistas/rcc/pdfs/r42\\_art2\\_c.pdf](http://www.unab.edu.co/editorialunab/revistas/rcc/pdfs/r42_art2_c.pdf)>

GÓMEZ, Omar Iván, Glosario sobre robótica, 2000 [citado en 2004-02-25] Disponible en Internet <URL:<http://www.robotics.freesevers.com/glosario.htm>>

HAAREN-RETAGNE, Elke. A Semi-Infinite Programming Algorithm for Robot Trajectory Planning. Tesis Doctoral, Universität Trier, Alemania, Junio 1992.

INFANTE, Juan-Antonio, REY José María, Introducción a Matlab, 1998 [citado en 2004-02-17] Disponible en Internet <URL:<http://www.mat.ucm.es/~infante/matlab/notas.htm>>

LÓPEZ, Marco, GOBERNA, Miguel Ángel, Programación Semi Infinita, 1998 [citado en 2004-02-13] Disponible en Internet <URL:[http://www.ua.es/otri/servicios/oferta\\_cientifica/ot\\_ual/prog\\_inf.DOC](http://www.ua.es/otri/servicios/oferta_cientifica/ot_ual/prog_inf.DOC)>

MENDOZA, Carlos, Robótica en Mendoza, [en línea]. 2003 [citado en 2004-02-18] Disponible en Internet <URL:[http://www.roboticajoven.mendoza.edu.ar/rob\\_car3.htm](http://www.roboticajoven.mendoza.edu.ar/rob_car3.htm)>

MUÑOZ, Antonio J, Introducción a la Robótica. [en línea]. 2003 [citado en 2004-02-20]. Disponible en internet: <<http://www.isa.uma.es/personal/antonio/robotica/tema1%20-20introduccion.pdf>>

MOORE, Ross, Cinemática, [en línea]. 1999 [citado en 2004-02-11]. Disponible en Internet <URL:<http://bellota.ele.uva.es/~imartin/libro/node9.html>>

TALLER EN TECNOLOGÍAS DE OPTIMIZACIÓN (4º: 2004: Bogotá). Actividad complementaria al III Congreso Colombiano- I Conferencia Internacional Andina de Investigación de Operaciones (CCIO). Bogotá: Optimización, 2004.

UNIVERSIDAD DE ANTIOQUIA, Restricciones Activas y Pasivas, 2003.  
[citado en 2004-02-08] Disponible en Internet  
<URL:<http://docencia.udea.edu.co/ingenieria/plineal/solucionGrafica04.html>>

UNIVERSIDAD DE VIGO, España, Automatización y Robótica. Cinemática,  
2002 [citado en 2004-03-01] Disponible en Internet  
<URL:[www.aisa.uvigo.es/DOCENCIA/AyRobotica/cinematica.pdf](http://www.aisa.uvigo.es/DOCENCIA/AyRobotica/cinematica.pdf)>

UPM-DISAM, Robótica Industrial, Norma ISO 8373:1998, [citado en 2004-03-02] Disponible en Internet  
<URL:<http://www.disam.upm.es/~barrientos/Doctorado/Trasparencias/ISO%209283.pdf>>

VAZ A, Ismael F., Aplicación, Métodos y Herramientas para Programación Semi-infinita, Tesis Doctoral, Univeridade do Minho, Portugal, 2003. [citado en 2004-03-03] Disponible en Internet <URL:<http://www.norg.uminho.pt/aivaz/>>

VAZ A, Ismael F. Robot trajectory Planning with Semi-infinite Programming [online].Confenecia, Paris, Sep. 2001 [Citado en 2004-09-15] Disponible en Internet <URL:<http://www.norg.uminho.pt/aivaz/publications.html>.>

VOLKOV Y.V., Zavriev S.K., 1997. "A general Stochastic Outer Approximations Method". SIAM J. Control Optim., V. 35 (No. 4): p. 1387-1421. [citado en 2004-03-20] Disponible en Internet <URL:<http://epubs.siam.org/sam-bin/dbq/article/26320>>