

# PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

## FACULTAD DE CIENCIAS E INGENIERÍA



PONTIFICIA  
**UNIVERSIDAD  
CATÓLICA**  
DEL PERÚ

### DESARROLLO DE UN ALGORITMO QUE PERMITA LA IMPLEMENTACIÓN FUTURA DE UN SOFTWARE PARA EL ANÁLISIS CINEMÁTICO INVERSO DE MECANISMOS EN 3D

Tesis para optar el Título de **Ingeniero Mecánico** que presenta el bachiller:

**ÁNGEL AGUSTÍN CÉSAR PEINADO BRAVO**

**ASESOR: Dr. Rosendo Franco Rodríguez**

**Lima, Febrero de 2016**

## RESUMEN

La presente tesis tiene por objetivo la elaboración de un algoritmo para el análisis cinemático inverso de mecanismos en el espacio, el cual abarca mecanismos clásicos y mecanismos empleados en la actualidad, tales como brazos robóticos. Con el fin de realizar el análisis cinemático de diversos mecanismos usando el mismo algoritmo, se plantea el uso de un método iterativo para la evaluación de las ecuaciones de movimiento. En este proceso se usan los parámetros de Euler como sistema de coordenadas generalizadas, así como la pseudo-inversión para la resolución de la inversión del jacobiano y el método de Newton-Raphson como método de minimización. Además, se presenta una librería de juntas para el modelamiento de diferentes tipos de juntas entre eslabones, permitiendo el estudio de diversos mecanismos. El algoritmo se implementa en el programa de Matlab, emplea archivos tipo texto para el ingreso de información y ofrece una interfaz tipo GUI para la obtención de diversas gráficas requeridas por el usuario. Durante la elaboración del algoritmo se presentaron dificultades en la eliminación de restricciones redundantes y evasión de singularidades del mecanismo, en específico en mecanismos contenidos en un plano. Esta dificultad fue superada empleando modelos depurados por parte del usuario. Para la validación del algoritmo se desarrollaron dos ejemplos de aplicación, un mecanismo clásico Biela-Manivela-Corredera y un brazo robótico tipo esférico. Los resultados obtenidos en estos ejemplos usando el algoritmo implementado y los obtenidos por otros autores son similares, apreciándose una adecuada correspondencia en los valores de posición, velocidad y aceleración. El algoritmo elaborado e implementado presenta subrutinas específicas y una librería de juntas que pueden ser empleados en un programa para el análisis cinemático y dinámico de mecanismos espaciales a ser desarrollado en un futuro.



PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ  
FACULTAD DE CIENCIAS E INGENIERÍA

TEMA DE TESIS PARA OPTAR EL TÍTULO DE INGENIERO MECÁNICO

TÍTULO : **DESARROLLO DE UN ALGORITMO QUE PERMITA LA IMPLEMENTACIÓN FUTURA DE UN SOFTWARE PARA EL ANÁLISIS CINEMÁTICO INVERSO DE MECANISMOS EN 3D**

ÁREA : Diseño **H 263**

PROPUESTO POR : Dr. Rosendo Franco Rodríguez

ASESOR : Dr. Rosendo Franco Rodríguez

TESISTA : Ángel Agustín César Peinado Bravo

CÓDIGO : 20064760

FECHA : 22 de septiembre de 2014



**DESCRIPCIÓN Y OBJETIVOS:**

El análisis cinemático inverso es un paso previo al análisis dinámica inverso, que tiene por finalidad obtener el conjunto de fuerzas necesarias a aplicar sobre un mecanismo para obtener un movimiento deseado. Estos análisis son de gran interés en la ingeniería debido a su aplicación en temas de investigación y en la solución de problemas reales de la industria, como es el control y optimización de mecanismos y brazos robóticos. Una vez resuelto el problema cinemático inverso resulta relativamente sencillo resolver el problema dinámico inverso, por lo que el presente trabajo se enfocará en el primero de ellos. Existen diversos enfoques para resolver el problema cinemático inverso, entre los cuales se destacan los métodos de solución de forma cerrada o específica y los métodos iterativos. En el presente trabajo se pretende realizar una búsqueda y revisión exhaustiva de los métodos existentes y seleccionar los más apropiados para realizar la modelación cinemática inversa de diversos mecanismos. El objetivo de la tesis es desarrollar un algoritmo coherente y programable que permita la implementación futura de un software para realizar el análisis cinemático inverso de mecanismos en tres dimensiones. Con la finalidad de validar el algoritmo propuesto se realizará una implementación básica que será aplicada en el análisis cinemático inverso de mecanismos típicos.

↑

Q



PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ  
FACULTAD DE CIENCIAS E INGENIERÍA

TEMA DE TESIS PARA OPTAR EL TÍTULO DE INGENIERO MECÁNICO

DESARROLLO DE UN ALGORITMO QUE PERMITA LA IMPLEMENTACIÓN  
FUTURA DE UN SOFTWARE PARA EL ANÁLISIS CINEMÁTICO INVERSO DE  
MECANISMOS EN 3D

Introducción

Capítulo 1. Estado del arte

Capítulo 2. Desarrollo de algoritmo para el análisis cinemático inverso de mecanismos en  
3D

Capítulo 3. Implementación básica y ejemplos de aplicación

Conclusiones y Recomendaciones

Bibliografía

*Máximo: 100 páginas*



Dr. Rosendo Franco Rodríguez  
Asesor





*A mi madre y padre,  
de cuyos auténticos cabellos grises  
yo soy una de las principales causas*

## INDICE DE CONTENIDO

	Pág.
<b>RESUMEN</b>	i
<b>APROBACION DE TEMARIO DE TESIS</b>	ii
<b>INTRODUCCIÓN</b>	1
<b>CAPÍTULO 1: ESTADO DEL ARTE</b>	5
1.1 Reseña Histórica	5
1.2 Mecanismos y Máquinas	8
1.2.1 Pares Cinemáticos	9
1.2.2 Cadena Cinemática	9
1.2.3 Clasificación de mecanismos	10
1.2.4 Coordenadas Generalizadas	10
1.2.5 Restricciones	14
1.2.5.1 Caracterización de Restricciones	15
1.2.6 Grados de Libertad (DoF)	17
1.3 Análisis Cinemático	17
1.3.1 Método de Análisis cinemático en sistemas Multicuerpos	18
1.3.1.1 Método Solución de Forma Cerrada	19
1.3.1.2 Método Iterativo	20
1.4 Jacobiano	21
1.4.1 Inversión del Jacobiano	22
1.4.1.1 Método de la Transpuesta del Jacobiano	23
1.4.1.2 Pseudo Inversión	24

1.4.1.3	Método de los mínimos cuadrados amortiguados	25
1.5	Método de minimización	26
1.5.1	Método Newton-Raphson	26
1.5.2	Método Quasi-Newton	28
1.6	Modelación del mecanismo	29
1.6.1	Cadena Cinemática Abierta	32
1.6.2	Cadena Cinemática Cerrada	33
<b>CAPÍTULO 2: DESARROLLO DE ALGORITMO PARA EL ANÁLISIS CINEMÁTICO INVERSO DE MECANISMOS EN 3D</b>		<b>35</b>
2.1	Selección de los métodos	35
2.1.1	Inversión del Jacobiano	36
2.1.2	Método de Minimización	36
2.1.3	Coordenadas Generalizadas	37
2.2	Descripción y desarrollo del algoritmo	37
2.2.1	Entradas – Modelo cinemático	38
2.2.2	Diagrama de Flujo	39
2.2.3	Pseudocódigo	41
2.2.3.1	Modelo Cinemático	42
2.2.3.2	Parámetros de Euler	44
2.2.3.3	Matriz de Rotación	47
2.2.3.4	Propiedades de los Parámetros de Euler y relación con velocidad angular	51
2.2.3.5	Librería de Juntas	54
2.2.3.5.1	Producto punto tipo 1 (DP1)	55

2.2.3.5.2	Producto punto tipo 2 (DP2)	57
2.2.3.5.3	Distancia (D)	59
2.2.3.5.4	Diferencia de coordenadas (CD)	60
2.2.3.5.5	Relación de Perpendicularidad a un plano ( $\perp 1$ )	62
2.2.3.5.6	Relación de Perpendicularidad a un plano ( $\perp 2$ )	64
2.2.3.5.7	Junta Esférica (SJ)	65
2.2.3.5.8	Junta Cilíndrica (CJ)	67
2.2.3.5.9	Junta de Revolución (RJ)	68
2.2.3.5.10	Junta de Traslación (TJ)	70
2.2.3.6	Juntas Motrices	72
2.2.4	Salidas	72
2.3	Eficiencia y Precisión del Algoritmo	75
2.3.1	Eficiencia del Algoritmo	75
2.3.2	Precisión del Algoritmo	76
<b>CAPÍTULO 3: IMPLEMENTACIÓN BÁSICA Y EJEMPLOS DE APLICACIÓN</b>		77
3.1	Implementación Básica del Algoritmo	77
3.1.1	Datos de Entrada	78
3.1.2	Desarrollo del Algoritmo	80
3.2	Ejemplos Desarrollados	83
3.2.1	Mecanismo Plano (Biela-Manivela-Corredera)	84
3.2.2	Mecanismo Tridimensional (Brazo Robótico)	90
<b>CONCLUSIONES Y RECOMENDACIONES</b>		96
<b>BIBLIOGRAFÍA</b>		98



## INTRODUCCIÓN

Los mecanismos son la base de la industria y se han ido desarrollando a lo largo de la historia. Su desarrollo puede datar desde antes de la época del renacimiento con Leonardo Da Vinci, el cual diseñó un robot con forma humana usando principalmente poleas. Por lo que su estudio se ha desarrollado por bastante tiempo siendo uno de sus principales investigadores el Ingeniero mecánico Franz Reuleaux, que define un mecanismo como una:

“combinación de cuerpos resistentes conectados por medio de articulaciones móviles para formar una cadena cinemática cerrada con un eslabón fijo, y cuyo propósito es *transformar* el movimiento” [1]

Según la definición de mecanismos del ingeniero Reuleaux no se incluye los robots dentro de estos, ya que son cadenas cinemáticas abiertas. En la presente investigación se va a modificar la definición de mecanismos para poder incluir a los robots. Por ello, los mecanismos son una combinación de cuerpos resistentes conectados por medio de articulaciones móviles para formar una cadena cinemática con un eslabón fijo, y cuyo propósito es transformar el movimiento y fuerzas.

Según la definición anterior se puede concluir que el principal objetivo de los mecanismos es transformar el movimiento, por lo cual el primer paso en el estudio de ellos es el análisis del comportamiento cinemático, posteriormente se procede a analizar el comportamiento dinámico de los eslabones y juntas.

Dentro del análisis del comportamiento cinemático existen dos estudios principales en el área de los mecanismos:

- Cinemática Directa, cuyo fin es obtener la cinemática del actuador final teniendo el comportamiento de todos los eslabones motrices.

- Cinemática Inversa, cuyo fin es obtener la cinemática de todos los eslabones motrices para obtener la cinemática del actuador final.

La aplicación principal de la Cinemática Inversa es desarrollar el comportamiento cinemático de todos los eslabones motrices. Otra aplicación es el estudio de todos los eslabones que pueden componer el mecanismo, facilitando el estudio de dinámica en el mecanismo. Una aplicación muy interesante en el estudio de la cinemática inversa es el comportamiento del cuerpo humano al realizar ciertas actividades, como en el deporte de tenis [2].

Existen diversos programas que realizan el análisis cinemático de mecanismos y sistemas multicuerpo, entre los cuales destaca el programa ADAMS. En la actualidad se desarrollan los modelos cinemáticos inversos de sistemas multicuerpo en tesis o investigaciones, usando diversos métodos para lograr una mayor eficiencia en el algoritmo.

Existen diversas formas de realizar el análisis cinemático de mecanismos entre los cuales destaca “Soluciones de Forma-Específica” (*Closed-Form Solutions*). Siendo este método el más usado en la actualidad debido a la velocidad de respuesta y la identificación de todas las soluciones posibles [23]. Debido a ser un análisis que depende de la topología y las características del mecanismo, son únicos o específicos para cada uno de ellos. Por lo que no son versátiles para el análisis de diferentes mecanismos en un mismo algoritmo [23]. En la actualidad, se han desarrollado diversos métodos para agilizar la obtención de las características cinemáticas necesarias para obtener el movimiento deseado, desarrollando diversos algoritmos usando métodos numéricos; basándose en generalizar ciertos tipos de juntas y teniendo una biblioteca general de juntas con facilidad de expansión. Siendo su principal desventaja la complejidad matemática que pueden llegar a tener y la sensibilidad del jacobiano o método de resolución. Sin embargo, con el avance de la tecnología se ha popularizado este método en programas versátiles, permitiendo el estudio de diversos mecanismos.

En el presente trabajo se desarrolla un algoritmo eficiente de fácil implementación y uso. Siendo su principal objetivo facilitar el cálculo en mecanismos en diversas aplicaciones de automatización y control de equipos. Además, como objetivo secundario se buscaría la motivación de desarrollo de tesis en este ámbito y en las aplicaciones que se le pueden dar en el campo de la biomecánica o animación por computadora.

La tesis se estructura en tres capítulos. El Capítulo 1 se hace una revisión de las definiciones y estado actual tanto de los mecanismos, formas de programación y del estudio de la cinemática inversa. De esta manera, se podrá definir, seleccionar y desarrollar el método de análisis y los tipos de entradas que se deben de considerar. En el Capítulo 2, se realiza un análisis detallado sobre el método seleccionado para realizar el modelo cinemático inverso, el desarrollo del algoritmo para su implementación. En el Capítulo 3 se desarrollara una implementación básica del algoritmo obtenido, usando el programa Matlab como base, además se desarrollan ejemplos de aplicación y se comparan los resultados con otros métodos ya existentes.

De acuerdo a lo planteado anteriormente, en la presente tesis se proponen los siguientes objetivos.

El objetivo general de la tesis es:

- Establecer un algoritmo coherente y programable que permita desarrollar un software para realizar el análisis cinemático inverso de mecanismos tridimensionales.

Como objetivos específicos se tiene:

- Seleccionar el método más eficiente y sencillo para la obtención del modelo cinemático inverso.

- Establecer las entradas que debe tener el algoritmo para realizar el análisis cinemático inverso de formar correcta.
- Desarrollar una librería de juntas.
- Realizar una implementación básica del algoritmo usando Matlab.
- Desarrollar un ejemplo de un mecanismo clásico y de una aplicación en la actualidad de un brazo robótico.



## CAPÍTULO 1

### ESTADO DEL ARTE

En el presente capítulo, en primera instancia se dará una reseña historia de la evolución de los mecanismos y sus análisis hasta la actualidad. Posteriormente se procederá a definir a que nos referimos con mecanismos y máquinas; la definición de pares cinemáticos, restricciones, cadenas cinemáticas y como se clasifica cada uno de ellos. Se dará un énfasis en la definición de coordenadas generalizadas y se analizará las desventajas y ventajas entre las coordenadas según los ángulos de Euler y los parámetros de Euler.

Teniendo definido lo básico, se detallara el análisis cinemático de mecanismos y sus métodos de análisis. Dando un énfasis en el análisis cinemático inverso, el desarrollo del jacobiano y su inversión; y los métodos de minimización usados en la actualidad para obtener el modelo cinemático inverso de mecanismos.

#### 1.1 Reseña Histórica

A lo largo de la historia se ha desarrollado diversos mecanismos y maquinas desde la época de Aristóteles hasta la era actual. Muchas de las máquinas y mecanismos desarrollados en la antigüedad fueron máquinas de guerra, tales como catapultas. Cabe destacar que además de máquinas de guerra, se desarrollaron mecanismos

ingeniosos e interesantes tales como los relojes de agua de Ctesibius, desarrollados en la época del apogeo de la ciudad de Alejandría.

Desde la época del renacimiento se vio un incremento en el desarrollo y estudio de las máquinas y mecanismos, con las famosas escuelas de las ciudades de Siena y Florencia, Italia. En esta época vivió Leonardo da Vinci, Taccola y Francesco di Giorgio Martini quienes fueron grandes ingenieros, en el campo de desarrollo de máquinas de guerra en específico.

En los posteriores siglos al renacimiento se incrementó el interés y el desarrollo de máquinas no tan aplicadas a la guerra, entre los cuales puede destacarse mecanismos aplicado a bombas desarrollados por Besson, en su libro *Theater of Machines*. Posteriormente llega la época de la revolución industrial en la cual tanto las maquinas como los mecanismos llegan a su apogeo.

Con respecto al análisis de los mecanismos, en la antigüedad no se buscaba entender su movimiento más bien se buscaba entender la relación de fuerzas que generaban, por lo que no se entendía por completo los mecanismos.

Durante el periodo (1750-1900) se desarrollaron el análisis de máquinas y mecanismos sin el uso de matemática compleja y potente por diversos ingenieros tales como James Watt, J.A. Bognis, Haton de la Goupillie're, Julius Weisbach, Franz Reuleaux, M.J. Calion, entre otros. Siendo el más destacado Franz Reuleaux a quien se le considera como el padre de la cinemática de mecanismos. Sus publicaciones mostraban un análisis de los mecanismos y maquinas desde un punto de vista geométrico más que de un análisis dinámico [6].

En el siglo XX se desarrolló más a fondo el análisis de la vibración y balanceo en máquinas y mecanismos debido a la llegada del automóvil, los generadores eléctricos y otros equipos que requerían generar grandes velocidades. Entre sus principales exponentes se encuentran Steven Timoshenko con su libro *Vibration Problems in Engineering* y James B. Hartman quien publico *Dynamic of Machinery* [6].

Con la llegada de la computadora se ha incrementado el interés en la generación y resolución de las ecuaciones de movimiento de sistemas multicuerpos interconectados. Se ha desarrollado en diversos programas que lograr ese objetivo tales como DAP, DADS, ADAMS, COMPAMM y SIMPACK [7]. Cuya principal diferencia es el método de cálculo, el tipo de coordenadas y la forma de manejo de los tipos de junta.

El programa ADAMS es reconocido como uno de los programas más completos en el análisis de sistemas multicuerpos, siendo este el más usado para el cálculo y diseño de estos [15]. Fue desarrollada en la década de los 70' por Orlandea y Chace pero salió a la venta en 1980 en Estados Unidos. El programa tiene como entrada el número de cuerpos que compone el sistema mecánico modelado. Siendo estos cuerpos considerados como eslabone o cuerpos independientes a los cuales se le debe definir sus propiedades y si son rígidos o flexibles. En el cual se debe especificar el tipo de junta, ya que este dato define el grado de libertad de un cuerpo con respecto a otro. Además, se puede agregar suspensión, resortes entre otros elementos. El programa genera la ecuación matemática que define o gobierna el movimiento del mecanismo, con la cual calcula la simulación para cada paso establecido, genera una imagen del comportamiento del mecanismo en 3D. Cuando se concluye la simulación se puede obtener todas las fuerzas y momentos que actúan en cada elemento del sistema mecánico analizado [15].

Por lo que el programa ADAMS es una de las herramientas más importantes para el análisis cinemático y dinámico de cualquier sistema mecánico a analizar.

Otro programa importante es AUTOSIM que genera unos códigos computacionales para simular el comportamiento cinemático y dinámico del sistema multicuerpos. En sus inicios se desarrolló para el cálculo en específico de los componentes mecánicos de vehículos y su simulación, pero a lo largo de los años se ha ido generalizando para su uso en cualquier tipo de sistema multicuerpos.

## 1.2 Mecanismos y Máquinas

Debido a la larga historia de mecanismos y maquinas la definición ha variado a lo largo del tiempo. La primera definición formal de mecanismos fue formulada por el Ingeniero Franz Reuleaux [1], quien define:

Maquina: “Combinación de cuerpos resistentes de tal manera que, por medio de ellos, las fuerzas mecánicas de la naturaleza se pueden encausar un trabajo acompañado de movimientos determinados” [1]

Mecanismo: “Combinación de cuerpos resistentes conectados por medio de articulaciones móviles para formar una cadena cinemática cerrada con un eslabón fijo, y cuyo propósito es transformar el movimiento” [1]

Debemos considerar que Reuleaux propone 17 definiciones para el término máquinas. Por ello, no existe un concepto absoluto en la definición apropiada de máquina y mecanismos.

Otra definición importante es la propuesta por Roque Calero Pérez, que propone “un mecanismo es una cadena cinemática a la que se ha inmovilizado uno de sus miembros” [11].

Se debe tomar en cuenta que los mecanismos son una parte de la familia de los sistemas multicuerpos. En la cual consideran que un mecanismo es:

“Un sistema mecánico que consiste en cuerpos rígidos llamados eslabones que están conectados a través de juntas” [9]

Es necesario definir primero que es un mecanismo en la presente tesis:



Mecanismo es una combinación de cuerpos resistentes conectados por medio de articulaciones móviles para formar una cadena cinemática con un eslabón fijo, y cuyo propósito es transformar el movimiento y fuerzas.

### 1.2.1 Pares Cinemáticos

Los mecanismos están conformados por eslabones que están conectados entre sí mediante juntas o llamados también pares cinemáticos [9, 10]. Los pares cinemáticos permiten transmitir el movimiento entre los eslabones. Existen diversas clasificaciones según la superficie de contacto, el tipo de movimiento relativo, tipo de rozamiento, el número de grados de libertad que posee el movimiento relativo entre los eslabones y la cantidad de eslabones interconectados [11].

Por lo que se va a clasificar los pares cinemáticos según el número de grados de libertad que posee el movimiento relativo entre los eslabones que conforman la junta o par cinemático. Entonces se clasifica en par de I, II, III, IV, V grados de libertad.

### 1.2.2 Cadena Cinemática

El conjunto de eslabones conectados entre sí por pares cinemáticos o juntas es denominado cadena cinemática. Las cadenas cinemáticas se pueden clasificar en dos tipos:

**Cadena cinemática cerrada:** Cuando todos los eslabones están conectados por lo menos con otros dos. Por ejemplo, los mecanismos de 4 barras.

**Cadena cinemática abierta:** Cuando por lo menos uno de los eslabones solo está conectado con otro. Por ejemplo, los brazos robóticos.

### 1.2.3 Clasificación de mecanismos

La clasificación a seguir se basa en la propuesta de Roque Calero Pérez [11], quien propone dos posibles clasificaciones.

Atendiendo a la movilidad de sus miembros, se clasifica en desmodrómico y no desmodrómico. En el primer caso solo se necesita definir el movimiento de uno de sus eslabones y en el segundo caso se requiere definir el movimiento de dos o más eslabones del mecanismo.

Atendiendo al tipo de movimiento de sus miembros, se clasifican en mecanismos planos y espaciales. En el caso de mecanismos planos los eslabones se mueven siempre en un plano, no necesariamente los eslabones se mueven en el mismo plano. En los mecanismos espaciales algunos de los eslabones no se mueven en un plano.

### 1.2.4 Coordenadas Generalizadas

La configuración del mecanismo es representada por un grupo de variables conocidas como coordenadas generalizadas que poseen la información de posición y orientación de cada eslabón en el mecanismo.

Para la definición de la posición de una partícula solo se requiere de 3 variables que otorgan solamente la posición en el espacio de esta, no se le considera la rotación debido a la consideración de ser tan pequeña que solo se requiere de un punto para poder determinar su posición. Con respecto a cuerpos rígidos, en este caso los eslabones, se requiere tener también las variables de rotación en el espacio esto quiere decir que requiere seis variables, tres con respecto a su posición y tres para determinar su orientación, siendo los ángulos de Euler usados regularmente para este propósito. Otra forma de determinar la posición y rotación del cuerpo rígido es usar dos puntos cualesquiera del cuerpo rígido y definir según la posición de uno de ellos y la posición relativa del segundo punto con el primero [17, 19].

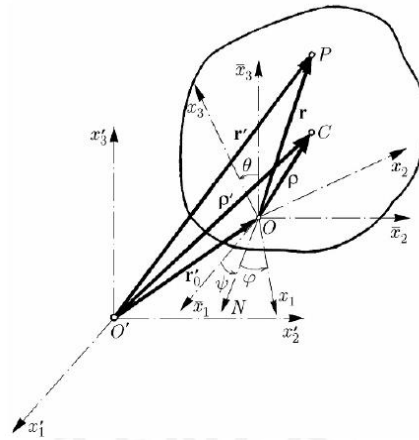


Figura 1.1: Cuerpo rígido representado en sus marcos de referencia inercial  $R'$  y sus dos marcos referenciales no inerciales  $R$  y  $\bar{R}$  [19]

En la Figura 1.1 se puede observar que se tiene un marco inercial  $R'$  donde se ubica el cuerpo rígido, por lo que se traslada el marco de referencia del punto  $O'$  al punto  $O$ , siendo este punto perteneciente al cuerpo rígido, usando el vector  $r'_{o'}$  y se genera un marco no inercial  $\bar{R}$ . Entonces para definir por completo la posición del cuerpo rígido se requiere de otro punto o usar los ángulos de Euler (el ángulo de precesión  $\psi$ ,  $0 \leq \psi \leq 2\pi$ , el ángulo de mutación  $\theta$ ,  $0 \leq \theta \leq \pi$  y el ángulo de rotación  $\varphi$ ,  $0 \leq \varphi \leq 2\pi$ ) que especifica la rotación del cuerpo rígido con respecto al punto  $O$ . Por lo tanto, se obtiene un nuevo marco de referencia no inercial  $\mathfrak{R}$  que está unido al cuerpo rígido en el punto  $O$ . Entonces se puede tener la siguiente notación [17]:

$$q'_r = [R^{i_1} \quad R^{i_2} \quad R^{i_3} \quad \theta^{iT}]^T \dots\dots\dots (1.1)$$

Si se requiere usar dos puntos para definir la posición del cuerpo rígido se tiene la siguiente notación [17]:

$$|\overline{P_P P_C}| = |r' - r'_{o'}| \dots\dots\dots (1.2)$$

Siendo la más usada la notación (1.1) debido a su fácil manipulación para la programación, donde  $R^{i_1}$ ,  $R^{i_2}$  y  $R^{i_3}$  definen la posición del punto  $O$  y  $\theta^{iT}$  representa los tres ángulos de Euler empleados. Entonces  $q'_r$  es un vector de seis dimensiones. Entonces la posición del punto  $P$  se puede expresar de la siguiente forma [17]:

$$r^i_p = R^i + A^i \bar{u}^i \quad \dots\dots\dots (1.3)$$

Donde  $R^i$  es la posición del punto O en el marco de referencia inercial  $R$  y el vector  $A^i$  es la matriz para la transformación del sistema de referencia no inercial  $R$  al sistema de referencia inercial  $R'$  y  $\bar{u}^i$  es el vector de posición del punto P desde el marco de referencia del cuerpo rígido  $R$  [17].

Entonces para un mecanismo compuesto por N eslabones se requiere 6N coordenadas para poder definir por completo la posición de cada eslabón del mecanismo. Por lo que se define el mecanismo con el siguiente vector, donde n es el número de coordenadas [17]:

$$q = [q_1 \quad q_2 \quad q_3 \quad \dots \quad q_n] \quad \dots\dots\dots (1.4)$$

Existen otros métodos de representar la orientación de un cuerpo rígido dentro del espacio basado en el Teorema de Euler:

*“Si el origen de dos marcos de referencia Cartesianos coinciden, entonces estos pueden ser llevados a la coincidencia mediante una rotación simple alrededor de un eje” [24]*

Este teorema explica que mediante un vector unitario y una rotación referente a un plano perpendicular a este se puede definir un marco de referencia. Según este teorema se desarrolló los parámetros de Euler o cuaternion unitario. En la Figura 1.2 se puede observar una representación espacial del vector unitario y el ángulo de giro requerido para alineación entre dos marcos de referencia cartesianos. [24]

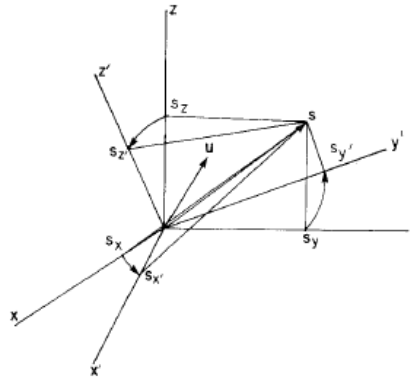


Figura 1.2: Vector unitario  $u$  y el ángulo de rotación  $x$ , requeridos para poder hacer coincidir dos marcos de referencia. [24]

Los parámetros de Euler o cuaterniones unitarios se definen con lo siguiente [24]:

$$\begin{aligned} \epsilon_1 &= u_x \operatorname{sen} \frac{x}{2} \\ \epsilon_2 &= u_y \operatorname{sen} \frac{x}{2} \\ \epsilon_3 &= u_z \operatorname{sen} \frac{x}{2} \\ \epsilon_0 &= \cos \frac{x}{2} \end{aligned}$$

Donde el vector  $u$  es llamado vector unitario de rotación y  $x$  es ángulo de rotación, se puede observar que los parámetros de Euler no son independientes entre sí. Por lo que se requiere una ecuación de normalización de los parámetros de Euler que siempre se debe cumplir [24].

$$P^T P = \epsilon_0^2 + \epsilon_1^2 + \epsilon_2^2 + \epsilon_3^2 = 1$$

Para poder definir la posición de un punto dentro del marco referencial no inicial se requiere de la misma fórmula (1.3), claro está que la matriz de rotación tendría como base los parámetros de Euler. Entonces para mecanismos por  $N$  eslabones se requiere de  $7N$  coordenadas para poder definir la posición de todos los eslabones que lo componen [24].

Ambas representaciones de la rotación tienen sus ventajas y desventajas dentro del desarrollo del algoritmo, las cuales se explicaran más a detalle en la selección del algoritmo a implementar.

### 1.2.5 Restricciones

Los sistemas multicuerpos o en este caso mecanismos están sujetos a diversos tipos de restricciones que pueden ser de origen geométrico o cinemática. Presentando una limitación con respecto al movimiento y desplazamiento de ciertos puntos característicos de los eslabones, mayormente las juntas. Estas relaciones se pueden expresar mediante ecuaciones en las cuales interaccionan vectores de posición y de velocidad.

Existen diversas clasificación de las restricciones, además de las presentadas en anteriormente, entre las cuales destacan las de tipo interna y externa, de contacto o de distancia y de tipo ideal o real. La ultima clasificación es la más importante ya que en las restricciones ideales no se considera la fricción y en la real si se toma en cuenta. Otra clasificación son las restricciones del tipo estacionario (no varían en el tiempo) o las que varían en el tiempo [18].

La clasificación más importante en los tipos de restricciones es si pertenecen al grupo de restricciones *Holonomic* o *Nonholonomic*, según el autor la definición de este tipo de restricciones puede variar ligeramente. La principal característica de las ecuaciones *Nonholonomic* no se pueden expresar o resolver usando las coordenadas generalizadas, sino deben ser adaptadas para poder solucionarlas [17,18].

Otro tipo de restricciones son dadas por igualdades o desigualdades esto quiere decir que cierto punto de un cuerpo rígido o un eslabón pertenece a una curva o no puede pertenecer a esta. Estos tipos de restricciones se le conocen como restrictivas y no restrictivas, no solo se puede aplicar a la ubicación sino también a la velocidad que puede alcanzar dichos puntos.

**1.2.5.1 Caracterización de Restricciones**

De acuerdo al número de restricciones,  $n_c$ , en el mecanismo se puede obtener el número de grados de libertad. Pero primero se debe poder caracterizar las restricciones del tipo geométricas y cinemáticas, para lo cual primero se debe diferenciar entre las restricciones del tipo *Holonomic* y *Nonholonomic* [17].

Considerando que todas las restricciones son del tipo *Holonomic* se puede escribir las ecuaciones de las restricciones de la siguiente forma:

$$C(q_1, q_2, q_3, \dots, q_{nc}; t) = C(1; t) = 0 \dots\dots\dots (1.5)$$

Donde  $C = [C_1(q; t) \ C_2(q; t) \ \dots \ C_{nc}(q; t)]^T$  es el grupo de las ecuaciones independientes de cada restricción.

Si no existe explícitamente la variable tiempo en la ecuación (1.5), se considera el sistema de tipo *scleronomic*, en caso contrario que se encuentra la variable tiempo se considera de tipo *rheonomic* [17].

Un ejemplo de una restricción *scleronomic* son los tipos de junta de tipo revolución. En la figura 1.3 se muestra dos eslabones unidos por este tipo de junta (de revolución) ubicada en el punto P, al ser dos cuerpos rígidos diferentes (i y j) se tiene dos marcos de referencia diferentes  $\mathfrak{R}_i$  y  $\mathfrak{R}_j$ . Debido a este tipo de junta la posición del punto P en el marco de referencia no inercial por lo tanto:

$$r_p^i = r_p^j \rightarrow R^i + A^i \bar{u}^i = R^j + A^j \bar{u}^j \dots\dots\dots (1.6)$$

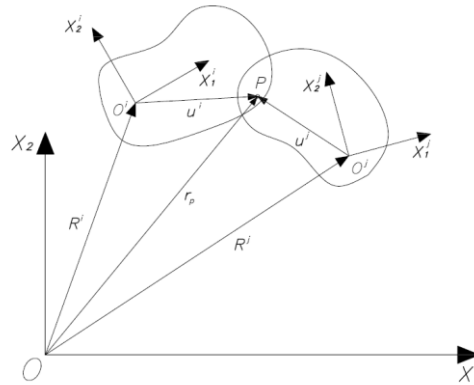


Figura 1.3: Junta tipo revolución, en un sistema de dos dimensiones

Una restricción del tipo *rheonomic* son los manipuladores o brazos robóticos que tiene como objetivo realizar una determinada curva en el tiempo. Tomando como ejemplo el manipulador mostrado en la figura 1.4, en donde si se analiza el último eslabón  $i$  se tiene la siguiente ecuación referente a la restricción, considerando la posición que debe realizar a lo largo del tiempo:

$$r_p^i = R^i + A^i \bar{u}^i = f(t) \dots\dots\dots (1.7)$$

Donde  $f(t) = [F_1(t) \ F_2(t) \ F_3(t)]^T$ , siendo la ubicación del eslabón analizado a lo largo del tiempo.

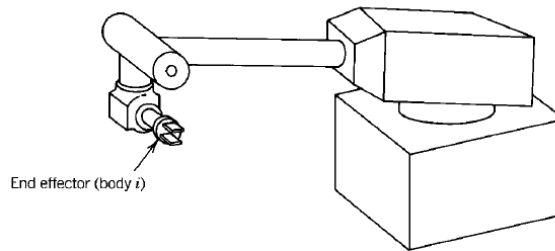


Figura 1.4: Manipulador [17]

Las restricciones del tipo Nonholonomic no pueden ser escritas de la forma de la ecuación (5) debido a su naturaleza, mencionada en la anterior parte, las restricciones de este tipo se puede escribir de la siguiente forma:



$$a_o + B\dot{q} = 0 \quad \dots\dots\dots (1.8)$$

Donde  $a_o = a_o(q, t) = [a_{o1} \ a_{o2} \ a_{o3} \ \dots \ a_{onc}]^T$  y  $\dot{q} = [\dot{q}_1 \ \dot{q}_2 \ \dots \ \dot{q}_{nc}]^T$  siendo el vector de las velocidades del sistema o mecanismo y B es un coeficiente con forma matricial  $n_c \times n$ . La ecuación (1.8) no puede ser integrada con respecto a las coordenadas generalizadas [17].

### 1.2.6 Grados de Libertad (DoF)

Debido a las restricciones originadas por las juntas entre eslabones y las originadas debido a la trayectoria deseada, considerando estas restricciones del tipo *Holonomic* se puede restar el número de restricciones del número de coordenadas independientes que definen la posición del mecanismo entonces para el caso general de mecanismos en tres dimensiones se define:

$$DOF = \#_R N - n_c \quad \dots\dots\dots (1.9)$$

Donde  $\#_R$  es el número de grados de libertad según las coordenadas generalizadas,  $N$  es el número de eslabones y  $n_c$  es el número de restricciones. Entonces se puede entender que el Grado de Libertad es la cantidad de parámetros que se debe definir para poder controlar por completo el mecanismo.

## 1.3 Análisis Cinemático

Dentro del estudio del comportamiento de los mecanismos y sistemas multicuerpos siempre se busca obtener las ecuaciones de movimiento y poder predecir el comportamiento cinemático del mecanismo analizado. Dentro del estudio de la cinemática existen dos problemas principales, en la figura 1.5 se puede apreciar un diagrama con ambas definiciones:

**Cinemática Directa:** Obtener el comportamiento del último eslabón o el llamado actuador final, teniendo como datos de entrada la posición, velocidad y aceleración de las juntas motrices.

**Cinemática Inversa:** Obtener las posiciones, velocidades y aceleración de los eslabones motrices, teniendo como datos de entrada el comportamiento del actuador final.

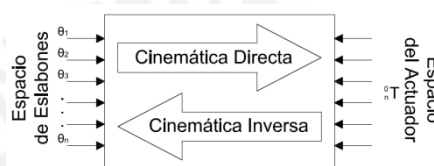


Figura 1.5: Definición de Cinemática Directa e Inversa

Existen diversas maneras de obtener la ecuación de movimiento según el tipo de análisis, el sistema de coordenadas a usar y como se maneje los tipos de junta. Según la combinación que se realiza se ha ido formando diversas formas de análisis, siendo las más populares el uso de soluciones cerradas según la topografía y características de cada mecanismo a analizar, se debe tomar en cuenta que la eficiencia de los algoritmos depende fundamentalmente de la forma en la que se manejan los datos y cálculos que en el principio en la que se basan [23].

### 1.3.1 Método de Análisis cinemático en sistemas Multicuerpos

Existen diversos métodos para la resolución de la cinemática inversa y directa de mecanismos, los cuales se clasifican en dos subgrupos principales, el método de solución de forma cerrada y el método iterativo. El método de solución de Forma Cerrada se basa en analizar las características típicas de cada mecanismo para poder dar una solución concreta, su principal ventaja es la respuesta rápida y la posibilidad de obtener todas las soluciones posibles pero este tipo de algoritmos son únicos para cada tipo de mecanismo. Por otro lado el método iterativo a través de métodos numéricos nos facilita obtener la solución más cercana en base a una primera aproximación, siendo su principal ventaja su aplicación a diversos mecanismos mediante un solo algoritmo.

### 1.3.1.1 Método Solución de Forma Cerrada

Su principio básico es generar ecuaciones según las características típicas del mecanismo, pueden ser tanto algebraicas como geométricas. Dando ecuaciones y parámetros según el mecanismo a analizar, siendo este método el más utilizado en la robótica. En la actualidad se ha generado diversas publicaciones para la solución de diversos brazos robóticos y mecanismos típicos producidos por empresas de automatización como los brazos robóticos tipo PUMA [23].

La principal ventaja de este método de solución es la obtención de todas las soluciones posibles del mecanismo y la respuesta rápida que se puede generar, siendo ideal para situaciones donde se requiere respuestas inmediatas o uso en tiempo real. La desventaja de estos algoritmos se basa en la dificultad de obtener las ecuaciones características del mecanismo en base a su topología y siendo complicado su uso en mecanismos complejos. Estos algoritmos son únicos para cada tipo de mecanismo no dando posibilidad en su uso en programas para el estudio de la cinemática de mecanismo en general. Existen dos métodos tipos para la obtención del algoritmo.

Método Algebraico.- busca relacionar todos los eslabones, en base a sus características cinemáticas, para obtener las diferentes soluciones para la posición del actuador o el eslabón deseado. Para el caso de brazos robóticos se tiene N ecuaciones igual a los grados de libertad del mecanismo, teniendo la gran dificultad de tener que resolver N ecuaciones con N variables [26].

Una de las estrategias más usadas es simplificar todas las ecuaciones a una simple ecuación con una sola incógnita característica del eslabón a estudiar, debido a ciertas particularidades de los mecanismos y la búsqueda de evitar singularidades se propones usar dos ecuaciones.

Su principal complejidad es poder relacionar todas las ecuaciones y poder reducirlas a una sola ecuación. Diferentes publicaciones acerca de este método se estudia casos simples de brazos articulados, siendo sus principales exponentes Craig [27], Manocha

[28] y Zhu [29] los cuales desarrollaron una propuesta generalizada de solución cerrada para mecanismos con 6 (o menos) GDL (Grados de Libertad) de una cadena cinemática cerrada.

Método Geométrico, busca resolver o sintetizar la solución en parámetros requeridos del actuador o eslabón de interés. Busca relacionar la geometría del mecanismo usando puntos de referencia en diversos planos. Reduce la complejidad del espacio cartesiano en sub-planos de solo dos dimensiones y pocas variables [23].

Entre sus exponentes esta Lee [30] que usa diversos teoremas en coordenadas geométricas espaciales para obtener la solución cerrada de mecanismos con seis grados de libertad. Tiene ciertas complicaciones y limitantes tal como el requerimiento que exista una referencia geométrica entre los tres primeros eslabones. La principal desventaja de este método es la limitación de la aplicación del algoritmo para un tipo de mecanismo o brazo robótico, por lo que no se puede aplicar a diversos mecanismo [23].

#### 1.3.1.2 Método Iterativo

Al contrario que los métodos de solución cerrada, el método iterativo se puede aplicar a la gran variedad de mecanismos existentes sin tener limitaciones en su aplicación. Siendo su principal desventaja la complejidad del algoritmo y por ende ser más lentos que los otros métodos en la obtención de la respuesta deseada. Este método solo converge a una solución dado por la aproximación inicial del mecanismo.

Existen tres componentes dentro del análisis a través del método iterativo el Jacobiano, la inversión del Jacobiano y el método de minimización [26]. Estos tres componentes del algoritmo son importantes, ya que definen las características del mismo (robustez y velocidad) y la evasión de singularidades.

Para un correcto desarrollo del algoritmo para el método iterativo se requiere seleccionar un correcto método de inversión del Jacobiano, el cual incide en la evasión

de singularidades, y método de minimización, el cual incide en la rapidez de convergencia del algoritmo a la solución deseada.

#### 1.4 Jacobiano

El jacobiano se puede definir como una matriz de todas las derivadas parciales de un vector función. El vector función en nuestro caso contiene las ecuaciones de movimiento de cada tipo de junta presente en el mecanismo. Esto quiero decir que el jacobiano en este caso describe cómo cambia cada eslabón con respecto a cada coordenada generalizada (traslación o rotación) [26]

Para mayor entendimiento del uso y del origen del jacobiano se debe entender el modelo cinemático o ecuación de movimiento de un mecanismo, el cual está en función de las ecuaciones de restricción de movimiento debido a las juntas y a las características de los eslabones o elementos que lo conforman. Para un mecanismo en un momento dado, teniendo la posición o valor de las coordenadas generalizadas se tiene la siguiente expresión [24]:

$$\Phi_{(q,t)} = \begin{bmatrix} \Phi_{(q,t)}^K \\ \Phi_{(q,t)}^D \end{bmatrix} = 0 \dots \dots \dots (1.10)$$

Dónde:

$\Phi_{(q,t)}^K$  : Ecuaciones debido a la geometría y pares cinemáticos, en funciones de las coordenadas generalizadas y del tiempo.

$\Phi_{(q,t)}^D$  : Ecuaciones con respecto a ciertos eslabones motrices o movimiento deseado de algún eslabón.

Entonces teniendo la ecuación de posición para todo instante dado del mecanismo se puede obtener la velocidad derivando la ecuación de posición (1.10). Debido a que las coordenadas generalizadas no son funciones explícitas del tiempo se procede a utilizar la regla de cadena de diferenciación, con lo cual se puede obtener:

$$\begin{aligned}\frac{d}{dt}\phi(q,t) &= 0 \\ \dot{\phi} &= \phi_q \dot{q} + \phi_t = 0 \\ \phi_q \dot{q} &= -\phi_t \equiv v \dots \dots \dots (1.11)\end{aligned}$$

Por lo que se debe poseer cierta información de las ecuaciones debido a las restricciones y ecuaciones motrices o características de movimiento, tal es el caso en los brazos robóticos donde  $v$  es la velocidad del último eslabón, ya que es el parámetro buscado o característico del movimiento, en el caso de mecanismo regularmente es la velocidad angular del motor con el cual se da movimiento.

El jacobiano es la matriz más importante dentro del desarrollo de un método iterativo tanto para cinemática como dinámica, ya que describe el comportamiento del mecanismo en primer orden.

#### 1.4.1 Inversión del Jacobiano

Debido a su aplicación en el análisis cinemático inverso se requiere poder aplicar la inversa del jacobiano presentándose diversas complejidades para poder lograrlo, basándose en primer lugar si es reversible (determinante diferente a cero) y que la matriz sea cuadrada, además de posibles degeneraciones en el jacobiano debido a la presencia de múltiples soluciones que no aportan información valiosa al modelo.

Debido a estas singularidades y la búsqueda de poder superar dichos problemas se han ido desarrollando múltiples algoritmo o artilugios para poder solucionarlo. Este hecho dio paso al origen de motivar a diferentes autores como Balestrino, de Maria y Sciavicco en desarrollar el método de la transpuesta del Jacobiano, el desarrollo del método de la pseudo inversión por Baillieul [33] y el método mínimos cuadrados amortiguados por Wampler [34] y Nakamura y Hanafusa [35] los cuales se usan en diversas aplicaciones para poder realizar la inversión del jacobiano y poder sobrellevar las diferentes singularidades expuestas.

### 1.4.1.1 Método de la Transpuesta del Jacobiano

El método de la transpuesta del Jacobiano fue usado por primera vez en el análisis cinemático de mecanismos por Balestrino, de Maria y Sciavicco en 1984 [31] y por Wolovich y Elliot [32] en el mismo año.

Se sabe que la transpuesta no es igual a transposición pero es similar y generar un pequeño error por lo que para mecanismo en los que no se requiere demasiada precisión en la respuesta se puede justificar su uso. Para el incremento de la precisión y poder considerar que la transpuesta es similar a la inversa los diferentes autores propusieron la siguiente fórmula:

$$\Delta\theta = \alpha J^T \vec{e} \dots\dots\dots (1.12)$$

Dónde:

- $\Delta\theta$  : Variación de las coordenadas generalizadas del sistema
- $\alpha$  : Escalar mayor a cero
- $J$  : Jacobiano del mecanismo
- $e$  : Vector con las ecuaciones características del mecanismo

Para lo cual se debe obtener un  $\alpha$  el cual debe minimizar el error en el sistema reduciendo el valor del vector  $e$  al mínimo posible. Esta conclusión requiere que  $\alpha$  se actualizado constantemente con cada nueva aproximación usando la siguiente fórmula [30]:

$$\alpha = \frac{[\vec{e}, J J^T \vec{e}]}{[J J^T \vec{e}, J J^T \vec{e}]} \dots\dots\dots (1.13)$$

Este método es usado principalmente para el cálculo de fuerzas virtuales y no mucho en el análisis cinemático de mecanismo o robots, debido a su precisión en la respuesta.

### 1.4.1.2 Método de la Pseudo Inversión

Este método desarrolla la siguiente fórmula:

$$\Delta\theta = J^\dagger \vec{e} \dots\dots\dots (1.14)$$

Donde  $J^\dagger$  es la pseudo inversión del jacobiano siendo una matriz de  $n \times m$ . Esta matriz brinda grandes ventajas como la inversión de matrices no cuadradas, también es conocido como la inversión del jacobiano de Moore-Penrose [30].

Este método tiene como principal característica la estabilidad frente a singularidades y la inversión de un jacobiano no cuadrado (robots redundantes). Una peculiaridad de este método es al encontrarse exactamente en una singularidad no tendera a moverse extremadamente en alguna dirección sino se mantendrá cerca de la solución pero si se acerca a la singularidad tiende a dar variaciones de ángulos bastante bruscos aun en pequeños desplazamientos del mecanismo. Por lo que se debe buscar las singularidades en base al determinante del jacobiano acercándose a cero.

Existen diversas publicaciones sobre el desarrollo de sub espacios para mejorar el comportamiento de la pseudo inversión del jacobiano los cuales facilitan el desarrollo de la cinemática inversa para la evasión de obstáculos dentro de la robótica [4]. Desarrollándose un método llamado extensión del jacobiano desarrollado por Baillieul [4] el cual busca el mínimo valor de una función es buscada alternativamente para evitar las singularidades.

Se desarrolla de la siguiente manera la inversión del jacobiano a través de la pseudo inversión del mismo:

$$J^T J \Delta\theta = J^T \vec{e} \dots\dots\dots (1.15)$$



Esta ecuación garantiza la existencia de una respuesta debido a que la parte derecha siempre va a estar en el rango  $J^T J$ . Este método permite solucionar jacobianos que no son cuadrados pero tiene aún la inestabilidad cerca de las singularidades. Para la inversión del jacobiano se requiere que el mismo posea un determinante diferente a cero y teniendo esta condición se tiene la siguiente fórmula:

$$\Delta\theta = J^T (J^T J)^{-1} \vec{e} \dots\dots\dots(1.16)$$

**1.4.1.3 Método de los mínimos cuadrados amortiguados**

El método de los mínimos cuadrados amortiguados busca eliminar las singularidades encontradas en el método de la pseudo inversión, este método fue propuesto en las diversas publicaciones de Nakamura y Hanafusa [35] y Wampler [34]. Se propone no buscar la respuesta más pequeña para satisfacer la ecuación característica del mecanismo ante pequeñas variaciones de las coordenadas generalizadas, sino buscar el valor mínimo de las variaciones de las coordenadas generalizadas en la siguiente ecuación:

$$\|J \Delta\theta - \vec{e}\|^2 + \lambda^2 \|\Delta\theta\|^2 \dots\dots\dots (1.17)$$

Donde  $\lambda \in \mathbb{R}$  y es una constante de amortiguación diferente a cero. Por lo tanto es igual a minimizar la siguiente ecuación:

$$\left\| \begin{pmatrix} J \\ \lambda I \end{pmatrix} \Delta\theta - \begin{pmatrix} \vec{e} \\ 0 \end{pmatrix} \right\| \dots\dots\dots (1.18)$$

Normalizando la ecuación y reduciéndola se puede representar de la siguiente forma:

$$(J^T J + \lambda^2 I) \Delta\theta = J^T \vec{e} \dots\dots\dots (1.19)$$

Por ende para la inversión del Jacobiano solo se requiere que  $(J^T J + \lambda^2 I)$  no sea singular y no tenga un determinante igual a cero. Por lo que se puede representar de la siguiente manera:

$$\Delta\theta = (J^T J + \lambda^2 I)^{-1} J^T \dot{e} \dots \dots \dots (1.20)$$

La constante de amortiguación debe tener ciertas características para poder evitar las singularidades y la divergencia de la solución cerca de una singularidad para lo cual la constante debe ser de grande pero no demasiado, ya que originaría que la convergencia de la solución sea muy lenta. Nakamura y Hanafusa realiza diversas propuestas para poder definir el valor de la constante de amortiguación [35]. Obteniéndose un valor correcto de la constante de amortiguación se puede generar un algoritmo robusto y sin singularidades.

## 1.5 Método de minimización

El método de minimización se refiere en específico al método de obtención de la respuesta o la posición de cada eslabón según las coordenadas generalizadas a emplear. Este método aplica tanto al Jacobiano como a su inversión para obtener la respuesta en cada instante analizado, este método caracteriza al algoritmo la velocidad de resolución y obtención de las características cinemáticas del mecanismo en un intervalo de tiempo analizado [23].

Existen diversas propuestas para obtener la solución deseada de las cuales destacan dos principalmente el método Newton-Raphson y el método Quasi-Newton. Ambos métodos presentan diversas ventajas en cuanto a la convergencia a la solución deseada [30].

### 1.5.1 Método Newton-Raphson

El método Newton-Raphson se ha empleado en diversos trabajos para el desarrollo óptimo de un algoritmo para obtener el comportamiento cinemático de mecanismo.

Fue expuesto inicialmente por Pieper [36]. Usando una aproximación lineal de una serie de ecuaciones no lineales (ecuaciones características del mecanismo), para lo cual utiliza la expansión de Taylor empleando la inversión del Jacobiano. Parte de una aproximación inicial y a través de varias iteraciones llega a la solución que cumple con un error establecido por el usuario.

$$\Phi(q) = 0 \dots\dots\dots (1.21)$$

Donde el vector  $\Phi$  contiene todas las ecuaciones características del mecanismo (restricciones entre eslabones y restricciones motrices o movimiento) el cual está en función de las coordenadas generalizadas. Para lo cual se debe cumplir la ecuación 1.21 en todo instante.

Debido a que las ecuaciones que caracterizan al mecanismo no son lineales no se puede resolver directamente la ecuación 11 para lo cual se utiliza la expansión de Taylor obteniendo la siguiente ecuación:

$$\Phi_q \times \Delta q = \Phi(q_0) \dots\dots\dots (1.22)$$

Donde se observa la primera derivada de las ecuaciones características del mecanismo con respecto a las coordenadas generalizadas dando origen al Jacobiano mencionado anteriormente. Además se puede observar  $\Phi(q_0)$  que es la ecuación característica del mecanismo para una primera aproximación inicial y  $\Delta q$  representa el error entre la primera aproximación lineal y el valor verdadero. Entonces se debe llegar a obtener un error permisible el cual es definido por el usuario, si no se cumple con dicho error se vuelve a iterar hasta llegar a la solución correcta.

$$\begin{aligned}
 q_1 &= q_0 - \left(\Phi_q(q_0)\right)^{-1} x \Phi(q_0) \\
 q_2 &= q_1 - \left(\Phi_q(q_1)\right)^{-1} x \Phi(q_1) \\
 q_3 &= q_2 - \left(\Phi_q(q_2)\right)^{-1} x \Phi(q_2) \\
 &\vdots \\
 &etc
 \end{aligned}$$

Se debe entender que la convergencia de este método depende directamente de una correcta aproximación inicial. Además, debido a que existen múltiples soluciones para el valor buscado se puede llegar a una solución no deseada, eso se debe principalmente debido a una mala elección de aproximación inicial. Por último, se puede presentar una convergencia lenta debido a la cercanía de una singularidad del Jacobiano [24, 25]

### 1.5.2 Método Quasi-Newton

Es llamado también el método BFGS (Broyden-Fletcher-Goldfarb-Shanno) [24, 25]. El cual no utiliza la inversión del jacobiano para obtener una nueva aproximación, en cambio utiliza una aproximación de una matriz que es actualizada en cada iteración. Esto trae una ventaja de no tener que recalcular el jacobiano ni su inversión en cada iteración. Por otro lado este método no tiene la convergencia cuadrática y no se autocorrigue en cada iteración

Este método utiliza la inversión de la matriz de Hessian (Hessian Matrix) [24, 25] es calculada aproximadamente en cada iteración. La matriz de Hessian se define de la siguiente forma

$$G_{i,j} = \frac{\delta^2 F(x)}{\delta x_i \delta x_j} \dots (i, j = 1, \dots, n)$$

Siendo la segunda derivada parcial del vector función con respecto a las coordenadas generalizadas. La inversión de la matriz de Hessian es reemplazada por  $H^r$  donde

$r = 0 \dots \infty$ , ya que cuando tiende al infinito se aproxima al valor de la matriz de Hessian invertida. Entonces la ecuación iterativa se define como:

$$x^{i+1} = x^i + \alpha^i d^i \dots\dots (1.23)$$

Donde  $d^i$  se define como la nueva dirección de aproximación:

$$d^i = - H^i g^i \dots\dots (1.24)$$

La ecuación 13 converge si la matriz  $H^r$  es positiva y el escalar  $\alpha$  se elige de tal manera que reduce la ecuación función. Se debe destacar que se realiza una aproximación de la matriz de Hessian de la siguiente forma [24, 25]:

$$H_{i+1} = H_i + Term\ 1 - Term\ 2 \dots\dots\dots (1.25)$$

Dónde:

$$Term\ 1 = \frac{(x_{i+1} - x_i) \otimes (x_{i+1} - x_i)}{(x_{i+1} - x_i) \cdot (\nabla f_{i+1} - \nabla f_i)}$$

$$Term\ 2 = \frac{[H_i \cdot (\nabla f_{i+1} - \nabla f_i)] \otimes [H_i \cdot (\nabla f_{i+1} - \nabla f_i)]}{(\nabla f_{i+1} - \nabla f_i) \cdot H_i \cdot (\nabla f_{i+1} - \nabla f_i)}$$

Donde  $\nabla f_i$  es la gradiente de la ecuación objetiva en la iva iteración. Todos estos artilugios pueden volver compleja la implementación de este método iterativo [24, 25].

## 1.6 Modelación del mecanismo

Un mecanismo es una serie de elementos interconectados mediante juntas, las cuales generan las restricciones en el mecanismo. Para obtener el argumento o la entrada de la configuración del mecanismo se tiene dos opciones, propuestas por Featherstone [3]:

- Trabajar el sistema aparte simbólicamente e ingresarlo a un programa para el cálculo del sistema de ecuaciones dado
- Crear un modelo que describa el sistema a analizar e ingresar lo obtenido a:
  - Un programa base para el cálculo cinemático
  - Un programa base para la obtención de la ecuación de movimiento.

Siendo la segunda posibilidad la más recomendada para el desarrollo de un algoritmo para el análisis cinemático de mecanismos en tres dimensiones.

El modelo del mecanismo debe poseer la descripción de los componentes del mecanismo y como es la junta que poseen dos eslabones. Se puede usar un gráfico de conexiones [3]:

- Un punto representa un eslabón
- Los arcos representan las juntas
- Uno de los puntos representa el bastidor
- El grafico no tiene dirección
- Existe siempre conexión entre nodos

Se presenta un ejemplo en la Fig 1.6:

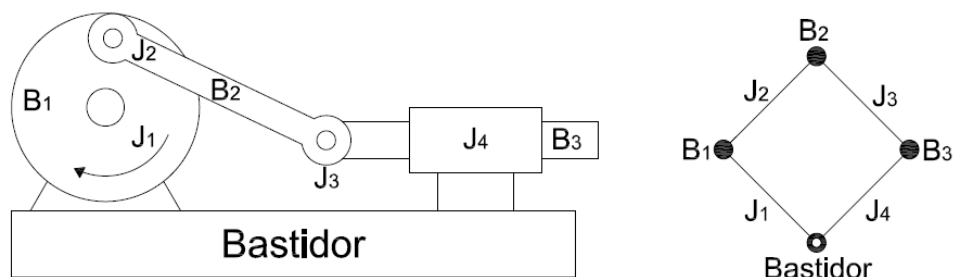


Figura 1.6: Mecanismo tipo biela-manivela con su respectivo grafico de conexiones. En el cual cada punto representa un eslabón y cada línea su junta

Para poder ingresar la geometría y la relación de las restricciones debido a las juntas del mecanismo a la computadora se debe seguir ciertos pasos, guiándose de lo propuesto por Featherstone [3]:

- Determinar el predecesor y sucesor en la junta dependiendo del tipo de junta puede llegar a ser importante su correcta definición.
- Determinar la relación de geométrica de cada eslabón con los tipos de juntas, donde cada eslabón posee sus propias coordenadas y un sistema de referencia general relacionada al bastidor.
- Relacionar cada marco de referencia o coordenadas de cada eslabón con su predecesor u con otro.
- Modelación de las juntas, para lo cual se requiere una librería con los tipos de juntas. Donde para cada tipo de junta se debe especificar de qué tipo es usando como base la librería mencionada anteriormente.
- Verificación de la relación entre predecesor y sucesor es correcta, de predecesor a sucesor, ya que si se obvia esta parte puede originarse ciertos problemas.

Por lo que el ingreso de datos sobre el mecanismo o sistema multicuerpos a analizar se debe realizar cuidadosamente y minuciosamente, en la figura 1.7 se muestra un gráfico sobre la modelación de un sistema compuesto por cuerpos rígidos.

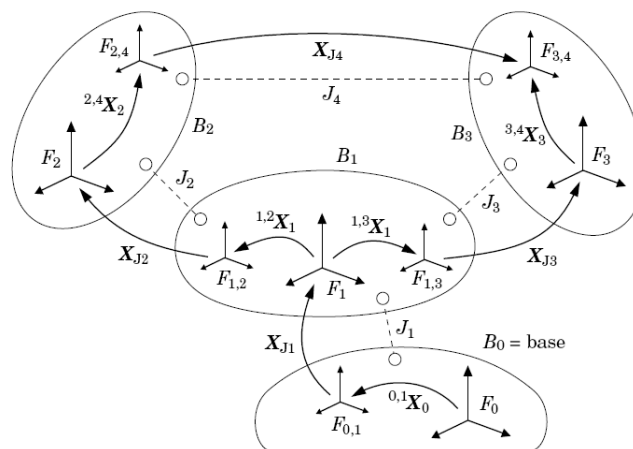


Figura 1.7: Modelo geométrico de un sistema compuesto por cuerpos rígidos [3].

Según el tipo de mecanismos se debe tomar ciertas consideraciones o tener cierta cantidad de información de la configuración sistema para realizar un correcto modelado y obtener una solución o simulación correcta. Para lo cual se debe diferenciar en dos tipos de mecanismos de tipo cadena cinemática abierta (tipo árbol o serial) o cadena cinemática cerrada.

### 1.6.1 Cadena Cinemática Abierta

La cadena cinemática abierta se puede diferenciar según su configuración puede ser de tipo árbol o de tipo serial, siendo la más importante la última. Mostrándose un ejemplo en la figura 1.8 sobre cadena cinemática abierta.

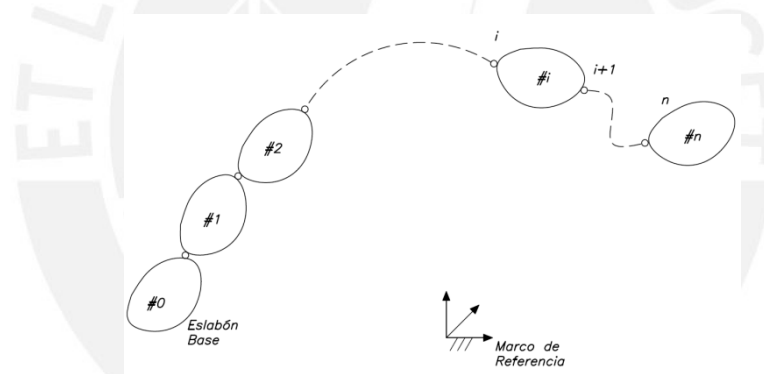


Figura 1.8: Se muestra un sistema multicuerpos compuesto por  $n$  eslabones y de tipo serial [5].

Para el cual se puede obtener un gráfico de conexión mostrado en la figura 1.9, en este caso para la correcta modelación del sistema se requiere los siguientes datos:

- Número de elementos que conforma el mecanismo
- El tipo de junta y su modelación respectiva en una matriz
- La *matriz padre*
- Una matriz de la transformación del sistema.



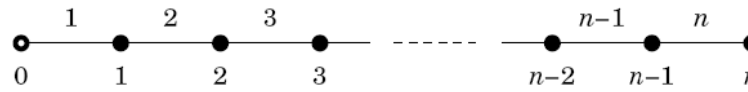


Figura 1.9: Se muestra el grafico de conexión para el caso de un sistema multicuerpos del tipo cadena cinemática abierta [3].

### 1.6.2 Cadena Cinemática Cerrada

Para el caso de cadena cinemática cerrada se debe realizar ciertos arreglos y ciertas consideraciones para poder calcularla como un sistema tipo cadena cinemática abierta. Para lo cual se debe abrir la cadena generando árboles separados (spanning tree) que son un subconjunto del sistema total, existen varios árboles separados los cuales contiene todos los elementos del sistema pero sin cierta cantidad de uniones. En la figura 1.10 se puede observar un ejemplo de cómo se abre un sistema del tipo cadena cinemática cerrada en varios árboles separados, los cuales se pueden tratar como cadenas cinemáticas abiertas [3].

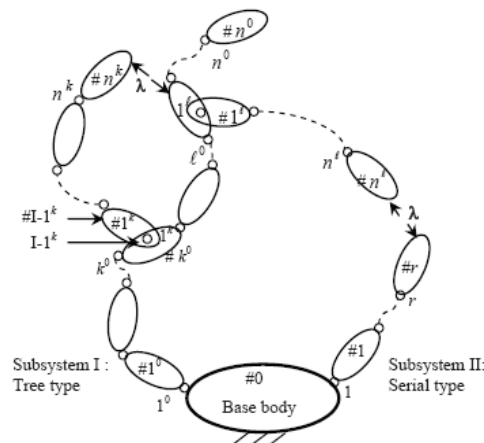


Figura 1.10: Se muestra una cadena cinemática cerrada la cual se ha eliminado ciertas juntas para transformarlo en una cadena cinemática abierta [5].

Se debe entender que cada cadena cinemática abierta puede generar diversos árboles separados de los cuales se debe seleccionar el más fácil de calcular, buscando que estén interconectados en línea. Otro elemento a notar al eliminar cierta cantidad de juntas es la creación de ciclos los cuales son iguales al número de juntas eliminadas. Estos ciclos indican la cantidad de cadenas cinemáticas cerradas existentes, siendo este el número mínimo de cadenas o consideraciones al realizar el análisis cinemático

del sistema analizado, en la figura 1.11 se muestra un ejemplo de los ciclos y las juntas eliminadas (chords) [3].

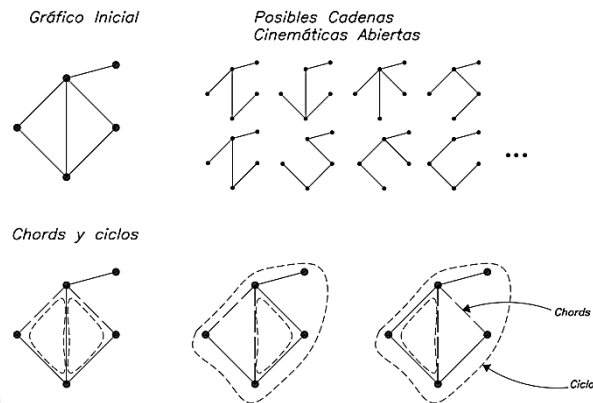


Figura 1.11: Se muestra la cantidad de árboles separados (spanning trees) que se pueden obtener de una cadena cinemática abierta y las juntas eliminadas y ciclos.

Para poder realizar la correcta modelación del mecanismo se requiere ciertos datos a ingresar extras, además de los pedidos para cadenas cinemáticas abiertas son:

- El número de cadenas independientes en el sistema, que es igual al número de juntas eliminadas para formar la cadena cinemática abierta.
- Para cada tipo de junta en la cadena se requiere:
  - Definir el predecesor y sucesor
  - El tipo de junta
  - La transformación de sucesor y predecesor

## CAPÍTULO 2

### DESARROLLO DE ALGORITMO PARA EL ANÁLISIS CINEMÁTICO INVERSO DE MECANISMOS EN 3D

En el presente capítulo se procederá a la selección de métodos para el desarrollo del algoritmo para la obtención del modelo cinemático inverso de mecanismo. Este algoritmo debe poder analizar mecanismos en 3D y cadenas cinemáticas abiertas y cerradas.

Una vez definido los métodos básicos o claves para el algoritmo, se procede a desarrollar el algoritmo general. En primera instancia se establece las entradas, el diagrama de flujo y el pseudocódigo necesario para poder procesar toda la información. En este punto se desarrolla la matemática a ser usada, las formulaciones intermedias requeridas, el desarrollo del modelo cinemático, la inter-relación de las coordenadas generalizadas con la velocidad y aceleración de cuerpos rígidos en el espacio. También se desarrolla las librerías de juntas a ser implementadas, en la cual se desarrollan las juntas bases y las juntas complejas.

Por último se define las salidas que se obtienen del algoritmo y la incidencia de la modelación con respecto a la precisión y eficiencia del algoritmo.

#### 2.1 Selección de los métodos

El algoritmo a ser implementado se basara en el método iterativo, ya que se requiere su aplicación a diversos mecanismos. Por lo que se ha de seleccionar tanto el método

de la inversión del jacobiano, método de minimización y la selección de las coordenadas generalizadas para poder definir el marco teórico del algoritmo.

### **2.1.1 Inversión del Jacobiano**

Debido a las grandes dificultades en la inversión del jacobiano y la búsqueda de la mejor manera para su inversión priorizando su exactitud y su buen comportamiento en la cercanía a singularidades. En esta tesis se ha optado el uso del método de la pseudo inversión el cual garantiza una mejor exactitud de respuesta y su resultado con respecto a la transposición del jacobiano, además de su amplio uso en el campo de la cinemática inversa.

Se ha de tomar en cuenta que en el algoritmo se tendrá una subrutina en base a la magnitud del determinante del jacobiano, ya que al aproximarse demasiado a cero o ser de un valor demasiado grande (mayor a 1000) muestra una alarma frente a la cercanía de una singularidad. Con dicha subrutina se podrá gestionar mejor el algoritmo dando alertas frente a posibles singularidad para lo cual el usuario podrá generar un mejor modelo del mecanismo o la búsqueda de una solución frente a ella.

### **2.1.2 Método de Minimización**

El método de minimización a utilizar es el método Newton-Raphson detallado en el Capítulo 1. Se selecciona este tipo de minimización debido a su fácil implementación y su convergencia cuadrática, el cual reduce la cantidad de iteración para obtener la respuesta aumentando así su velocidad de respuesta.

El error permisible para la respuesta será un dato de entrada estipulado por el usuario para poder obtener una respuesta satisfactoria y según los requerimientos del usuario. Pudiendo priorizar entre la exactitud o velocidad de resolución del análisis cinemático del mecanismo (siendo el error estándar del algoritmo de 0.00001)

### 2.1.3 Coordenadas Generalizadas

Para el sistema de coordenadas generalizadas se procederá a usar los cuaterniones o conocidos también como los parámetros de Euler, los cuales brindan diversas ventajas respecto a los ángulos de Euler tales como evasión de singularidad y orientación de cuerpos no deseados por la sucesión de giros a realizar.

Por lo que se tendrá 3 parámetros cinemáticos para la posición (coordenadas cartesianas) y 4 parámetros para la orientación del mecanismo (parámetros de Euler). En el anterior capítulo se explicó que los parámetros de Euler no son independientes entre sí, por lo que se tendrá una ecuación de normalización de los parámetros, desarrollándose como una restricción, a detallar más adelante.

## 2.2 Descripción y desarrollo del algoritmo

Seleccionado los métodos de resolución del sistema se procede a desarrollar el algoritmo, el cual será implementado de manera básica en Matlab.

Para el correcto desarrollo del algoritmo y la obtención de la cinemática inversa del mecanismo de manera integral (posición, velocidad y aceleración), se requiere realizar los siguientes pasos:

1. Calcular la posición de todos los eslabones del mecanismo
2. Calcular la velocidad de todos los eslabones del mecanismo
3. Calcular la aceleración de todos los eslabones del mecanismo

Para poder ejecutar el algoritmo se requerirá cierta información del mecanismo al cual nos referiremos en adelante como modelo del mecanismo, el cual debe poseer la relación entre los eslabones, el tipo de junta o par cinemático, las matrices para la transformación de coordenadas entre los eslabones, entre otros parámetros que se van a detallar más adelante.

### 2.2.1 Entradas – Modelo cinemático

El usuario del programa debe ingresar cierta información vital para poder obtener un correcto modelo cinemático del mecanismo, entre los cuales están:

- **Características de los Mecanismo**, la información a ingresar básicamente es la cantidad de eslabones que lo componen.
- **Parámetros de estudio**, se debe ingresar el intervalo de estudio y la cantidad de iteración a estudiar.
- **Identificación de juntas**, se ingresa la información sobre el tipo de junta o par cinemático que representa por ejemplo de revolución, prismático, etc. Para lo cual se posee una biblioteca con las propiedades del par cinemático.
- **Relación entre las juntas y los eslabones**, se ingresa la información sobre la relación entre el eslabón predecesor y sucesor del par cinemático y la posición de las juntas en el eslabón.
- **Identificación de las Juntas Motrices**, se ingresa la información sobre las juntas motrices que genera el movimiento o movimiento deseado del mecanismo a estudiar. Para lo cual se posee una biblioteca de juntas similar a las juntas entre los eslabones.
- **Características de las juntas Motrices**, se ingresa las características cinemáticas de las juntas motrices tales como la función en el tiempo de sus posición y por ende de su velocidad y aceleración.
- **Posición Inicial**, se debe ingresar una primera estimación de la posición inicial del mecanismo para poder comenzar con el proceso iterativo seleccionado. Se resalta que es un punto crítico en el análisis cinemático para su correcto desarrollo.

### 2.2.2 Diagrama de Flujo

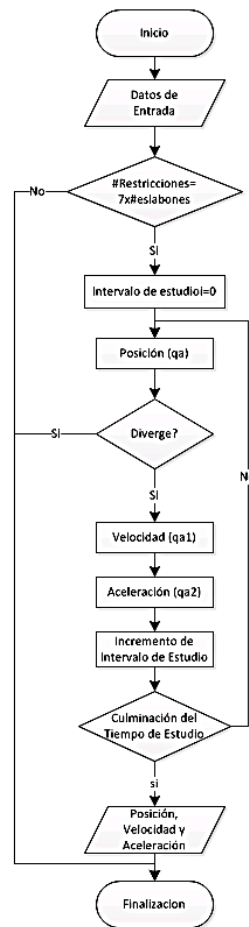
Se procede a presentar los diagramas de flujo del algoritmo principal para la cinemática inversa. Posteriormente se procede a mostrar los diagramas de flujo de las sub-rutinas que realiza y se indica el diagrama de flujo principal.

#### Cinemática Inversa

En el siguiente diagrama de flujo se puede observar que primero se debe ingresar cierta información sobre el mecanismo (número de eslabones), información del proceso como el intervalo a analizar y la cantidad de puntos a calcular. También se debe ingresar los tipos de juntas y cierta información que permita definirla.

Si se cumple que la cantidad de restricciones, originada por las juntas, normalización de los parámetros de euler y las juntas motrices, sea igual al número de coordenadas generalizadas, en este caso al usar los parámetros de euler es igual a siete veces el número de eslabones. Se procede a calcular la posición inicial, siendo una sub-rutina presentada como un proceso. Si el proceso empieza a diverger se cierra el programa, ya que ha sido mal modelado, si converge se procede a calcular la velocidad y aceleración en ese periodo de tiempo.

Luego se repite el proceso hasta alcanzar el periodo de tiempo establecido y la cantidad de puntos analizados (t). Obteniéndose al final la posición, velocidad y aceleración de todos los eslabones en el intervalo de tiempo ingresado.

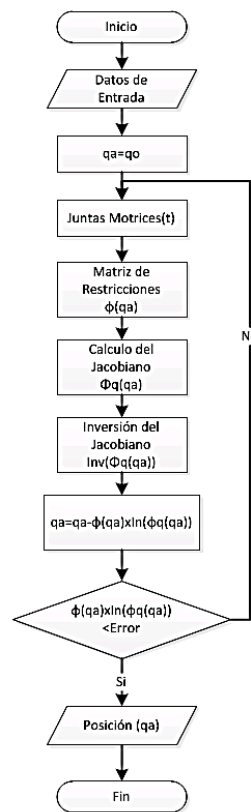


Subrutina de posición. La subrutina para obtener la posición inicial, y la posición para cada intervalo tiene el siguiente diagrama de flujo. Donde se tiene una posición inicial ingresada anteriormente sobre los tipos de juntas, número de eslabones, juntas motrices y la primera estimación inicial.

Donde se tiene subrutinas para obtener el jacobiano y la matriz de restricciones para una estimación inicial. Cuando el error obtenido debido a una estimación inicial de posición (matriz de restricción / jacobiano) es menor a una tolerancia especificada por el usuario, se procede a la siguiente posición. Si no cumple vuelve a iterar los valores de la matriz de restricción y el jacobiano para una nueva estimación, ese proceso se repite hasta obtener un error aceptable.



Para obtener la siguiente posición se usa la posición anterior, como primera estimación, considerando que su variación es baja. Por lo que el número de puntos a analizar debe ser razonable para poder obtener una convergencia rápida.



### 2.2.3 Pseudocódigo

El algoritmo a desarrollar se divide en sub-rutinas para los diferentes procesos que realiza el programa para obtener el resultado deseado. Por lo que se va a presentar un algoritmo básico, en el cual se incluyen sub-rutinas, y posteriormente un algoritmo de fácil entendimiento de las sub-rutinas y su labor en el programa desarrollado.

Las limitaciones del algoritmo se basan en las juntas desarrolladas y seleccionadas, en este caso juntas de revolución, esférica, cilíndrica, de traslación y distancia, para las cuales se ha desarrollado una librería de fácil acceso la cual puede ser ampliada para el desarrollo de otros tipos de junta tales como pares de engranajes entre otros.

### 2.2.3.1 Modelo Cinemático

Para el desarrollo del algoritmo de análisis cinemático y usando el diagrama de flujo propuesto en el acápite 2.2.2 se debe analizar en primera instancia la posición de las juntas y eslabones del mecanismo posteriormente se analiza la velocidad y la aceleración una vez encontrada la posición correcta.

Mencionado lo anterior se debe obtener la posición inicial a partir de una aproximación inicial para lo cual se generara un vector que contenga las ecuaciones características de cada tipo de junta (tanto motriz como de restricción) y a su vez de las restricciones originadas al usar los parámetros de Euler (normalización de los parámetros de Euler). Dicho esto se procede a detallar la ecuación característica del mecanismo:

$$\phi_{(q,t)} = \begin{bmatrix} \phi_{(q)}^K \\ \phi_{(q,t)}^D \\ \phi_{(p)}^P \end{bmatrix} = \mathbf{0}_{7nb} \dots \dots \dots (2.1)$$

Dónde:

- $\phi_{(q)}^K$  : Ecuaciones debido a la geometría y pares cinemáticos, en funciones de las coordenadas generalizadas y del tiempo.
- $\phi_{(q,t)}^D$  : Ecuaciones con respecto a ciertos eslabones motrices o movimiento deseado de algún eslabón.
- $\phi_{(p)}^P$  : Ecuaciones de la normalización de los parámetros de Euler de cada eslabón que lo conforman.

Se debe tomar en cuenta que esta matriz es igual al número de grados de libertad del mecanismo (7 x número de eslabones,  $7nb$ ). En caso no se cumpla se termina el programa, como se muestra en el diagrama de flujo de la cinemática inversa.

Obtenida este vector con las ecuaciones de la librería de juntas (más adelante se detallara las ecuaciones a usar) se procede a usar el método de minimización Newton-

Raphson, mostrada en el diagrama de flujo de subrutina de posición, para lo cual se genera el Jacobiano.

Superado el error estipulado por el usuario se procede a obtener la posición del mecanismo para un determinado tiempo, posteriormente se procede a calcular la velocidad con la siguiente ecuación (derivada con respecto al tiempo de la ecuación:

$$\phi_q \dot{q} = -\phi_t \dots \dots \dots (2.2)$$

Dónde:

- $\phi_t$  : Es el vector que contiene a las derivadas de las ecuaciones de movimiento del mecanismo con respecto al tiempo, más adelante llamada  $v$ .
- $\phi_q$  : La matriz que contiene el Jacobiano del mecanismo.
- $\dot{q}$  : El vector que contiene velocidad de los eslabones del mecanismo con respecto a las coordenadas generalizadas usadas.

De la ecuación (2.2) se puede obtener el vector velocidad  $\dot{q}$  aplicando la inversa del Jacobiano y multiplicándolo por el vector  $v$ , observándose en la siguiente ecuación:

$$\dot{q} = \phi_q^{-1} v \dots \dots \dots (2.3)$$

Obtenida la velocidad se procede al cálculo de la aceleración para lo cual se requiere realizar la segunda derivada con respecto al tiempo de la ecuación (2.1), obteniéndose la siguiente ecuación:

$$\phi_q \ddot{q} = -(\phi_q \dot{q})_q \dot{q} - 2\phi_{qt} \dot{q} - \phi_{tt} = \gamma \dots \dots \dots (2.4)$$

Dónde:

- $\gamma$  : Es un vector que contiene una serie de derivadas, multiplicaciones y subtracciones de matrices, las cuales se detallan según el tipo de junta más adelante.
- $\phi_q$  : La matriz que contiene el Jacobiano del mecanismo.
- $\phi_{qt}$  : La matriz que contiene la derivada del Jacobiano del mecanismo con respecto al tiempo.
- $\phi_{tt}$  : El vector que contiene la segunda derivada con respecto del tiempo de las ecuaciones del movimiento del mecanismo.
- $\ddot{q}$  : El vector que contiene velocidad de los eslabones del mecanismo con respecto a las coordenadas generalizadas usadas.

Tal como se hizo para la velocidad se puede obtener una ecuación simplificada de la ecuación (2.4) para obtener el vector velocidad de los eslabones del mecanismo, tal como se muestra en la ecuación (2.5):

$$\ddot{q} = \phi_q^{-1} \gamma \dots \dots \dots (2.5)$$

Se puede observar y concluir que el análisis de posición es uno de los más complicados, ya que se requiere obtener una aproximación inicial, usar el método Newton-Raphson y la inversión del Jacobiano para obtener la posición real del mecanismo. Posterior a este cálculo el análisis de velocidad es relativamente sencillo, ya que se tiene previamente la inversión del Jacobiano, mientras que el análisis de aceleración es el más tedioso de todos debido a la complejidad del vector  $\gamma$ .

### 2.2.3.2 Parámetros de Euler

El acápite 1.2.4 se detalló las características de los parámetros de Euler, los cuales representan a un vector y un giro con respecto a él para definir la rotación en el espacio del cuerpo rígido. El vector posición de cada eslabón se puede definir como:

$$q = [\mathbf{r}, \mathbf{p}]^T = [\mathbf{r}, e_0, \mathbf{e}^T]^T = [r_x, r_y, r_z, e_0, e_1, e_2, e_3]^T \dots\dots\dots (2.6)$$

Dónde:

$\mathbf{r}$  : Es el vector posición, compuesto por las tres coordenadas cartesianas conocidas ( $\mathbf{r} = [r_x, r_y, r_z]^T$ ).

$\mathbf{p}$  : Es el vector de rotación, compuesto por los cuatro parámetros de Euler ( $\mathbf{p} = [e_0, \mathbf{e}^T]^T$ ).

$e_0$  : Es el parámetro de Euler que define principalmente el giro ( $e_0 = \cos(x/2)$ ).

$\mathbf{e}$  : Es el vector que contiene el resto de los parámetros de Euler, el cual contiene el eje de rotación ( $\mathbf{u}$ ) del eslabón ( $\mathbf{e} = [e_1, e_2, e_3]^T = \mathbf{u} \sin(x/2)$ ).

Para mayor comprensión del eje de rotación y de los parámetros de Euler se puede apreciar la figura 2.1, donde se puede apreciar que el marco de referencia  $x - y - z$  gira con respecto al eje  $\mathbf{u}$ , dando origen al nuevo sistema de referencias  $x' - y' - z'$ .

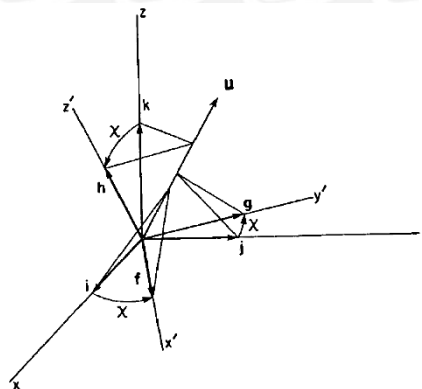


Figura 2.1, esquema donde se puede apreciar el eje de rotación y el ángulo de rotación, definiendo así los parámetros de Euler [24].

Como se mencionó en el acápite 1.1.4 y 2.1.2.1 se requiere utilizar una ecuación que interrelacione los parámetros de Euler, ya que son interdependiente, la cual se le llama la normalización de los parámetros de Euler. Dicha ecuación se muestra a continuación:

$$\mathbf{p} \mathbf{p}^T = 1 \dots\dots\dots (2.7)$$

La ecuación (2.7) se debe cumplir para todos los eslabones en todo instante de análisis de la cinemática inversa del mecanismo. Entonces se tiene que  $\phi_{(P)}^P$  debe ser:

$$\phi_{(P)}^P = \begin{bmatrix} P_1^T P_1 - 1 \\ \vdots \\ P_{nb}^T P_{nb} - 1 \end{bmatrix} = \mathbf{0}_{nb} \dots\dots\dots (2.8)$$

Definida la ecuación (2.8) se procede a desarrollar su Jacobiano con respecto a las coordenadas generalizadas para cada eslabón:

$$\phi_{iq_i}^P = [\mathbf{0}, 2P_i^T] \dots\dots\dots (2.9)$$

Además de obtener el Jacobiano, es requerido obtener el vector  $v$  para el análisis de la velocidad y obtener la velocidad de los eslabones que componen el mecanismo. Entonces se debe proceder a derivar la ecuación (2.7) con respecto al tiempo y se obtiene lo siguiente:

$$v^P = \mathbf{0}_{nb} \dots\dots\dots (2.10)$$

Posteriormente es requerido obtener el vector  $\gamma$  el cual serviría para el cálculo de las aceleraciones ( $\ddot{q}$ ), entonces:

$$\gamma^P = \begin{bmatrix} -2 \dot{p}_1^T \dot{p}_1 \\ \vdots \\ -2 \dot{p}_{nb}^T \dot{p}_{nb} \end{bmatrix} \dots\dots\dots (2.11)$$

Donde  $\dot{p}$  se obtiene del análisis de velocidad del mecanismo.

### 2.2.3.3 Matriz de Rotación

Las matrices de rotación nos permiten cambiar nuestro sistema de referencia de un eslabón o cuerpo a otro, este método nos permite desplazarnos entre diferentes sistemas de coordenadas o referencia a otras para la facilidad e interpretación de la geometría y ubicación de eslabones.

Se debe tomar en cuenta que la matriz de rotación depende directamente de las coordenadas generalizadas a usar por ejemplo ángulos de Euler, cuaterniones o parámetros de Denavit-Hartenberg [24,25].

Para la rotación simple de un sistema de referencia a otro de un vector definido en el primer eslabón, teniendo el mismo origen ambos sistema de referencia se emplea la siguiente ecuación:

$$s^1 = A^{12} s^2 \dots\dots\dots(2.12)$$

Dónde:

- $s^1$  : es el vector en el marco de referencia uno,  $s = [s_x \ s_y \ s_z]^T$
- $s^2$  : es el vector en el marco de referencia dos,  $s = [s_{x'} \ s_{y'} \ s_{z'}]^T$
- $A^{12}$  : es la matriz de rotación del marco de referencia dos al uno.

Donde la matriz de rotación es:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \dots\dots\dots (2.13)$$

La matriz de rotación es ortogonal por lo tanto se tiene la siguiente propiedad [24]:

$$A^T A = I \dots\dots\dots (2.14)$$

Esta propiedad permite la fácil inversión de la matriz,  $A^T = A^{-1}$ , con la cual obtendremos la siguiente ecuación:

$$s^2 = A^{12T} s^1 \dots\dots\dots(2.15)$$

Cuando el origen de ambos sistemas de coordenadas no coinciden, como se muestra en la figura 2.1, se utiliza la siguiente formula:

$$r^P = r + A s'^P \dots\dots\dots (2.16)$$

Dónde:

- $r^P$  : es el vector del punto P en el marco de referencia uno
- $r$  : es el vector del origen del sistema de referencia dos en el marco de referencia uno
- $A$  : es la matriz de rotación del segundo marco de referencia con respecto al primero
- $s'^P$  : es el vector del punto P en el marco de referencia numero dos

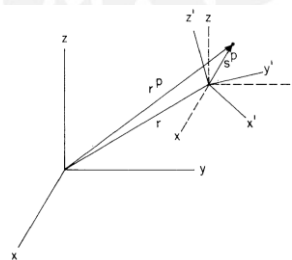


Figura 2.2, Traslación y Rotación del marco de Referencia [24].

Además de ver las matrices de rotación desde un punto análisis de posición, se debe tomar en cuenta su importancia e impacto en el análisis de velocidad, aceleración y velocidad angular del mecanismo y eslabones.



Se debe tomar en cuenta que en un mecanismo la matriz de posición está en función del tiempo y también el vector de posición  $r$ , mientras que  $s'^P$  no necesariamente está en función del tiempo, ya que representa la posición relativa de la junta con respecto al centro de gravedad del eslabón (considerando los eslabones como cuerpos rígidos).

Entonces la derivada de la ecuación (2.16) con respecto al tiempo es:

$$\dot{r}^P = \dot{r} + \dot{A} s'^P \dots\dots\dots (2.17)$$

Usando la ecuación (2.15) se puede simplificar la ecuación (2.17) en un solo sistema de referencia:

$$\dot{r}^P = \dot{r} + \dot{A} A^T s^P \dots\dots\dots (2.18)$$

Para su mayor comprensión de la derivada y su utilización en la elaboración del pseudocódigo, por lo que se derivara la ecuación (2.14) con respecto al tiempo:

$$\dot{A} A^T + A \dot{A}^T = 0 \dots\dots\dots (2.19)$$

Esto implica que:

$$(\dot{A} A^T)^T = A \dot{A}^T = -\dot{A} A^T \dots\dots\dots (2.20)$$

Analizando más las ecuaciones obtenidas podemos observar que la matriz  $\dot{A} A^T$  es un matriz anti simétrica. Entonces al ser una matriz anti simétrica, existe un vector  $\omega$  llamado velocidad angular del segundo sistema de referencia tal que:

$$\tilde{\omega} = \dot{A} A^T \dots\dots\dots (2.21)$$

Reemplazando la ecuación (2.21) en la ecuación (2.18) se tiene lo siguiente:

$$\dot{r}^P = \dot{r} + \tilde{\omega} s^P \dots\dots\dots (2.22)$$

La ecuación (2.22) es muy similar a la ecuación de análisis cinemático usado regularmente mediante vectores  $\dot{r}^P = \dot{r} + \vec{\omega} \times s^P$ . Donde  $\vec{\omega}$  es la velocidad angular del segundo sistema de referencia, relativo al primer marco de referencia. Tomándose el producto vector en el segundo marco de referencia y posteriormente pasando al primer sistema de referencia la ecuación 2.11 se puede expresar como:

$$\dot{r}^P = \dot{r} + A \tilde{\omega}' s'^P \dots\dots\dots (2.23)$$

Otra relación importante se presenta al multiplicar ambos lados de la ecuación (2.21) por la matriz de rotación A:

$$\tilde{\omega} A = \dot{A} \dots\dots\dots (2.24)$$

Aplicando las propiedades de las matrices asimétricas y de la siguiente relación  $(A s') = \tilde{s} = A \tilde{s}' A^T$ , la ecuación 2.13 se puede simplificar en la siguiente ecuación:

$$\tilde{\omega}' = A^T \dot{A} \dots\dots\dots (2.25)$$

Las ecuaciones o relaciones (2.24) y (2.25) se usaran frecuentemente en el desarrollo del algoritmo.

Diferenciando una vez más la ecuación (2.17) se logra obtener la ecuación de la aceleración:

$$\ddot{r}^P = \ddot{r} + \ddot{A} s'^P \dots\dots\dots (2.26)$$

Para obtener mayor concepción de  $\ddot{A}$  derivamos la ecuación (2.24) obteniendo la siguiente ecuación:

$$\ddot{A} = \tilde{\omega} A + \tilde{\omega} \dot{A} = \tilde{\omega} A + \tilde{\omega} \tilde{\omega} A \dots\dots\dots (2.27)$$

Aplicando la relación  $(A s') = \tilde{s} = A \tilde{s}' A^T$ , en la ecuación (2.27) se puede obtener la segunda derivada con respecto del tiempo de la matriz A con respecto al segundo marco de referencia:

$$\ddot{A} = A \tilde{\omega}' + A \tilde{\omega}' \tilde{\omega}' \dots\dots\dots (2.28)$$

En esta tesis se usarán las coordenadas generales conocidas como Parámetros de Euler el cual posee 4 valores que definen la rotación del marco de referencia del eslabón, tal como se mencionó en el capítulo 1. A continuación se detalla las subrutinas para la elaboración de la matriz de rotación usando parámetros de Euler. La matriz de rotación se puede definir con la siguiente ecuación:

$$A = (2e_0^2 - 1)I + 2(e e^T + e_0 \tilde{e}) \dots\dots\dots (2.29)$$

$$A = 2 \begin{bmatrix} e_0^2 + e_1^2 - 0.5 & e_1 e_2 - e_0 e_3 & e_1 e_3 + e_0 e_2 \\ e_1 e_2 + e_0 e_3 & e_0^2 + e_2^2 - 0.5 & e_2 e_3 - e_0 e_1 \\ e_1 e_3 - e_0 e_2 & e_2 e_3 + e_0 e_1 & e_0^2 + e_3^2 - 0.5 \end{bmatrix} \dots\dots\dots (2.30)$$

**2.2.3.4 Propiedades de los Parámetros de Euler y relación con velocidad angular**

Los parámetros de Euler tienen ciertas propiedades que benefician el cálculo y aumentan la robustez y velocidad del algoritmo, los cuales tendrán subrutinas para su elaboración y su aplicación. En las subrutinas de la matriz de rotación y la modelación de las juntas.

Se define dos matrices E y G (matrices de 3 x 4):

$$\mathbf{E} = [-\mathbf{e}, \tilde{\mathbf{e}} + e_0 \mathbf{I}] = \begin{bmatrix} -e_1 & e_0 & -e_3 & e_2 \\ -e_2 & e_3 & e_0 & -e_1 \\ -e_3 & -e_2 & e_1 & e_0 \end{bmatrix} \dots\dots\dots (2.31)$$

$$\mathbf{G} = [-\mathbf{e}, -\tilde{\mathbf{e}} + e_0 \mathbf{I}] = \begin{bmatrix} -e_1 & e_0 & e_3 & -e_2 \\ -e_2 & -e_3 & e_0 & e_1 \\ -e_3 & e_2 & -e_1 & e_0 \end{bmatrix} \dots\dots\dots (2.32)$$

Obsérvese que estas matrices tienen las siguientes relaciones, se debe enfatizar que  $\mathbf{I}_4$  es una matriz identidad de 4x4:

$$\mathbf{E}\mathbf{p} = \mathbf{G}\mathbf{p} = \mathbf{0} \dots\dots\dots (2.33)$$

$$\mathbf{E}\mathbf{E}^T = \mathbf{G}\mathbf{G}^T = \mathbf{I} \dots\dots\dots (2.34)$$

$$\mathbf{E}^T \mathbf{E} = \mathbf{E}^T \mathbf{E} = \mathbf{I}_4 - \mathbf{p}\mathbf{p}^T \dots\dots\dots (2.35)$$

Comparando la multiplicación de las matrices  $\mathbf{E}$  y  $\mathbf{G}$  ( $\mathbf{E}\mathbf{G}^T$ ) y comparándola con la ecuación (2.29) se concluye en la siguiente relación:

$$\mathbf{A} = \mathbf{E}\mathbf{G}^T \dots\dots\dots (2.36)$$

Derivando las ecuaciones (2.33) y (2.25) se puede obtener las siguientes ecuaciones:

$$\mathbf{E}\dot{\mathbf{p}} = -\dot{\mathbf{E}}\mathbf{p} \dots\dots\dots (2.36)$$

$$\mathbf{G}\dot{\mathbf{p}} = -\dot{\mathbf{G}}\mathbf{p} \dots\dots\dots (2.37)$$

$$\dot{\mathbf{p}}\mathbf{p}^T = \dot{\mathbf{p}}^T\mathbf{p} = \mathbf{0} \dots\dots\dots (2.38)$$

Haciendo una expansión directa y ciertos artificios en la ecuación (2.31) y (2.32) se puede obtener la siguiente ecuación:

$$\mathbf{E}\dot{\mathbf{G}}^T = \dot{\mathbf{E}}\mathbf{G}^T \dots\dots\dots (2.39)$$

Se derivaba ambas parte de la ecuación (2.36) y usando la relación encontrada anteriormente, ecuación (2.39), se obtiene la siguiente relación:

$$\dot{\mathbf{A}} = \dot{\mathbf{E}}\mathbf{G}^T + \mathbf{E}\dot{\mathbf{G}}^T = 2\mathbf{E}\dot{\mathbf{G}}^T \dots\dots\dots (2.40)$$

Empleando las ecuaciones (2.36) y (2.40) en la ecuación (2.25) se puede obtener la siguiente relación:

$$\tilde{\omega}' = \mathbf{A}^T\dot{\mathbf{A}} = 2\mathbf{G}\mathbf{E}^T\mathbf{E}\dot{\mathbf{G}}^T \dots\dots\dots (2.41)$$

Y reemplazando en la ecuación (2.41) las ecuaciones (2.33) y (2.35) se obtiene la siguiente ecuación:

$$\tilde{\omega}' = 2\mathbf{G}\dot{\mathbf{G}}^T = 2\widetilde{(\mathbf{G}\dot{\mathbf{p}})} \dots\dots\dots (2.42)$$

Entonces se puede obtener la siguiente relación para obtener una relación entre la velocidad angular y la primera derivada con respecto al tiempo de los parámetros de Euler:

$$\omega' = 2\mathbf{G}\dot{\mathbf{p}} \dots\dots\dots (2.43)$$

Multiplicado  $G^T$  ambos lados de la ecuación (2.43) y aplicando las ecuaciones (2.35) y (2.38) se obtiene la siguiente relación:

$$\dot{\mathbf{p}} = \frac{1}{2} \mathbf{G}^T \boldsymbol{\omega}' \dots\dots\dots (2.44)$$

Aplicando las ecuaciones (2.36), (2.43), (2.35) y la ecuación (2.38) se puede obtener la velocidad angular en el marco de referencia general:

$$\boldsymbol{\omega} = \mathbf{A}\boldsymbol{\omega}' = 2\mathbf{E}\mathbf{G}^T \mathbf{G}\dot{\mathbf{p}} = 2\mathbf{E}\dot{\mathbf{p}} \dots\dots\dots (2.45)$$

E invirtiendo la ecuación (3.45) se puede obtener una relación directa de la primera derivada con respecto al tiempo de los parámetros de Euler con la velocidad angular en el marco de referencia.

$$\dot{\mathbf{p}} = \frac{1}{2} \mathbf{E}^T \boldsymbol{\omega} \dots\dots\dots (2.46)$$

**2.2.3.5 Librería de Juntas**

Para la modelación de las juntas o relaciones entre eslabones se va generar cuatro juntas básicas de las cuales se desarrollaran las juntas de mayor complejidad. Estas juntas permitirán desarrollar el modelo del mecanismo a analizar, generando un vector con las ecuaciones de restricciones [24].

Hay que tener en cuenta que las juntas básicas podrán caracterizar las juntas motrices o desplazamiento y rotación del actuador. Las juntas tendrán parámetros requeridos para su modelación los cuales serán mencionados para cada tipo de junta, todas estas ecuaciones se desarrollaran usando las coordenadas generalizadas en base a los cuaterniones o parámetros de Euler.

En la tabla 2.1 se puede observar la relación de las juntas base con respecto a la modelación de juntas más complejas.

	<i>DP1</i>	<i>DP2</i>	<i>D</i>	<i>CD</i>
$\perp_1$	××			
$\perp_1$		××		
<i>SJ</i>				×××
<i>UJ</i>	×			[×××] <sub><i>SJ</i></sub>
<i>CJ</i>	[××] <sub><math>\perp_1</math></sub>	[××] <sub><math>\perp_2</math></sub>		
<i>RJ</i>	[××] <sub><math>\perp_1</math></sub>			[×××] <sub><i>SJ</i></sub>
<i>TJ</i>	× [××] <sub><math>\perp_1</math></sub>	[××] <sub><math>\perp_2</math></sub>		

Tabla 2.1, Tabla de relación entre juntas base con juntas de mayor complejidad (DP1: Producto punto tipo 1, DP2: Producto Punto tipo 2, D: Distancia, CD: Diferencia de Coordenadas,  $\perp_1$  Relación de Perpendicularidad a un plano tipo 1,  $\perp_2$  Relación de Perpendicularidad a un plano tipo 2, SJ: Junta Esférica, UJ: Junta Universal, CJ: Junta Cilíndrica, RJ: Junta de Revolución, TJ: Junta de Traslación [24].

### 2.2.3.5.1 Producto punto tipo 1 (DP1)

La junta Producto Punto Tipo 1 representa la relación (orientación) entre dos vectores definidos en dos eslabones diferentes ( $\vec{a}_i, \vec{a}_j$ ). Dicha relación se interpreta mediante el producto punto entre ambos vectores en un marco de referencia general, el valor del producto punto define la orientación entre ambos vectores y dicho valor puede variar con respecto al tiempo. Por ejemplo, una junta motriz o junta de revolución. En la figura 2.3 se puede apreciar este tipo de junta donde el producto punto de los vectores  $\vec{a}_i$  (perteneciente al cuerpo *i*) y  $\vec{a}_j$  (perteneciente al cuerpo *j*) tienen un determinado valor.

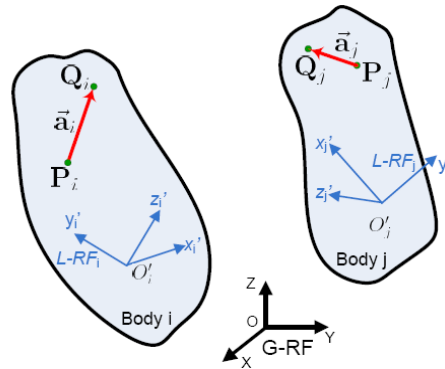


Figura 2.3, Junta tipo DP1 (Producto punto tipo 1) [25].

Dicho esto se procede a definir la ecuación de movimiento como:

$$\phi^{DP1}(i, \bar{a}_i, j, \bar{a}_j, f(t)) = \bar{a}_i^T A_i^T A_j \bar{a}_j - f(t) = 0 \quad \dots\dots\dots (2.47)$$

En la ecuación de movimiento se puede apreciar el vector  $\bar{a}_i$  definido en el eslabón  $i$  y el vector  $\bar{a}_j$  definido en el eslabón  $j$ . Ambos vectores son representados en un marco de referencia general multiplicándolos por su respectiva matriz de rotación, logrando así la ecuación (2.47), la cual puede tener un valor constante o variable en el tiempo definido por la función  $f(t)$ .

Definida la ecuación de movimiento se procede a desarrollar la ecuación de velocidad derivando la ecuación (2.47) con respecto al tiempo. Obteniéndose la siguiente ecuación:

$$v^{DP1}(i, \bar{a}_i, j, \bar{a}_j, f(t)) = -\frac{\delta \phi^{DP1}}{\delta t} = \frac{\delta f}{\delta t} \quad \dots\dots\dots (2.48)$$

Terminado las ecuaciones de movimiento se requiere el desarrollo de la ecuación de la aceleración, la cual se obtiene derivando dos veces la ecuación (2.47) con respecto al tiempo:

$$\gamma^{DP1} = -\bar{a}_j^T (A_j^T A_i \widetilde{\omega}_i \widetilde{\omega}_j + \widetilde{\omega}_j \widetilde{\omega}_j A_j^T A_i) \bar{a}_i + 2\widetilde{\omega}_j^T \bar{a}_j A_j^T A_i \widetilde{\omega}_i \bar{a}_i + \frac{\delta^2 f}{\delta t^2} \quad \dots\dots\dots (2.49)$$



Por último se requiere el desarrollo de la matriz del jacobiano, la cual se obtiene derivando la ecuación (2.47) con respecto a todas las coordenadas generalizadas, creando la siguiente matriz:

$$\Phi_{q_i, q_j}^{DP1}(\mathbf{a}_i, \mathbf{a}_j) = [\mathbf{0}_{1 \times 3} \quad \bar{\mathbf{a}}_j^T \mathbf{B}(p_i, \bar{\mathbf{a}}_i) \quad \mathbf{0}_{1 \times 3} \quad \bar{\mathbf{a}}_i^T \mathbf{B}(p_j, \bar{\mathbf{a}}_j)] \dots\dots (2.50)$$

Cabe resaltar que el jacobiano debe ser ordenado según los eslabones y las coordenadas generalizadas involucradas en cada tipo de junta desarrollada.

### 2.2.3.5.2 Producto punto tipo 2 (DP2)

Se define como el producto punto de dos vectores ( $\bar{\mathbf{a}}_i$ ), definido en el eslabón  $i$ , y el un vector ( $d_{ij}$ ), definido por dos puntos  $P_i$  y  $Q_j$  perteneciente a los eslabones  $i$  y  $j$ . Dicho producto punto puede tener un valor constante o variable en el tiempo, definido mediante una función  $f(t)$  pudiendo representar una junta motriz. Se puede apreciar la junta DP2 en la Figura 2.4

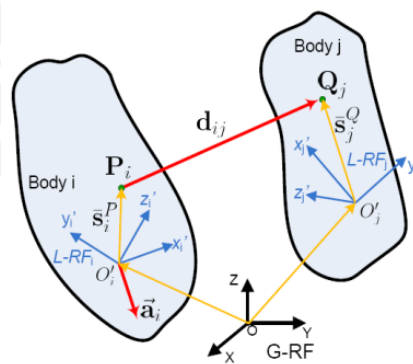


Figura 2.4, Junta tipo DP2 (Producto punto tipo 2) [25]

El vector  $d_{ij}$  se encuentra ubicado en el espacio mediante un punto  $P$  ubicado en el eslabón  $i$  (ubicado mediante el vector  $\bar{s}_i^P$  definido en el eslabón  $i$ ) y el punto  $Q$  ubicado en el eslabón  $j$  (ubicando mediante el vector  $\bar{s}_j^Q$  definido en el eslabón  $j$ ).

Entonces se define la ecuación de movimiento identificando los eslabones relacionados  $i$  y  $j$ , los vectores  $\bar{\mathbf{a}}_i$  (definido en el eslabón  $i$ ),  $\bar{\mathbf{s}}_i^P$  (que define la ubicación del punto  $P$  en el eslabón  $i$ ) y  $\bar{\mathbf{s}}_j^Q$  (que define la ubicación del punto  $Q$  en el eslabón  $j$ ):

$$\phi^{DP2}(i, \bar{\mathbf{a}}_i, \bar{\mathbf{s}}_i^P, j, \bar{\mathbf{s}}_j^Q, f(t)) = \bar{\mathbf{a}}_i^T \mathbf{A}_i^T \mathbf{d}_{ij} - f(t)$$

$$\phi^{DP2}(i, \bar{\mathbf{a}}_i, \bar{\mathbf{s}}_i^P, j, \bar{\mathbf{s}}_j^Q, f(t)) = \bar{\mathbf{a}}_i^T \mathbf{A}_i^T (\mathbf{r}_j + \mathbf{A}_j \bar{\mathbf{s}}_j^Q - \mathbf{r}_i + \mathbf{A}_i \bar{\mathbf{s}}_i^P) - f(t) = 0 \dots (2.50)$$

Obtenida la ecuación de movimiento se procede a obtener la ecuación de velocidad, derivando la ecuación (2.50) con respecto al tiempo:

$$v^{DP2} = \frac{\delta f}{\delta t} \dots (2.51)$$

Posteriormente se procede con el desarrollo de la ecuación de aceleración derivando dos veces la ecuación (2.50) con respecto al tiempo:

$$\begin{aligned} \gamma^{DP2} &= 2\bar{\omega}_i^T \tilde{\omega}_i \mathbf{A}_i^T (\dot{\mathbf{r}}_i - \dot{\mathbf{r}}_j) + 2[\bar{\mathbf{s}}_j^Q]^T \tilde{\omega}_j \mathbf{A}_j^T \mathbf{A}_i \tilde{\omega}_i \bar{\mathbf{a}}_i - [\bar{\mathbf{s}}_i^P]^T \tilde{\omega}_i \tilde{\omega}_i \bar{\mathbf{a}}_i \\ &\quad - [\bar{\mathbf{s}}_i^Q]^T \tilde{\omega}_j \tilde{\omega}_j \mathbf{A}_j^T \mathbf{A}_i \bar{\mathbf{a}}_i - \mathbf{d}_{ij}^T \mathbf{A}_i \tilde{\omega}_i \tilde{\omega}_i \bar{\mathbf{a}}_i + \frac{\delta^2 f}{\delta t^2} \dots (2.52) \end{aligned}$$

En última instancia se procede con la obtención del jacobiano, la cual se obtiene derivando la ecuación de movimiento (2.50) con respecto a las coordenadas generalizadas:

$$\begin{aligned} \Phi_{q_i, q_j}^{DP2}(\mathbf{a}_i, \mathbf{d}_{ij}) &= \mathbf{a}_i^T (\mathbf{d}_{ij})_{q_i, q_j} + \mathbf{d}_{ij}^T (\mathbf{a}_i)_{q_i, q_j} \\ &= \mathbf{a}_i^T [-I_3 \quad -\mathbf{B}(\mathbf{p}_i, \bar{\mathbf{s}}_i^P) \quad I_3 \quad \mathbf{B}(\mathbf{p}_i, \bar{\mathbf{s}}_i^Q)] + \mathbf{d}_{ij}^T [0 \quad \mathbf{B}(\mathbf{p}_i, \bar{\mathbf{a}}_i) \quad 0 \quad 0] \\ &= [-\mathbf{a}_i^T \quad \mathbf{d}_{ij}^T \mathbf{B}(\mathbf{p}_i, \bar{\mathbf{s}}_i^P) - \mathbf{a}_i^T \mathbf{B}(\mathbf{p}_i, \bar{\mathbf{s}}_i^P) \quad \mathbf{a}_i^T \quad \mathbf{a}_i^T \mathbf{B}(\mathbf{p}_j, \bar{\mathbf{s}}_j^Q)] \dots (2.53) \end{aligned}$$

### 2.2.3.5.3 Distancia (D)

Se define que la distancia entre dos puntos  $P_i$  y  $Q_j$ , pertenecientes al eslabón  $i$  y  $j$  respectivamente, tiene un cierto valor. Se debe tomar en cuenta que los puntos  $P_i$  y  $Q_j$  no deben coincidir en el espacio. El valor del módulo del vector  $\vec{d}_{ij}$  ( $\vec{Q}_j - \vec{P}_i$ ) puede ser una constante o puede variar en el tiempo ( $f(t)$ ) pudiendo representar una junta motriz, por ejemplo un cilindro hidráulico, la junta distancia se puede apreciar en la Figura 2.5.

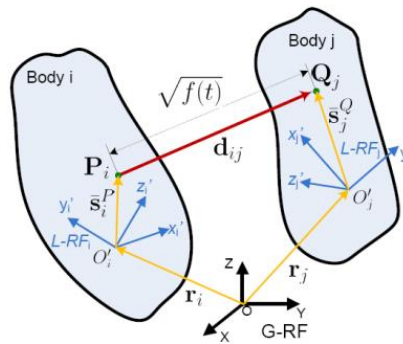


Figura 2.5, Junta tipo D (Distancia) [25]

Los puntos son definidos en el marco de referencia de cada eslabón mediante los vectores  $\vec{s}_i^P$  y  $\vec{s}_j^Q$ , respectivamente.

Con este concepto se define la ecuación de movimiento, para lo cual se requiere tener definidos los puntos  $P$  y  $Q$  ubicados en los eslabones  $i$  y  $j$  mediante los vectores anteriormente mencionados.

$$\begin{aligned} \phi^D(i, \vec{s}_i^P, j, \vec{s}_j^Q, f(t)) &= \mathbf{d}_{ij}^T \mathbf{d}_{ij} - f(t) \\ &= (\mathbf{r}_j + \mathbf{A}_j \vec{s}_j^Q - \mathbf{r}_i + \mathbf{A}_i \vec{s}_i^P)^T (\mathbf{r}_j + \mathbf{A}_j \vec{s}_j^Q - \mathbf{r}_i + \mathbf{A}_i \vec{s}_i^P) - f(t) = 0 \dots (2.54) \end{aligned}$$

Obtenida la ecuación de movimiento se puede proceder a obtener la ecuación de velocidad derivándola con respecto al tiempo:

$$v^D = \frac{\delta f}{\delta t} \dots \dots \dots (2.55)$$

Se procede con el desarrollo de la ecuación de aceleración derivando dos veces la ecuación de movimiento (2.55) con respecto al tiempo

$$\begin{aligned} \gamma^{DP2} = & -2(\dot{\mathbf{r}}_j - \dot{\mathbf{r}}_i)^T (\dot{\mathbf{r}}_j - \dot{\mathbf{r}}_i) + 2[\bar{\mathbf{s}}_j^Q]^T \tilde{\omega}_j \tilde{\omega}_j \bar{\mathbf{s}}_j^Q + 2[\bar{\mathbf{s}}_i^P]^T \tilde{\omega}_i \tilde{\omega}_i \bar{\mathbf{s}}_i^P - 4[\bar{\mathbf{s}}_j^Q]^T \tilde{\omega}_j \mathbf{A}_j^T \mathbf{A}_i \tilde{\omega}_i \bar{\mathbf{s}}_i^P + \\ & 4(\dot{\mathbf{r}}_j - \dot{\mathbf{r}}_i)^T (\mathbf{A}_i \tilde{\omega}_i \bar{\mathbf{s}}_i^P - \mathbf{A}_j \tilde{\omega}_j \bar{\mathbf{s}}_j^Q) - 2\mathbf{d}_{ij} (\mathbf{A}_j \tilde{\omega}_j \tilde{\omega}_j \bar{\mathbf{s}}_j^Q - \mathbf{A}_i \tilde{\omega}_i \tilde{\omega}_i \bar{\mathbf{s}}_i^P) + \frac{\partial^2 f}{\partial t^2} \dots \dots \dots (2.56) \end{aligned}$$

Por último se procede con el análisis del jacobiano de la ecuación de movimiento Distancia (D), derivando la ecuación (2.54) con respecto a las coordenadas generalizadas.

$$\begin{aligned} \Phi_{q_i, q_j}^D &= (\mathbf{d}_{ij}^T \mathbf{d}_{ij})_{q_i, q_j} = 2\mathbf{d}_{ij}^T [\mathbf{d}_{ij}]_{q_i, q_j} \\ &= 2\mathbf{d}_{ij}^T [-\mathbf{I}_3 \quad -\mathbf{B}(\mathbf{p}_i, \bar{\mathbf{s}}_i^P) \quad \mathbf{I}_3 \quad \mathbf{B}(\mathbf{p}_j, \bar{\mathbf{s}}_j^Q)] \\ &= [-2\mathbf{d}_{ij}^T \quad 2\mathbf{d}_{ij}^T \mathbf{B}(\mathbf{p}_i, \bar{\mathbf{s}}_i^P) \quad \mathbf{d}_{ij}^T \quad 2\mathbf{d}_{ij}^T \mathbf{B}(\mathbf{p}_j, \bar{\mathbf{s}}_j^Q)] \dots \dots \dots (2.57) \end{aligned}$$

**2.2.3.5.4 Diferencia de coordenadas (CD)**

Define que la distancia con respecto a uno de los ejes del marco de referencia general entre dos puntos  $P_i$  y  $Q_j$ , pertenecientes a los eslabones  $i$  y  $j$ , tiene un cierto valor. El valor puede ser una constante o variable con respecto al tiempo, definida como una función  $(f(t))$ . Por ejemplo se puede representar la trayectoria deseada del último eslabón de una cadena cinemática abierta en el espacio, en la Figura 2.6 se puede apreciar dicha relación.

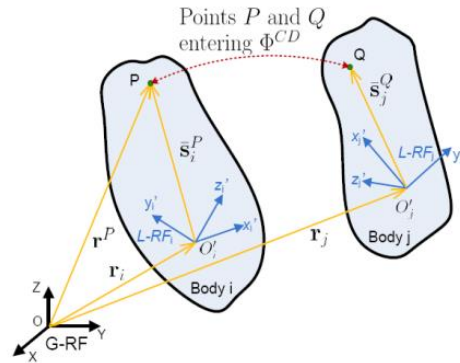


Figura 2.6, Junta tipo CD (Diferencia de Coordenadas) [25]

Este tipo de junta es de gran utilidad, ya que permite describir la relación de los eslabones en el marco de referencia general, permitiendo el desarrollo de la junta esférica y descripción de la ubicación de la junta de revolución, entre otros.

Analizando a mayor detalle este tipo de junta se requiere de la definición de dos puntos  $P$  y  $Q$  ubicados en los eslabones  $i$  y  $j$  respectivamente mediante los vectores  $\bar{s}_i^P$  y  $\bar{s}_j^Q$ . La relación se define mediante la proyección del vector formado por los puntos  $P$  y  $Q$  (vector  $d_{ij}$ ) con respecto a una dirección definida anteriormente y contenida en el vector  $c$  (usualmente contiene los vectores que definen las coordenadas cartesianas). Esta relación puede ser variable en el tiempo o constante, por ejemplo si es constante y el valor es cero indica coincidencia entre ambos puntos con respecto a una coordenada. Se muestra a continuación la ecuación de movimiento.

$$\phi^{CD}(c, i, \bar{s}_i^P, j, \bar{s}_j^Q, f(t)) = c^T d_{ij} - f(t) = c^T (r_j + A_j \bar{s}_j^Q - r_i + A_i \bar{s}_i^P) - f(t) = 0 \dots (2.58)$$

Posteriormente se procede a derivar la ecuación (2.58) con respecto al tiempo para tener la ecuación de velocidad.

$$v^{DP1} = \frac{\delta f}{\delta t} \dots (2.59)$$

Se deriva dos veces con respecto al tiempo a la ecuación (2.58) para obtener la ecuación de aceleración.

$$\gamma^{DP2} = \mathbf{c}^T (\mathbf{A}_i \tilde{\omega}_i \tilde{\omega}_i \bar{\mathbf{s}}_i^P - \mathbf{A}_j \tilde{\omega}_j \tilde{\omega}_j \bar{\mathbf{s}}_j^Q) \dots \dots \dots (2.60)$$

Por último se debe analizar el jacobiano originado por este tipo de junta, por lo que se deriva con respecto a las coordenadas generalizadas a la ecuación (2.58).

$$\begin{aligned} \Phi_{q_i, q_j}^{CD} &= (\mathbf{c}^T \mathbf{d}_{ij})_{q_i, q_j} = \mathbf{c}^T [\mathbf{d}_{ij}]_{q_i, q_j} \\ &= \mathbf{c}^T [-\mathbf{I}_3 \quad -\mathbf{B}(\mathbf{p}_i, \bar{\mathbf{s}}_i^P) \quad \mathbf{I}_3 \quad \mathbf{B}(\mathbf{p}_j, \bar{\mathbf{s}}_j^Q)] \\ &= [-\mathbf{c}^T \quad \mathbf{c}^T \mathbf{B}(\mathbf{p}_i, \bar{\mathbf{s}}_i^P) \quad \mathbf{c}^T \quad \mathbf{c}^T \mathbf{B}(\mathbf{p}_j, \bar{\mathbf{s}}_j^Q)] \dots \dots \dots (2.61) \end{aligned}$$

### 2.2.3.5.5 Relación de Perpendicularidad a un plano ( $\perp 1$ )

Define que un vector  $c_i$ , definido en el cuerpo  $i$ , es perpendicular a un plano en el cuerpo  $j$ , este plano se define con los vectores  $a_j$  y  $b_j$ . Esta relación usa las funciones de la junta tipo producto punto (DP1) pero define que el valor es cero, ya que se requiere perpendicularidad. En la figura 2.7 se puede apreciar la junta  $\perp 1$ .

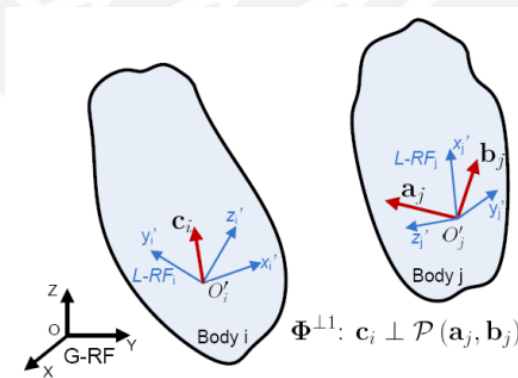


Figura 2.7, Junta tipo  $\perp 1$  (Relación de Perpendicularidad a un plano tipo 1) [25]

Este tipo de junta se emplea principalmente para definir la junta cilíndrica y de revolución las cuales son usadas usualmente en mecanismos.

Para poder definir este tipo de junta se requiere usar dos juntas básicas producto punto tipo 1 (DP1), como se muestra en la tabla 2.1. Detallando más la naturaleza de este tipo de junta se requiere definir la perpendicular del vector  $c_i$  con los dos vectores  $a_j$  y  $b_j$  mediante la junta básica DP1. Por lo que la ecuación de movimiento se define por una matriz compuesta por dos ecuaciones de movimiento DP1 como se muestra a continuación:

$$\phi^{+1}(i, \bar{c}_i, j, \bar{a}_j, \bar{b}_j) = \begin{bmatrix} \phi^{DP1}(i, \bar{c}_i, j, \bar{a}_j, 0) \\ \phi^{DP1}(i, \bar{c}_i, j, \bar{b}_j, 0) \end{bmatrix} = \begin{bmatrix} \bar{c}_i^T A_i^T A_j \bar{a}_j \\ \bar{c}_i^T A_i^T A_j \bar{b}_j \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \dots\dots (2.62)$$

A su vez la matriz que de la ecuación de la velocidad está compuesta por las ecuaciones de velocidad de las dos juntas básicas DP1, tomes en cuenta que el valor del producto punto entre los vectores es constante con respecto al tiempo:

$$v^{+1}(i, \bar{c}_i, j, \bar{a}_j, \bar{b}_j) = \begin{bmatrix} v^{DP1}(i, \bar{c}_i, j, \bar{a}_j, 0) \\ v^{DP1}(i, \bar{c}_i, j, \bar{b}_j, 0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \dots\dots (2.63)$$

Posterior al cálculo de la velocidad se procede a calcular la matriz que corresponde al análisis de las ecuaciones de aceleración que la componen:

$$\gamma^{+1}(i, \bar{c}_i, j, \bar{a}_j, \bar{b}_j) = \begin{bmatrix} \gamma^{DP1}(i, \bar{c}_i, j, \bar{a}_j, 0) \\ \gamma^{DP1}(i, \bar{c}_i, j, \bar{b}_j, 0) \end{bmatrix} \dots\dots (2.63)$$

Por último, se desarrolla la matriz que representa el jacobiano, la cual está compuesta por los jacobianos de las juntas simples DP1 que componen la junta o relación de perpendicularidad tipo 1:

$$\Phi_{q_i, q_j}^{+1} = \begin{bmatrix} \Phi_{q_i, q_j}^{DP1}(i, \bar{c}_i, j, \bar{a}_j, 0) \\ \Phi_{q_i, q_j}^{DP1}(i, \bar{c}_i, j, \bar{b}_j, 0) \end{bmatrix} \dots\dots (2.64)$$

### 2.2.3.5.6 Relación de Perpendicularidad a un plano ( $\perp 2$ )

Define que un vector ( $d_{ij}$ ) definido por dos puntos,  $P_i$  y  $Q_j$ , ubicados en el cuerpo  $i$  y  $j$  es perpendicular a un plano en el cuerpo  $i$ , este plano se define con los vectores  $a_i$  y  $b_i$ . Esta relación usa las funciones de la junta tipo producto punto (DP2) entre los vectores  $d_{ij}$  y los vectores  $a_i$  y  $b_i$  definiendo que el valor del producto punto es cero en todo instante, ya que se requiere perpendicularidad. En la figura 2.8 se puede apreciar la junta  $\perp 2$ .

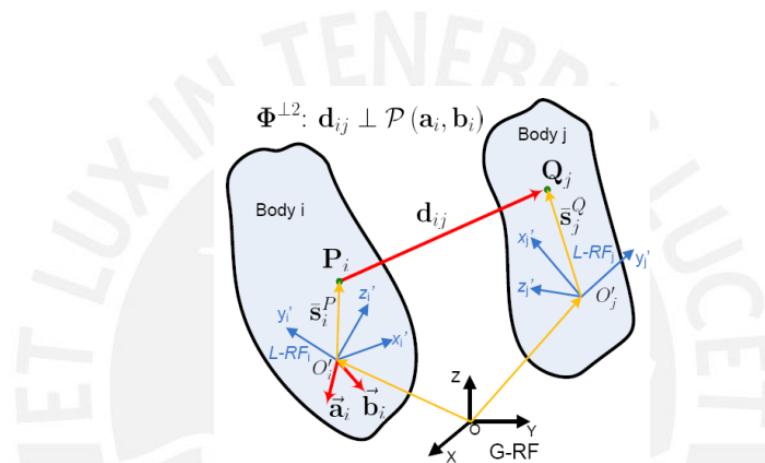


Figura 2.8, Junta tipo  $\perp 2$  (Relación de Perpendicularidad a un plano tipo 2) [25]

Este tipo de junta es muy útil, ya que define un eje de translación entre dos eslabones de manera fácil y sencilla, es utilizado principalmente en las juntas de traslación y cilíndricas.

Al realizar un mayor análisis de este tipo de junta se puede concluir que requiere la definición de dos puntos fijos en los dos eslabones mediante los vectores  $\vec{s}_i^P$  y  $\vec{s}_j^Q$ . Y para definir el plano se emplea la misma lógica de la Relación de Perpendicularidad a un Plano ( $\perp 1$ ), usando dos vectores ( $a_i$  y  $b_i$ ) contenidos en el plano y generando una perpendicular entre ellos y el vector eje ( $d_{ij}$ ).

Con el concepto de las interrelaciones requeridas para poder definir este tipo de junta, se procede a desarrollar la ecuación de movimiento:



$$\phi^{+2}(i, \bar{\mathbf{a}}_i, \bar{\mathbf{b}}_i, \bar{\mathbf{s}}_i^P, j, \bar{\mathbf{s}}_j^Q) = \begin{bmatrix} \phi^{DP2}(i, \bar{\mathbf{a}}_i, \bar{\mathbf{s}}_i^P, j, \bar{\mathbf{s}}_j^Q, 0) \\ \phi^{DP2}(i, \bar{\mathbf{b}}_i, \bar{\mathbf{s}}_i^P, j, \bar{\mathbf{s}}_j^Q, 0) \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{a}}_i^T \mathbf{A}_i^T \mathbf{d}_{ij} \\ \bar{\mathbf{b}}_i^T \mathbf{A}_i^T \mathbf{d}_{ij} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \dots\dots (2.65)$$

Al igual que en la junta Relación de Perpendicularidad a un Plano ( $\perp 1$ ), la ecuación de velocidad está compuesta por las ecuaciones de velocidad de dos juntas tipo DP2 teniendo la función igual a cero (0) a lo largo del tiempo:

$$v^{-2}(i, \bar{\mathbf{a}}_i, \bar{\mathbf{b}}_i, \bar{\mathbf{s}}_i^P, j, \bar{\mathbf{s}}_j^Q) = \begin{bmatrix} v^{DP2}(i, \bar{\mathbf{a}}_i, \bar{\mathbf{s}}_i^P, j, \bar{\mathbf{s}}_j^Q, 0) \\ v^{DP2}(i, \bar{\mathbf{b}}_i, \bar{\mathbf{s}}_i^P, j, \bar{\mathbf{s}}_j^Q, 0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \dots\dots (2.67)$$

Posterior a la obtención de la ecuación de velocidad se procede a desarrollar la ecuación de aceleración de la junta, compuesta por dos ecuaciones de aceleración:

$$\gamma^{+2}(i, \bar{\mathbf{a}}_i, \bar{\mathbf{b}}_i, \bar{\mathbf{s}}_i^P, j, \bar{\mathbf{s}}_j^Q) = \begin{bmatrix} \gamma^{DP2}(i, \bar{\mathbf{a}}_i, \bar{\mathbf{s}}_i^P, j, \bar{\mathbf{s}}_j^Q, 0) \\ \gamma^{DP2}(i, \bar{\mathbf{b}}_i, \bar{\mathbf{s}}_i^P, j, \bar{\mathbf{s}}_j^Q, 0) \end{bmatrix} \dots\dots (2.68)$$

Por último se desarrolla la matriz que contiene al jacobiano:

$$\Phi_{q_i, q_j}^{+2} = \begin{bmatrix} \Phi_{q_i, q_j}^{DP1}(i, \bar{\mathbf{a}}_i, \bar{\mathbf{s}}_i^P, j, \bar{\mathbf{s}}_j^Q, 0) \\ \Phi_{q_i, q_j}^{DP1}(i, \bar{\mathbf{b}}_i, \bar{\mathbf{s}}_i^P, j, \bar{\mathbf{s}}_j^Q, 0) \end{bmatrix} \dots\dots (2.69)$$

### 2.2.3.5.7 Junta Esférica (SJ)

Este tipo de junta define la coincidencia de dos puntos en el marco de referencia general, por lo que se debe definir los eslabones relacionados y el punto de coincidencia de ambos. Este tipo de junta elimina solamente tres grados de libertad [25]. Los dos puntos  $P_i$  y  $Q_j$  de los eslabones diferentes  $i$  y  $j$ , coinciden en el espacio (marco de referencia general) en todo el tiempo, este tipo de junta se representa en la figura 2.9.

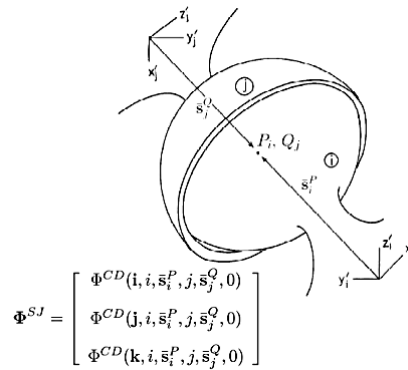


Figura 2.9, Junta tipo SJ (Junta Esférica) [25]

Para poder definir esta junta se requiere definir la ubicación de los puntos  $P_i$  y  $Q_j$  en sus respectivos eslabones mediante dos vectores  $\bar{s}_i^P$  y  $\bar{s}_j^Q$ . La coincidencia se puede representar mediante las juntas CD, definiendo que la distancia entre coordenadas es igual cero. Dicho lo anterior se puede proceder a desarrollar la ecuación de movimiento de la junta esférica:

$$\phi^{SJ}(i, \bar{s}_i^P, j, \bar{s}_j^Q) = \begin{bmatrix} \phi^{CD}(i, i, \bar{s}_i^P, j, \bar{s}_j^Q, 0) \\ \phi^{CD}(j, i, \bar{s}_i^P, j, \bar{s}_j^Q, 0) \\ \phi^{CD}(k, i, \bar{s}_i^P, j, \bar{s}_j^Q, 0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \dots\dots\dots (2.70)$$

Obtenida la ecuación de velocidad se procede a calcular la matriz de velocidad (compuesta por tres ecuaciones de velocidad de la junta CD):

$$v^{SJ}(i, \bar{s}_i^P, j, \bar{s}_j^Q) = \begin{bmatrix} v^{CD}(i, i, \bar{s}_i^P, j, \bar{s}_j^Q, 0) \\ v^{CD}(j, i, \bar{s}_i^P, j, \bar{s}_j^Q, 0) \\ v^{CD}(k, i, \bar{s}_i^P, j, \bar{s}_j^Q, 0) \end{bmatrix} \dots\dots\dots (2.71)$$

Posteriormente se obtiene la ecuación de aceleración de la junta:

$$\gamma^{SJ}(i, \bar{s}_i^P, j, \bar{s}_j^Q) = \begin{bmatrix} \gamma^{CD}(i, i, \bar{s}_i^P, j, \bar{s}_j^Q, 0) \\ \gamma^{CD}(j, i, \bar{s}_i^P, j, \bar{s}_j^Q, 0) \\ \gamma^{CD}(k, i, \bar{s}_i^P, j, \bar{s}_j^Q, 0) \end{bmatrix} \dots\dots\dots (2.72)$$

Por último se desarrolla el jacobiano de la junta:

$$\Phi_{q_i, q_j}^{SJ} = \begin{bmatrix} \Phi_{q_i, q_j}^{SJ}(i, i, \bar{s}_i^P, j, \bar{s}_j^Q, 0) \\ \Phi_{q_i, q_j}^{SJ}(j, i, \bar{s}_i^P, j, \bar{s}_j^Q, 0) \\ \Phi_{q_i, q_j}^{SJ}(k, i, \bar{s}_i^P, j, \bar{s}_j^Q, 0) \end{bmatrix} \dots\dots\dots (2.73)$$

**2.2.3.5.8 Junta Cilíndrica (CJ)**

Este tipo de junta define que un eslabón puede trasladarse y rotar con respecto a otro alrededor de un eje. Este tipo de junta solo elimina cuatro grados de libertad.

Entonces se define en el cuerpo  $j$  el eje de revolución  $c_j$ , el cual debe ser perpendicular a un plano definido en el cuerpo  $i$ , definido por los vectores  $a_i$  y  $b_i$ . Por último, se define un vector que representa al eje de traslación  $d_{ij}$ , definido por los puntos  $P_i$  y  $Q_j$ . Todos estos elementos se pueden observar en la figura 2.10.

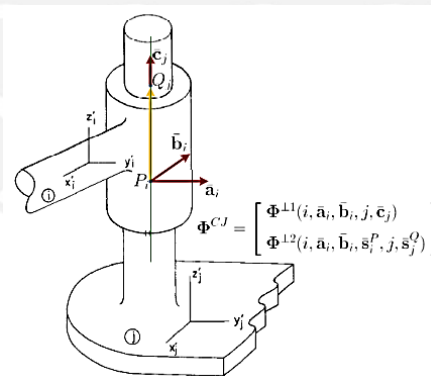


Figura 2.10, Junta tipo CJ (Junta Cilíndrica) [25]

Analizando a mayor detalle este tipo de junta, se puede desarrollar mediante el uso de dos juntas desarrolladas anteriormente, Relación de Perpendicularidad a un Plano ( $\perp 1$ ), con el cual se definiría el vector  $\bar{c}_j$ , y Relación de Perpendicularidad a un Plano ( $\perp 2$ ), el cual definiría el eje de traslación. Esto quiere decir que tanto el eje de traslación y el vector  $\bar{c}_j$  serian perpendiculares a un plano definido en el eslabón  $i$  en todo momento, logrando que tanto el eje como el vector  $\bar{c}_j$  sean colineales en todo

momento. Para poder definir estas relaciones o juntas se requiere definir dos vectores  $\mathbf{a}_i$  y  $\mathbf{b}_i$  los cuales definirían el plano en el eslabón  $i$ , un vector  $\bar{\mathbf{c}}_j$  definido en el eslabón  $j$  y por último los puntos  $P_i$  y  $Q_j$  definidos en sus respectivos eslabones mediante los vectores  $\bar{\mathbf{s}}_i^P$  y  $\bar{\mathbf{s}}_j^Q$ . Definido el modo de procesar este tipo de junta se procede a desarrollar su ecuación de movimiento:

$$\phi^{CJ}(i, \bar{\mathbf{s}}_i^P, \bar{\mathbf{a}}_i, \bar{\mathbf{b}}_i, j, \bar{\mathbf{s}}_j^Q, \bar{\mathbf{c}}_j) = \begin{bmatrix} \phi^{+1}(i, \bar{\mathbf{a}}_i, \bar{\mathbf{b}}_i, j, \bar{\mathbf{c}}_j) \\ \phi^{+2}(i, \bar{\mathbf{a}}_i, \bar{\mathbf{b}}_i, \bar{\mathbf{s}}_i^P, j, \bar{\mathbf{s}}_j^Q) \end{bmatrix} = \begin{bmatrix} \mathbf{0}_2 \\ \mathbf{0}_2 \end{bmatrix} \dots\dots\dots (2.74)$$

A continuación se desarrolla la ecuación de velocidad de la junta:

$$v^{CJ}(i, \bar{\mathbf{s}}_i^P, \bar{\mathbf{a}}_i, \bar{\mathbf{b}}_i, j, \bar{\mathbf{s}}_j^Q, \bar{\mathbf{c}}_j) = \begin{bmatrix} v^{+1}(i, \bar{\mathbf{a}}_i, \bar{\mathbf{b}}_i, j, \bar{\mathbf{c}}_j) \\ v^{+2}(i, \bar{\mathbf{a}}_i, \bar{\mathbf{b}}_i, \bar{\mathbf{s}}_i^P, j, \bar{\mathbf{s}}_j^Q) \end{bmatrix} \dots\dots\dots (2.75)$$

Concluida la ecuación de velocidad se desarrolla la ecuación de aceleración de la junta:

$$\gamma^{CJ}(i, \bar{\mathbf{s}}_i^P, \bar{\mathbf{a}}_i, \bar{\mathbf{b}}_i, j, \bar{\mathbf{s}}_j^Q, \bar{\mathbf{c}}_j) = \begin{bmatrix} \gamma^{+1}(i, \bar{\mathbf{a}}_i, \bar{\mathbf{b}}_i, j, \bar{\mathbf{c}}_j) \\ \gamma^{+2}(i, \bar{\mathbf{a}}_i, \bar{\mathbf{b}}_i, \bar{\mathbf{s}}_i^P, j, \bar{\mathbf{s}}_j^Q) \end{bmatrix} \dots\dots\dots (2.76)$$

Por último se desarrolla el jacobiano:

$$\Phi_{q_i, q_j}^{CJ} = \begin{bmatrix} \Phi_{q_i, q_j}^{+1}(i, \bar{\mathbf{a}}_i, \bar{\mathbf{b}}_i, j, \bar{\mathbf{c}}_j) \\ \Phi_{q_i, q_j}^{+2}(i, \bar{\mathbf{a}}_i, \bar{\mathbf{b}}_i, \bar{\mathbf{s}}_i^P, j, \bar{\mathbf{s}}_j^Q) \end{bmatrix} \dots\dots\dots (2.77)$$

### 2.2.3.5.9 Junta de Revolución (RJ)

Este tipo de junta define que un eslabón puede girar con respecto a otro alrededor de un eje de revolución. Este tipo de junta solo elimina cinco grados de libertad.

Entonces para desarrollar este tipo de junta se requiere definir el eje de revolución, usando un vector en el eslabón  $j$  ( $c_j$ ), el cual debe ser perpendicular a un plano definido en el eslabón  $i$ , usando los vectores  $a_i$  y  $b_i$ . Además, se requiere definir un punto de coincidencia entre ambos eslabones, definido por los puntos Q y P. Todo esto se puede observar en la figura 2.11.

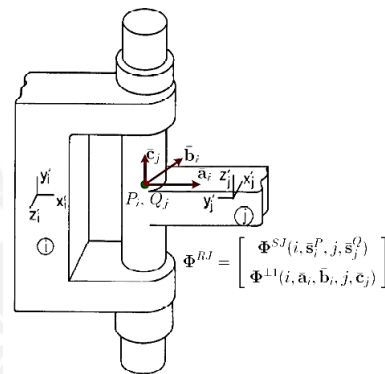


Figura 2.11, Junta tipo RJ (Junta de Revolución) [25]

Entrando a más detalle de esta junta, se requiere definir el punto de coincidencia entre los dos eslabones mediante una junta esférica (SJ), para lo cual se requiere definir la ubicación de los puntos  $P_i$  y  $Q_j$  mediante los vectores  $\bar{s}_i^P$  y  $\bar{s}_j^Q$  y el eje de rotación se puede definir mediante la junta Relación de Perpendicularidad a un Plano ( $\perp 1$ ) con el cual se define el eje de rotación  $\bar{c}_j$  en uno de los eslabones y un plano definido con los vectores  $a_i$  y  $b_i$  en otro eslabón. Dicho esto se procede a desarrollar la ecuación de movimiento:

$$\phi^{RJ}(i, \bar{s}_i^P, \bar{a}_i, \bar{b}_i, j, \bar{s}_j^Q, \bar{c}_j) = \begin{bmatrix} \phi^{SJ}(i, \bar{s}_i^P, j, \bar{s}_j^Q) \\ \phi^{\perp 1}(i, \bar{a}_i, \bar{b}_i, j, \bar{c}_j) \end{bmatrix} = \begin{bmatrix} \mathbf{0}_3 \\ \mathbf{0}_2 \end{bmatrix} \dots \dots \dots (2.78)$$

Desarrollada la ecuación de movimiento se procede a desarrollar la ecuación de velocidad:

$$v^{RJ}(i, \bar{s}_i^P, \bar{a}_i, \bar{b}_i, j, \bar{s}_j^Q, \bar{c}_j) = \begin{bmatrix} v^{SJ}(i, \bar{s}_i^P, j, \bar{s}_j^Q) \\ v^{\perp 1}(i, \bar{a}_i, \bar{b}_i, j, \bar{c}_j) \end{bmatrix} \dots \dots \dots (2.79)$$

Consecutivamente se desarrolla la ecuación de aceleración de la junta:

$$\gamma^{CJ}(i, \bar{s}_i^P, \bar{a}_i, \bar{b}_i, j, \bar{s}_j^Q, \bar{c}_j) = \left[ \begin{matrix} \gamma^{SJ}(i, \bar{s}_i^P, j, \bar{s}_j^Q) \\ \gamma^{+1}(i, \bar{a}_i, \bar{b}_i, j, \bar{c}_j) \end{matrix} \right] \dots\dots\dots (2.80)$$

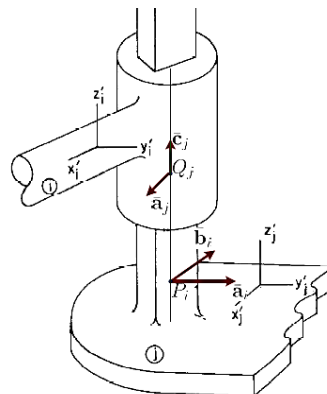
Por último se desarrolla el jacobiano:

$$\Phi_{q_i, q_j}^{CJ} = \left[ \begin{matrix} \Phi_{q_i, q_j}^{SJ}(i, \bar{s}_i^P, j, \bar{s}_j^Q) \\ \Phi_{q_i, q_j}^{+1}(i, \bar{a}_i, \bar{b}_i, j, \bar{c}_j) \end{matrix} \right] \dots\dots\dots (2.81)$$

**2.2.3.5.10 Junta de Traslación (TJ)**

Esta junta define que un eslabón se traslada a través de un eje con respecto a otro eslabón. Este tipo de junta elimina 5 grados de libertad.

Entonces se requiere definir el eje de traslación usando dos puntos, P y Q, y usando un vector  $c_j$  en eslabón  $j$ , ambos deben ser perpendiculares a un plano en el eslabón  $i$ . Además, se requiere definir un vector en el cuerpo  $j$  ( $a_j$ ) que sea perpendicular o su producto punto sea constante con respecto a otro en el cuerpo  $i$  ( $a_i$ ). Estos elementos se pueden observar en la figura 2.12.



$$\Phi^{TJ} = \left[ \begin{matrix} \Phi^{CJ}(i, \bar{s}_i^P, \bar{a}_i, \bar{b}_i, j, \bar{s}_j^Q, \bar{c}_j) \\ \Phi^{DPI}(i, \bar{a}_i, j, \bar{a}_j, \text{const.}) \end{matrix} \right]$$

Figura 2.12, Junta tipo TJ (Junta de Traslación) [25]

La junta de translación se puede definir como una junta cilíndrica pero que no puede rotar, simplemente permite translación entre dos puntos  $P_i$  y  $Q_j$  ubicados en los cada eslabón.

Para poder definir este tipo de junta, se procede a desarrollar la junta cilíndrica y adicionar un vector en el eslabón  $j$  ( $\mathbf{a}_j$ ) el cual tiene un cierto ángulo con respecto al vector  $\mathbf{a}_i$ , entonces para definir esta última relación se puede emplear la junta tipo DP1 (Para mayor facilidad dicho ángulo se puede definir con  $90^\circ$ , dando un valor de la función igual cero). Definido la anterior se puede desarrollar la ecuación de movimiento:

$$\phi^{TJ}(i, \bar{s}_i^P, \bar{\mathbf{a}}_i, \bar{\mathbf{b}}_i, j, \bar{s}_j^Q, \bar{\mathbf{a}}_j, \bar{\mathbf{c}}_j) = \begin{bmatrix} \phi^{CJ}(i, \bar{s}_i^P, \bar{\mathbf{a}}_i, \bar{\mathbf{b}}_i, j, \bar{s}_j^Q, \bar{\mathbf{c}}_j) \\ \phi^{DP1}(i, \bar{\mathbf{a}}_i, j, \bar{\mathbf{a}}_j, 0) \end{bmatrix} = \begin{bmatrix} \mathbf{0}_4 \\ 0 \end{bmatrix} \dots \dots \dots (2.82)$$

Obtenida la ecuación de movimiento se procede a desarrollar la ecuación de velocidad:

$$v^{TJ}(i, \bar{s}_i^P, \bar{\mathbf{a}}_i, \bar{\mathbf{b}}_i, j, \bar{s}_j^Q, \bar{\mathbf{a}}_j, \bar{\mathbf{c}}_j) = \begin{bmatrix} v^{CJ}(i, \bar{s}_i^P, \bar{\mathbf{a}}_i, \bar{\mathbf{b}}_i, j, \bar{s}_j^Q, \bar{\mathbf{c}}_j) \\ v^{DP1}(i, \bar{\mathbf{a}}_i, j, \bar{\mathbf{a}}_j, 0) \end{bmatrix} \dots \dots \dots (2.83)$$

Posteriormente la ecuación de aceleración:

$$\gamma^{TJ}(i, \bar{s}_i^P, \bar{\mathbf{a}}_i, \bar{\mathbf{b}}_i, j, \bar{s}_j^Q, \bar{\mathbf{a}}_j, \bar{\mathbf{c}}_j) = \begin{bmatrix} \gamma^{CJ}(i, \bar{s}_i^P, \bar{\mathbf{a}}_i, \bar{\mathbf{b}}_i, j, \bar{s}_j^Q, \bar{\mathbf{c}}_j) \\ \gamma^{DP1}(i, \bar{\mathbf{a}}_i, j, \bar{\mathbf{a}}_j, 0) \end{bmatrix} = \begin{bmatrix} \mathbf{0}_4 \\ 0 \end{bmatrix} \dots \dots \dots (2.84)$$

Por último se desarrolla el jacobiano:

$$\Phi_{q_i, q_j}^{TJ} = \begin{bmatrix} \Phi_{q_i, q_j}^{CJ}(i, \bar{s}_i^P, \bar{\mathbf{a}}_i, \bar{\mathbf{b}}_i, j, \bar{s}_j^Q, \bar{\mathbf{c}}_j) \\ \Phi_{q_i, q_j}^{DP1}(i, \bar{\mathbf{a}}_i, j, \bar{\mathbf{a}}_j, 0) \end{bmatrix} \dots \dots \dots (2.85)$$

### 2.2.3.6 Juntas Motrices

Las juntas motrices definen el movimiento característico del mecanismo o movimiento deseado. Para lo cual se puede emplear las juntas previamente definidas para definir dicho movimiento, por ejemplo se tiene un mecanismo el cual tiene un grado de libertad y es impulsado por un motor, el motor genera una rotación específica en uno de los eslabones el cual puede ser representado por una junta tipo DP1 que varía el valor de la función con respecto al tiempo el cual contiene a la velocidad del motor y su ángulo inicial.

Por lo tanto las junta motrices pueden ser definidas por una o por varias juntas base (DP1, DP2, CD o D) en las cuales la función con respecto al tiempo define el movimiento deseado a ser analizado. Entonces se puede obtener tanto la ecuación de movimiento, velocidad, aceleración y Jacobiano de ellos usando juntas base mostradas en la tabla 2.1.

### 2.2.4 Salidas

Al efectuar el algoritmo la salida que se puede obtener son los valores de las aceleración, velocidad y posición de los eslabones y juntas, las cuales pueden ser graficadas en función del tiempo o de alguna variable, por ejemplo una coordenada generalizada como ángulo de giro de algún eslabón.

Teniendo estos valores se puede proceder a un análisis cinemático del mecanismo para obtener la velocidad y aceleraciones requeridas para el movimiento deseado de un eslabón característico. A continuación se muestra algunos gráficos obtenidos del uso un algoritmo para el análisis cinemático de mecanismos:



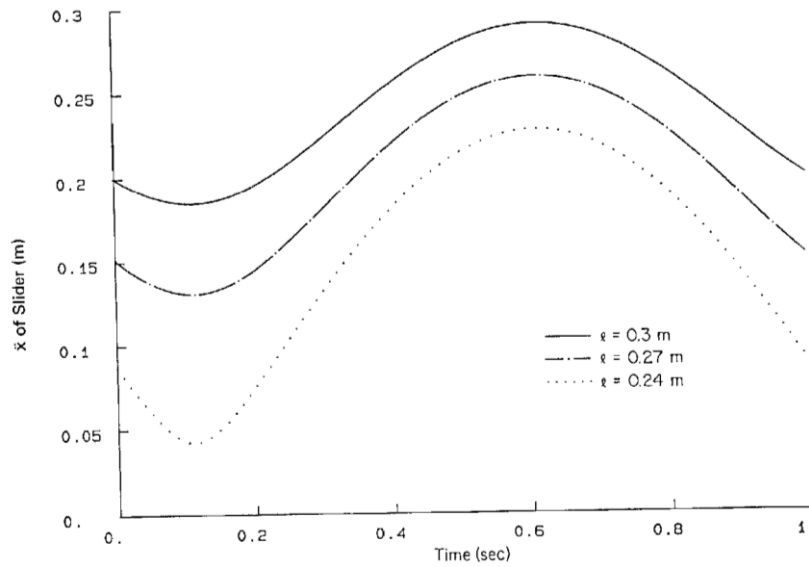


Figura 2.13, Diagrama de la posición del Pistón vs tiempo de un mecanismo biela manivela espacial, para un giro constante de  $\omega_1$  de  $2\pi$  manivela con diversas longitudes de la biela  $l_2$  [24].

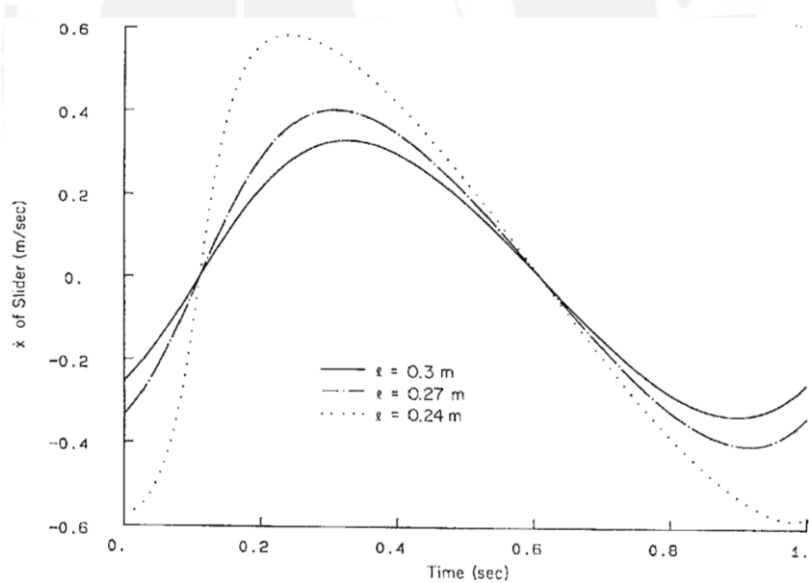


Figura 2.14, Diagrama de la velocidad del Pistón vs tiempo de un mecanismo biela manivela espacial, para un giro constante de  $\omega_1$  de  $2\pi$  manivela con diversas longitudes de la biela  $l_2$  [24].

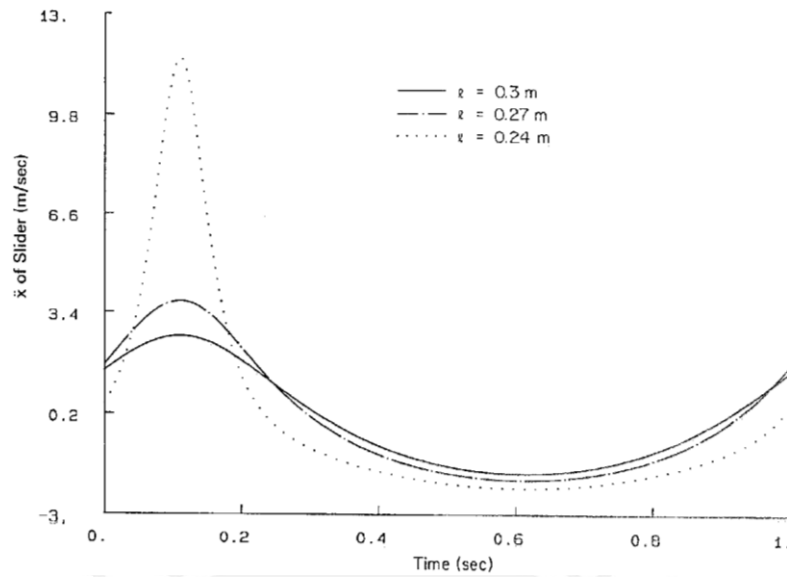


Figura 2.15, Diagrama de la aceleración del Pistón vs tiempo de un mecanismo biela manivela espacial, para un giro constante de  $\omega_1$  de  $2\pi$  manivela con diversas longitudes de la biela  $l_2$  [24].

Las figuras 2.13, 2.14, y 2.15 son graficas que pertenecen a un mismo mecanismo biela manivela espacial, las gráficas se presentan para ver la variación de la posición, velocidad y aceleración del pistón con respecto al tiempo, para una velocidad angular de  $2\pi$  de la manivela. Los gráficos muestran tres escenarios respecto a la longitud de la biela, siendo la longitud 0.24m una posición crítica y de posible singularidad. En la figura 2.16 se muestra el mecanismo biela-manivela espacial.

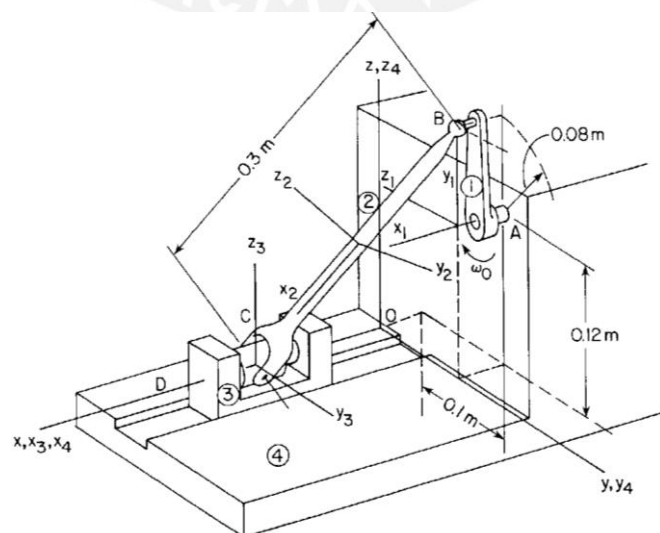


Figura 2.16, Mecanismo biela-manivela espacial [24].

## 2.3 Eficiencia y Precisión del Algoritmo

Existen diversas características típicas en el diseño o desarrollo de algoritmos entre las cuales destaca la eficiencia y la precisión. La precisión del algoritmo se refiere a los errores generados dentro del algoritmo los cuales pueden originarse debido a aproximaciones de valores, errores de modelación y valores pequeños o grandes originados en operaciones intermedias (*round-off error*). La eficiencia del algoritmo tiene el criterio de la cantidad de operaciones realizadas dentro del algoritmo para obtener la respuesta.

### 2.3.1 Eficiencia del Algoritmo

La eficiencia usa el criterio de la cantidad de operaciones intermedias que realiza, tales como sustracción, adición, multiplicación y división de valores. Normalmente se divide estos valores en dos grupos:

- Multiplicación (**m**), donde esta las operación de multiplicación y división
- Adición (**a**), están las operaciones de sustracción y adición.

En la figura 2.17 se muestra graficas sobre el costos o la cantidad de operaciones realizadas para diversos tipos de algoritmos ABA (articulated-body algorithm), F&S (Factorize and solve equation of motion), RNEA (recursive Newton-Euler) y CRBA (composite-rigid-body algorithm), usando los algoritmos más eficaces que hay en este tipo de cálculo, versus la cantidad de cuerpos rígidos que posee el algoritmo. Se debe tomar en cuenta que tanto la multiplicación y adición poseen el mismo peso en la figura 2.17

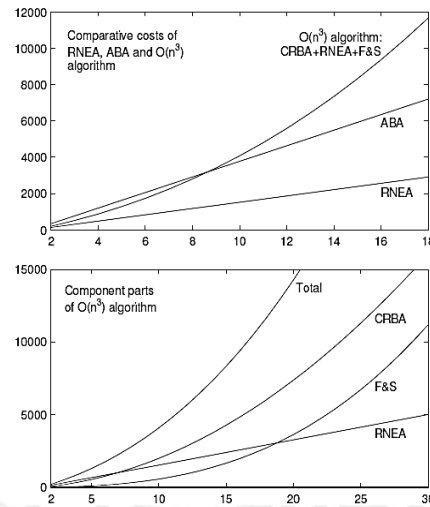


Figura 2.17, Diagrama de costo versus número de cuerpos rígidos en un mecanismo usando diversos algoritmos [3].

### 2.3.2 Precisión del Algoritmo

En el desarrollo de una solución en un algoritmo se puede tener ciertas fuentes de error, tales como [3]:

- Truncamiento o aproximación, este tipo de error se refiere principalmente a la variación de la posición, aceleración y velocidad a lo largo del tiempo, si el intervalo de tiempo para diferentes posiciones es muy pequeño la precisión aumenta pero la eficiencia tiende a disminuir debido a la cantidad de operaciones a realizar.
- Errores de modelación, este tipo de error hace que el modelo matemático empleado difiera de la realidad o del mecanismo en físico. Estos errores se deben a la consideración de que son cuerpos rígidos, juntas ideales, modelación incorrecta y a una falta de precisión en la modelación y sus parámetros.
- Operaciones Round-off, son muy pocas pero surgen al emplear grandes aceleraciones, velocidades, distancias de las coordenadas generalizadas, entre otros, cuando se sustraer generar valores muy pequeños que pueden originar ciertos errores.

## CAPÍTULO 3

### IMPLEMENTACIÓN BÁSICA Y EJEMPLOS DE APLICACIÓN

En este Capítulo se procede a realizar la implementación básica del algoritmo, a su vez del detalle de cómo se emplea y se ingresa la información requerida para su uso. Por último, se presenta dos ejemplos conjuntamente con sus resultados.

Se debe tomar en cuenta que se representara un ejemplo de mecanismo plano, para poder explicar y demostrar la complejidad de su implementación dentro de un algoritmo de análisis cinemático en 3D, y una cadena cinemática abierta, mostrando el uso del algoritmo para brazos robóticos.

#### 3.1. Implementación Básica del Algoritmo

El algoritmo de Análisis Cinemático Inverso se implementó en el programa de Matlab, debido a la gran variedad de herramientas que posee y su versatilidad para desarrollar programas e interfaces para el usuario. Además, se usó archivos tipo texto (.txt) para el ingreso de información por parte del usuario, eso debido a que se plantea la integración del programa con softwares de diseño y modelación en 3D.

Para el programa desarrollado se definió que información debe ingresar el usuario, la información que debe contener el algoritmo o programa (librería de Juntas) y las herramientas de minimización e inversión de jacobiano que requiere.

### 3.1.1. Datos de Entrada

En primera instancia se procede a detallar la información inicial requerida por el usuario para la ejecución correcta del algoritmo, la cual es ingresada en el formato tipo texto.

- I. **Datos Generales** esta información detalla la cantidad de eslabones que comprenden el mecanismo (nb), el intervalo de estudio (T) definido en segundo y la cantidad de iteraciones a ser realizadas o intervalos de estudio (t). Se presenta un ejemplo en la figura 3.1.

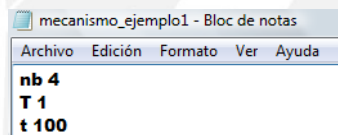


Figura 3.1, Datos Generales

- II. **Datos de Juntas** esta información detalla los tipos de juntas presentes. Para la identificación de las juntas se usa la primera columna, la segunda y tercera identifican que eslabones están involucrados, las columnas 3 hasta 21 representan 6 puntos, los 3 primeros correspondientes al eslabón *i* y los otros 3 puntos corresponden al eslabón *j* cada uno en su marco de referencia respectivo y por último en la columna 22 representa el valor que toma una variable, por ejemplo en una junta tipo distancia la columna 22 representa al valor de la distancia entre los dos puntos definidos. Se debe tomar en cuenta que no todas las juntas requieren el uso de los 6 puntos definidos en el archivo, cada implementada en matlab contiene una explicación para poder definir cada punto según el tipo de junta. Se presenta un ejemplo en la figura 3.2.

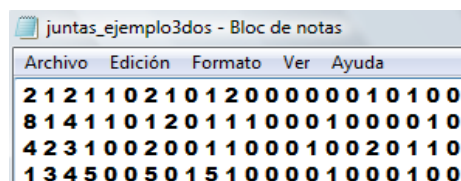


Figura 3.2, Datos de Juntas

- III. **Datos de Juntas Adicionales**, esta información detalla los tipos de juntas motrices involucradas en el mecanismo. La cual presenta un formato similar al de la Información Básica de Juntas con excepción del ítem 22 que no se requiere. Además, en el algoritmo se cuenta con una subrutina de juntas motrices en la cual se debe detallar la ecuación de movimiento, velocidad y aceleración en base al tiempo. Tal como se muestra en la figura 3.3.

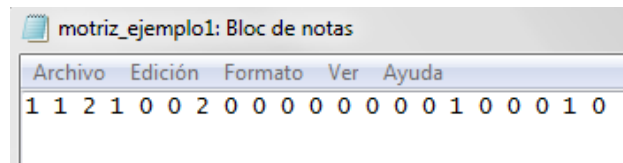


Figura 3.3, Datos de Juntas Adicionales

- IV. **Posición Inicial del Mecanismo**, en este archivo se obtiene la posición y orientación de todos los eslabones a ser usados. En el archivo texto se requiere colocar ( $q_0$ ) al inicio y posteriormente las coordenadas generalizadas de todos los eslabones, en el ejemplo a continuación se cuenta con tres eslabones (21 coordenadas generalizadas). Se presenta un ejemplo en la figura 3.4.

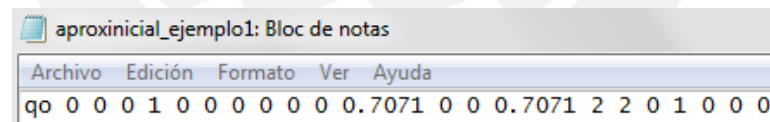


Figura 3.4, Información básica Posición Inicial del Mecanismo

En base a los cuatro archivos previamente descritos se procede a iniciar el algoritmo.

La información se importa al Matlab y se definen las variables en la importación de información y usando un código que separa la información para luego procesarla. En la figura 3.5 (a) se puede observar la importación de la información básica del mecanismo y en la figura 3.5 (b) la importación de la información de las juntas.

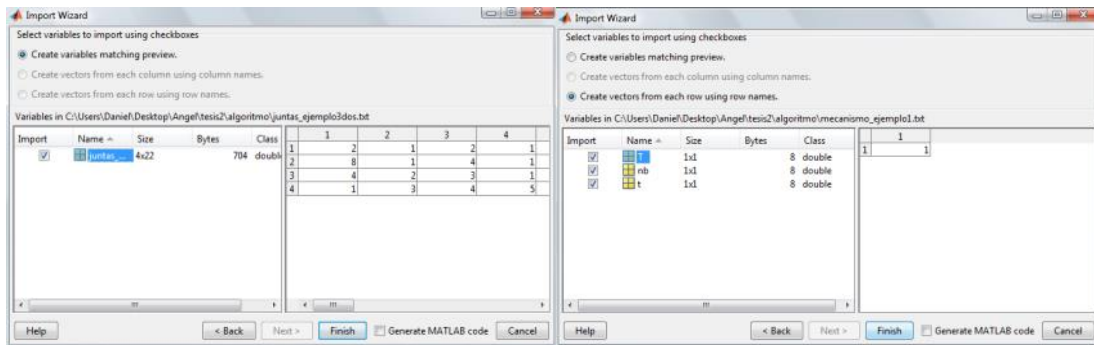


Figura 3.5, Importación de la información a Matlab, (a) Información básica de las juntas; (b) Información básica del mecanismo

### 3.1.2. Desarrollo del Algoritmo

El algoritmo se desarrolla según lo detallado en el acápite 2.2.2. En la cual se tiene un algoritmo marco (“Cinemática Inversa”) y la subrutina de posición.

En una primera instancia el algoritmo verifica que la información inicial ingresada sea correcta y suficiente para la ejecución del algoritmo (Filtro Inicial) y posteriormente se ingresa a la subrutina de posición, velocidad y aceleración, la última subrutina se repite hasta completar el intervalo de tiempo de estudio.

- I. **Filtro Inicial**, el filtro inicial verifica que los grados de libertad del mecanismo y la cantidad de grados de libertad restringidos por las juntas sean iguales. Esta subrutina requiere obtener información de las juntas ingresadas (Datos de Juntas) y juntas motrices (Datos de Juntas Adicionales), además de la normalización de los parámetros de Euler. La subrutina se muestra a continuación.

```

69     %como la juntas motrices en este caso solo quitan un grado de libertad solo
70     %se suma la cantidad de restricciones motrices que se ingreso
71     %se debe incorporar la restriccion de la base o ground
72     groundr=6;
73     restricciones=cont+nb+size(JMinfo,1)+groundr;
74     if 7*nb==restricciones
75         %aquí comienza el programa luego de verificar si se cumple la condicion
76         %para el intervalo de tiempo a analizar se adquiere las matrices q,qlyq2

```

Figura 3.6, código implementado que define si las restricciones ingresadas son igual al número coordenadas generalizadas.



El algoritmo implementado cuenta con una subrutina que obtiene la información de los grados de libertad restringidos por los tipos de juntas ingresados en base a una librería de juntas, la cual tiene como resultado la variable *cont*. Las restricciones por la normalización de parámetros de Euler es igual a la cantidad de eslabones presentes en el mecanismo ( $n_b$ ), las restricciones por el tipo de junta motriz es igual a la cantidad de juntas motrices en el mecanismo ( $size(JMinfo,1)$ ) y se debe tomar en cuenta que el bastidor o marco de referencia general está restringido en todas las coordenadas (*ground*).

- II. **Subrutina de Posición, Velocidad y Aceleración**, la subrutina se describe en el acápite 2.2.2. En esta subrutina se debe tomar en cuenta que la principal es la subrutina de posición. En esta subrutina se procesa para cada instante del tiempo la matriz de la ecuación de movimiento o modelo cinemático y el jacobiano. Esta subrutina se basa en el método de minimización a ser usado.

Para obtener las matrices de la ecuación de movimiento se debe proceder a analizar cada tipo de junta entre eslabones, juntas motrices y la respectiva normalización de los parámetros de Euler en base a la posición inicial ingresada por el usuario ( $q_0$ ). En la figura 3.7 se muestra la subrutina del análisis de las juntas motrices para obtener la ecuación de movimiento.

```

175 %segun las juntas motrices
176 for H=1:size(JMinfo,1)
177     i=JMdata(H,1);
178     j=JMdata(H,2);
179     Pi=JMdata(H,3:5)';
180     Qi=JMdata(H,6:8)';
181     Pj=JMdata(H,9:11)';
182     Qj=JMdata(H,12:14)';
183     res1=0;
184     if JMinfo(H)==1
185         res1=rvm(i,j,Pi,Qi,Pj,Qj,qa,motriz,pos);
186     else
187         if JMinfo(H)==2
188             res1=trm(i,j,Pi,Pj,qa,qal,motriz,pos);
189         else
190             if JMinfo(H)==3
191                 res1=posm(i,j,Pi,Pj,'i',qa,qal,motriz,pos);
192             else
193                 if JMinfo(H)==4
194                     res1=posm(i,j,Pi,Pj,'j',qa,qal,motriz,pos);
195                 else
196                     if JMinfo(H)==5
197                         res1=posm(i,j,Pi,Pj,'k',qa,qal,motriz,pos);
198                     else
199                         end
200                     end
201                 end
202             end
203         end
204     end
205     cont=cont+H;
206     res(cont,1)=res1;
207 end
208 %con lo realizado se obtiene las la primera aproximacion de la
209 %matriz de restricciones.

```

Figura 3.7, código implementado que permite la obtención de la matriz de la ecuación de movimiento (*res*) originada por las juntas motrices.

Posterior a la obtención de la matriz de la ecuación de movimiento o modelo cinemático, se procede a obtener el jacobiano originado por las juntas entre eslabones, juntas motrices y la normalización de los parámetros de Euler, en base a la posición inicial ingresada por el usuario ( $q_0$ ). En la figura 3.8 se muestra el algoritmo para obtener el jacobiano de los parámetros de Euler

```

292      %segun la normalizacion de los parametros de euler
293 -   for H=1:nb
294 -       J1 = normali(H, pos, qa, qal);
295 -       cont=a+H;
296 -       b=7*(H-1)+4;
297 -       d=7*H;
298 -       jacobiano(cont,b:d) = J1;
299
300 -   end
  
```

Figura 3.8, código implementado que permite la obtención del jacobiano originada por la normalización de los parámetros de Euler.

Obtenida ambas matrices se procede a obtener una nueva aproximación en base al método de minimización usado, Newton-Raphson el cual es detallado en el acápite 1.5.1. Este método se basa en la expansión de Taylor para obtener una nueva aproximación de la posición que cumpla con la ecuación de movimiento o modelo cinemático.

Esta iteración se realiza hasta que la normal de la ecuación de movimiento de la aproximación inicial anterior sea menor que el valor de error estipulado en el algoritmo (en este caso 0.00001). El código implementado se muestra en la figura 3.9.

```

440      %obtenido el jacobiano para la aprximacion inicial se procede a
441      %calcular el error.
442 -   invjacob=jacobiano'*inv((jacobiano*jacobiano'));
443 -   correc=jacobiano'*inv((jacobiano*jacobiano'))*res;
444 -   jacos=norm(jacobiano);
445 -   normcorre=norm(res);
  
```

Figura 3.9, código implementado que permite la obtención del error y la nueva aproximación.

Obtenida la posición se procede a obtener la velocidad y aceleración de cada eslabón usando el jacobiano obtenido previamente. En la figura 3.10 se muestra el proceso para obtener la velocidad de todos los eslabones. En

primer lugar se debe obtener la ecuación de velocidad de todas las juntas entre eslabones, juntas motrices y de la normalización de los parámetros de Euler.

```

755 |                                     %segun la normalizacion de los parametros de euler
756 - |                                     for H=1:nb
757 - |                                         veloc1=normali(H, pos, qa, qal);
758 - |                                         cont=a+H;
759 - |                                         veloc(cont,1)=veloc1;

```

Figura 3.10, código implementado que permite la obtención de la matriz de la ecuación de velocidad originada por la normalización de los parámetros de Euler.

Obtenida la ecuación de velocidad y con el jacobiano previamente calculado se procede a obtener la matriz de velocidad de todos los eslabones, en este caso se denomina como la matriz (qa1), tal como se muestra en la figura 3.11.

```

808 - |     qa1=jacobiano'*inv((jacobiano*jacobiano'))*veloc;

```

Figura 3.11, código implementado que permite la obtención de la matriz de velocidad de todos los eslabones.

Este proceso se repite iterativamente hasta completar el tiempo y cantidad de iteraciones que ingreso el usuario al inicio. El resultado del algoritmo se sintetiza en tres matrices principales  $q$ ,  $q1$ ,  $q2$  las cuales son las matrices que contienen la ubicación y orientación, velocidad y aceleración de todos los eslabones en el mecanismo, respectivamente. De estas matrices se pueden obtener de manera gráfica la posición vs tiempo, entre otros tipos de gráficas que requiera el usuario.

### 3.2. Ejemplos Desarrollados

En el presente acápite se procede a desarrollar dos ejemplos que representan los mecanismo que el algoritmo puede resolver, en los cuales se realiza un ejemplo de mecanismo clásico en la cual se detalla los problemas que se pueden presentar en la modelación de un mecanismo en un plano en el espacio y un brazo robótico que representa una cadena cinemática abierta en el espacio

### 3.2.1. Mecanismo Plano (Biela-Manivela-Corredera)

Se procede a desarrollar el mecanismo tipo Biela-Manivela-Corredera, en la figura 3.12 se presenta el diagrama del mecanismo con las características físicas, identificación de eslabones y posición inicial. Se debe tomar en cuenta que este modelo se desarrolla también en el libro de Haug [24] y el eslabón numero uno es el marco de referencia o bastidor.

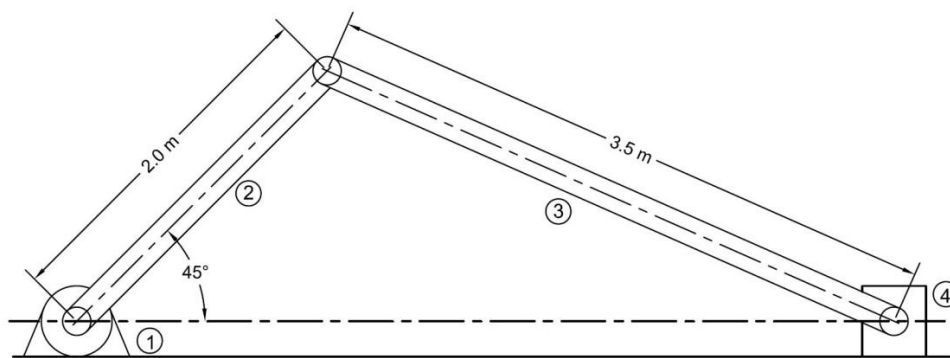


Figura 3.12, diagrama del mecanismo, ejemplo 1

La función que rige a la junta motriz, variación del ángulo entre el eslabón 2 y el bastidor con respecto al tiempo es la siguiente función, se tiene dos velocidades, ya que se analizara dos escenarios:

$$\theta(t)_1 = \frac{\pi}{4} + 2\pi t$$

$$\theta(t)_2 = \frac{\pi}{4} + 4\pi t$$

En primero lugar se debe tomar en cuenta que el mecanismo está compuesto por cuatro (04) eslabones, considerando el bastidor como un eslabón (marco de referencia), el tiempo de análisis será de un (01) segundo y con cien (100) intervalos de estudio. Toda esta información (Datos Generales) se ingresa en forma de texto en un block de notas tal como se muestra en la figura 3.13.

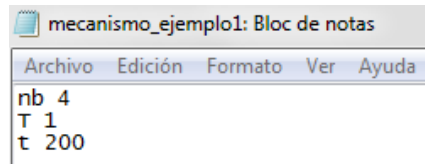


Figura 3.13, Datos Generales.

Como segundo paso se procede a realizar la modelación del mecanismo con las juntas desarrolladas en el Capítulo 2. Guiándose por un modelamiento típico, las juntas del mecanismo son cuatro (04), se considera el marco de referencia como eslabón 1. Las juntas del mecanismo se detallan en la tabla 3.1.

Tabla 3.1 Modelo de Juntas Mecanismo Clásico

Item	Descripción	Eslabón 1	Eslabón 2
RJ – 01	Junta de Revolución	1	2
RJ – 02	Junta de Revolución	2	3
RJ – 03	Junta de Revolución	3	4
TJ – 01	Junta de Traslación	1	4

Con el modelo desarrollado se procede a calcular los grados de Libertad del mecanismo según los parámetros de Euler (para mayor detalle revisar el Capítulo 2):

Grados de Libertad del mecanismo (nb=4) = 28

Restricciones/Juntas

RJ 01 (eslabones 1 y 2)	:	5
RJ 02 (eslabones 2 y 3)	:	5
RJ 03 (eslabones 3 y 4)	:	5
TJ 01 (eslabones 1 y 4)	:	5

Restricciones del marco de referencia (eslabón 1) : 6

Normalización de los Parámetros de Euler (nb=4) : 4

Grados de libertad eliminados = 30

Grados de Libertad (DOF) = -2

Este resultado nos indica que el sistema tiene demasiadas restricción y por ende existe relaciones redundantes. Por lo que el primer modelo tiene exceso de relaciones y si se emplea presentaría un Jacobiano no cuadrado, ya que se tiene mayor cantidad de restricciones que grados de libertad.

El problema de las restricciones redundantes es abordado por diversos autores [24], quienes explican que es requerida la implementación de un programa que pueda identificar y eliminar las relaciones redundantes dentro de la definición y modelación de las juntas. Debido a la complejidad de este algoritmo y al no encontrarse dentro del alcance de esta tesis se procede a desarrollar un modelo cinemático del mecanismo sin relaciones redundantes.

En búsqueda de un modelo sin restricciones redundantes, se procede a realizar el siguiente modelo compuesto por cinco (05) juntas como se muestra en la tabla 3.2.

Tabla 3.2 Modelo de Juntas Mecanismo Clásico

Item	Descripción	Eslabón 1	Eslabón 2
RJ – 01	Junta de Revolución	1	2
RJ – 02	Junta de Revolución	2	3
TJ – 01	Junta de Traslación	1	4
CD (i)	Coincidencia en el eje X	3	4
CD (j)	Coincidencia en el eje Y	1	4

Se procede a analizar los grados de libertad del mecanismo:

$$\text{Grados de Libertad del mecanismo (nb=4)} = 28$$

Restricciones/Juntas

RJ 01 (eslabones 1 y 2)	:	5
RJ 02 (eslabones 2 y 3)	:	5
TJ 01 (eslabones 1 y 4)	:	5
CD (i) (eslabones 3 y 4)	:	1
CD (j) (eslabones 1 y 4)	:	1

Restricciones del marco de referencia (eslabón 1)	:	6
Normalización de los Parámetros de Euler (nb=4)	:	4
Grados de libertad eliminados	=	27
Grados de Libertad (DOF)	=	1

Se debe tomar en cuenta que existen diversas aproximaciones para obtener el modelo del mecanismo, teniendo varios modelos satisfactorios para un mismo mecanismo.

Habiendo desarrollado modelo que cumpla satisfactoriamente con la cantidad de juntas, grados de libertad y sin restricciones redundantes, se procede a definir los Datos de Juntas. Los Datos de Juntas se ingresan mediante el formato tipo texto (block de notas) como se muestra en la figura 3.14.



Figura 3.14, Datos de Juntas

Como se muestra en el cálculo de grados de libertad del modelo seleccionado se tiene un grado de libertad adicional. El grado de libertad adicional es la junta motriz del mecanismo, esta junta motriz es una junta de revolución la cual se define con la junta tipo DP1 entre el bastidor y el eslabón 2. Por lo que se ingresa esta información como block de notas Datos de Juntas Adicionales, tal como se muestra en la figura 3.15.

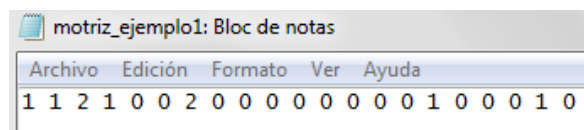


Figura 3.15, Datos de Juntas Adicionales

La ecuación que cumple la junta motriz se implementa en la subrutina MOTRIZ. En la subrutina se ingresa la ecuación de posición que describe la junta motriz (MOTRIZ1),

la ecuación de velocidad (MOTRIZ2) y la ecuación de la aceleración (MOTRIZ3). Esta información se muestra en la figura 3.16.

```

MOTRIZ.m: Bloc de notas
Archivo Edición Formato Ver Ayuda
function MOTRIZ=MOTRIZ(T,H,step,Mo)
%funcion para obtener los valores de posicion, velocidad, aceleracion y
%jacobiano segun la ecuacion de movimiento impuesta por la junta motriz
%se debe ingresar manualmente por el momento y colocar las funcion y su
%derivada correspondiente con respecto al tiempo;
if H==1;
    MOTRIZ1=cos(pi/4+2*pi()*T);
    MOTRIZ2=-sin(pi/4+2*pi()*T)*2*pi();
    MOTRIZ3=-cos(pi/4+2*pi()*T)*(2*pi())^2;
end
else
MOTRIZ1=0;
MOTRIZ2=0;
MOTRIZ3=0;
end
MOTRIZ=[MOTRIZ1 MOTRIZ2 MOTRIZ3];
    
```

Figura 3.16, Sub-Rutina de Junta Motriz

Como último requerimiento del algoritmo es necesario definir una posición inicial de los eslabones. Tomando en cuenta que cuando el tiempo es cero, el ángulo entre el bastidor y el eslabón 2 es de 45° (figura 3.12). Para lograr dicha posición se desarrolla la siguiente matriz como Posición Inicial del Mecanismo:

$$q_0 = \begin{bmatrix} [0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0]^T \\ [0 \ 0 \ 0 \ 0.9 \ 0 \ 0 \ 0.4]^T \\ [3 \ 0.7 \ 0 \ -0.97 \ 0 \ 0 \ 0.2]^T \\ [4.6 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0]^T \end{bmatrix}$$

Para ingresar dicha información al algoritmo se desarrolla su respectivo block de notas, mostrado en la figura 3.17.

```

aproxinicial_ejemplo1: Bloc de notas
Archivo Edición Formato Ver Ayuda
qo 0 0 0 1 0 0 0 0 0 0 0.9 0 0 0.4 3 0.7 0 -0.97 0 0 0.2 4.6 0 0 1 0 0 0
    
```

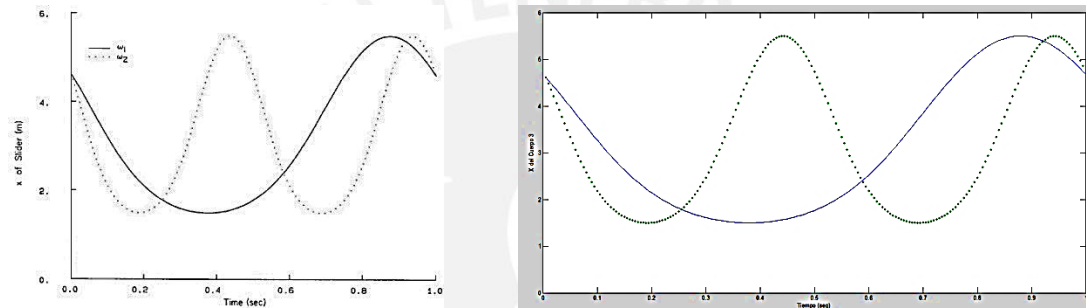
Figura 3.17, Posición Inicial del Mecanismo

Se ejecutó el programa con la información desarrollada y se obtuvo los resultados en un tiempo de aproximadamente 20 segundos.

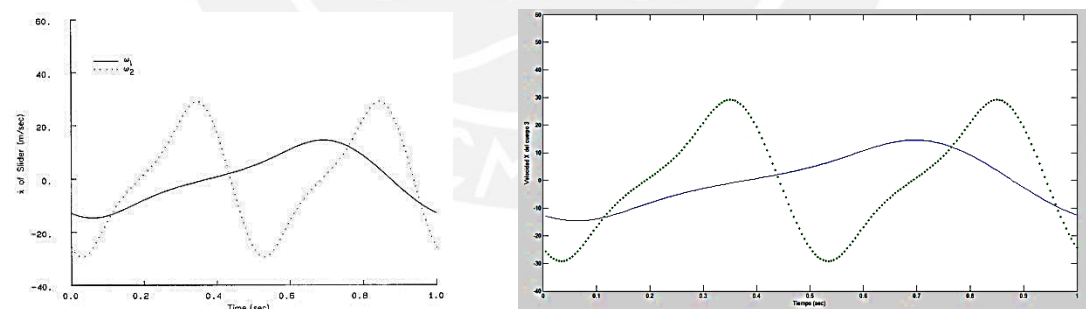


Los resultados obtenidos del Matlab (figura en el lado derecho) se muestran a continuación conjuntamente con los gráficos mostrados en el Capítulo 5 del libro “Computer Aided Kinematics and Dynamics of Mechanical Systems, Volumen I: Basic Methods” [24], gráficos de la parte izquierda (figura en el lado izquierdo). Tomar en cuenta que  $w_1$  representa a la ecuación  $\theta(t)_1$  (línea azul sólida) y  $w_2$  representa a la ecuación  $\theta(t)_2$ :

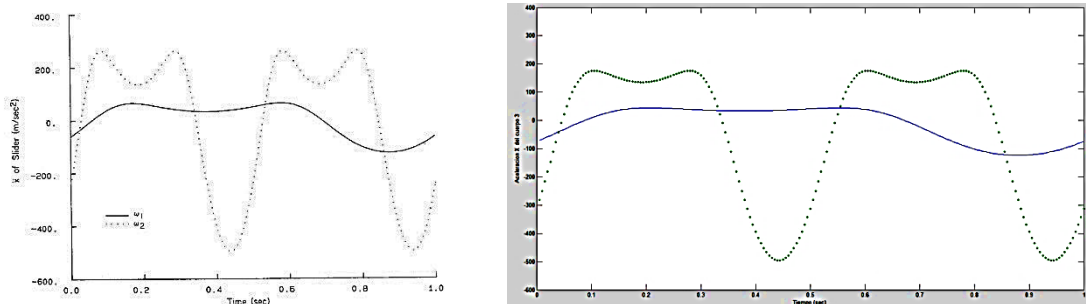
Posición en el eje X del Pistón o eslabón 4:



Velocidad en el eje X del Pistón o eslabón 4:



Aceleración en el eje X del Pistón o eslabón 4:



Se puede apreciar que el cambio de la velocidad angular de la junta motriz impacta directamente en la aceleración del eslabón 4. Esto implica en mayores esfuerzos presentes en el mecanismo. El desplazamiento máximo de ambos modelos del mecanismo es de 5.5 metros pero en la variación en la velocidad y aceleración llega a valores de 5 veces. Esta información es vital para poder desarrollar la dinámica inversa y poder diseñar correcta el mecanismo.

### 3.2.2. Mecanismo Tridimensional (Brazo Robótico)

El brazo robótico a ser analizado para la modelación de mecanismo de cadena cinemáticas abiertas en el espacio, es tipo articulado esférico. Dicho cual presenta una junta de revolución en la base con sentido en el eje Z y a dos juntas de revolución en cada articulación del brazo robótico, como se muestra en la figura 3.18.

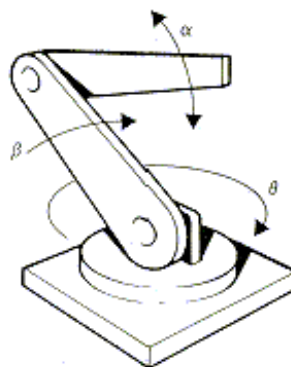


Figura 3.18, Brazo Robótico Articulado

Las características físicas y sus dimensiones se pueden apreciar en la figura 3.19 (a) con vista frontal, en la figura 3.19 (b) se puede apreciar el recorrido a ser analizado por el algoritmo.

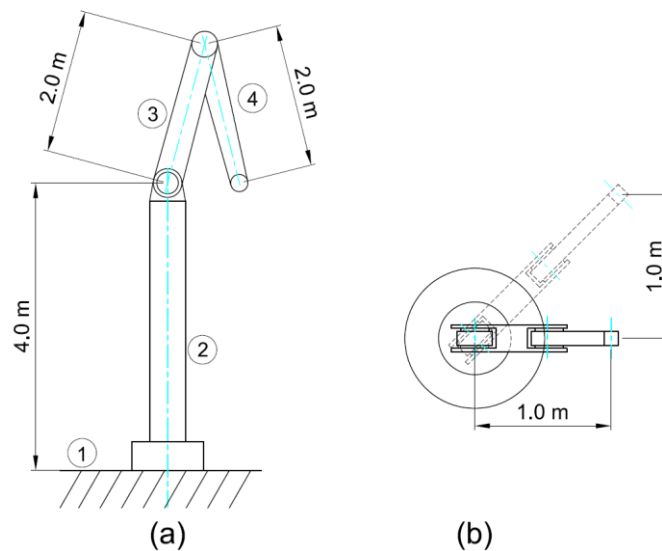


Figura 3.19, Brazo Robótico. (a) Características Físicas del Brazo Robótico; (b) Recorrido a ser analizado.

El recorrido del actuador será desarrollado por un intervalo de dos (2) segundos y con cien (100) intervalos para un análisis detallado. El brazo robótico tiene cuatro eslabones, considerando un eslabón como el marco de referencia. Entonces, para poder utilizar el programa se ingresa la siguiente información de análisis:

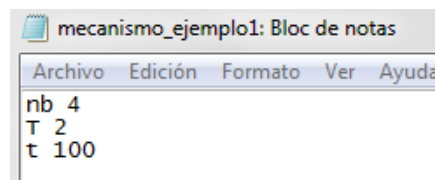
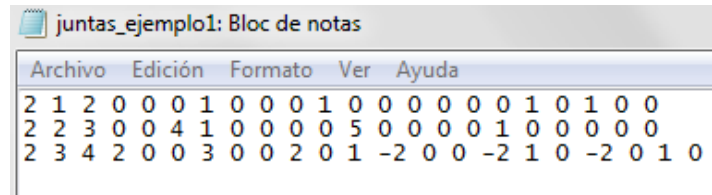


Figura 3.20, Datos Generales.

A su vez se debe determinar las juntas y los eslabones correspondientes, en este caso se tiene juntas de revolución entre el cuerpo 1 (marco de referencia) y el eslabón 2, entre el eslabón 2 y 3 y entre el eslabón 3 y 4.



Archivo	Edición	Formato	Ver	Ayuda																	
2	1	2	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0	1	0	0	
2	2	3	0	0	4	1	0	0	0	0	5	0	0	0	0	1	0	0	0	0	
2	3	4	2	0	0	3	0	0	2	0	1	-2	0	0	-2	1	0	-2	0	1	0

Figura 3.21, Datos de Juntas

Ya definido el mecanismo, se procede a analizar los grados de libertad del mecanismo:

Grados de Libertad del mecanismo (nb=4) = 28

Restricciones/Juntas

- RJ 01 (eslabones 1 y 2) : 5
- RJ 02 (eslabones 2 y 3) : 5
- RJ 01 (eslabones 3 y 4) : 5

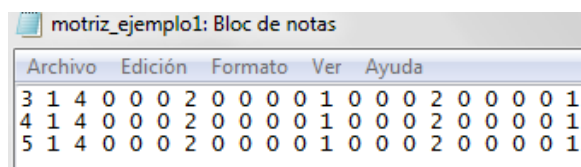
Restricciones del marco de referencia (eslabón 1) : 6

Normalización de los Parámetros de Euler (nb=4) : 4

Grados de libertad eliminados = 25

Grados de Libertad (DOF) = 3

Como se muestra el cálculo de grados de libertad se tiene tres (03) adicionales, lo que hace que se requiera tener tres parámetros que establezcan el movimiento del brazo robótico. Las juntas motrices a ser empleadas en este ejemplo son las siguientes, las cuales están ubicadas en el actuador:



Archivo	Edición	Formato	Ver	Ayuda																	
3	1	4	0	0	0	2	0	0	0	0	1	0	0	0	2	0	0	0	0	0	1
4	1	4	0	0	0	2	0	0	0	0	1	0	0	0	2	0	0	0	0	0	1
5	1	4	0	0	0	2	0	0	0	0	1	0	0	0	2	0	0	0	0	0	1

Figura 3.22, Datos de Juntas Adicionales

Se debe tomar en cuenta que el actuador se va a desplazar en el eje y y por un tiempo de 2 segundos a una velocidad constante de 0.5 m/s, en el resto de los ejes el brazo robótico mantiene su posición inicial (en el eje x una posición constante de 1 m y en el eje z una posición constante de 4 m). Por lo que las juntas motrices representan el desplazamiento del actuador en el espacio. Toda esta información se ingresa en la sub-rutina MOTRIZ, tal como se realizó en el ejemplo anterior, donde H 1, H 2 y H 3 representan la posición del actuador en los ejes x, y y z.

```
MOTRIZ.m: Bloc de notas
Archivo Edición Formato Ver Ayuda
function MOTRIZ=MOTRIZ(T,H,step,Mo)
%funcion para obtener los valores de posicion, velocidad, aceleracion y
%jacobiano segun la ecuacion de movimiento impuesta por la junta motriz
%se debe ingresar manualmente por el momento y colocar las funcion y su
%derivada correspondiente con respecto al tiempo;
if H==1;
    MOTRIZ1=1;
    MOTRIZ2=0;
    MOTRIZ3=0;
else
    if H==2;
        MOTRIZ1=0+0.5*T;
        MOTRIZ2=0.5;
        MOTRIZ3=0;
    else
        if H==3;
            MOTRIZ1=4;
            MOTRIZ2=0;
            MOTRIZ3=0;
        else
            MOTRIZ1=0;
            MOTRIZ2=0;
            MOTRIZ3=0;
        end
    end
end
MOTRIZ=[MOTRIZ1 MOTRIZ2 MOTRIZ3];
```

Figura 3.23, Sub-Rutina de Junta Motriz

Por último se procede a analizar la posición inicial del mecanismo, la cual se detalla en la siguiente matriz.

$$q_0 = \begin{bmatrix} [0 & 0 & 0 & 1 & 0 & 0 & 0]^T \\ [0 & 0 & 0 & 1 & 0 & 0 & 0]^T \\ [0 & 0 & 4 & 0.79 & 0 & -0.62 & 0]^T \\ [1 & 0 & 4 & 0.79 & 0 & 0.62 & 0]^T \end{bmatrix}$$

La información obtenida se ingresa al Matlab en formato de block de notas, tómease en cuenta que se toma la posición del actuador como el origen del cuarto eslabón, tal como se muestra en la figura 3.24:

```
aproxinicial_ejemplo1.txt: Bloc de notas
Archivo Edición Formato Ver Ayuda
qo 0 0 0 1 0 0 0 0 4 1 0 0 0 0 0 4 0.79 0 -0.62 0 1 0 4 0.79 0 0.62 0
```

Figura 3.24, Posición Inicial del Mecanismo

Con la información desarrollada se procede a ejecutar el algoritmo, el tiempo aproximado de procesamiento de la información es de 20 segundos.

Se procede a graficar la posición del actuador en el plano X-Y, con el fin de verificar su correcto desplazamiento. Conjuntamente se presenta la posición en el eje Y del actuador.

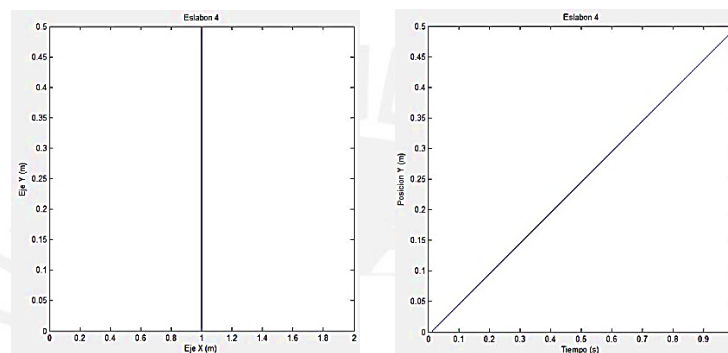


Figura 3.25,(a) Posición del actuador en el plano X-Y (b) Posición del actuador en el eje Y con respecto al tiempo

En la figura 3.25 (a) se puede observar que el actuador se desplaza únicamente en el plano Y de la posición inicial 0 a 0.5 y en la figura 3.25 (b) se puede apreciar que se desplaza de manera constante. Con esta información se corrobora que el actuador se desplaza según lo deseado.

A continuación se va a analizar la rotación de los eslabones principales. En primer lugar analizaremos es eslabón 2, el cual gira con respecto al eje Z, en la figura 3.26 se puede observar la rotación de dicho eslabón.

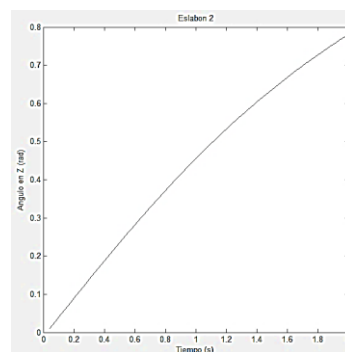


Figura 3.26. Rotación del eslabón 2 respecto al eje Z

Como se puede observar la rotación se da desde el ángulo 0 a 0.7804 radianes, mostrando el giro estimado para obtener la posición final del actuador.

Por último se procede a analizar la rotación con respecto al eje Y de los eslabones 3 y 4, con el fin de verificar su orientación y posición. Este paso nos ayudara a verificar si ocurrió inversión del mecanismo en algún punto.

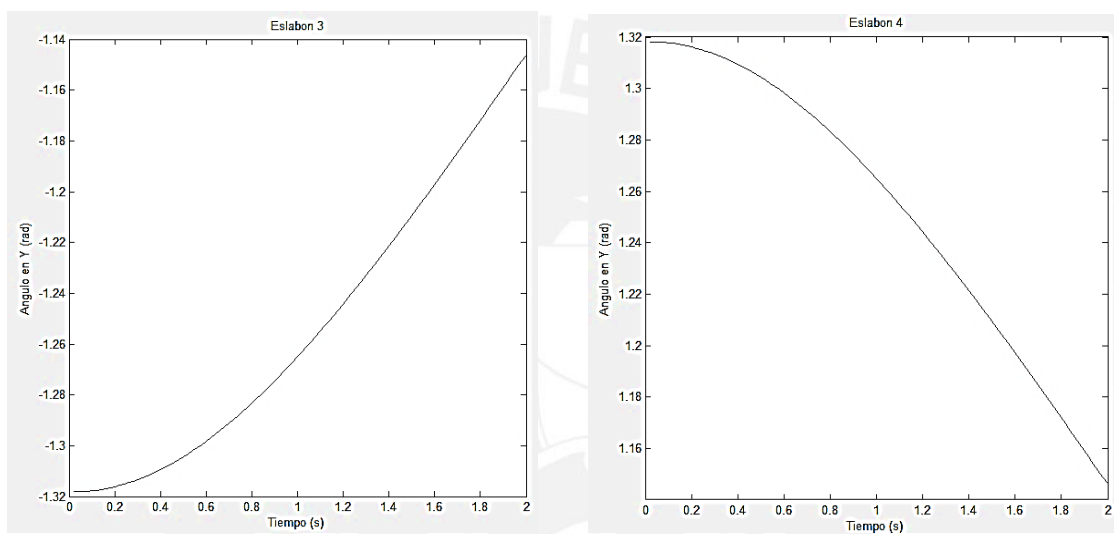


Figura 3.27 (a) Rotación del eslabón 3 respecto al eje Y (b) Rotación del eslabón 4 respecto al eje Y

Como se puede observar en el gráfico 3.27(a) el ángulo de rotación del eslabón 3 es de la misma magnitud pero de signo contrario que el ángulo de rotación del eslabón 4. Este efecto se debe a la disposición de los eslabones.

Analizando las gráficas presentadas se puede apreciar que el comportamiento del mecanismo es según lo esperado en el desplazamiento del actuador y la rotación de los eslabones respecto al tiempo.

## CONCLUSIONES Y RECOMENDACIONES

Se definió el método de análisis más apropiado para una fácil implementación en Matlab. En particular se encontró que el uso de parámetros de Euler fue el más óptimo para poder evitar las singularidades presentes en la modelación y ejecución del programa, además de dar robustez al algoritmo desarrollado.

Una vez definido el tipo de aproximación a ser realizada, se logró realizar una base matemática fuerte para poder ejecutar y describir los tipos de juntas más comunes. Con esta información se creó una librería de juntas en la que se detallan diez (10) tipos de juntas entre las cuales se destaca la junta de revolución, esférica, cilíndrica y de posición en el espacio. Hay que resaltar que dentro de las juntas desarrolladas se implementó juntas básicas, las cuales permiten ampliar la librería con juntas más avanzadas como tornillos de potencia, engranajes, etc.

Con la creación de la librería de juntas se implementó el algoritmo base para la obtención del modelamiento cinemático inverso de mecanismo en Matlab. En este proceso, se definió los tipos de entrada requerido para su ejecución y el formato en la que será ingresada por el usuario. Esta información básica se ingresa de manera rápida al algoritmo mediante la importación de información usando el archivo de texto.

Con el algoritmo implementado en el Matlab se desarrolló dos ejemplos representativos, un mecanismo en un plano y brazo robótico en el espacio. Los cuales obtuvieron soluciones satisfactorias y convergencias en tiempo reducidos. Durante el modelamiento y desarrollo del mecanismo en un plano se presentó diversas dificultades, tal como la presencia de relaciones redundantes dentro de los tipos de juntas. Para evitar dicho problema se planteó el uso de la ecuación del cálculo de grados de libertad, siendo esta aproximación errónea debido a la falta de garantía de evasión de relaciones redundantes. Para superar la problemática, se realizó un análisis preliminar, por parte del usuario, para eliminar las relaciones redundantes



mediante un correcto modelamiento del mecanismo, esto implicó un incremento de dificultad al usar el algoritmo.

Incluso con un correcto modelamiento del mecanismo se observó divergencia en la resolución de la inversión del jacobiano, esto debido a la presencia de singularidades cercanas a puntos de cambio de giro de las juntas motrices de rotación, puntos donde se pueden generar una inversión del mecanismo y modelos funcionales del mecanismo pero sin la robustez requerida para la evasión de las singularidades. Todo esto conllevó a generar modelos robustos y sin singularidades durante el modelamiento inicial del mecanismo, elevando la complejidad en el uso del algoritmo. En consecuencia, para la implementación futura del algoritmo en un software de mayor alcance se requerirá de una subrutina que evite las relaciones redundantes, evasión de singularidades en el espacio de trabajo del mecanismo y detección de posibles fuentes de error durante el modelamiento y resolución de la inversión del jacobiano.

En suma, se desarrolló el algoritmo mediante subrutinas y módulos de fácil implementación para ser utilizados en un software para el análisis cinemático y dinámico de mecanismo a ser desarrollado en un futuro

## BIBLIOGRAFÍA

- [1] A.B.W. Kennedy, "Reuleaux, Kinematics of Machinery", Macmillan, Londres 1876
- [2] F.H. Mohammad and A. Sham, "Modeling of a An Arm Via Kane's Method: An Inverse Dynamic Approach", *European Journal of Scientific Research* ISSN 1450.216X Vol.33 No.2 (2009), pp 358-364.
- [3] R. Featherstone, "Rigid Body Dinamics", Springer New York, 2008
- [4] J. Angeles and O. Ma, "An Algorithm for the Inverse Dynamics of n-Axis General Manipulators Using Kane's Equation", *Computer Math Application* Vol. 17, No12, pp 1545-1561, 1989.
- [5] H. Chaudhary and S. Kumar, "Dynamics and Balancing Multibody Systems" Springer New York, 2009.
- [6] Hong-Sen Yan and M. Ceccarelli, "International Symposium on History of Machines and Mechanisms", Springer New York, 2009
- [7] P. Flores, J. Ambrosio, J.C. Pimienta Claro and H.M. Lankarani, "Kinematics and Dynamics of Multibody Systems with Imperfect Joints", Springer, 2008
- [8] V.Mata, S. Provenzano, J.L. Cuadrado and F. Valero, "Inverse Dynamic problem in robot using Gibbs-Appell equations", *Robotica* Vol. 20, pp 59-67.
- [9] B. Fabien, "Analytical System Dynamics", Springer, 2009
- [10] J.E. Shigley and J.J. Uicker, "Teoría de maquinas y mecanismos", McGraw Hill, 1983.
- [11] R. Calero y J. Carta, "Fundamentos de mecanismos y máquinas para Ingenieros", McGraw Hill, 1998
- [12] W.F. Feehery, J.R. Banga y P.I. Barton, "A Novel Approach to Dynamic optimization of ODE and DAE Systems as High-Index Problems", AICHE Annual Meeting, Miami, 1995
- [13] Subir Kumar Saha, "Recursive Dynamics Algorithms for Serial, Parallel, and Closed-chain Multibody Systems", Department of Mechanical Engineering, IIT Delhi Hauz Khas, New Delhi 110 016.
- [14] <http://www.slimy.com/~steuard/teaching/tutorials/Lagrange.html>
- [15] Dr H. Prem, E. Ramsay, Dr J. Pont, Dr J. McLean, J. Woodrooffe, "Comparison of Modelling Systems for Performance-Based Assessments of Heavy Vehicles", ARRB Transport Research Ltd.
- [16] G. Jelenic and J.J. Muñoz, "Modelling Multi-Body Systems Using the Master-Slave Approach", *Computational Methods in Applied Sciences* Vol. 4

- [17] A.A. Shabana, "Dynamic of Multibody Systems", Cambridge University Press 2005
- [18] P.P Teodorescu, "Mechanical System, Classical Model" Tomo 1
- [19] P.P Teodorescu, "Mechanical System, Classical Model" Tomo 2
- [20] P.P Teodorescu, "Mechanical System, Classical Model" Tomo 3
- [21] A.A Shabana, "Computational Dynamics" segunda edición, John Wiley & Sons, Inc. 2001
- [22] A. Ollero, "Robótica. Manipuladores y robots móviles", Alfaomega y marcombo, 2007
- [23] B. Siciliano, O. Khatib, "Handbook of Robotics", Springer 2008
- [24] E.J. Haug, "Computer Aided Kinematics and Dynamics of Mechanical Systems, Volumen I: Basic Methods", Allyn and Bacon 1989.
- [25] Dan Negrut, "Advanced Computational Multibody Dynamics" Universidad de Wisconsin, Madison 2010. "<http://sbel.wisc.edu/Courses/>"
- [26] S. Tejomurtula, S. Kak, "Inverse Kinematics in robotics using neural Network", Information Science 116 (1999) 147-164
- [27] J.J. Craig, "Introduction to Robotic: Mechanisms and Controls", Addison-Wesley, Reading, MA, 1989
- [28] D. Manocha, J.F Canny, "Efficient inverse kinematics for general 6r manipulators", IEEE Transactions on Robotics and Automation 10 (5) (1994) 648-657
- [29] D. Manocha, Y. Zhu, "A fast algorithm and system for the inverse kinematics of general serial manipulator", IEEE Conference on Robotics and Automation 94 (1994) 3348-3354.
- [30] S.R. Buss, "Introduction to Inverse Kinematics with Jacobian Transpose, Pseudoinverse and Damped Least Square Methods", Department of Mathematics, University of California, San Diego, October 7, 2009.
- [31] A. Balestrino, G. De Maria and L. Sciavicco, "Robust control of robotic manipulators", Proc. 9<sup>th</sup> IFAC World Congress, Vol. 5, 1984, pp. 2435-2440
- [32] W.A. Wolovich and H. Elliot, "A computational technique for inverse kinematics", Proc. 23<sup>rd</sup> IEEE Conference on Decision and Control, 1984, pp. 1359-1363
- [33] J. Baillieul, "Kinematic programming alternatives for redundant manipulators", Proc IEEE International Conference on Robotics and Automation, 1985, pp 722-728
- [34] C.W. Wampler, "Manipulator inverse kinematic solutions based on vector formulations and damped least square methods", IEEE Transactions on Systems, Man, and Cybernetics, 16 (1986) pp. 93-101

- [35] Y. Nakamura and H. Hanafusa, “Inverse kinematics solution with singularity robustness for robot manipulator control”, *Journal of Dynamic Systems, Measurement, and Control*, 108 (1986), pp. 163-171.
- [36] D. Pieper, “The kinematics of Manipulators Under Computer Control”, Ph D. Thesis (Stanford University, Stanford 1968)

