

Departamento de Automática, Ingeniería Electrónica
e Informática Industrial

Escuela Técnica Superior de Ingenieros Industriales

Modelado, Control y Diseño de Robots Submarinos de Estructura Paralela con Impulsores Vectorizados

Autor

Héctor Alonso Moreno Ávalos

Ingeniero Mecánico

Directores

Roque Saltarén

Doctor Ingeniero Industrial

Rafael Aracil

Doctor Ingeniero Industrial

2013

Resumen

En años recientes, se han realizado algunos esfuerzos para equipar a robots submarinos con impulsores vectorizados. Este sistema de propulsión permite el uso de una menor cantidad de impulsores, esto a su vez tiene consecuencias favorables en el volumen y costo del robot a medida que una mayor potencia de propulsión es requerida. El propósito de esta tesis es realizar un estudio sobre el modelado, control y diseño de robots submarinos de estructura paralela con impulsores vectorizados. De esta manera exponer los aspectos más importantes relativos a estos puntos, y proponer soluciones a los problemas que plantea la arquitectura de estos robots.

En esta tesis se tomo como objeto de estudio el robot Remo 2, cuya estructura paralela representa una gran parte del volumen del robot y esto hace que su análisis sea el más complejo que se pueda tener en los robots de esta categoría. El diseño de este robot es un concepto radicalmente diferente al de los robots submarinos convencionales. Sus características son prometedoras, pero para poder sacar provecho de estas potencialidades es necesario un entendimiento de la dinámica del robot. En este trabajo se presenta el desarrollo y análisis de modelos analíticos, y el desarrollo de herramientas de simulación para este robot. El propósito de estas herramientas es identificar las oportunidades y restricciones impuestas por la estructura y la dinámica del vehículo.

Se presenta el planteamiento (y solución) de los problemas cinemático y dinámico inverso para un robot submarino de estructura paralela. Por otro lado, se demostró por primera vez el funcionamiento del concepto del robot submarino a través de una herramienta de simulación. Haciendo uso de esta herramienta se exploró el desempeño del robot bajo diversos esquemas de control, se encontró que el robot es capaz de ejecutar con éxito diversas maniobras empleando controladores clásicos. Se presento un controlador basado en el modelo del robot y se demostro sus ventajas. Posteriormente se presento un apartado cuyo propósito es exponer aspectos importantes sobre el diseño de este tipo de robots, sobre todo aquellos que influyen en su desempeño cinetostático. Se plantea el problema del diseño óptimo de un robot tipo Remo 2, y se proponen un par de índices de desempeño globales. Otra contribución fue, en condición de coautoría, el diseño y análisis de una nueva estructura paralela la cual no ha sido considerada anteriormente en la literatura.

Abstract

In recent years, there have been some efforts to equip underwater robots with vectored thrusters. This propulsion system allows the use of a smaller number of thrusters, which in turn, has favorable consequences in the cost and volume when a greater propulsive power is required. The purpose of this thesis is to study the modeling, control and design of underwater robots of parallel structure with vectored thrusters. Thus exposing the most important aspects related to these issues, and propose solutions to the problems given by the architecture of these robots.

In this thesis the robot Remo 2 is the object of study. Its parallel structure represents a large part of the volume of the robot and this makes the analysis the most complex for this class of robots. The design of this robot is a radically different concept of conventional underwater robots. Its features are promising, but in order to exploit these potentials it is necessary to understand the dynamics of the robot. This thesis presents the development of analytical models and simulation tools for this robot. The purpose of these tools is to identify opportunities and constraints imposed by the dynamics of the structure of the vehicle.

In this thesis the solution of the inverse kinematic and dynamic problems for an underwater robot with parallel structure is presented. Furthermore, it was demonstrated, for the first time, the functioning of concept of Remo 2 using a powerful simulation package whose reliability have been validated by various institutions worldwide. Using this tool we explored the robot's performance under various control schemes, it was found that the robot is able to successfully execute various maneuvers using classic controllers. A model based controller was proposed and its advantages demonstrated. Subsequently, a section whose purpose is to expose important aspects of the design of such robots, especially those that affect its kinetostatic performance, is presented. The problem of the optimum design of the robot Remo 2 is stated, and we suggests a series of global performance indices. Another contribution was, in condition of coauthorship, the design and analysis of a novel parallel structure which has not been considered previously in the literature.

**A Dios, Isela y Nina.
A mis padres y hermanos.**

Índice general

Resumen	III
Abstract	V
Índice general	IX
Contents	IX
Lista de Figuras	XIII
Lista de Tablas	XVII
1. Introducción	1
1.1. Robots Submarinos con Impulsores Vectorizados de Estructura Paralela	2
1.2. Fortalezas y Debilidades de los Robots Submarinos con Impulsores Vectorizados de Estructura Paralela	5
1.3. Objetivos	8
1.4. Principales Aportaciones	9
1.5. Organización de la Tesis	10
I Fundamentos	11
2. Robótica Submarina: Generalidades	13
2.1. Introducción	13
2.2. Clasificación de los Robots Submarinos	14
2.2.1. Robots Submarinos Operados Remotamente, ROVs	14
2.2.2. Robots Submarinos Autónomos, AUVs	15
2.2.3. Robots Submarinos Autónomos para Intervenciones, IAUVs	16
2.3. Estructura de un robot submarino	16
2.4. Sensores	18
2.5. Sistemas de propulsión	18
2.5.1. Impulsores de Hélice	19
2.5.2. Planeador Acuático (Glider)	20
2.5.3. Bioinspirados	21
2.5.4. Inyección	22

2.5.5. Impulsor Magnetohidrodinámico, MHD	22
2.5.6. Tracción con el fondo marino	23
2.6. Brazos Robóticos	23
2.7. Fuentes de Energía	23
2.8. Conceptos Elementales y Antecedentes del Control de Robots Submarinos.	24
2.9. Resumen	27
3. Modelado de Robots Submarinos	29
3.1. Introducción	29
3.2. Modelado Cinemático de Robots Submarinos	31
3.2.1. Matriz de Rotación	31
3.2.2. Ángulos de Euler	31
3.2.3. Parámetros de Euler	33
3.3. Dinámica de Robots Submarinos	34
3.3.1. Fuerzas Inerciales	35
3.3.2. Fuerzas Hidrodinámicas	37
3.3.3. Masa Añadida	37
3.3.3.1. Fuerzas Viscosas	39
3.3.4. Fuerzas de Restitución	40
3.4. Resumen	41
II Desarrollo	43
4. Modelado del Robots Submarinos de Estructura Paralela con Impulsores Vectorizados	45
4.1. Cinemática del Robot Remo II	45
4.1.1. Analisis de Posición	46
4.1.2. Analisis de Velocidad	47
4.1.3. Analisis de Aceleración	48
4.1.4. Analisis del Jacobiano	49
4.2. Dinámica del Robot Remo II	50
4.3. Parámetros del Robot	54
4.3.1. Consideraciones previas	54
4.3.2. Parámetros Geométricos	56
4.3.3. Parámetros Inerciales e Hidrodinámicos	57
4.4. Simulación	61
4.4.1. Trayectoria en línea recta hacia el frente	61
4.4.2. Trayectoria Circular utilizando un CMG	64
4.5. Resumen	72

5. Control de Robots Submarinos con IVEP	73
5.1. Introducción	73
5.2. Control del Robot considerado como una estructura rígida	76
5.2.1. Controlador P con retroalimentación de velocidad	77
5.2.1.1. Línea Recta	77
5.2.1.2. Traslación Pura Lateral	78
5.2.1.3. Rotación Pura	81
5.2.2. Controlador PID	84
5.3. Control de la Estructura Paralela durante la navegación	88
5.3.1. Controlador articular P con retroalimentación de velocidad	90
5.3.1.1. Línea Recta	90
5.3.1.2. Rotación Pura	91
5.3.2. Controlador articular PID	91
5.4. Controlador PD con Precompensación	94
5.5. Resumen	97
6. Diseño de Robots Submarinos con IVEP	99
6.1. Introducción	99
6.2. Análisis del Desempeño Cinetostático	99
6.2.1. Número de Condición	101
6.2.2. Homogeneización de la Matriz Jacobiana	103
6.2.3. Singularidades de la plataforma de Stewart-Gough	104
6.2.4. Espacio de Trabajo	105
6.2.5. Análisis del espacio de trabajo del robot Remo 2	107
6.2.5.1. Espacio de Trabajo de Traslación	107
6.2.5.2. Espacio de Trabajo de Orientación	109
6.3. Diseño de la Estructura del Robot	115
6.3.1. Índices de desempeño globales	116
6.3.2. Optimización del diseño	119
6.4. Resumen	124
7. Análisis y Simulación de Robots Submarinos con Múltiples Impulsores Vectorizados	127
7.1. Introducción	127
7.2. Múltiples impulsores vectorizados	127
7.3. Simulación	133
7.3.1. Simulador de robots submarinos	133
7.3.2. Simulación del Robot	138
7.4. Resumen	139
III Conclusiones	143
8. Conclusiones Generales	145

IV Apéndice	149
A. Análisis del Robot de 6 g.d.l. desacoplado	151
A.1. Análisis Cinemático	153
A.1.1. Cinemática Directa	154
A.1.1.1. Problema traslacional	154
A.1.1.2. Problema rotacional	155
A.1.2. Cinemática Inversa	158
A.1.3. Análisis de Velocidad	158
B. Librería de Simulación Dinámica	159
B.1. Simulación Basada en Penalizaciones	159
B.2. Geometrias consideradas	160
B.3. Descripción de la Librería	162
B.4. Paquete de Detección de Colisiones	162
B.4.0.1. Clase Gometry	162
B.5. Paquete de Fuerzas de Contacto	167
B.6. Paquete de Simulación	170
B.7. Uso de la Librería	173
C. Detección de Colisiones	179
C.1. Detección de colisiones	179
C.2. Calculo de la Distancia Mínima entre Elementos Geométricos	180

Índice de figuras

1.1. Remo I.	3
1.2. Remo II.	4
1.3. Operación del Remo II en diferentes configuraciones.	5
1.4. Remo 2.	8
2.1. Clasificación de los robots submarinos.	15
2.2. Sistemas de propulsión. a) Seaeye Panther-XT (Seaeye, 2012), b) REMO, c) Slocum Glider, d) Seaglider, e) Tuna Robot, f) AQUA.	21
2.3. Sistema de Control de Misión	25
3.1. Notación en robótica submarina.	30
3.2. Notación en robótica submarina.	31
3.3. Marcos de referencia en un robot sumarino.	36
4.1. Remo 2	45
4.2. Esquema de Remo 2.	46
4.3. Diagrama de cuerpo libre de las piernas de Remo II	50
4.4. Diagrama de cuerpo libre de la Plataforma 1	53
4.5. Diagrama de cuerpo libre de la Plataforma 2	53
4.6. Trayectoria en linea recta.	61
4.7. x_t , \dot{x}_t and \ddot{x}_t vs t	62
4.8. x_t , \dot{x}_t and \ddot{x}_t vs t	63
4.9. x_t , \dot{x}_t and \ddot{x}_t vs t	63
4.10. Trayectoria circular	64
4.11. θ_t , $\dot{\theta}_t$ and $\ddot{\theta}_t$ vs t	65
4.12. Position of leg 1	66
4.13. Velocidades de la pierna 1	66
4.14. Aceleraciones de la pierna 1	67
4.15. Fuerzas radiales en las articulaciones de la pierna 1	68
4.16. Torque radial en la articulación prismática de la pierna 1	69
4.17. Fuerzas en los impulsores 1 y 2	69
4.18. Fuerzas axiales	70
4.19. Torque generado por el CMG	70
4.20. Procesos concurrente para calcular la cinemática y dinámica inversa	71

5.1. Modelo en ADAMS.	76
5.2. Controlador P con retroalimentación de velocidad.	77
5.3. Fuerzas por masa añadida y por arrastre.	78
5.4. Comportamiento del robot empleando un controlador tipo proporcional.	78
5.5. Comportamiento del robot empleando un controlador P con retroalimentación de velocidad, $K_p = 50$	79
5.6. Fuerza en el primer impulsor, para $K_p=50$ y $K_v= 30$	79
5.7. Configuración para traslación pura lateral.	80
5.8. Respuesta del sistema para una referencia de 1 m, con ganancias $K_p = 50$ $K_v = 30$	81
5.9. Postura final del robot a) cuando se controla únicamente la posición de la plataforma 1, b) cuando se controla la posición de ambas plataformas.	82
5.10. Configuración para rotación pura.	82
5.11. Comportamiento del robot empleando un controlador tipo proporcional.	83
5.12. Comportamiento del robot para $K_p = 0,05$	83
5.13. Comportamiento del robot para $K_p = 10,0$	84
5.14. Comportamiento del robot para $K_v = 5,0$ y diversos valores de K_p	84
5.15. Controlador PID.	85
5.16. Comportamiento del robot empleando un controlador PID, $K_p = 50$ y $K_v = 30$	86
5.17. Fuerzas debido a la componente proporcional (f_{kp}) y a la componente integral (f_{ki}).	86
5.18. Movimiento lateral empleando un PID.	87
5.19. Movimiento rotacional empleando un PID.	88
5.20. Control articular de la estructura paralela.	89
5.21. Control articular P con retroalimentación de velocidad. a) Comportamiento de las variables articulares para diferentes ganancias proporcionales y $K_v = 10$; b) comportamiento de las variables articulares para diferentes ganancias derivativas y $K_p = 100$; c) movimiento global del robot cuando se varían las ganancias proporcionales; d) movimiento global del robot cuando se varían las ganancias integrales.	89
5.22. Controlador articular P con retroalimentación de velocidad, $K_p = 50$ y $K_v = 30$. a) Un impulsor y b) Dos impulsores.	90
5.23. Comportamiento de las variables articulares para el movimiento rotacional del robot. Se muestra como cada articulación llega a la posición deseada, la cual corresponde a la configuración del robot que permite un movimiento rotacional.	92
5.24. Controlador articular PID, $K_p = 50$, $K_v = 1$, $K_{pp} = 100$ y $K_{vp} = 100$	93
5.25. Comportamiento de las variables articulares cuando el robot se mueve a velocidad constante. a) Empleando con controlador P con retroalimentación de velocidad; b) empleando un controlador PID. Ganancias $K_{pp} = 100$ y $K_{vp} = 100$ y $K_{i\rho} = 50$	93
5.26. Controlador PD con Precompensación.	94

5.27. Trayectoria deseada. a) Posición y b) velocidad deseada de la plataforma 1 del robot; c) fuerza del impulsor 1 y d) fuerza en los actuadores lineales obtenidas a partir de la dinámica inversa.	95
5.28. Errores de posición y velocidad empleando un controlador PD.	96
5.29. Errores de posición y velocidad empleando un controlador PD con Precompensación.	96
6.1. Espacio de trabajo de traslación cuando ambas plataformas son paralelas.	107
6.2. Comportamiento de κ^{-1} cuando ambas plataformas son paralelas.	108
6.3. Espacio de trabajo para una orientación equivalente a un giro de 90° sobre el eje y	109
6.4. Comportamiento de κ^{-1} para una orientación equivalente a un giro de 90° sobre el eje y	110
6.5. Espacio de trabajo para una orientación equivalente a un giro de 90° sobre el eje x	110
6.6. Comportamiento de κ^{-1} para una orientación equivalente a un giro de 90° sobre el eje x	111
6.7. Volumen del espacio de trabajo para diversas alturas de la plataforma.	113
6.8. Comportamiento del GCI para diversas alturas de la plataforma.	113
6.9. Espacio de trabajo de orientación para $z = 0,55m$	114
6.10. Valores de κ para $z = 0,55m$	114
6.11. Robot de 6 grados de libertad con cinemática desacoplada.	116
6.12. Mapeo del índice μ para dos diseños diferentes de un manipulador.	118
6.13. Volumen del espacio de trabajo para diversos valores de los parámetros de diseño.	120
6.14. Máximo de κ^{-1} en el espacio de trabajo de diversos diseños.	120
6.15. Mínimo de κ^{-1} en el espacio de trabajo de diversos diseños.	121
6.16. GCI para diversos diseños.	121
6.17. Valor de $d(\kappa^{-1})$ para diversos diseños.	122
6.18. Valor de $R(\kappa^{-1})$ para diversos diseños.	123
6.19. Valor de $R(\kappa^{-1})$ para diversos diseños con $\alpha_2 = 100$	123
6.20. Sumatoria de κ^{-1} en el espacio de trabajo de diversos diseños.	124
7.1. Mecanismo paralelo Agile Eye.	128
7.2. Esquema de robot con dos impulsores vectorizados.	129
7.3. Notación del impulsor vectorizado.	129
7.4. Esquema de robot con tres impulsores vectorizados.	131
7.5. Esquema de robot con cuatro impulsores vectorizados.	133
7.6. Diagrama de clases del simulador MinPBS.	135
7.7. Primitivas geométricas del simulador empleadas para representar la geometría de robots submarinos, (<i>cube</i> , <i>sphere</i> y <i>stick</i>).	136
7.8. Diagrama de clases del simulador MinPBS.	137
7.9. Posición en el eje x , $t_1 = t_2 = 20N$	139
7.10. Velocidad en el eje x , $t_1 = t_2 = 20N$	140

7.11. Aceleración en el eje x , $t_1 = t_2 = 20N$	140
7.12. Rotación en el eje z , $t_1 = t_2 = 1N$	141
7.13. Velocidad angular en el eje z , $t_1 = t_2 = 1N$	141
A.1. Robot de 6 grados de libertad con cinemática desacoplada	152
A.2. Pierna interna	153
A.3. Articulaciones rotacionales en la plataforma superior	153
A.4. Diagrama esquemático del robot	154
A.5. Cadena cinemática interna de 3 grados de libertad.	155
A.6. Cadena cinemática externa de 3 grados de libertad.	156
B.1. Geometrías consideradas para la detección de colisiones	161
B.2. Diagrama de casos de uso de la librería MinPBS.	161
B.3. Diagrama de Clases del simulador MinPBS (repetida).	163
B.4. Paquete de Detección de Colisiones.	164
B.5. Diagrama de secuencia que se ejecuta cuando el usuario llama a la función <code>compute_collision</code>	168
B.6. Diagrama de secuencia de <code>contact_determination</code>	169
B.7. Paquete de cálculo de fuerzas de contacto.	170
B.8. Diagrama de secuencia de <code>compute_force</code>	171
B.9. Diagrama de Clases del paquete de simulación.	173
B.10. Diagrama de secuencia de la función <code>simulate</code>	175
B.11. Simulación de varios objetos cayendo desde una altura.	178
C.1. Tres tipos de regiones de Voronoi de un cubo tridimensional. c) región de una arista, b) región de un vértice c) región de una cara.	183
C.2. Condiciones en las diferentes regiones de Voronoi.	184
C.3. Distancia mínima entre un punto y un cubo.	186

Índice de cuadros

1.1. Movimientos instantaneos	7
2.1. Materiales para cabinas, extraído de (Wang et al., 2009)	17
3.1. Notación del movimiento de robots submarinos	30
4.1. Parámetros Geométricos del Robot	56
4.2. Parámetros Inerciales	60
4.3. Coeficientes de la Masa Añadida	60
4.4. Parámetros de Arrastre	60
6.1. Parámetros óptimos	122
7.1. Parámetros del robot simulado	138
B.1. Descripción de las clases	165
B.2. Descripción de las funciones de la clase Geomtry	166
B.3. Descripción de las funciones de la clase GeometryList	166
B.4. Descripción de las funciones de la clase ContactsQueue	167
B.5. Descripción de las funciones de la clase CollitionDetection	167
B.6. Descripción de las clases para el cálculo de fuerzas de contacto	170
B.7. Descripción de las clases	172
B.8. Descripción de las clases del Paquete de Simulación	172
B.9. Funciones de la clase RigidBody	174
B.10.Descripción de las funciones de MinPBS	174

Capítulo 1

Introducción

Como es bien sabido, cerca de dos terceras partes de la superficie de la tierra están cubiertas por agua. Los océanos permiten el transporte de productos y materia prima entre países, representan fuentes críticas de alimentos y otros recursos como los son el petróleo y el gas, y tienen un gran efecto en el clima y el medio ambiente.

El conocimiento científico de los mares profundos esta creciendo rápidamente mediante el uso de una variedad de tecnologías. Las primeras exploraciones científicas se llevaron a cabo mediante vehículos submarinos ocupados por humanos. Recientemente los robots submarinos han cambiado sustancialmente la exploración del fondo marino, ofreciendo mejor información a un menor costo. Por otro lado, estos robots han permitido realizar operaciones a bajas profundidades y además intervenir en desastres como son las fugas en instalaciones petrolíferas.

La historia de los robots submarinos comienza a principios de la década de los 50 del siglo pasado con la construcción de un robot submarino operado remotamente nombrado POODLE desarrollado por Dimitri Rebikoff en Francia. Desde ese entonces varios robots submarinos teleoperados y autónomos han sido desarrollados.

En las últimas décadas se han propuesto varias arquitecturas de robots submarinos. Algunos conceptos son los robots submarinos bioinspirados (Anderson and Chhabra, 2002; Dudek et al., 2007; Boyer et al., 2009) y los planeadores acuáticos (Webb et al., 2001; Zhang et al., 2006; Caffaz et al., 2010). Sin embargo, el sistema de propulsión de la vasta mayoría de los robots submarinos consiste de múltiples impulsores. Estas arquitecturas, varían principalmente en el número y disposición de los impulsores (Yuh, 2000), (Wang et al., 2009).

En años recientes, se han realizado algunos esfuerzos para equipar a robots submarinos con impulsores vectorizados (Le Page and Holappa, 2000), (Damus et al., 2002), (Morel and Leonessa, 2003). La impulsión vectorial se ha utilizado exitosamente para incrementar la maniobrabilidad de aviones. Por consecuencia, algunos investigadores creen que este método debe proveer estos mismos resultados en ambientes submarinos.

En (Cavallo and Micheline, 2004) y (Saltaren et al., 2007a) se presentó un nuevo concepto de diseño de robots submarinos que provee impulsión vectorial se ha propuesto: robots submarinos con impulsores vectorizados de estructura paralela. En

(Saltaren et al., 2007a) y (Alvarez, 2008a) se describe el primer desarrollo de un robot submarino con un impulsor vectorial de estructura paralela de 6 grados de libertad. El impulsor estaba unido a la cabina principal mediante un mecanismo paralelo de Stewart-Gough. Tal mecanismo permitía controlar la posición y orientación de la fuerza de impulsión con respecto al centro de masa del robot. Por lo tanto se podían realizar maniobras sofisticadas en 5 grados de libertad.

El propósito de esta tesis es realizar el primer estudio formal sobre el modelado, control y diseño de robots submarinos con impulsores vectorizados de estructura paralela. En las siguientes secciones se presentaran algunas definiciones y un estudio sobre las ventajas y desventajas que tienen esta clase de robots. Posteriormente se presentaran los objetivos y aportaciones de este trabajo, y finalmente la organización de la tesis.

1.1. Robots Submarinos con Impulsores Vectorizados de Estructura Paralela

En esta tesis consideramos la *impulsión vectorial* como *la capacidad de un robot submarino para manipular la dirección de un impulsor con el objetivo de controlar su propia posición y orientación*. En embarcaciones marinas una de las formas más básicas de impulsión vectorial son los impulsores acimutales. Por otro lado, en robótica submarina el uso de impulsores rotatorios ha probado ser útil, ejemplos de esto son los robots Sentry y Odyssey (Kinsey et al., 2011) (Desset et al., 2005).

Un impulsor vectorizado de estructura paralela (IVEP), consiste principalmente de dos elementos: un impulsor de hélice, y un mecanismo paralelo que controla la posición y orientación de este impulsor con respecto al cuerpo de la cabina.

En (Cavallo, 2003) se propuso el uso de un impulsor vectorizado de estructura paralela de dos grados de libertad para un robot submarino autónomo. Sin embargo, el primer desarrollo de un robot submarino con un IVEP fue el Remo I, construido en la Universidad Politécnica de Madrid. Remo I consistía de una cabina principal, una estructura paralela y un impulsor de hélice.

En la figura 1.1 se puede observar una imagen del robot Remo I. La cabina del robot constituye la parte delantera y aquí se situaban los circuitos electrónicos de medición, control, la fuente de alimentación y los amplificadores. El impulsor del robot se situaba en la parte trasera. La cabina y el impulsor estaban unidos a través una estructura tipo Stewart-Gough. Esta estructura está compuesta de 6 piernas con articulaciones tipo Universal-Prismática-Esférica, donde la articulación prismática esta actuada por un motor eléctrico.

Remo I tenía 170 mm de ancho y su largo podía variar entre 165 y 200 mm. Su peso era de aproximadamente de 90 kg en el aire, y en el agua se neutraliza por medio de unas bolsas de aire que habia en las aletas, las cuales también se utilizaban para su sustentación en el agua. El control de los actuadores, el impulsor y el lastrado se realizaba por medio de una palanca de mando.

La navegación de este vehículo se realizaba por medio del impulsor (que

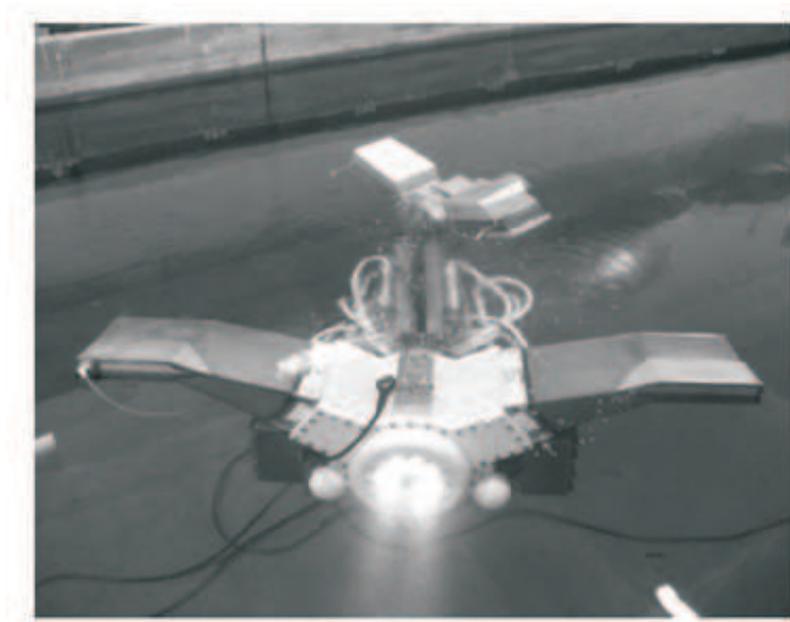


Figura 1.1: Remo I.

proporcionaba la fuerza de empuje del robot), y de los actuadores (que permitían orientar y desplazar la estructura); de esta manera era posible realizar giros, ascender o descender. Los actuadores tenían la capacidad de actuar de forma independiente; sin embargo, era necesario que los seis se muevan de manera coordinada, para obtener la orientación y desplazamiento requeridos entre la base y el efector final, y con ello lograr el movimiento deseado para la navegación.

Posteriormente, se desarrolló el robot Remo II (ver fig. 1.2). Al igual que el Remo I, la estructura del IVEP consistía de un mecanismo paralelo de Stewart-Gough.

El sistema de navegación de Remo II consistía de dos impulsores y un giróscopo de control de momento (CMG, por sus siglas en inglés). Los impulsores estaban ubicados en cada una de las plataformas respectivamente. Por lo tanto, el control de la posición y orientación de ambas plataformas permitían el control de una fuerza resultante en una gran variedad de direcciones.

El CMG estaba compuesto de cuatro giróscopos actuados con una configuración piramidal. El CMG era capaz de generar un momento resultante que podría los compensar los momentos generados por las fuerzas hidrodinámicas durante la navegación.

La estructura del mecanismo SG estaba hecha de aluminio, pesaba 15 kg y sus dimensiones eran 500x300x300 mm. La electrónica del robot estaba contenida en dos cabinas hechas de PVC que se encontraban en cada una de las plataformas. Estas cabinas tenían una placa de aluminio que permitía disipar el calor generado por los dispositivos electrónicos.

La figura 1.3 muestra la operación del robot en diferentes configuraciones de su estructura. La fig. 1.3a) ilustra a ambos impulsores alineados y permitiendo al robot

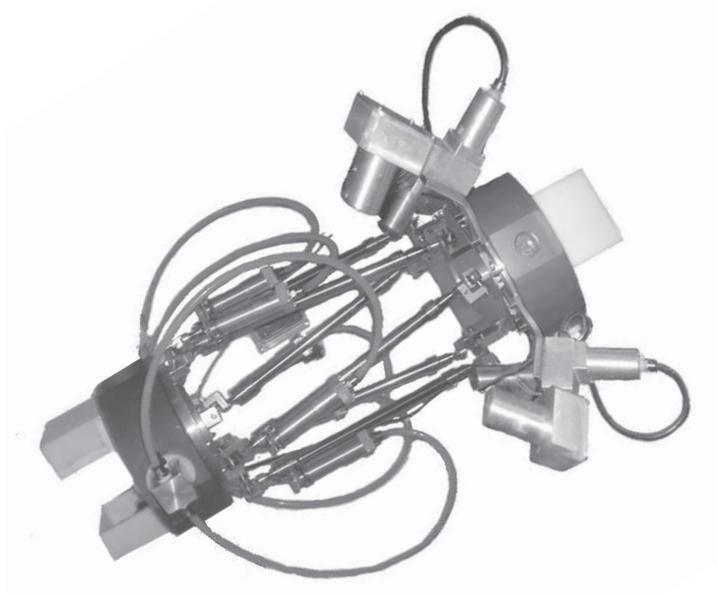


Figura 1.2: Remo II.

moverse en línea recta mientras se distribuye la fuerza requerida por ambos impulsores para moverse; la fig. 1.3b) muestra dos configuraciones del robot y a un impulsor genera una fuerza, lo que causa un movimiento lineal, y uno rotacional perpendicular al plano simultáneamente; la fig. 1.3c) describe la combinación de fuerzas en los impulsores que permiten un movimiento lateral del robot, debido a que los componentes cuya dirección es ortogonal a la dirección de movimiento se cancelan, y los componentes paralelos se suman; la fig. 1.3d) muestra una configuración del robot con el objetivo de hacer un movimiento vertical; la fig. 1.3e) muestra un movimiento general controlado por las fuerzas de impulsión y el torque generado por el CMG.

En base a los trabajos previamente presentados, se podría hacer una clasificación de los robots submarinos con IVEP de la siguiente manera:

Número grados de libertad. Una estructura paralela puede tener entre 2 y 6 grados de libertad. En este sentido, en un impulsor vectorizado se pueden controlar a diferentes niveles la posición y orientación relativa de la fuerza de empuje con respecto al centro de masa del robot. Generalmente, un AUV posee un mecanismo que permite cambiar la orientación de fuerza de empuje en dos ejes. Por otro lado, en el Remo I se demostró que cambiar la posición relativa puede ser útil, ya que esto permite variar el brazo de palanca de la fuerza del impulsor con respecto a la cabina del robot.

Tipo de movimiento relativo. Un mecanismo para un impulsor vectorial pudiera combinar diferentes tipos de movimiento o grados de libertad de diferente naturaleza. Por ejemplo, se podría tener un mecanismo paralelo de 4 grados de libertad, que controle la orientación en dos ejes y la posición en un plano de la

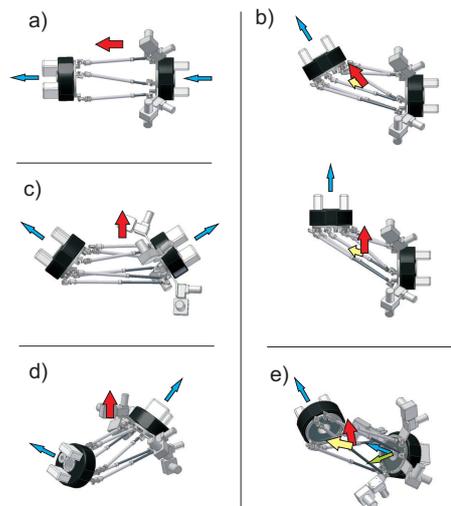


Figura 1.3: Operación del Remo II en diferentes configuraciones.

fuerza de impulsión con respecto al centro de masa. Diferentes combinaciones pudieran obtenerse dependiendo del tipo de movimientos deseados.

Relación de masa impulsor/cabina. La relación de masa (o tamaño) impulsor/cabina puede ser una de las características más importantes de un robot submarino con impulsores de este tipo. Se pueden considerar tres niveles de relación de masa impulsor/cabina: *baja*, *media*, o *alta*. Estos niveles tienen implicaciones tanto en el estudio de estos robots como en las prestaciones que la estructura paralela puede ofrecer. Un robot con una relación de masa baja indica que la estructura es pequeña con respecto a la masa principal del robot. En este caso la dinámica del robot es despreciable para cualquier análisis. Cuando la relación de masa es alta la dinámica de la estructura no puede ser despreciada, este es el caso del robot Remo II. En este caso la estructura paralela puede ser útil para realizar otras operaciones que unicamente asistir en la navegacion.

1.2. Fortalezas y Debilidades de los Robots Submarinos con Impulsores Vectorizados de Estructura Paralela

A continuación se realiza una discusión sobre las ventajas y desventajas que tienen estos robots con respecto a otras tecnologías de robots submarinos.

En submarinos de gran envergadura y AUVs se usa un sistema de propulsión basado en un impulsor y superficies de control. En este esquema se utilizan aletas con un solo grado de libertad para obtener los movimientos de cabeceo, viraje y balanceo. Este sistema presenta una gran simplicidad, pero la maniobrabilidad es muy reducida, especialmente a baja velocidad.

Un impulsor vectorizado además de proveer los movimientos previamente citados, es

capaz de realizar maniobras a baja velocidad, pues en cada momento se puede controlar la dirección de la fuerza de impulsión. Para orientar el impulsor se pueden emplear un mecanismo con dos actuadores rotacionales en serie.

Un mecanismo paralelo tiene por su parte varias ventajas sobre los mecanismos en serie. Estas son que tiene mayor rigidez y puede alcanzar mayores aceleraciones.

Se ha considerado que la principal característica que una estructura paralela aporta a un robot es su capacidad para deformar su estructura. En este sentido, un robot tipo Remo II puede aportar mayores ventajas, que un robot con una estructura que solo se utiliza para realizar un cambio de orientación o de posición.

Algunos atributos que hacen interesante la arquitectura son los siguientes:

- Debido a la capacidad de Remo II para controlar la orientación de los impulsores se pueden realizar maniobras a baja velocidad en múltiples grados de libertad, lo cual permite operaciones en aéreas confinadas.
- La rigidez estructural del mecanismo paralelo de SG permite que se puedan utilizar actuadores prismáticos pequeños y ligeros, y que estos puedan soportar las fuerzas hidrodinámicas que actúan en el robot durante el movimiento del mismo. Además, debido a que la estructura es ligera, las fuerzas inerciales que se generan durante el movimiento son relativamente pequeñas.
- El espacio entre piernas de la estructura paralela es amplio, lo cual permite que un amplio flujo de agua alimente a los impulsores.
- El robot puede generar movimientos en 6 g.d.l. con solo dos impulsores de alta potencia y seis actuadores prismáticos de baja potencia.
- El robot puede tener dispositivos de agarre en cada plataforma. Estos dispositivos de sujeción pueden permitir al robot trepar en tuberías o estructuras robóticas submarinas. Además, si solo una de las plataformas estuviera sujeta, la estructura puede controlar la posición y orientación de la otra plataforma, esto puede ser útil cuando se realizan operaciones en instalaciones submarinas.

Comparar el Remo II con otras arquitecturas es una tarea difícil, debido a que el diseño y principio de funcionamiento es radicalmente diferente. Sin embargo, a continuación se realiza una breve discusión para aclarar las capacidades de este robot. Para hacer esto, se comparan los movimientos que puede generar el robot Remo II y otras arquitecturas. Las arquitecturas elegidas para esta comparación se encuentran regularmente en el diseño de robots submarinos. Estas arquitecturas con las siguientes:

- Tipo I, un robot con un impulsor vectorial de dos grados de libertad.
 - Tipo II, un robot con 4 impulsores y estabilidad en roll pitch.
 - Tipo III, un robot con 6 impulsores y 6 grados de libertad.
-

Tabla 1.1: Movimientos instantaneos

Arquitectura	Movimientos puros						Movimientos combinados				
	Rotaciones			Traslaciones			2 dof	3 dof	4 dof	5 dof	6 dof
	X	Y	Z	X	Y	Z					
Tipo I				+							
Tipo II			+	+	+	+	+	+	+		
Tipo III	+	+	+	+	+	+	+	+	+	+	+
Remo I				+							
Remo II - CMG				+							
Remo II + CMG	+	+	+	+	+	+	+	+	+	+	

Adicionalmente, se incluye al Remo I. Consideramos un movimiento instantáneo puede ser realizado cuando el robot es capaz de generar fuerzas para acelerar el robot en ciertos grados de libertad. Se consideran movimientos puros y movimientos combinados. Los movimientos puros permiten moverse y rotar alrededor de un sólo eje. Estos tipos de movimientos son importante en maniobras en espacios confinados o donde un movimiento combinado no se puede realizar. Por ejemplo, un robot con una arquitectura del Tipo I con un impulsor vectorizado en la proa no puede realizar una rotación pura de 180° alrededor del eje z . Para realizar un cambio de orientación el robot tendría que generar un momento, el cual se produce por la fuerza del impulsor y el brazo de palanca. Sin embargo, esta fuerza también generara una aceleración lineal la cual a su vez hará que el robot se traslade. La tabla muestra los movimientos instantáneos que las diferentes arquitecturas pueden realizar. Las celdas de color gris indican los tipos de movimientos que las arquitecturas pueden realizar. Por otro lado, el símbolo + en la celda, indica que los movimientos se pueden realizar sin una reconfiguración del robot, i.e. en el caso de robots con impulsores vectoriales estos no requieren un cambio en la posición y orientación de los impulsores.

La arquitectura Tipo I puede genera solamente un movimiento puro en el eje x (cuando el impulsor está alineado con el centro de masa) y movimientos combinados en 2 g.d.l. (una aceleración en la dirección de la fuerza y una aceleración angular perpendicular a esta fuerza) y 3 g.d.l. Esta arquitectura es útil en misiones de exploración, sin embargo es insuficiente para realizar maniobras complejas en 6 g.d.l. Por otro lado, la arquitectura tipo II puede realizar movimientos en 4 g.d.l.

Puede realizar movimientos lineales en tres direcciones y rotar en roll, sin embargo debido a que esta arquitectura tiene estabilidad en roll-pitch, esta no puede realizar rotaciones en esos ejes.

La arquitectura Tipo III tiene 6 impulsores que pueden generar movimientos en 6 g.d.l. Por otro lado, Remo I puede generar movimientos instantáneos en 6 g.d.l., debido a que puede controlar la magnitud, posición y orientación de un impulsor con respecto al centro de masa de del robot. Sin embargo, solo puede generar un movimiento puro

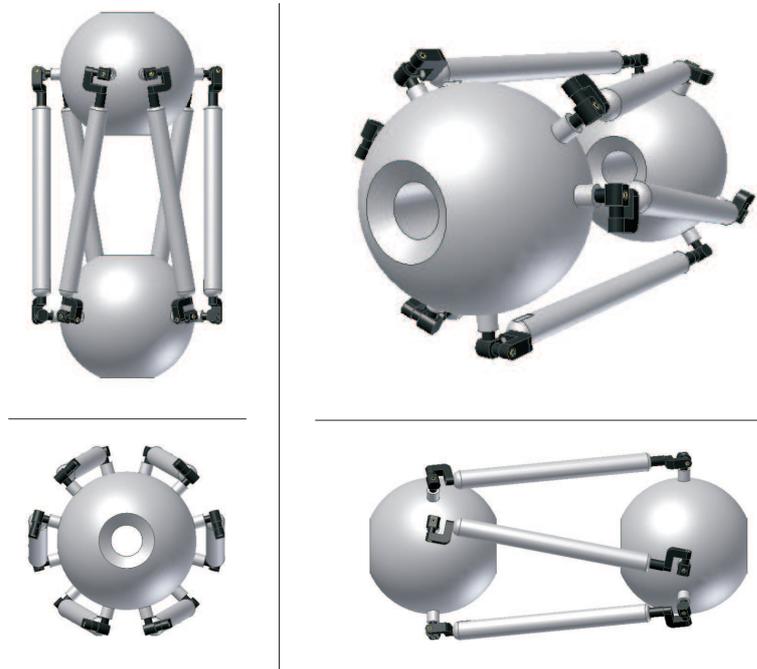


Figura 1.4: Remo 2.

en el eje x , y no puede realizar rotaciones puras.

Remo II puede realizar traslaciones puras, rotaciones puras, y movimientos combinados en 6 g.d.l., mediante la reconfiguración de su estructura. Cuando Remo II usa el CMG este robots puede relizar cambios en su movimiento en 5 g.d.l. sin la reconfiguración de su estructura, bajo la condición de que ambas platafomras no sean paralelas.

La habilidad de Remo II para realizar movimientos en 6 g.d.l. es superior a la de las arquitecturas Tipo I y Tipo 2. Por otro lado, la principal diferencia entre Remo II y Remo I, es la capacidad del primero para realizar movimientos puros. Esta habilidad da al Remo II una gran maniobrabilidad en espacios pequeños.

Mientras que la arquitectura Tipo III consiste de 6 impulsores de gran potencia, Remo II consiste de 2 impulsores de alta potencia y 6 actuadores lineales de baja potencia. Se puede demostrar que ambas alternativas son similares económicamente.

1.3. Objetivos

El diseño del robot Remo II es un concepto radicalmente diferente al de los robots submarinos convencionales. Como se explicó en la sección anterior, las características del diseño son prometedoras, pero sacar provecho de estas potencialidades requiere un entendimiento de la dinámica del robot. El objetivo central de esta tesis es el desarrollo y análisis de modelos analíticos, y el desarrollo de herramientas de simulación del robot Remo II. Estas herramientas a su vez deben ser útiles para identificar las oportunidades

y restricciones impuestas por la dinámica del vehículo.

De manera concisa los objetivos planteados originalmente para la presente tesis de doctorado son:

1. Desarrollar herramientas de simulación para robots submarinos tipo Remo II.
2. Desarrollar algoritmos para el cálculo en múltiples procesadores de la dinámica inversa de robots submarinos de estructura paralela.
3. Analizar y proponer controladores basados en el modelo para un robot tipo Remo II.
4. Demostrar la factibilidad del concepto de un robot tipo Remo II para realizar diversas maniobras de interés.

Estos objetivos fueron los que se persiguieron en esta tesis. Un cambio que se realizó durante el desarrollo de la tesis fue que en lugar de estudiar la dinámica del robot Remo II, se eligió como arquitectura a estudiar un robot con una arquitectura similar, pero con ciertas características hidrodinámicas especiales, de entre ellas la que mayor influencia tiene es que la flotabilidad de sus elementos es neutra. La geometría de este robot de estudio es similar a la de Remo II (radio de las plataformas y distribución de las articulaciones esféricas y universales). Para diferenciar la arquitectura del robot que se estudia en esta tesis, se le nombra como Remo 2, ver fig. 4.1.

1.4. Principales Aportaciones

Del trabajo realizado en esta tesis se consideran como contribuciones:

1. El planteamiento del modelado cinemático y dinámico inverso de un robot submarino de estructura es paralela.
2. Una estrategia para el cálculo paralelizado de la dinámica inversa.
3. Un estudio sobre el diseño de robots submarinos de estructura paralela.
4. Se plantean controladores para estos robots y se realiza un estudio de su comportamiento.
5. Un análisis sobre el empleo de múltiples impulsores vectorizados en robots submarinos.
6. Se desarrolló una librería en C++ para la simulación de robots submarinos, que incluye las funciones de detección de colisiones y fuerzas de contacto.

Por otro lado, el autor de esta tesis fue coautor del diseño de una nueva estructura paralela cuya topología no ha sido reportada anteriormente en la literatura.

Producto de labor del autor se realizaron 23 publicaciones científicas: 4 en revistas con factor de impacto (JCR), 16 en memorias de congresos arbitrados y 3 capítulos de libro.

1.5. Organización de la Tesis

Esta tesis está dividida en cuatro partes, la primera presenta conceptos fundamentales sobre robots submarinos, en la segunda se presenta el cuerpo principal de este trabajo, posteriormente se establecen las conclusiones, y en la cuarta parte se presentan como apéndices desarrollos que se realizaron complementarios a este trabajo.

En el Capítulo 2, se hace una revisión del estado de la técnica de la robótica submarina. Se describen las tecnologías actuales para determinar el estado del robot (sensores), los sistemas de propulsión (actuadores), y las fuentes de alimentación. Además, se presenta una descripción de las estrategias que se han propuesto para controlar estos dispositivos.

En el Capítulo 3, se presentan los fundamentos matemáticos sobre el modelado de robots submarinos. Brevemente se explican las convenciones para describir el movimiento de estos robots. En la sección de modelado dinámico se presenta de manera concisa la derivación de las ecuaciones de movimiento con el objetivo esclarecer su forma final.

En el Capítulo 4, se presenta la derivación de las ecuaciones de movimiento de la estructura de robot que se ha elegido en esta tesis para estudiar. Primero se presenta el modelado cinemático inverso y posteriormente el modelado dinámico inverso. Se analizan diversos casos de estudio sobre el movimiento del robot.

En el Capítulo 5, se presenta un estudio del espacio de trabajo del robot y se realiza un ejercicio de síntesis dimensional dadas las características de las tareas para cuyo propósito es propuesto este robot. En esta sección se incluyen diversos estudios del desempeño (cinetostático y dinámico) del robot bajo diversas características de su diseño.

En el Capítulo 6, se presentan algunas estrategias de control del robot submarino. Por un lado se analiza el desempeño del robot empleando diversas leyes de control y por otro lado mediante la simulación del robot es posible analizar su dinámica en el ambiente submarino.

En el Capítulo 7, se propone y se analiza el diseño de un robot submarino con un par de impulsores vectorizados. Se presentan los resultados de diversas simulaciones realizadas para explorar el comportamiento.

En el Capítulo 8, se presenta una discusión sobre los resultados obtenidos en esta tesis, estableciendo las principales conclusiones y posteriormente se proponen futuras líneas de investigación.

En los apéndices se presenta el análisis de un robot desacoplado de 6 grados de libertad, el diseño de la librería de simulación y fundamentos de detección de colisiones entre primitivas geométricas.

Parte I

Fundamentos

Capítulo 2

Robótica Submarina: Generalidades

2.1. Introducción

Un robot submarino está diseñado para realizar tareas bajo el agua, las cuales pueden ser realizadas mientras navega o al llegar a un lugar prefijado mediante algún tipo de manipulador. Desde este punto de vista, los robots submarinos pueden hacer dos tipos de misiones:

Misiones de Inspección. Son aquellas misiones que se realizan durante la navegación del robot submarino. En este tipo de tareas no se requiere un brazo manipulador, ni mecanismos para interactuar con el ambiente. Una misión de inspección consistiría en recoger imágenes con una o varias cámaras mientras el robot navega en el agua. Por otro lado, la observación del lecho marino mediante la obtención de la cartografía acústica o en la obtención de datos relativos a la calidad del agua donde navega. Además, dentro de las misiones de inspección se encuentra la revisión de instalaciones submarinas (como lo son las estructuras metálicas, tuberías, cables, etc.).

Misiones de Manipulación. Son aquellas misiones en las que el robot submarino interviene brazos manipuladores y/o herramientas. Para el desarrollo de estas misiones se deberá contar con un sistema de visión en tiempo real (en el caso de ROVs), que proporciona al operador las imágenes en directo del entorno de operación. Las tareas típicas de manipulación comprenden: el mantenimiento de estructuras subacuáticas; la apertura y cierre de válvulas en instalaciones subacuáticas; la desactivación de minas; el ensamble y desensamble de componentes; la recolección de muestras para estudios arqueológicos, geológicos o ecológicos; la intervención en desastres para controlar las fugas de material contaminante o ayudar en el rescate de personas.

En este capítulo se presentarán algunos conceptos elementales de robótica submarina. Se realizará una clasificación de estos robots y se describirán sus componentes principales. Durante el desarrollo de la tesis se hará referencia a estos conceptos por lo

cual es útil describirlos con antelación para que el lector los conozca. El contenido de este capítulo se ha extraído de diversas fuentes, resaltando principalmente (Antonelli et al., 2008), (Wang et al., 2009), (Fossen, 2002), y (Alvarez, 2008b).

2.2. Clasificación de los Robots Submarinos

En la fig. 2.1 se presenta una clasificación de los robots submarinos. Los robots submarinos se clasifican principalmente por su nivel de autonomía. Existen robots completamente autónomos y otros que deben ser controlados continuamente por un operador, estos son los AUVs y los ROVs, respectivamente. Por otro lado, están los IAUVs que pueden considerarse actualmente en un nivel intermedio de autonomía, ya que los prototipos desarrollados hasta ahora requieren de un Control Supervisado o de un operador que realice la tarea de razonamiento para determinar las acciones que se deben llevar a cabo para completar la misión. No obstante, el objetivo final de estos robots es que lleguen a ser completamente autónomos y solo sea el operador el que al inicio defina la misión a realizar mediante comandos de alto nivel.

Estos robots se pueden clasificar por el tipo de misión a realizar. Las misiones pueden ser de inspección (u observación) o de manipulación (o intervención). La principal diferencia entre un robot diseñado para realizar intervenciones y otro que solo realiza inspecciones, es que el primero debe poseer herramientas y/o un brazo robótico. La misión para la que está diseñado un robot submarino definirá el tipo de sensores, actuadores, estructura que deberá poseer.

El sistema de propulsión de un robot submarino define completamente el tipo de movimientos y maniobras que éste puede realizar. Los sistemas de propulsión también tienen implicaciones en el consumo de energía, el hardware del robot, y el efecto que el robot genera en el ambiente marino. Encontramos que los principales sistemas de propulsión son los Impulsores de Hélice, los Planeadores Acuáticos (basados en Sistemas de Lastre y Aletas), y los Bioinspirados. En fase experimental o cuya tecnología no se ha madurado completamente están los basados en Inyección de Agua y los Magnetohidrodinámicos. En la siguiente sección describiremos en mayor detalle éstos sistemas de propulsión.

2.2.1. Robots Submarinos Operados Remotamente, ROVs

Los robots submarinos operados remotamente (ROVs, por las siglas en inglés de *Remotely Operated Vehicle*) están conectados a la superficie mediante un cordón umbilical (un conjunto de cables unidos), el cual permite el intercambio de datos y la alimentación de energía al robot. A través de una interface gráfica del ordenador situado en la superficie, el usuario define los comandos que el robot deberá ejecutar. A su vez el ROV envía las señales de sus sensores (presión, temperatura, imágenes, etc.) al ordenador en superficie, para que el usuario conozca el estado del robot y el ambiente que lo rodea.

Actualmente a las instalaciones petroleras o de gas se les presta servicio mediante ROVs. Las exigencias del trabajo realizado en estructuras submarinas son altas, ya

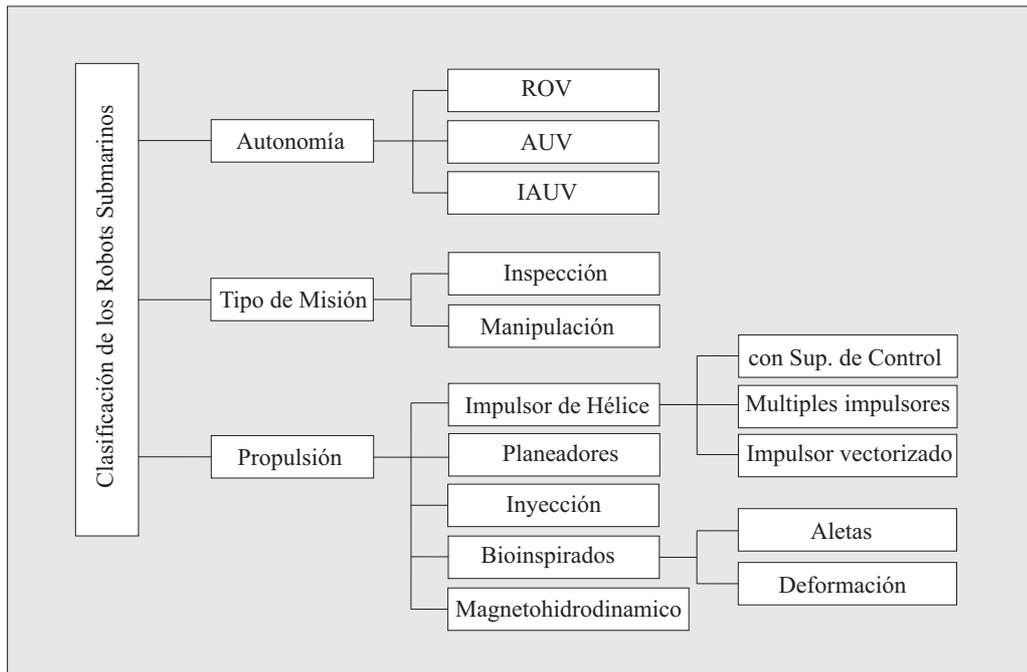


Figura 2.1: Clasificación de los robots submarinos.

que requieren de inspección frecuente e intervención para realizar las operaciones de perforación, manipular válvulas, reparar o reemplazar componentes submarinos, y realizar una variedad de tareas requeridas para mantener el ritmo de producción y la calidad del producto. La tendencia en el uso de ROVs aumentará a medida que la producción de gas y petróleo en instalaciones cerca de la costa se traslade hacia aguas profundas.

Las tareas de este tipo de robots se vuelven más complejas cuando se realizan a mayores profundidades. Debido al aumento de la cantidad de datos y señales de control a transferir a través del cordón umbilical, los cables han aumentado su diámetro, lo cual hace que el vehículo sea menos manejable ya que las fuerzas de arrastre del agua son mayores. No obstante, se han presentado algunas soluciones como construir un Sistema de Manejo de Cable (TMS, por las siglas en inglés de *Tether Management System*) el cual se ancla en el fondo marino y soporta las fuerzas de arrastre del cable de mayor distancia (el que va desde la embarcación en la superficie al TMS) y a la vez permite que el ROV navegue con más facilidad.

2.2.2. Robots Submarinos Autónomos, AUVs

Los robots submarinos autónomos (AUVs, por las siglas en inglés de *Autonomous Underwater Vehicle*) poseen una arquitectura de control que les permite realizar misiones sin la supervisión de un operador. Además contienen su propia fuente de energía generalmente basada en baterías recargables.

Por lo general no hay una línea de comunicación entre el vehículo y la superficie, ya que se le suele programar con tareas y misiones predefinidas. No obstante, cuando

se requiere un intercambio de información con la superficie la comunicación se puede realizar a través de dispositivos acústicos.

Estos robots pueden resolver las limitaciones impuestas por los cables de los ROVs para algunas tareas. Los AUVs se utilizan actualmente para tareas de exploración científica, muestreo oceanográfico, arqueología submarina y exploración debajo del hielo. Los datos que recopila el vehículo se almacenan en su memoria interna para luego ser analizados.

Por otro lado, también se han utilizado para operaciones militares, como lo son la detección de minas, y se están desarrollando aplicaciones más elaboradas como lo son vigilancia submarina.

Aproximadamente 200 AUVs están en operación muchos de ellos experimentalmente. Sin embargo, esta tecnología está madurando rápidamente y algunas compañías ya ofrecen servicios con este tipo de robots (Antonelli et al., 2008).

2.2.3. Robots Submarinos Autónomos para Intervenciones, IAUVs

Los AUVs han sido diseñados para realizar misiones de observación, sin embargo, recientemente se ha generado un interés en que estos puedan realizar tareas de manipulación. Esta es la idea detrás de los robots submarinos autónomos para intervención (IAUVs, por sus siglas en ingles). Con este tipo de robots las misiones serían mas económicas que en el caso de los ROVs, además su maniobrabilidad sería superior ya que el IAUV estaría libre de las restricciones que impone el cordón umbilical.

Ejemplos de desarrollos este tipo de robots son los proyectos ALIVE (Evan et al., 2003), SAUVIM (Marani et al., 2009) y RAUVI (DeNovi et al., 2010). El vehículo ALIVE navega autónomamente hasta el lugar donde se realizará la intervención, una vez que llega a la posición deseada cambia su operación a Control Supervisado y mediante comunicación acústica realiza las tareas de manipulación. Por su parte, el robot GIRONA 500 (del proyecto RAUVI) explora primero la región de interés y toma información acústica y visual del fondo. Posteriormente el robot sube a la superficie donde la información recolectada es procesada para realizar una reconstrucción de la región explorada. A través de una interface gráfica de usuario un operador identifica el objeto de interés y da los comandos al robot para que realice la intervención. Hasta el momento se han realizado exitosamente pruebas experimentales en un escenario en el cual la misión del robot es recuperar una caja negra de avión (Prats et al., 2012).

2.3. Estructura de un robot submarino

La cabina o casco (en inglés *Hull*) consiste de la estructura y el material que reviste al robot submarino. En la cabina están montados y/o contenidos, el ordenador abordo, el equipo eléctrico y electrónico (motores eléctricos, controladores, convertidores de señales, sensores, tarjetas electronicas, etc), y los elementos mecánicos del robot (como lo son el sistema de lastre, las aletas y los impulsores).

Los factores que determinan el diseño de una cabina son: la profundidad a la que va operar (y por lo tanto la presión hidrostática que va a soportar); los niveles de

Tabla 2.1: Materiales para cabinas, extraído de (Wang et al., 2009)

Material	Densidad (kg/dm ³)	Resistencia a la fluencia (MPa)	Modulo de tensión (GPa)	Resistencia Es- pecífica (kNm/kg)
Acero alta resistencia (HY80)	7.86	550	207	70
Aleación de Aluminio (7075)	2.9	503	70	173
Aleación de Titanio (6-4 STOA)	4.5	830	120	184
CFRP (Epoxy/S-lass)	2.1	1200	65	571
GFRP (Epoxy/HS)	1.7	1200	210	706
Acrílico	1.2	103	3.1	86
PVC	1.4	48	35	34

temperatura que va resistir; la resistencia a la corrosión; el volumen de los elementos que contendrá; la facilidad para ensamblar/desensamblar; la factibilidad de su construcción y el costo.

Por otro lado, otro factor que se debe considerar son las fuerzas de arrastre que se generan por el movimiento del robot. Estas fuerzas son proporcionales al cuadrado de la velocidad del robot. Cuando el robot se mueve a velocidad constante, la fuerza generada por el impulsor es igual a las fuerzas de arrastre. Por lo tanto, a menores fuerzas de arrastre menor es la energía consumida durante la navegación del robot. Las fuerzas de arrastre dependen en gran medida de la geometría de la cabina.

En el diseño de AUVs, una cabina con forma cilíndrica es una elección muy popular. Esto se debe a que (para un determinado volumen del robot) las fuerzas de arrastre que se generan son pequeñas cuando se comparan con otras geometrías. Por otro lado, esta forma geométrica es adecuada para resistir la presión hidrostática generada a grandes profundidades (Ross, 2006). Además, la manufactura de una cabina con esta geometría es relativamente simple y esto a su vez tiene consecuencias favorables en el costo.

Por otro lado, una parte elemental del diseño de la cabina es la forma del extremo frontal. Un extremo frontal plano no es aconsejable para un robot que realizará tareas altas velocidades, ya que en las aristas de la cara frontal se lleva a cabo el fenómeno llamado cavitación¹ que puede provocar la erosión o ruptura del material de la cabina. En (Paster, 1986) se aconseja el uso de un extremo ovalado para reducir la cavitación y las fuerzas de arrastre en el robot.

Otro aspecto importante del diseño de la cabina es el tipo de material. En (Wang et al., 2009) se presenta una discusión sobre los diferentes tipos de materiales mayormente utilizados en la construcción de robots submarinos. Estos materiales son

¹La cavitación es un cambio de fase muy rápido de líquido a vapor que se genera cuando el agua pasa a alta velocidad por una arista afilada del cuerpo del vehículo. Entonces se generan burbujas las implotan cuando son transportadas a regiones de mayor presión y esto puede provocar un arranque del material de la superficie en la que se genera este fenómeno (Potter et al., 1998).

principalmente metálicos, compuestos y plásticos. En la tabla 2.1 se presentan las propiedades de algunos materiales utilizados en la construcción de cabinas.

2.4. Sensores

Los robots submarinos están equipados con sistemas de sensores dedicados a determinar la localización del vehículo, su velocidad y aceleración, así como su estado de funcionamiento. Los sistemas de visión entran en esta categoría de dispositivos cuando son utilizados para determinar la posición y orientación de objetos con respecto al vehículo. Los sensores permiten controlar los movimientos del robot lo cual a su vez permite que éste pueda realizar la misión que le fue comandada. Podemos clasificarlos como:

Propioceptivos Los sensores propioceptivos son aquellos que permiten determinar la posición, orientación, velocidad y aceleración del robot submarino con respecto a un marco de referencia inercial o con respecto a otros objetos de interés. A continuación se presentan algunos de los sensores que utilizan la mayoría de los robots submarinos. Algunos de estos sensores son: el GPS, la IMU, la Brújula de estado sólido, el Velocímetro Doppler de Navegación, el Sensor de profundidad, los Sistemas de Posicionamiento Acústico, etc.

Estado interno Estos sensores permiten determinar el consumo de energía del robot, una posible inundación, la temperatura interna del vehículo, entre otras variables. Algunos de estos sensores son por ejemplo: Los voltímetros y amperímetros, para medir el consumo eléctrico, y los detectores de inundación.

Medición del ambiente Los sensores de Medición del ambiente son aquellos que se utilizan para medir las características del agua que los rodea, algunos de éstos miden las siguientes variables: conductividad, pH, densidad, turbidez, oxígeno disuelto, temperatura, presencia de metales, etc.

2.5. Sistemas de propulsión

Los sistemas de propulsión son los dispositivos que permiten que los robots submarinos avancen en el medio acuático. Aquí consideramos como parte del sistema de propulsión, tanto los elementos que generan las fuerzas, como aquellos que controlan la dirección de estas fuerzas.

A pesar de que la mayoría de los sistemas de propulsión de robots submarinos consisten de impulsores de hélice, aletas móviles para el guiado, y sistemas de lastre para realizar las operaciones de viraje, se han propuesto otras formas de generar movimiento dentro del agua, a continuación se describen algunas de éstas.

2.5.1. Impulsores de Hélice

El sistema de empuje con hélice y motor es el más utilizado en la mayoría de robots submarinos. Consiste generalmente en un motor eléctrico, al cual se le acopla una hélice. Al girar la hélice produce un efecto de empuje al desplazar el fluido de adelante hacia atrás, debido a la diferencia de presión producida.

Los modelos matemáticos de impulsores que se han propuesto, son solo una aproximación para describir su comportamiento, ya que influyen muchos factores tales como: la forma, el diámetro y el área de las aspas; el área del ducto; la velocidad de giro de la hélice; las corrientes en el ambiente; la densidad y viscosidad del agua, etc. Diferentes modelos se han presentado en (Yoerger et al., 1990) (Bachmayer et al., 2000).

Frecuentemente, se utiliza un modelo matemático sencillo del impulsor en el cual la fuerza del impulsor es proporcional al cuadrado de la velocidad angular de la hélice, y a su vez esta velocidad es proporcional al voltaje del motor. Lo anterior bajo el supuesto que la dinámica de los impulsores tienen unas constantes de tiempo mucho menores que la dinámica de los vehículos (Yuh, 2000).

Las prestaciones que ofrece el impulsor de hélice han hecho de este sistema de propulsión el más utilizado en robots submarinos, sin embargo estos dispositivos producen una gran cantidad de ruido en el ambiente marino, lo cual altera en cierta medida la vida acuática (Low and Willy, 2005), es por ello que en el desarrollo de ciertos robots submarinos se proponen otros sistemas de propulsión.

Impulsor y superficies de control. Este tipo de sistema de propulsión es el que más se usa en submarinos de gran envergadura y AUVs. Utilizan aletas con un solo grado de libertad para obtener los movimientos de cabeceo, viraje y balanceo. Este sistema presenta una gran simplicidad, pero la maniobrabilidad es muy reducida, especialmente a baja velocidad.

Múltiples impulsores. Es el sistema más utilizado en los ROVs. Consiste en colocar número determinado de impulsores de manera de darle de maniobrabilidad en los grados de libertad requeridos. A pesar de que los impulsores de mayor importancia para ejecutar la navegación se colocan en la parte trasera o a los lados del robot submarino, no hay una regla específica para la ubicación y orientación de ellos. Una configuración que puede dotar de seis grados de libertad a un vehículo submarino, es la que posee el robot ODIN II (Yuh et al., 1999).

Esta disposición de los impulsores permite al robot girar sobre sí mismo. Se utiliza sobretodo en trabajos de manipulación e inspección por su precisión en el posicionamiento y su capacidad para gobernar los grados de libertad independientemente. Los robots submarinos que utilizan el sistema de arreglo de impulsores suelen tener forma cúbica o esférica y compacta. El inconveniente que presenta esta configuración es el alto consumo de energía debido a la gran cantidad de impulsores.

Impulsor vectorial. La impulsión vectorial es la capacidad de un robot submarino para orientar o posicionar un impulsor con el objetivo de controlar su propio movimiento. En los robots submarinos con impulsor vectorial, el propulsor se coloca en la parte trasera, de forma similar a la de los vehículos con un impulsor fijo y aletas de guiado, pero en este caso el impulsor no está fijo, ya que tiene la capacidad de

orientarse.

El desarrollo de robots submarinos con impulsores vectoriales es relativamente reciente. En algunos trabajos de investigación se ha analizado la dinámica de este tipo de vehículos y se ha encontrado que presenta grandes ventajas para la navegación y guiado con precisión (Le Page and Holappa, 2000), (Morel and Leonessa, 2003). En (Cavallo and Micheline, 2004) se presentó un impulsor vectorial situado en la parte de trasera que mediante una estructura paralela esférica permite los movimientos de cabeceo y viraje .

Por otro lado, en (Kinsey et al., 2011) y (Desset et al., 2005) se presentaron los AUVs SENTRY y Odyssey IV, respectivamente. Estos robots poseen impulsores rotatorios que permitían controlar sus movimientos de avance y viraje simultáneamente. Estos robots han sido utilizados exitosamente para realizar exploraciones del fondo marino.

En (Saltaren et al., 2007a) y (Alvarez, 2008b) se presentó el diseño de robot submarino cuyo impulsor estaba unido a la cabina a través de una plataforma de Stewart-Gough. La ventaja que presenta este robot, llamado Remo I, es que utiliza un solo impulsor, lo cual representa un menor gasto de energía en comparación con los robots de varios impulsores. El robot puede controlar la posición y orientación de la fuerza de impulsión (vea fig. 2.2b). Además, por el hecho de poder deformarse, este robot tiene la potencial capacidad de nadar.

En (Yime, 2008) y (Moreno et al., 2011) se presentó el diseño y modelado del robot Remo II. El sistema de propulsión de este robot es mejor que el de su predecesor. Tiene dos impulsores, uno en cada plataforma, y un Giróscopo de Control de Momento. La capacidad de este robot para realizar trayectorias en un plano es superior a la de Remo I, además puede realizar movimientos puros de rotación y traslación.

2.5.2. Planeador Acuático (Glider)

El concepto de planeador acuático fue presentado en (Stommel, 1989), en este artículo se describen las misiones del vehículo submarino Slocum, el cual se valía de los gradientes térmicos entre las aguas profundas del océano y la superficie para realizar movimientos cíclicos de inmersión - emersión. Posteriormente Stommel desarrolló un prototipo de robot planeador submarino.

Los robots submarinos planeadores no disponen de impulsor ni aleta móvil. Éstos han sido diseñados para deslizarse desde la superficie del mar hasta una profundidad programada, luego cambian su flotabilidad y cabeceo para ir en dirección ascendente hasta llegar a un punto prefijado donde vuelven a descender y así sucesivamente. El avance se obtiene por medio de la inclinación de las aletas, la cual se consigue al combinar pequeñas variaciones de la posición y magnitud de la fuerza de flotabilidad. Como resultado se obtiene un movimiento de desplazamiento diagonal con un mínimo consumo de energía. Mientras se realiza el recorrido, recogen información de temperatura, salinidad, corrientes y otras medidas a lo largo de su trayectoria (Davis and Jones, 2002), (Graver, 2005), (Zhang et al., 2006), (Caffaz et al., 2010).

A pesar de que la velocidad de estos robots es bastante reducida y el avance está inevitablemente ligado al movimiento vertical, su estructura es ideal para los

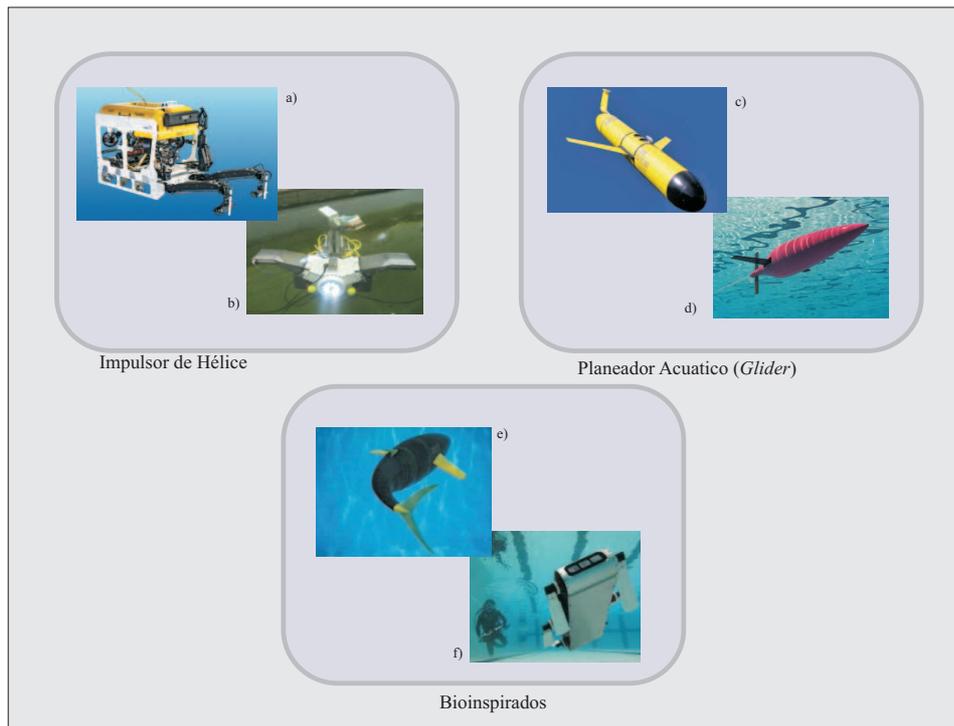


Figura 2.2: Sistemas de propulsión. a) Seaeye Panther-XT (Seaeeye, 2012), b) REMO, c) Slocum Glider, d) Seaglider, e) Tuna Robot, f) AQUA.

robots AUV dedicados a la observación y medición oceanográfica. El mínimo consumo incrementa el tiempo de las misiones a varios meses y aumenta el campo de acción a cientos de kilómetros, lo cual reduce en gran medida los costes de seguimiento.

En agosto de 2006, la marina de los Estados Unidos puso en funcionamiento un planeador submarino llamado Liberdade XRay cuya finalidad es recorrer el Océano Atlántico en busca de submarinos enemigos.

Otro caso de planeador acuático es el que se muestra en el robot MILANA, el cual funciona al ser remolcado por un barco, mientras el vehículo mantiene una altura con respecto al fondo para hacer tomas del fondo frente a la costa de Barcelona, España (Amat et al., 2006).

2.5.3. Bioinspirados

Este tipo de sistema de propulsión está inspirada en la fisonomía de los peces o los cetáceos, y con ello su forma de desplazarse en el agua. El sistema más simple consiste en una aleta con un grado de libertad colocada vertical u horizontalmente en la parte trasera del vehículo submarino. Su movimiento cíclico perpendicular al plano de las aletas, produce ondas de agua que impulsan el vehículo hacia adelante. Este movimiento produce empuje en un solo sentido.

En la fig. 2.2e se observa el prototipo de un robot nadador llamado Tuna Robot con una aleta trasera desarrollada por el Draper's Laboratory (Anderson and Chhabra,

2002). El diseño permite mover una aleta vertical de izquierda a derecha y con ello hacer navegar a este robot submarino. El desarrollo de un modelo matemático basado en la dinámica de un robot similar llamado PoTuna se puede encontrar en un artículo de (Kim and Yourn, 2004).

En la fig. 2.2f se presenta el robot llamado AQUA, el cual posee 6 aletas (tipo pala) cuyo movimiento sincronizado le permiten navegar en el agua (Dudek et al., 2007).

En (Boyer et al., 2009) se presentó el diseño de un robot anguila. Al igual que la anguila, este robot deforma completamente su cuerpo para generar ondas en el agua y propulsarse. En los robots tipo pez solo la parte trasera se deforma. El robot anguila tiene múltiples vertebras que permiten la deformación. Estas vertebras están compuestas de mecanismos paralelos de tres grados de libertad.

Otro tipo de vehículo submarino que se impulsa por deformación, es el desarrollado por la Universidad Tecnológica de Nanyang en Singapur, el cual consiste en un robot submarino con aletas flexibles modulares que imitan las aletas de un pez raya (Low and Willy, 2005). Cada pequeña aleta es capaz de girar sobre la aleta adyacente a la que se encuentra unida para moverse de forma sincronizada. Esto produce un movimiento ondulatorio que impulsa el conjunto en una dirección.

Otro concepto de navegación por deformación es el robot AMOEBOT (Chen et al., 1999), el cual es un vehículo que se impulsa en el agua por medio de cambios continuos en la forma de su cuerpo, parecido al movimiento de una medusa. La manera de navegar del vehículo consiste en inflar y desinflar determinados globos en una secuencia.

2.5.4. Inyección

Los vehículos acuáticos que utilizan el sistema de propulsión por inyección, recogen el agua, la cual se almacena en un depósito y una bomba expulsa el agua, la cual es inyectada por medio de una boquilla colocada en un tubo.

El agua es forzada a salir a alta presión de la boquilla y la fuerza de empuje del agua en vehículo se refleja en un empuje igual y opuesto, con lo que se produce una aceleración del vehículo. Debido a que el tubo puede girar en cualquier ángulo, es posible obtener un empuje en todas las direcciones según la orientación que se le dé a la boquilla de salida.

Los sistemas de propulsión a chorro son considerados como sistemas no convencionales, y se utilizan en aplicaciones de alta velocidad, que pueden proveer una alternativa viable al impulsor convencional de hélice. Por otro lado, puede proveer de mayor seguridad, ya que evita daños o heridas a seres acuáticos que pueden ser causadas por el contacto de una hélice en movimiento.

Para la implementación en pequeños robots submarinos se ha experimentado con sistemas de propulsión a chorro tipo pulsante (Polsenberg et al., 2005).

2.5.5. Impulsor Magnetohidrodinámico, MHD

El principio de funcionamiento de los impulsores magnetohidrodinámicos consiste en hacer circular corriente eléctrica por un fluido conductor (e.g. agua salada) el cual

es atravesado por un campo magnético. El impulsor MHD genera el movimiento del fluido debido a la interacción del campo magnético y la corriente que circula a través de él.

La ventaja que presenta este tipo de sistema de propulsión es que no tiene partes móviles, y por tanto será más fácil resolver problemas de estanqueidad y mantenimiento. Sin embargo, la desventaja que presenta es, que los electrodos reducen drásticamente el rendimiento electromecánico, debido a su disolución y desprendimiento de gases causada por la electrólisis de los materiales afectados.

Otro problema es que se requieren campos magnéticos de alta intensidad para obtener una fuerza de propulsión considerable en un vehículo submarino. Según (Lin and Gilbert, 1991) máxima eficiencia que puede dar este tipo de impulsores está por debajo del 50%, debido a que la fuerza de propulsión depende de la eficiencia de la inductancia.

2.5.6. Tracción con el fondo marino

La técnica del desplazamiento de un vehículo submarino por medio de la interacción con el fondo marino es similar a la técnica utilizada por los robots terrestres, con la diferencia que el peso del vehículo en el agua se puede neutralizar mediante la flotabilidad del mismo, y que en el agua hay fuerzas de fricción viscosa que son mucho mayores que en el aire. En (Iwasaki et al., 1987) se presento el Aquarobot, un robot caminador submarino de 6 patas simétricas unidas a al cuerpo del robot el cual se encuentra en el centro. Los vehículos submarinos podrían desplazarse mediante patas, ruedas u orugas (Alvarez, 2008b).

2.6. Brazos Robóticos

Para realizar misiones de manipulación los robots submarinos requieren de brazos robóticos manipuladores. En general, los brazos de un ROV son teleoperados utilizando un sistema Maestro-Esclavo. Los ROVs utilizados en la industria del petróleo frecuentemente tienen dos brazos, uno para sujetar el robot a la estructura y el otro para realizar las operaciones requeridas, como lo pueden ser el mantenimiento o la reparación. La mayoría de los manipuladores han sido diseñados para ROVs y éstos son controlados mediante actuadores hidráulicos. No obstante, también se han desarrollado brazos actuados por motores eléctricos, los cuales son más útiles para IAUVs ya que consumen menos energía que los actuadores hidráulicos (Yuh, 2000).

2.7. Fuentes de Energía

La fuente de energía es uno de los componentes mas críticos en un robot sumarino, principalmente en los AUVs. El tipo de fuente de energía define el tiempo de operación del robot, ademas del volumen y peso del vehículo.

Las fuentes de energía mas comunes son (Wang et al., 2009):

Baterías. Están compuestas por una o más celdas electro-químicas que convierten la energía química almacenada en energía eléctrica. Las mismas se clasifican en primarias o secundarias, dependiendo si se pueden recargar o no, respectivamente. Las baterías primarias poseen mejor rendimiento que las secundarias, sin embargo son más costosas. Las baterías primarias más comunes son las alcalinas, las cuales dentro de este género son más económicas y fáciles de emplear. Sin embargo las mismas son sensibles a los cambios de temperatura y pueden generar gas de hidrógeno, lo cual es altamente prohibitivo, debido a que la combinación con el oxígeno produce un compuesto explosivo.

Celdas de combustible. Generan energía eléctrica por medio de una reacción química entre un combustible (por lo general hidrógeno) y un agente oxidante (por lo general oxígeno). Si bien existen una gran diversidad de celdas de combustibles, todas están compuestas por un ánodo, un cátodo y un electrolito que permite el movimiento de cargas. El material empleado como electrolito da origen a los distintos tipos de celdas de combustibles. A diferencia de las baterías, una pila de combustible requiere de un flujo constante de combustible y elemento oxidante para generar energía eléctrica. Por lo general, poseen mayor autonomía que las baterías, pero es necesario que los tanques de combustibles estén presurizados.

En (Bradley et al., 2001), se presenta un resumen de los aspectos más importantes de las baterías y su empleo en sistemas submarinos.

2.8. Conceptos Elementales y Antecedentes del Control de Robots Submarinos.

En esta sección se hará una revisión de las estrategias, métodos y leyes de control que se han propuesto para robots submarinos. Antes de esto, se presentarán algunos conceptos relativos a los componentes de un sistema de control de robots submarinos.

En (Fossen, 2002) y (Antonelli et al., 2008) se presenta el concepto del Sistema de Control de la Misión (MCS por *Mision Control System*). El MCS es el conjunto de programas responsables de que se lleve a cabo la misión deseada. Un MCS está usualmente equipado con una interfaz gráfica que permite al operador comandar diferentes tareas que componen la misión. El MCS contiene los elementos de Guiado, Control y Navegación del robot. Las tareas de la misión son generalmente concurrentes y su manejo depende del estado del vehículo y de las condiciones ambientales; por lo tanto el MCS maneja las tareas, eventualmente suprimiéndolas y definiendo su secuencia (modificando y priorizando). Puede considerarse al MCS como el Sistema Operativo del robot submarino.

El MCS está compuesto por el Sistema de Guiado, el Sistema de Control y el Sistema de Navegación. En la fig. 2.3 se presentan las relaciones entre estos sistemas y el robot submarino. A continuación describimos cada uno de ellos:

Sistema de Guiado. El sistema de guiado genera las trayectorias que se desea que el robot siga durante una misión. Esta información (la posición, velocidad y

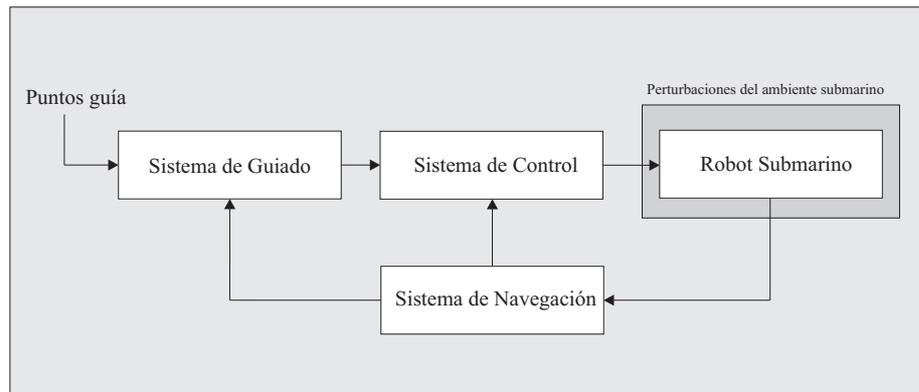


Figura 2.3: Sistema de Control de Misión

aceleración del robot) es enviada al Sistema de Control el cual tratará de alcanzar las referencias deseadas. El sistema de guiado tiene como entradas los Puntos Guía definidos por el usuario, e información proveniente del sistema de Navegación. Además, puede recibir información sobre la situación del ambiente (e.g. corrientes marinas), de la topografía del fondo marino, y datos provenientes de un sonar para la detección de obstáculos.

Sistema de Control. El sistema de control es el encargado de calcular las fuerzas necesarias que debe generar el sistema de propulsión para que el robot alcance las referencias deseadas. La construcción del sistema de control involucra el diseño y la sintonización de las Leyes de Control que regularán los movimientos del robot. El sistema de control recibe información la posición, velocidad y aceleración deseada y el estado actual de estas variables medidas a través de los sensores del robot. Este sistema puede consistir en leyes de control clásico, control no lineal, control inteligente, sistemas basados en comportamiento (behavior-based), etc.

Sistema de Navegación. El sistema de navegación recibe la información de los sensores de posición, velocidad y aceleración del robot submarino. Mediante un Observador u otros algoritmos procesa esta información y posteriormente la envía al Sistema de Guiado y al Sistema de Control.

En (Valavanis et al., 1997) se presenta una clasificación de los tipos de MSCs de AUVs en la cual se identificaron 4 arquitecturas de control: Jerárquica, Distribuida, Por Capas (subsumption) e Híbrida. Por otro lado en (Ridao et al., 2000), se hace una clasificación de arquitecturas de control en la que se considera los aspectos de inteligencia artificial e interacción con el medio ambiente. En este último trabajo la clasificación de las arquitecturas de control quedaría de la siguiente manera: Deliberativa, Reactiva e Híbrida. Consideramos que esta última clasificación es más general e incluye a la hecha en (Valavanis et al., 1997). A continuación se describe brevemente estas arquitecturas:

Arquitectura Deliberativa. Esta arquitectura, del tipo piramidal, divide el sistema

en niveles de mayor a menor responsabilidad. Los niveles más altos se encargan de la misión que debe realizar el robot, y los niveles más bajos resuelven problemas particulares. La comunicación sólo es posible entre dos capas adyacentes, la de mayor capa envía comandos al nivel más bajo subsiguiente. La arquitectura deliberativa se basa en el planeamiento y un modelo del mundo. Esto le permite razonar y hacer predicciones acerca del medio ambiente. El análisis de los datos obtenidos de los sensores identifica el mundo real y se usa para planear nuevas acciones.

Arquitectura Reactiva. La arquitectura reactiva o basada en comportamiento (*behavior based*) usa una estructura paralela, donde todos los módulos del sistema pueden comunicarse directamente entre ellos, sin necesidad de la supervisión de un nivel intermediario superior. El comportamiento obtenido del robot está basado en una secuencia de fases con un conjunto de conductas activas que reaccionan continuamente ante las situaciones detectadas. El comportamiento global surge de la combinación de los elementos activos, los cuales siguen el principio de detección - reacción, y que pueden interactuar con medios dinámicos. Debido a que las múltiples reacciones podrían cambiar la perspectiva global de la misión, a veces el robot podría comportarse de una manera no previsible. Para evitar esto, se establece un sistema de arbitraje de prioridades, para que en el momento de algún conflicto, se resuelva la situación y se retome la meta establecida.

Arquitectura Híbrida. Es una arquitectura que combina a las arquitecturas deliberativa y reactiva, en donde se toman las ventajas de cada una. El sistema se divide por lo general en tres niveles: la capa deliberativa, la cual está basada en la planificación; la capa de ejecución de control, y la capa funcional reactiva. Estas capas usan diferentes niveles de abstracción. El nivel alto utiliza la arquitectura jerárquica para implementar las estrategias y la funcionalidad del sistema. El nivel más bajo utiliza la arquitectura reactiva, para controlar el hardware del subsistema según la interacción con el medio ambiente, pero controladas por la capa de ejecución de control. La capa reactiva consiste de algunos módulos con autonomía propia que pueden ejecutar tareas de forma independiente. En una situación de emergencia, puede hacerse cambios, para que el nivel más alto asuma el control.

Una vez descritas las arquitecturas de control, a continuación se describe brevemente de los sistemas de control (de bajo nivel) que se han propuesto en la literatura.

Control Clásico. Para controlar robots submarinos se pueden emplear las distintas técnicas de control que previamente se han propuesto para otros sistemas. Este es el caso de los controladores en lazo cerrado que realimentan las señales medidas por los sensores del robot, como lo son los controladores PI, PD, PID, PID con realimentación de aceleración, etc. Dada la estructura del modelo dinámico presentado en forma matricial se podrían extrapolar las distintas leyes de control que se han propuesto para robot manipuladores.

Control No Lineal. En la literatura de robótica submarina se han propuesto dos técnicas de control no lineal: el control en modos deslizantes y el control adaptable. Las ventajas de un controlador en modos deslizantes son que trabaja directamente con los términos no lineales, es altamente robusto a modelos imprecisos y a la presencia de perturbaciones que son difíciles de medir o estimar. Por otro lado, diversos autores han propuesto controladores adaptables, ya que pueden ser una solución al problema de determinar los parámetros hidrodinámicos del robot, pues es difícil obtener estos parámetros mediante formulaciones matemáticas, y por otro lado, determinarlos experimentalmente conlleva a un costo económico y de tiempo.

Control Inteligente. En esta clasificación se encuentran los controladores basados en redes neuronales, lógica difusa, el empleo de algoritmos genéticos, etc. En (Yuh, 1994) se presentó un sistema de control que consistía de una red neuronal y un algoritmo de aprendizaje. El algoritmo de aprendizaje ajusta los valores de las parámetros de ponderación de las neuronas, en función del error existente entre el estado deseado y el real. Algunos beneficios de los controladores basados en lógica difusa son su simplicidad, debido que no se requiere un modelo dinámico del sistema, y mediante el uso de reglas lingüísticas se puede tener una fácil comprensión y manipulación del controlador. Se han aplicado algoritmos genéticos para sintonizar de reglas para un controlador de lógica difusa.

2.9. Resumen

En este capítulo se hizo una revisión del estado de la técnica de la robótica submarina. Se describen las tecnologías actuales para determinar el estado del robot, los actuadores, y las fuentes de alimentación. Se hizo una clasificación por el nivel de autonomía, el tipo de misión a realizar y el sistema de propulsión. Se presentaron los distintos sistemas de propulsión que se han propuesto. Además, se expuso una descripción de las estrategias que se han propuesto para controlar estos dispositivos. Los conceptos y términos mostrados aquí serán útiles para dar contexto al trabajo de investigación.

Capítulo 3

Modelado de Robots Submarinos

3.1. Introducción

En esta capítulo se presentará el modelado cinemático y dinámico de robots submarinos. El modelado cinemático consiste de las ecuaciones que relacionan los diferentes tipos de coordenadas utilizadas para expresar la velocidad del robot. Por otro lado, en el modelado dinámico se describen las fuerzas que actúan sobre el robot cuando este se navega.

Cuando se analiza el movimiento de un robot submarino, es necesario definir dos marcos de referencia para describir su movimiento, estos son:

Marco de referencia inercial NED. Este marco esta ubicado sobre la superficie de la tierra y su nombre indica las siglas en inglés de Norte-Este-Abajo (*North-East-Down*). En este marco el eje x apunta hacia el Norte, el eje y apunta hacia el este, y el eje z apunta hacia abajo y es normal a la superficie de la tierra. De esta forma el plano $x - y$ es tangente en la superficie de la tierra. En esta tesis nos referimos a este marco simplemente por N .

Marco de referencia del cuerpo B . Este marco de referencia esta fijo al cuerpo del robot. Las velocidades lineales y angulares se expresan generalmente en el marco de referencia del cuerpo. El origen O del este marco de referencia generalmente se hace coincidir con el centro de gravedad del cuerpo, cuando este se encuentra en el plano principal de simetría, ó en algún otro punto conveniente si este no es el caso. Los ejes de este marco generalmente se eligen de tal forma que coinciden con los ejes principales de inercia. Generalmente x_b es el eje longitudinal (que va de atrás hacia adelante) , y_b es el eje transversal (que va de un lado al otro), y z_b que va dirigido hacia abajo.

En navegación marítima y en robótica submarina existe una convención (SNAME) para expresar la posición, orientación del vehículo y las fuerzas que se ejercen sobre el.

Para representar la posición del vehículo con respecto al marco inercial N se tiene el vector:

$$\eta = \begin{bmatrix} \mathbf{r}_o^n \\ \Theta \end{bmatrix} \quad (3.1)$$

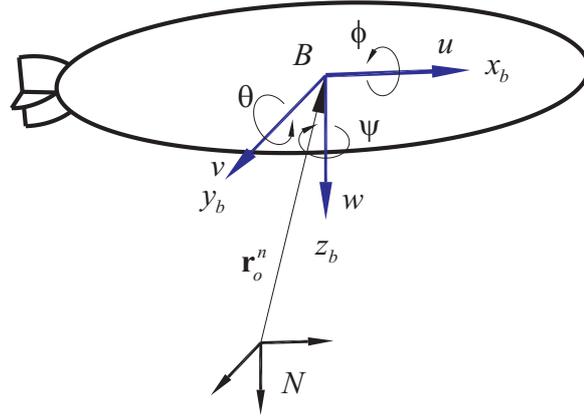


Figura 3.1: Notación en robótica submarina.

Tabla 3.1: Notación del movimiento de robots submarinos

Movimiento	Nombre		Posición	Velocidad	Fuerza
	Español	Inglés			
Traslación en x	Avance	Surge	x	u	X
Traslación en y	Desvío	Sway	y	v	Y
Traslación en z	Viraje	Heave	z	w	Z
Rotación en x	Alabeo	Roll	ϕ	p	K
Rotación en y	Cabeceo	Pitch	θ	q	M
Rotación en z	Guiñada	Yaw	ψ	r	N

donde $\mathbf{r}_o^n = [x \ y \ z]^T$ representa la posición del marco de referencia del cuerpo B con respecto al marco N , mientras que Θ es un vector de parámetros que permiten definir la orientación relativa entre estos dos marcos de referencia. En este caso, los parámetros de orientación pueden ser los ángulos de Euler ó el cuaternión de los parámetros de Euler \mathbf{p}_b .

La velocidad del vehículo se representa por el vector:

$$\boldsymbol{\nu} = \begin{bmatrix} \mathbf{v}_o^b \\ \boldsymbol{\omega}_b^b \end{bmatrix} \quad (3.2)$$

donde $\mathbf{v}_o^b = [u \ v \ w]^T$ es la velocidad lineal del vehículo medido en coordenadas del marco del cuerpo B , mientras que $\boldsymbol{\omega}_b^b = [p \ q \ r]^T$ representa la velocidad angular del cuerpo con respecto al marco inercial medido en el marco de referencia del cuerpo B . Estas velocidades también se suelen denotar por los vectores $\boldsymbol{\nu}_1 = \mathbf{v}_o^b$ y $\boldsymbol{\nu}_2 = \boldsymbol{\omega}_b^b$.

Para describir las fuerzas a las que está sujeto el vehículo se utiliza la notación :

$$\boldsymbol{\tau} = \begin{bmatrix} \mathbf{f}_o^b \\ \mathbf{n}_o^b \end{bmatrix} \quad (3.3)$$

donde $\mathbf{f}_o^b = [X \ Y \ Z]^T$ y $\mathbf{n}_o^b = [K \ M \ N]^T$ son los vectores de fuerza y momento ejercidos sobre el cuerpo medidos en el marco de referencia del cuerpo B .

En las siguientes secciones se presenta un revisión breve a la formulación presentada en (Fossen, 2002) sobre el modelado de cuerpos sumergidos.

3.2. Modelado Cinemático de Robots Submarinos

3.2.1. Matriz de Rotación

Las matrices de rotación son el metodo más extendido para la descripción de orientaciones, debido principalmente a la comodidad que proporciona el uso del algebra matricial. Estas matrices describen la orientación mutua entre dos sistemas coordenados; sus vectores columna son los cosenos directores de los ejes del sistema coordenado del cuerpo, con respecto al sistema coordenado inercial.

Dado un vector \mathbf{r}^b expresado en coordenadas del marco B , entonces ese mismo vector, expresado en coordenadas del marco N es dado por: $\mathbf{r}^a = \mathbf{R}_b^a \mathbf{r}^b$.

Sean A , B y C tres marcos coordenados. Si \mathbf{R}_b^a es la matriz de rotación de B a A y \mathbf{R}_c^b es la matriz de rotación de C a B , entonces la matriz de rotación de C a A es dada por: $\mathbf{R}_c^a = \mathbf{R}_b^a \mathbf{R}_c^b$ que define la regla de composición para rotaciones. Finalmente, se denomina matriz de rotación elemental a la matriz de rotación que representa una rotación alrededor de uno de los ejes del marco de referencia.

3.2.2. Ángulos de Euler

De acuerdo a L. Euler, cualesquiera dos marcos coordenados ortonormales independientes (con un origen común) pueden ser relacionados por una sucesión de no más de tres rotaciones alrededor de los ejes coordenados. Esto significa que si se conoce la secuencia de ejes a rotar, se necesitan sólo tres ángulos de Euler para definir completamente la rotación total.

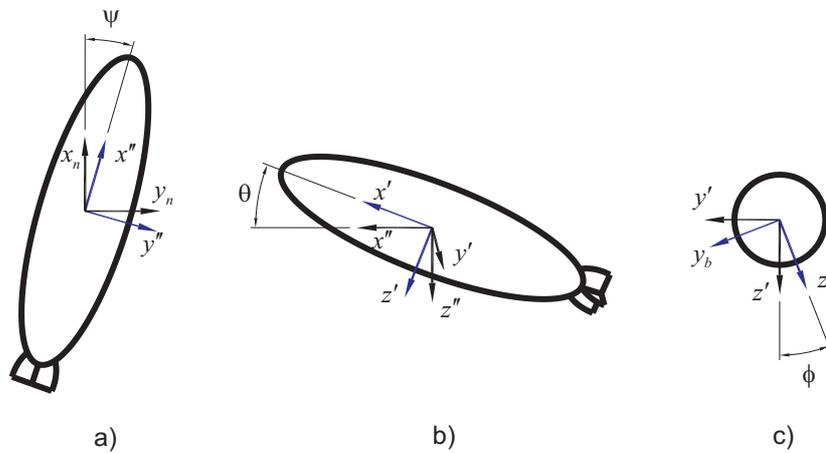


Figura 3.2: Notación en robótica submarina.

Dada una sucesión de ángulos de Euler, es posible encontrar la matriz de rotación correspondiente. La convención utilizada en robótica submarina para los ángulos de Euler consiste de la siguiente sucesión:

- Rotación del marco de referencia de un ángulo ψ alrededor del eje z para obtener el marco B'' .
- Rotación del marco de referencia de un ángulo θ alrededor del eje y para obtener el marco B' .
- Rotación del marco de referencia de un ángulo ϕ alrededor del eje x para obtener el marco B .

De esta forma:

$$\mathbf{R}_{x,\phi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & -s\phi \\ 0 & s\phi & c\phi \end{bmatrix} \mathbf{R}_{y,\theta} = \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix} \mathbf{R}_{z,\psi} = \begin{bmatrix} c\psi & -s\psi & 0 \\ s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

donde $s \cdot = \sin(\cdot)$ y $c \cdot = \cos(\cdot)$.

El orden de estas rotaciones no es arbitrario. Estas rotaciones se llevan a cabo desde el marco de referencia N al marco B . Esta matriz se denota por $\mathbf{R}_n^b = \mathbf{R}_b^{nT}$. La matriz transpuesta implica que el mismo resultado se obtiene de transformar un vector del marco de referencia B al marco N , invirtiendo el orden de rotación. Esta secuencia de rotaciones es matemáticamente equivalente a:

$$\mathbf{R}_b^n = \mathbf{R}_{z,\psi} \mathbf{R}_{y,\theta} \mathbf{R}_{x,\phi}$$

Realizando esta operación se obtiene:

$$\mathbf{R}_b^n = \begin{bmatrix} c\psi c\theta & -s\psi c\phi + c\psi s\theta s\phi & s\psi s\phi + c\psi c\phi s\theta \\ s\psi c\theta & c\psi c\phi + s\phi s\theta s\psi & -c\psi s\phi + s\theta s\psi c\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \quad (3.4)$$

Para realizar la operación inversa, es decir determinar los ángulos de euler una vez conocida la matriz de rotación, se pueden derivar las siguientes expresiones:

$$\phi = \text{atan2}(r_{32}, r_{33}) \quad (3.5)$$

$$\theta = -\sin^{-1}(r_{31}) = -\tan^{-1} \left(\frac{r_{31}}{\sqrt{1 - r_{31}^2}} \right) \quad (3.6)$$

$$\psi = \text{atan2}(r_{21}, r_{11}) \quad (3.7)$$

en donde r_{ij} denota el elemento (i, j) de la matriz de rotación \mathbf{R}_b^n , y $\text{atan2}(y, x)$ es la función arco tangente de dos argumentos reales x, y , y satisface que $\frac{\pi}{2} \leq \text{atan2} \leq \frac{\pi}{2}$.

Transformación de Velocidades

La transformación que relaciona el vector de velocidad lineal en un marco de referencia inercial a al marco de referencia del marco de referencia del cuerpo puede se expresada de la siguiente manera:

$$\dot{\mathbf{r}}_o^b = \mathbf{R}_b^n \mathbf{v}_o^b \quad (3.8)$$

Por otro lado, el vector de velocidad angular del cuerpo ω_b^b y el vector de velocidad de los ángulos de Euler están relacionados mediante una matriz de transformación \mathbf{T}_Θ como se muestra a continuación:

$$\dot{\Theta} = \mathbf{T}_\Theta \omega_b^b \quad (3.9)$$

donde \mathbf{T}_Θ esta dada por:

$$\mathbf{T}_\Theta = \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & s\phi/c\theta & c\phi/c\theta \end{bmatrix} \quad (3.10)$$

y $t \cdot = \tan(\cdot)$.

Las expresiones anteriores se pueden resumir de la siguiente manera:

$$\dot{\eta} = \mathbf{J}\nu \quad (3.11)$$

donde:

$$\mathbf{J}(\nu) = \begin{bmatrix} \mathbf{R}_b^n & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_\Theta \end{bmatrix} \quad (3.12)$$

donde $\dot{\eta} = \frac{d(\eta)}{dt}$.

3.2.3. Parámetros de Euler

Otro alternativa para representar la orientación mediante parámetros de Euler. La principal motivación de usar estos parámetros es evitar las singularidades que sufre la representación mediante los ángulos de Euler.

Un cuaternión esta compuesto de cuatro parámetros:

$$\mathbf{p} = [e_0 \ e_1 \ e_2 \ e_3]^T \quad (3.13)$$

que satisfacen una restricción de norma unitaria,

$$\mathbf{p}^T \mathbf{p} = 1.$$

La expresión de la matriz de rotación en términos de los parámetros de Euler es la siguiente:

$$\mathbf{R}(\mathbf{p}) = \begin{bmatrix} 1 - 2(e_2^2 + e_3^2) & 2(e_1 e_2 - e_0 e_3) & 2(e_1 e_3 + e_0 e_2) \\ 2(e_1 e_2 + e_0 e_3) & 1 - 2(e_1^2 + e_3^2) & 2(e_2 e_3 - e_0 e_1) \\ 2(e_1 e_3 - e_0 e_2) & 2(e_2 e_3 + e_0 e_1) & 1 - 2(e_1^2 + e_2^2) \end{bmatrix} \quad (3.14)$$

En forma compacta, se tiene:

$$\mathbf{R} = (e_0^2 - \mathbf{e}^T \mathbf{e}) \mathbf{I}_3 + 2e_0 \mathbf{S}(\mathbf{e}) + 2\mathbf{e}\mathbf{e}^T,$$

o equivalentemente

$$\mathbf{R} = \mathbf{I}_3 + 2e_0\mathbf{S}(\mathbf{e}) + 2\mathbf{S}(\mathbf{e})^2$$

donde $\mathbf{e} = [e_1 \ e_2 \ e_3]^T$ es un subvector de \mathbf{p} y $\mathbb{S}(\cdot)$ es el operador matricial del producto cruz. Dado el vector \mathbf{e} se tiene que

$$\mathbb{S}(\mathbf{e}) = \begin{bmatrix} 0 & -e_3 & e_2 \\ e_3 & 0 & -e_1 \\ -e_2 & e_1 & 0 \end{bmatrix} \quad (3.15)$$

Transformación de Velocidades

Para realizar la transformación de velocidades lineales se realiza la misma operación que en el caso de los ángulos de Euler, simplemente mediante el uso de la matriz de rotación.

La transformación de la velocidad angular se puede derivar sustituyendo la expresión (3.14) en la ecuación diferencial $\dot{\mathbf{R}}_b^n = \mathbf{R}_b^n \mathbf{S}(\boldsymbol{\omega}_b^b)$. Mediante algunas operaciones se tiene:

$$\dot{\mathbf{p}} = \mathbf{T}_p \boldsymbol{\omega}_b^b \quad (3.16)$$

donde:

$$\mathbf{T}_p(\mathbf{p}) = \frac{1}{2} \begin{bmatrix} -e_1 & -e_2 & -e_3 \\ e_0 & -e_3 & e_2 \\ e_3 & e_0 & -e_1 \\ -e_2 & e_1 & e_0 \end{bmatrix} \quad (3.17)$$

Las expresiones anteriores se pueden resumir de la siguiente manera:

$$\dot{\boldsymbol{\eta}} = \mathbf{J}\boldsymbol{\nu} \quad (3.18)$$

donde:

$$\mathbf{J}(\boldsymbol{\nu}) = \begin{bmatrix} \mathbf{R}_b^n & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_p \end{bmatrix} \quad (3.19)$$

En este caso $\dot{\boldsymbol{\eta}} = [\dot{\mathbf{r}}_o^n \ \dot{\mathbf{q}}]^T$.

3.3. Dinámica de Robots Submarinos

El modelo dinámico de un robot submarino describe la relación existente entre los movimientos del robot y las fuerzas que se ejercen sobre el mismo. Se presenta una revisión breve a la formulación presentada en (Fossen, 2002) sobre el modelado de cuerpos sumergidos. De esta manera, se pueden calcular las fuerzas externas necesarias para que el robot se mueva de una forma determinada, ó por el contrario, se puede determinar el movimiento generado por las fuerzas externas a las que el robot está sujeto.

En este capítulo se presentaran las ecuaciones de movimiento de Newton-Euler para cuerpos sumergidos en el agua. Los cuerpos sumergidos están sujetos a diferentes tipos

de fuerzas. Estas fuerzas son principalmente las fuerzas inerciales, las hidrodinámicas y las fuerzas de restitución.

El modelo dinámico de un robot submarino se puede escribir en su forma compacta como se muestra a continuación:

$$\mathbf{M}\dot{\boldsymbol{\nu}} + \mathbf{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{g}(\boldsymbol{\eta}) = \boldsymbol{\tau} + \mathbf{g}_o + \mathbf{w} \quad (3.20)$$

$$\boldsymbol{\tau} = \mathbf{B}_t \mathbf{u}_t \quad (3.21)$$

donde la matriz \mathbf{M} contiene los elementos de la matriz de inercia y la matriz de masa añadida (el concepto de la masa añadida se presentara posteriormente); por su parte la matriz \mathbf{C} consiste de la matriz de fuerzas centrífugas y de Coriolis del cuerpo rígido y de la masa añadida; la matriz de \mathbf{D} es la matriz de fuerzas viscosas; y el vector $\mathbf{g}(\boldsymbol{\eta})$ representa las fuerzas de restitución (las cuales están compuestas por la fuerza de gravedad y la fuerza de flotación).

Por otro lado el vector $\boldsymbol{\tau}$ representa las fuerzas que ejercen los impulsores (o cualquier otro elemento generador de fuerza) sobre el robot submarino. Los elementos de la matriz de control \mathbf{B}_t dependen de la configuración de cada robot, sus superficies de control, el numero de impulsores, y la ubicación y orientación de los impulsores. El vector \mathbf{u}_t esta compuesto de las fuerzas que se generan en cada impulsor. Por otra parte, el vector \mathbf{g}_o representa la fuerza generada por el control de lastre. El vector \mathbf{w} representa las perturbaciones del ambiente (como lo son, las olas, el viento y las corrientes marinas).

En las siguientes secciones se describen en mayor detalle las fuerzas anteriormente señaladas.

3.3.1. Fuerzas Inerciales

Las ecuaciones de Newton-Euler para un cuerpo rígido son las siguientes:

$$\mathbf{f}_c^b = \frac{d}{dt}(m\mathbf{v}_c^b) \quad (3.22)$$

$$\mathbf{n}_c^b = \frac{d}{dt}(\mathbf{I}_c\boldsymbol{\omega}_b^b) \quad (3.23)$$

donde \mathbf{f}_c^b y \mathbf{n}_c^b son la fuerza y el torque en el centro de masa del cuerpo; m e \mathbf{I} son la masa y la matriz de inercia del cuerpo, respectivamente. En este caso la matriz de masa esta dada en el marco de referencia del cuerpo. Los vectores \mathbf{v}_c^b y $\boldsymbol{\omega}_b^b$ representan la velocidad lineal y angular del cuerpo con respecto al marco de referencia inercial N pero medidas en el marco de referencia del cuerpo B .

Para robots submarinos, es deseable derivar las ecuaciones de movimiento dada una posición arbitraria del origen O del marco de referencia del cuerpo B . Esto se debe a que algunas propiedades hidrodinámicas se pueden calcular con mayor facilidad en puntos que pueden no coincidir con el centro de masa del robot.

La velocidad lineal del centro de masa con respecto al marco N medido en el marco de referencia del cuerpo B se puede expresar de la siguiente manera:

$$\mathbf{v}_c^b = \mathbf{v}_o^b + \boldsymbol{\omega}_b^b \times \mathbf{r}_g^b \quad (3.24)$$

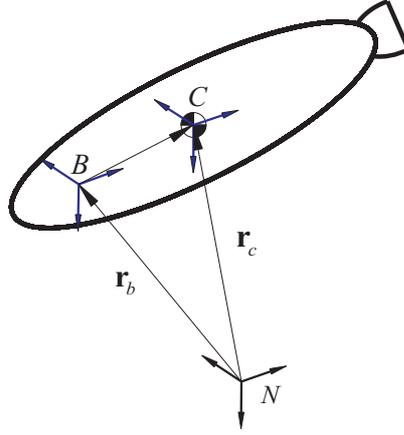


Figura 3.3: Marcos de referencia en un robot sumario.

donde \mathbf{r}_g^b es la posición del centro de masa con respecto al origen del marco B . Para determinar la aceleración del centro de masa con respecto al marco de referencia N se considera primero que $\mathbf{v}_c^n = \mathbf{R}_b^n \mathbf{v}_c^b = \mathbf{R}_b^n (\mathbf{v}_o^b + \boldsymbol{\omega}_b^b \times \mathbf{r}_g^b)$. Derivando \mathbf{v}_c^n con respecto al tiempo se obtiene la siguiente expresión:

$$\dot{\mathbf{v}}_c^n = \mathbf{R}_b^n [\dot{\mathbf{v}}_o^b + \dot{\boldsymbol{\omega}}_b^b \times \mathbf{r}_g^b + \boldsymbol{\omega}_b^b \times \mathbf{v}_o^b + \boldsymbol{\omega}_b^b \times (\boldsymbol{\omega}_b^b \times \mathbf{r}_g^b)] \quad (3.25)$$

Por lo tanto, sustituyendo el valor de la derivada (3.25) en la ecuación (3.22), y dado que $\mathbf{f}_o^b = \mathbf{f}_c^b$ para movimientos traslacionales obtenemos la siguiente expresión:

$$\mathbf{f}_o^b = m[\dot{\mathbf{v}}_o^b + \dot{\boldsymbol{\omega}}_b^b \times \mathbf{r}_g^b + \boldsymbol{\omega}_b^b \times \mathbf{v}_o^b + \boldsymbol{\omega}_b^b \times (\boldsymbol{\omega}_b^b \times \mathbf{r}_g^b)] \quad (3.26)$$

lo cual nos permite calcular la fuerza en el origen del marco de referencia del cuerpo, es decir O .

Por otra parte el momento angular con respecto al punto O y al punto C esta dado por las siguientes expresiones $\mathbf{h}_o^b = \mathbf{I}_o \boldsymbol{\omega}_b^b + m \mathbf{r}_g^b \times \mathbf{v}_o^b$ y $\mathbf{h}_c^b = \mathbf{h}_o^b - m \mathbf{r}_g^b \times \mathbf{v}_c^b$

Considerando que $\mathbf{h}_c^n = \mathbf{R}_b^n \mathbf{h}_c^b$, la derivada temporal del momentum angular con respecto al punto C es la siguiente:

$$\dot{\mathbf{h}}_c^n = \mathbf{R}_b^n [\mathbf{I}_o \dot{\boldsymbol{\omega}}_b^b + m \mathbf{r}_g^b \times (\dot{\mathbf{v}}_o^b - \dot{\mathbf{v}}_c^b) + \boldsymbol{\omega}_b^b \times (\mathbf{I}_o \boldsymbol{\omega}_b^b + m \mathbf{r}_g^b \times (\mathbf{v}_o^b - \mathbf{v}_c^b))].$$

Por otro lado, el momento \mathbf{n}_c^n es igual a $\mathbf{n}_c^n = \dot{\mathbf{h}}_c^n$ y $\mathbf{n}_c^n = \mathbf{R}_b^n (m \mathbf{r}_g^b - \mathbf{r}_g^b \times \mathbf{f}_c^b)$ siendo en este caso $\mathbf{f}_c^b = m(\dot{\mathbf{v}}_c^b + \boldsymbol{\omega}_b^b \times \mathbf{v}_c^b)$. Igualando las expresiones de \mathbf{n}_c^n y realizando operaciones algebraicas se obtiene finalmente la ecuación del momento en el origen del marco B :

$$\mathbf{n}_o^b = \mathbf{I}_o \dot{\boldsymbol{\omega}}_b^b + \boldsymbol{\omega}_b^b \times (\mathbf{I}_o \boldsymbol{\omega}_b^b) + m \mathbf{r}_g^b \times \dot{\mathbf{v}}_o^b + m \mathbf{r}_g^b \times (\boldsymbol{\omega}_b^b \times \mathbf{v}_o^b) \quad (3.27)$$

Las ecuaciones (3.26) y (3.27) se pueden escribir en forma matricial, como se muestra a continuación:

$$\begin{bmatrix} \mathbf{f}_o^b \\ \mathbf{n}_o^b \end{bmatrix} = \mathbf{M}_R \begin{bmatrix} \dot{\mathbf{v}}_o^b \\ \dot{\boldsymbol{\omega}}_b^b \end{bmatrix} + \mathbf{C}_R(\mathbf{v}_o^b, \boldsymbol{\omega}_b^b) \begin{bmatrix} \mathbf{v}_o^b \\ \boldsymbol{\omega}_b^b \end{bmatrix} \quad (3.28)$$

donde \mathbf{M}_R es la matriz de masas del cuerpo rígido, y $\mathbf{C}_R(\nu)$ es la matriz de fuerzas centrífugas y de Coriolis. La matriz de masas esta dada de la siguiente manera:

$$\mathbf{M}_R = \begin{bmatrix} m\mathbf{I}_3 & -m\mathbf{S}(\mathbf{r}_g^b) \\ m\mathbf{S}(\mathbf{r}_g^b) & \mathbf{I}_O \end{bmatrix} \quad (3.29)$$

La matriz de fuerzas centrífugas y de Coriolis puede adoptar diferentes formas. En este caso adoptamos una que hace que esta matriz sea antisimétrica, es decir $\mathbf{C}_R(\nu) = -\mathbf{C}_R(\nu)^T$. Por lo tanto, esta matriz esta dada por la siguiente expresión:

$$\mathbf{C}_R(\nu) = \begin{bmatrix} \mathbf{0}_3 & -\mathbf{S}(\mathbf{M}_{R11}\nu_1 + \mathbf{M}_{R12}\nu_2) \\ -\mathbf{S}(\mathbf{M}_{R11}\nu_1 + \mathbf{M}_{R12}\nu_2) & -\mathbf{S}(\mathbf{M}_{R21}\nu_1 + \mathbf{M}_{R22}\nu_2) \end{bmatrix} \quad (3.30)$$

3.3.2. Fuerzas Hidrodinámicas

Los fuerzas hidrodinámicas que afectan un cuerpo sumergido son bastantes complejas ya que dependen de muchas variables como son las características del fluido, la temperatura, la presión y la forma geométrica del cuerpo entre otras. Las fuerzas hidrodinámicas son la fuerzas de masa añadida y las fuerzas viscosas.

3.3.3. Masa Añadida

La masa añadida es una fuerza que se opone al movimiento del cuerpo en el fluido y depende de la aceleración del cuerpo, como también puede serlo de la velocidad. La masa añadida se suele asociar con la cantidad de fluido que se encuentra alrededor del cuerpo, y que debe acelerarse cuando el cuerpo también lo hace. Muchas veces se malinterpreta la masa añadida como una cantidad de fluido a agregar a la masa del cuerpo, sin embargo, la masa añadida es algo más complejo. A diferencia de la masa de cuerpo rígido, la masa añadida no siempre cumple con la propiedad de ser constante y simétrica, como tampoco es siempre diagonal, incluso se han reportado casos en donde posee elementos negativos. La manera más fiable de medir la masa añadida es a través de la experimentación. La forma como se hace es acelerar el cuerpo en dirección de los ejes y medir la variación en la fuerza requerida.

$$\mathbf{M}_A = - \begin{bmatrix} X_{\ddot{u}} & X_{\ddot{v}} & X_{\ddot{w}} & X_{\ddot{p}} & X_{\ddot{q}} & X_{\ddot{r}} \\ Y_{\ddot{u}} & Y_{\ddot{v}} & Y_{\ddot{w}} & Y_{\ddot{p}} & Y_{\ddot{q}} & Y_{\ddot{r}} \\ Z_{\ddot{u}} & Z_{\ddot{v}} & Z_{\ddot{w}} & Z_{\ddot{p}} & Z_{\ddot{q}} & Z_{\ddot{r}} \\ K_{\ddot{u}} & K_{\ddot{v}} & K_{\ddot{w}} & K_{\ddot{p}} & K_{\ddot{q}} & K_{\ddot{r}} \\ M_{\ddot{u}} & M_{\ddot{v}} & M_{\ddot{w}} & M_{\ddot{p}} & M_{\ddot{q}} & M_{\ddot{r}} \\ N_{\ddot{u}} & N_{\ddot{v}} & N_{\ddot{w}} & N_{\ddot{p}} & N_{\ddot{q}} & N_{\ddot{r}} \end{bmatrix} \quad (3.31)$$

Por otro lado a la matriz de masa añadida la podemos expresar en términos de 4 submatrices:

$$\mathbf{M}_A = \begin{bmatrix} \mathbf{M}_{A11} & \mathbf{M}_{A12} \\ \mathbf{M}_{A21} & \mathbf{M}_{A22} \end{bmatrix} \quad (3.32)$$

La matriz de fuerzas centrífugas y de Coriolis de la masa añadida puede adoptar diferentes formas. Al igual que en el caso de $\mathbf{C}_R(\boldsymbol{\nu})$, adoptamos una que hace que esta matriz sea antisimétrica, es decir $\mathbf{C}_A(\boldsymbol{\nu}) = -\mathbf{C}_A(\boldsymbol{\nu})^T$. Por lo tanto, esta matriz esta dada por la siguiente expresión:

$$\mathbf{C}_A(\boldsymbol{\nu}) = \begin{bmatrix} \mathbf{0} & -\mathbb{S}(\mathbf{M}_{A11}\boldsymbol{\nu}_1 + \mathbf{M}_{A12}\boldsymbol{\nu}_2) \\ -\mathbb{S}(\mathbf{M}_{A11}\boldsymbol{\nu}_1 + \mathbf{M}_{A12}\boldsymbol{\nu}_2) & -\mathbb{S}(\mathbf{M}_{A21}\boldsymbol{\nu}_1 + \mathbf{M}_{A22}\boldsymbol{\nu}_2) \end{bmatrix} \quad (3.33)$$

Masa añadida de una esferoide prolato

Imlay derivó las expresiones de la masa añadida para un elipsoide y un esferoide prolato. A continuación se presentan las ecuaciones de la masa añadida para un esferoide prolato. Estas expresiones son función únicamente de las características geométricas del esferoide.

Considérese un elipsoide totalmente sumergido descrito como $\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 0$, donde a , b , c son los semi ejes del elipsoide. Un esferoide prolato se obtiene de $b = c$ y $a > b$. La masa añadida esta dada de la siguiente manera:

$$X_{\dot{u}} = -\frac{\alpha_0}{2 - \alpha_0} m_s \quad (3.34)$$

$$Y_{\dot{v}} = Z_{\dot{w}} = -\frac{\beta_0}{2 - \beta_0} m_s \quad (3.35)$$

$$K_{\dot{p}} = 0 \quad (3.36)$$

$$N_{\dot{r}} = M_{\dot{q}} = -\frac{1}{5} \frac{(b^2 - a^2)^2 (\alpha_0 - \beta_0)}{2(b^2 - a^2) + (b^2 + a^2)(\beta_0 - \alpha_0)} m_s \quad (3.37)$$

donde el parámetro m_s es:

$$m_s = \frac{4}{3} \pi \rho a b^2 \quad (3.38)$$

y las constantes α_0 y β_0 puede calcularse como:

$$\alpha_0 = \frac{2(1 - e^2)}{e^3} \left(\frac{1}{2} \ln \frac{1 + e}{1 - e} - e \right) \quad (3.39)$$

$$\beta_0 = \frac{1}{e^2} - \frac{1 - e^2}{2e^3} \ln \frac{1 + e}{1 - e} \quad (3.40)$$

siendo e la excentricidad:

$$e^2 = 1 - (b/a)^2 \quad (3.41)$$

Estas expresiones pueden resultar útiles para calcular un estimado de la masa añadida de un robot submarino con forma de esferoide prolato (como lo son algunos AUVs).

3.3.3.1. Fuerzas Viscosas

Estas fuerzas son el resultado de la fricción que ejerce la viscosidad del fluido sobre el cuerpo. Se suelen clasificar según el efecto sobre el cuerpo en: fuerza de arrastre y fuerza de sustentación. Las fuerzas de arrastre se oponen al movimiento y actúan en dirección opuesta, mientras que las fuerzas de sustentación son perpendiculares al movimiento relativo del cuerpo sobre el fluido. Al igual que la masa añadida, la única forma de tener un conocimiento fiable de las magnitudes de dichas fuerzas es a través de la experimentación, con lo cual se obtendrá una matriz no lineal y compleja que varía en función de la velocidad del cuerpo.

Las fuerzas de amortiguación viscosa se pueden modelar de la siguiente manera:

$$f(u) = -\frac{1}{2}\rho_a C_D A_t |u|u \quad (3.42)$$

Donde u es la velocidad del vehículo, A es la area que se proyecta en un plano ortogonal a la dirección de la velocidad, C_D es el coeficiente de arrastre basado en una area representativa, y ρ_a es la densidad del agua. Esta expresión se reconoce como una lo los términos de la ecuaciones de morrison. El coeficiente de arrastre $CD(Rn)$ es una función de el numero de Reynolds: $Rn = UD/\nu$ Donde D es la longitud característica del cuerpo y μ es el coeficiente viscosidad cinemática ($\nu=1.56*10^{-6}$ para agua salada a 5° C con salinidad de 3.5

$$\mathbf{D}_n = \begin{bmatrix} |\nu| \mathbf{D}_{n1} \nu \\ |\nu| \mathbf{D}_{n2} \nu \\ |\nu| \mathbf{D}_{n3} \nu \\ |\nu| \mathbf{D}_{n4} \nu \\ |\nu| \mathbf{D}_{n5} \nu \\ |\nu| \mathbf{D}_{n6} \nu \end{bmatrix} \quad (3.43)$$

Donde $|\nu| = [|u|, |v|, |w|, |p|, |q|, |r|]^T$ y \mathbf{D}_{ni} son matrices cuadradas de dimensión 6 que dependen de ρ_a , C_D y A_t . Los diferentes términos de amortiguación contribuyen a las amortiguaciones lineales y cuadráticas. Sin embargo, es generalmente difícil de separar esos efectos. En muchos casos es conveniente escribir el amortiguamiento hidrodinámico total como:

$$\mathbf{D}(\nu) = \mathbf{D}_l + \mathbf{D}_n \quad (3.44)$$

Donde \mathbf{D}_l es la matriz de amortiguamiento lineal y \mathbf{D}_n es la matriz de amortiguación no lineal. La matriz \mathbf{D} tiene las siguientes propiedades: es real, no simétrica y estrictamente positiva.

En general, el amortiguamiento de un vehículo submarino moviéndose en el espacio tridimensional a altas velocidades es altamente no lineal y acoplado. Sin embargo, una aproximación a groso modo es asumir que el robot realiza un movimiento desacoplado. Esto sugiere que la estructura de la matriz \mathbf{D} con solo términos lineales o cuadráticos

en la diagonal:

$$\begin{aligned} \mathbf{D} = & -\text{diag}\{X_u, Y_v, Z_w, K_p, M_q, N_r\} \\ & -\text{diag}\{X_{|u|u}|u|, Y_{|v|v}|v|, Z_{|w|w}|w|, K_{|p|p}|p|, M_{|q|q}|q|, N_{|r|r}|r|\} \end{aligned} \quad (3.45)$$

3.3.4. Fuerzas de Restitución

Además de las fuerzas hidrodinámicas, los vehículos submarinos son afectados por la gravedad y las fuerza de flotación. Estas fuerzas son llamadas fuerzas de restitución.

La fuerza gravitacional \mathbf{f}_g^b actúan sobre el centro de gravedad el cual esta definido por el vector $\mathbf{r}_g^b = [x_g \ y_g \ z_g]^T$. De manera similar, la fuerza de flotación \mathbf{f}_{by}^b actuan sobre el centro de flotación definido por el vector $\mathbf{r}_{by}^b = [x_{by} \ y_{by} \ z_{by}]^T$. El peso de un cuerpo sumergido en el agua y la fuerza de flotación están definidos de la siguiente manera:

$$W = mg \quad (3.46)$$

$$B = \rho_a g V \quad (3.47)$$

donde g es la aceleración de la gravedad y V es el volumen desplazado por el robot.

Por lo tanto:

$$\mathbf{f}_g^n = \begin{bmatrix} 0 & 0 & W \end{bmatrix}^T \quad (3.48)$$

y

$$\mathbf{f}_{by}^n = - \begin{bmatrix} 0 & 0 & B \end{bmatrix}^T \quad (3.49)$$

Las fuerzas de gravedad y flotación se pueden transformar al marco de referencia del cuerpo simplemente multiplicando por la inversa de matriz de rotación \mathbf{R}_b^n , i.e. $\mathbf{f}_g^b = \mathbf{R}_b^{n-1} \mathbf{f}_g^n$ y $\mathbf{f}_{by}^b = \mathbf{R}_b^{n-1} \mathbf{f}_{by}^n$.

Por lo tanto la fuerza y el momento de restitución en el marco de referencia del cuerpo esta dado de la siguiente manera:

$$\mathbf{g}(\boldsymbol{\nu}) = - \begin{bmatrix} \mathbf{f}_g^b + \mathbf{f}_{by}^b \\ \mathbf{r}_g^b \times \mathbf{f}_g^b + \mathbf{r}_{by}^b \times \mathbf{f}_{by}^b \end{bmatrix} \quad (3.50)$$

sustituyendo los elementos que cada vector, se tiene los siguientes valores para el vector de fuerzas de restitución:

$$\mathbf{g}(\boldsymbol{\nu}) = \begin{bmatrix} (W - B) \sin \theta \\ -(W - B) \cos \theta \sin \phi \\ -(W - B) \cos \theta \cos \phi \\ -(y_g W - y_{by} B) \cos \theta \cos \phi + (z_g W - z_{by} B) \cos \theta \sin \phi \\ (z_g W - z_{by} B) \sin \theta + (x_g W - x_{by} B) \cos \theta \cos \phi \\ -(x_g W - x_{by} B) \cos \theta \sin \phi - (y_g W - y_{by} B) \sin \theta \end{bmatrix} \quad (3.51)$$

3.4. Resumen

En este capítulo se hizo una revisión de los fundamentos matemáticos para el modelado de robots submarinos. Brevemente se explicaron las convenciones para describir el movimiento de estos robots. El modelado cinemático consiste de las ecuaciones que relacionan los diferentes tipos de coordenadas utilizadas para expresar la velocidad del robot. Por otro lado, en el modelado dinámico se describen las fuerzas que actúan sobre el robot cuando este se navega. En la sección de modelado dinámico se presentó de manera concisa la derivación de las ecuaciones de movimiento con el objetivo aclarar su forma final.

Parte II

Desarrollo

Capítulo 4

Modelado del Robots Submarinos de Estructura Paralela con Impulsores Vectorizados

En este capítulo se presenta la derivación de las ecuaciones de movimiento de la estructura de robot que se ha elegido en esta tesis para estudiar (ver fig. 4.1). Primero se presenta el modelado cinemático inverso y posteriormente el modelado dinámico inverso. Se analizan diversos casos de estudio sobre el movimiento del robot. Al final de este capítulo se presenta una estrategia para el cálculo paralelizado de la dinámica inversa.



Figura 4.1: Remo 2

4.1. Cinemática del Robot Remo II

En esta sección se resuelve la cinemática inversa del robot Remo II. Por lo tanto, dado el movimiento deseado del robot, se deben calcular la posición, velocidad y

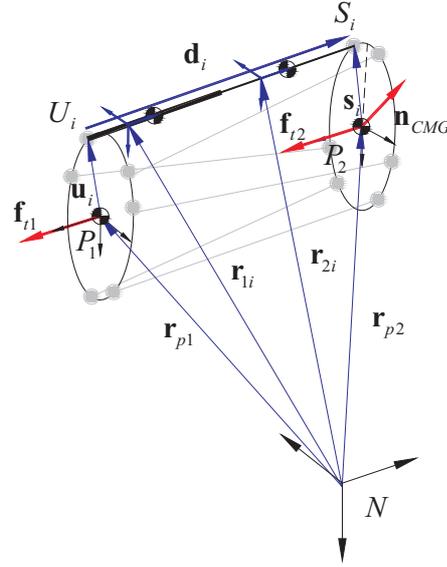


Figura 4.2: Esquema de Remo 2.

aceleración de los cuerpos de cada pierna, y además los valores de las variables articulares. La trayectoria deseada del robot puede ser expresada en términos de la posición y orientación de ambas plataformas (i.e. η_{p1} y η_{p2}) y sus derivadas con respecto al tiempo. Para el análisis, se fija un marco de referencia P_1 a la plataforma 1, y otro marco de referencia P_2 a la plataforma 2. La articulaciones universales U_i esta ubicada sobre la plataforma 1, y las articulaciones esféricas S_i están ubicadas sobre la plataforma 2. Adicionalmente definimos dos marcos de referencia en el centro de masa de cada eslabón de la i -ésima pierna L_{1i} y L_{2i} . Debido a que ambos marcos de referencia tienen la misma orientación definimos un marco de referencia auxiliar L_i a cada pierna (con origen en U_i) con el objetivo de simplificar la nomenclatura de las ecuaciones. Véase la figura 4.2.

4.1.1. Analisis de Posición

Considerese la figura 4.2, el vector asociado a la i -ésima pierna puede ser encontrado de la siguiente forma

$$\mathbf{d}_i^n = \mathbf{r}_{p2}^n + \mathbf{R}_{p2}^n \mathbf{s}_i - (\mathbf{r}_{p1}^n + \mathbf{R}_{p1}^n \mathbf{u}_i) \quad (4.1)$$

Donde \mathbf{r}_{p1} es el vector de posición del marco P_1 con respecto al marco de referencia fijo, \mathbf{r}_{p2} es el vector de posición del marco P_2 con respecto al marco de referencia fijo. \mathbf{R}_{p1} y \mathbf{R}_{p2} son las matrices de rotación de la plataforma 1 y la plataforma 2 con respecto al marco de referencia fijo. El vector \mathbf{u}_i denota la posición de la junta universal U_i en la plataforma 1, y el vector \mathbf{s}_i representa la posición del vector de la articulación esférica S_i en la plataforma 2. La dirección del vector unitario asociado a la i -ésima pierna esta dado por $\hat{\mathbf{d}}_i^n = \mathbf{d}_i^n / \|\mathbf{d}_i^n\|$

Dada la posición y orientación de ambas plataformas, la posición de los marcos de referencia de los eslabones se calcula de la siguiente manera:

$$\mathbf{r}_{1i}^n = \mathbf{r}_{p1}^n + \mathbf{R}_{p1}^n \mathbf{u}_i - \mathbf{r}_{ui}^n \quad (4.2)$$

and

$$\mathbf{r}_{2i}^n = \mathbf{r}_{p2}^n + \mathbf{R}_{p2}^n \mathbf{s}_i - \mathbf{r}_{si}^n \quad (4.3)$$

donde $\mathbf{r}_{ui}^n = -l_u \hat{\mathbf{d}}_i^n$ y $\mathbf{r}_{si}^n = l_s \hat{\mathbf{d}}_i^n$. l_u es la distancia entre el centro de masa del eslabon 1 y la articulación universal. l_s es la distancia entre el centro de masa del eslabón 2 y la articulación esférica. Los marcos de referencia unidos a ambos cuerpos de la pierna se definen considerando que el eje $\hat{\mathbf{z}}_i^n$ esta a lo lardo de la dirección del vector $\hat{\mathbf{d}}_i^n$; el eje $\hat{\mathbf{y}}_i^n$ es normal al plano dado por $\hat{\mathbf{z}}_i^n$ y \mathbf{s}_i ; finalmente $\hat{\mathbf{x}}_i^n$ es ortogonal a $\hat{\mathbf{z}}_i^n$ y $\hat{\mathbf{y}}_i^n$. Por lo tanto, la orientación de la i-esima pierna con respecto al marco de referencia N puede ser expresada de acuerdo a la matriz de rotación dada por :

$$\mathbf{R}_i^n = \begin{bmatrix} \hat{\mathbf{x}}_i^n & \hat{\mathbf{y}}_i^n & \hat{\mathbf{z}}_i^n \end{bmatrix} \quad (4.4)$$

donde $\hat{\mathbf{z}}_i^n = \hat{\mathbf{d}}_i^n$, $\hat{\mathbf{y}}_i^n = (\mathbf{s}_i^n \times \hat{\mathbf{z}}_i^n) / \|\mathbf{s}_i^n \times \hat{\mathbf{z}}_i^n\|$ y $\hat{\mathbf{x}}_i^n = \hat{\mathbf{y}}_i^n \times \hat{\mathbf{z}}_i^n$. Debido a que L_{1i} y L_{2i} tienen la misma orientación que L_i , entonces $\mathbf{R}_{1i}^n = \mathbf{R}_{2i}^n = \mathbf{R}_i^n$. La variable articular se obtiene de la siguiente manera:

$$\rho_i = \|\mathbf{d}_i^n\| - l_m \quad (4.5)$$

donde l_m es la longitud muerta del actuador

4.1.2. Analisis de Velocidad

Con el objetivo obtener el modelo de velocidad del robot, se calcula primero la derivada de (4.1):

$$\mathbf{v}_{di}^n = \mathbf{v}_{p2}^n + \omega_{p2}^n \times \mathbf{s}_i^n - (\mathbf{v}_{p1}^n + \omega_{p1}^n \times \mathbf{u}_i^n) \quad (4.6)$$

donde \mathbf{v}_{p1}^n y ω_{p1}^n son las velocidades lineales y angulares del marco de referencia P_1 , respectivamente, y \mathbf{v}_{p2}^n y ω_{p2}^n son las velocidades lineales y angulares del marco de referencia P_2 . Al mismo tiempo \mathbf{v}_{di}^n tiene dos componentes:

$$\mathbf{v}_{di}^n = \rho_i \omega_{di}^n \times \hat{\mathbf{d}}_i^n + \dot{\mathbf{d}}_i^n \quad (4.7)$$

donde ω_{di}^n es la velocidad angular de la i-esima pierna y $\dot{\mathbf{d}}_i^n$ es la velocidad lineal a lo largo del eje de la i-esima pierna. La velocidad angular se obtiene de la siguiente manera:

$$\omega_{di}^n = \frac{\mathbf{d}_i^n \times \mathbf{v}_{di}^n}{\|\mathbf{d}_i^n\|^2} \quad (4.8)$$

y $\dot{\mathbf{d}}_i^n$:

$$\dot{\mathbf{d}}_i^n = \dot{\rho} \hat{\mathbf{d}}_i^n \quad (4.9)$$

La velocidad de los marcos de referencia de los eslabones puede ser encontrada de la siguiente manera:

$$\mathbf{v}_{1i}^n = \mathbf{v}_{p1}^n + \omega_{p1}^n \times \mathbf{u}_i^n - \omega_{di}^n \times \mathbf{r}_u^n \quad (4.10)$$

y

$$\mathbf{v}_{2i}^n = \mathbf{v}_{p2}^n + \omega_{p2}^n \times \mathbf{s}_i^n - \omega_{di}^n \times \mathbf{r}_s^n \quad (4.11)$$

La velocidad de las articulaciones se puede calcular haciendo el producto cruz de $\hat{\mathbf{d}}_i^n$ por (4.6):

$$\dot{\rho}_i = \mathbf{v}_{p2}^n \cdot \hat{\mathbf{d}}_i^n + (\mathbf{s}_i^n \times \hat{\mathbf{d}}_i^n) \cdot \omega_{p2}^n - (\mathbf{v}_{p1}^n \cdot \hat{\mathbf{d}}_i^n + (\mathbf{u}_i^n \times \hat{\mathbf{d}}_i^n) \cdot \omega_{p1}^n) \quad (4.12)$$

La expresión anterior obtiene la velocidad del actuador dada la velocidad lineal y angular de ambas plataformas.

4.1.3. Analisis de Aceleración

Con el objetivo de obtener la aceleración de los modelos del robot, las ecuaciones de las subsecciones anteriores pueden ser derivadas. Derivando (4.6) obtenemos:

$$\mathbf{a}_{di}^n = \mathbf{a}_{si}^n - \mathbf{a}_{ui}^n \quad (4.13)$$

donde \mathbf{a}_{di}^n es la aceleración del vector de la pierna,

$$\mathbf{a}_{di}^n = \ddot{\mathbf{d}}_i^n + 2\omega_{di}^n \times \dot{\mathbf{d}}_i^n + \dot{\omega}_{di}^n \times \mathbf{d}_i^n + \omega_{di}^n \times (\omega_{di}^n \times \mathbf{d}_i^n) \quad (4.14)$$

y, \mathbf{a}_{ui} y \mathbf{a}_{si} son las aceleraciones de los puntos U_i y S_i ,

$$\mathbf{a}_{ui}^n = \mathbf{a}_{p1}^n + \dot{\omega}_{p1}^n \times \mathbf{u}_i^n + \omega_{p1}^n \times (\omega_{p1}^n \times \mathbf{u}_i^n) \quad (4.15)$$

y

$$\mathbf{a}_{si}^n = \mathbf{a}_{p2}^n + \dot{\omega}_{p2}^n \times \mathbf{s}_i^n + \omega_{p2}^n \times (\omega_{p2}^n \times \mathbf{s}_i^n) \quad (4.16)$$

Los vectores \mathbf{a}_{p1}^n y $\dot{\omega}_{p1}^n$ representan las aceleraciones lineales y angulares de la plataforma 1, \mathbf{a}_{p2}^n y $\dot{\omega}_{p2}^n$ denotan las aceleraciones lineales y angulares de la plataforma 2.

La aceleración angular de la i-esima pierna es:

$$\dot{\omega}_{di}^n = \hat{\mathbf{d}}_i^n \times (\mathbf{a}_{si}^n - \mathbf{a}_{ui}^n) - 2\dot{\rho}_i \omega_{di}^n \quad (4.17)$$

La aceleración de los marcos unidos a los eslabones de la pierna se obtiene de la siguiente manera:

$$\mathbf{a}_{1i}^n = \mathbf{a}_{ui}^n - (\dot{\omega}_{di}^n \times \mathbf{r}_{ui}^n + \omega_{di}^n \times (\omega_{di}^n \times \mathbf{r}_{ui}^n)) \quad (4.18)$$

y

$$\mathbf{a}_{2i}^n = \mathbf{a}_{si}^n - (\dot{\omega}_{di}^n \times \mathbf{r}_{si}^n + \omega_{di}^n \times (\omega_{di}^n \times \mathbf{r}_{si}^n)) \quad (4.19)$$

Generalmente, las ecuaciones de movimiento de robots submarinos se expresan en terminus de la velocidad lineal expresada en el marco de referencia del cuerpo y su derivada con respecto al tiempo Fossen (1991). Por lo tanto, para los eslabones de las piernas tenemos que:

$$\dot{\mathbf{v}}_{1i}^i = \mathbf{R}_i^{nT} \mathbf{a}_{1i}^n - \omega_{di}^i \times \mathbf{v}_{1i}^i \quad (4.20)$$

y

$$\dot{\mathbf{v}}_{2i}^i = \mathbf{R}_i^{nT} \mathbf{a}_{2i}^n - \omega_{di}^i \times \mathbf{v}_{2i}^i \quad (4.21)$$

Finalmente, la aceleración de la articulación se obtiene haciendo el producto punto entre $\hat{\mathbf{d}}_i^n$ y \mathbf{a}_{di}^n de la ecuación (4.13):

$$\ddot{\rho} = \hat{\mathbf{d}}_i^n \cdot (\mathbf{a}_{si}^n - \mathbf{a}_{ui}^n) \quad (4.22)$$

4.1.4. Análisis del Jacobiano

Arreglando (4.12) para cada pierna en forma matricial, el siguiente modelo cinemático es obtenido:

$$\mathbf{J}_{\rho i} \dot{\rho} = \mathbf{J}_{p2} \mathbf{t}_{p2} - \mathbf{J}_{p1} \mathbf{t}_{p1} \quad (4.23)$$

donde $\dot{\rho} = [\dot{\rho}_1 \quad \dot{\rho}_2 \quad \dots \quad \dot{\rho}_6]^T$ es el vector de velocidades articulares $\mathbf{t}_{p1} = [\mathbf{v}_{p1}^T \quad \omega_{p1}^T]^T$ y $\mathbf{t}_{p2} = [\mathbf{v}_{p2}^T \quad \omega_{p2}^T]^T$ son las velocidades de las plataformas 1 y 2, respectivamente, las matrices $\mathbf{J}_{\rho i}$, \mathbf{J}_{p1} y \mathbf{J}_{p2} son las matrices Jacobiana del robot. Las matrices Jacobianas tienen la siguiente forma:

$$\mathbf{J}_{\rho i} = \mathbf{I}_{6 \times 6}, \quad (4.24)$$

donde $\mathbf{I}_{6 \times 6}$ es la matriz de identidad de dimensión 6.

$$\mathbf{J}_{p1} = \begin{bmatrix} \hat{\mathbf{d}}_1^{nT} & (\mathbf{u}_1^n \times \hat{\mathbf{d}}_1^n)^T \\ \hat{\mathbf{d}}_2^{nT} & (\mathbf{u}_2^n \times \hat{\mathbf{d}}_2^n)^T \\ \vdots & \vdots \\ \hat{\mathbf{d}}_6^{nT} & (\mathbf{u}_6^n \times \hat{\mathbf{d}}_6^n)^T \end{bmatrix} \quad (4.25)$$

y

$$\mathbf{J}_{p2} = \begin{bmatrix} \hat{\mathbf{d}}_1^{nT} & (\mathbf{s}_1^n \times \hat{\mathbf{d}}_1^n)^T \\ \hat{\mathbf{d}}_2^{nT} & (\mathbf{s}_2^n \times \hat{\mathbf{d}}_2^n)^T \\ \vdots & \vdots \\ \hat{\mathbf{d}}_6^{nT} & (\mathbf{s}_6^n \times \hat{\mathbf{d}}_6^n)^T \end{bmatrix} \quad (4.26)$$

Además, expandiendo (4.22), el modelo de aceleración en la forma matricial puede ser de la siguiente manera:

$$\mathbf{J}_{\rho} \ddot{\rho} = \mathbf{J}_{p2} \dot{\mathbf{t}}_{p2} - \mathbf{J}_{p1} \dot{\mathbf{t}}_{p1} + \mathbf{K} \quad (4.27)$$

donde $\ddot{\rho} = [\ddot{\rho}_1 \quad \ddot{\rho}_2 \quad \dots \quad \ddot{\rho}_6]^T$ es el vector de aceleraciones articulares $\dot{\mathbf{t}}_{p1} = [\mathbf{a}_{p1}^{nT} \quad \dot{\omega}_{p1}^{nT}]^T$ y $\dot{\mathbf{t}}_{p2} = [\mathbf{a}_{p2}^{nT} \quad \dot{\omega}_{p2}^{nT}]^T$ son las derivadas con respecto al tiempo de los vectores de velocidad, y \mathbf{K} es un vector que tiene la siguiente forma:

$$\mathbf{K} = \begin{bmatrix} \omega_1 d_1 + (\omega_{p1}^n \times \mathbf{u}_1^n \times \omega_{p1}^n - \omega_{p2}^n \times \mathbf{s}_1^n \times \omega_{p2}^n)^T \mathbf{d}_1^n \\ \omega_2 d_2 + (\omega_{p1}^n \times \mathbf{u}_2^n \times \omega_{p1}^n - \omega_{p2}^n \times \mathbf{s}_2^n \times \omega_{p2}^n)^T \mathbf{d}_2^n \\ \vdots \\ \omega_6 d_6 + (\omega_{p1}^n \times \mathbf{u}_6^n \times \omega_{p1}^n - \omega_{p2}^n \times \mathbf{s}_6^n \times \omega_{p2}^n)^T \mathbf{d}_6^n \end{bmatrix} \quad (4.28)$$

donde $\omega_1 = \|\omega_{d1}^n\|$ y $d_i = \|\mathbf{d}_i^n\|$.

Las matrices (4.25) y (4.26) tienen propiedades interesantes para evaluar el desempeño de el robot. Cuando $\det(\mathbf{J}_{p1}) = 0$ or $\det(\mathbf{J}_{p2}) = 0$ la estructura paralela del robot esta en una configuración singular, y esto puede ocasionar una pérdida de rigidez Gosselin and Angeles (1990). Por lo tanto es necesario evitar las configuraciones singulares en el cual las matrices (4.25) and (4.26) se vuelven singulares.

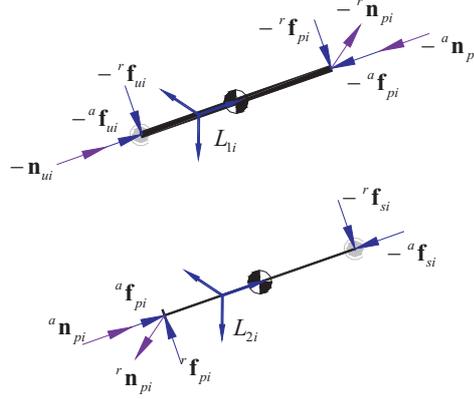


Figura 4.3: Diagrama de cuerpo libre de las piernas de Remo II

4.2. Dinámica del Robot Remo II

El análisis dinámico del robot paralelo es complicado debido a que tiene múltiples cadenas cinemáticas. Diferentes enfoques pueden ser encontrados en la literatura, como lo son la formulación de Newton-Euler Khalil and Guegan (2004); Gosselin (1996), la formulación Lagrangiana Bhattacharya et al. (1997); Liu et al. (2000), y el principio del trabajo virtual Staicu (2011); Tsai (2000). En este trabajo estamos interesados en calcular la dinámica del robot en múltiples procesadores de manera similar al enfoque presentado en Gosselin (1996). El objetivo es usar la estrategia presentada en esta sección para el control y la planificación de trayectorias.

Esta estrategia consiste en separa la ecuación de movimiento de cada pierna en las componentes radiales y axiales. Por lo tanto, las fuerzas de contacto entre las piernas y las plataformas se dividen en dos componentes, axiales y radiales. Primero, considérese el diagrama de cuerpo libre de la i -ésima pierna, véase figura 4.3. Se obtiene la siguiente ecuación para el eslabón 1:

$$-{}^r\mathbf{f}_{pi}^i - {}^r\mathbf{f}_{ui}^i = {}^r\mathbf{f}_{T1i}^i \quad (4.29)$$

donde ${}^r\mathbf{f}_{T1i}^i$ es el componente radial de la fuerza externa debido a los efectos, inerciales, hidrodinámicos y de restitución.

Adicionalmente, existe un torque generado por los efectos inerciales, hidrodinámicos y restitución, \mathbf{n}_{T1i}^i . Si $\bar{\mathbf{n}}_{T1i}^i = [n_{T1ix} \quad n_{T1iy}]^T$ es un subvector de \mathbf{n}_{T1i}^i , entonces la ecuación de movimiento se puede escribir de la siguiente forma:

$$-{}^i\mathbf{r}_{1pi}^i \times {}^r\mathbf{f}_{pi}^i - {}^i\mathbf{r}_{ui}^i \times {}^r\mathbf{f}_{ui}^i - {}^r\mathbf{n}_{pi}^i = \begin{bmatrix} \bar{\mathbf{n}}_{T1i}^i \\ 0 \end{bmatrix} \quad (4.30)$$

donde ${}^i\mathbf{r}_{1pi}^i$ es el vector de posición de la articulación prismática P_i con respecto a los marcos de referencia de los eslabones, ${}^i\mathbf{r}_{ui}^i = [0 \quad 0 \quad -l_u]^T$ es el vector de posición de la articulación S_i con respecto al marco de referencia de la pierna, ${}^r\mathbf{f}_{pi}^i = [{}^r f_{pix}^i \quad {}^r f_{piy}^i \quad 0]^T$ es la fuerza radial aplicada en la articulación prismática y

referido al marco de referencia de la pierna, ${}^r\mathbf{f}_{ui}^i = \begin{bmatrix} {}^r f_{uix}^i & {}^r f_{uiy}^i & 0 \end{bmatrix}^T$ es la fuerza radial aplicada en el punto U_i con respecto al marco de referencia de la pierna.

Por otro lado, se tiene el siguiente conjunto de ecuaciones de los componentes radiales para el eslabón 2:

$${}^r\mathbf{f}_{pi}^i - {}^r\mathbf{f}_{si}^i = {}^r\mathbf{f}_{T2i}^i \quad (4.31)$$

donde ${}^r\mathbf{f}_{T2i}^i$ es la componente radial de la fuerza externa total sobre el eslabón 2 de la i -ésima pierna.

Adicionalmente, existe un torque generado por los efectos inerciales, hidrodinámicos, y de restitución, \mathbf{n}_{T2i}^i . Si $\bar{\mathbf{n}}_{T2i}^i = \begin{bmatrix} n_{T2ix} & n_{T2iy} \end{bmatrix}^T$ es un subvector de \mathbf{n}_{T2i}^i , entonces se puede escribir la siguiente ecuación de movimiento:

$${}^i\mathbf{r}_{2pi}^i \times {}^r\mathbf{f}_{pi}^i - {}^i\mathbf{r}_{si}^i \times {}^r\mathbf{f}_{si}^i + {}^r\mathbf{n}_{pi}^i = \begin{bmatrix} \bar{\mathbf{n}}_{T2i}^i \\ 0 \end{bmatrix} \quad (4.32)$$

donde ${}^i\mathbf{r}_{2pi}^i$ es un vector de posición de la articulación prismática P_i con respecto al marco de referencia de la pierna, ${}^i\mathbf{r}_{si}^i = \begin{bmatrix} 0 & 0 & l_s \end{bmatrix}^T$ es un vector de posición de la articulación S_i con respecto al marco de referencia de la pierna, ${}^r\mathbf{f}_{si}^i = \begin{bmatrix} {}^r f_{uix}^i & {}^r f_{uiy}^i & 0 \end{bmatrix}^T$ es la fuerza radial aplicada en el punto S_i con respecto al marco de referencia de la pierna

El conjunto de ecuaciones (4.29) a (4.32) pueden escribirse en una forma mas compacta. Si ${}^r\mathbf{F}_i$ es el conjunto de las fuerzas radiales aplicadas en las articulaciones de la i -ésima pierna:

$${}^r\mathbf{F}_i = \begin{bmatrix} {}^r\bar{\mathbf{f}}_{ui}^i \\ {}^r\bar{\mathbf{f}}_{pi}^i \\ {}^r\bar{\mathbf{f}}_{si}^i \\ {}^r\bar{\mathbf{n}}_{pi}^i \end{bmatrix}$$

si el vector \mathbf{f} es el conjunto de fuerzas inerciales e hidrodinámicas generadas en un plano ortogonal del eje de la i -ésima pierna:

$${}^r\mathbf{f}_{T12i} = \begin{bmatrix} {}^r\bar{\mathbf{f}}_{T1i}^i \\ {}^r\bar{\mathbf{f}}_{T2i}^i \\ {}^r\bar{\mathbf{n}}_{T1i}^i \\ {}^r\bar{\mathbf{n}}_{T2i}^i \end{bmatrix}$$

donde ${}^i\bar{\mathbf{f}}_{ui}^i$, ${}^i\bar{\mathbf{f}}_{pi}^i$, ${}^i\bar{\mathbf{f}}_{si}^i$ y ${}^i\bar{\mathbf{n}}_{pi}^i$, son vectores de dos dimensiones que representan las fuerzas y torques radiales aplicados en las articulaciones.

De esta manera, se puede obtener un sistema de 8 ecuaciones con 8 incógnitas para cada pierna:

$$\mathbf{L}_i {}^r\mathbf{F}_i = {}^r\mathbf{f}_{T12i} \quad (4.33)$$

\mathbf{L}_i es una matriz de coeficientes que depende de la posición de las fuerzas aplicadas :

$$\mathbf{L}_i = \begin{bmatrix} -\mathbf{I}_2 & -\mathbf{I}_2 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_2 & -\mathbf{I}_2 & \mathbf{0} \\ l_{ui}\mathbf{E} & -l_{1pi}(\rho_i)\mathbf{E} & \mathbf{0} & -\mathbf{I}_2 \\ \mathbf{0} & -l_{2pi}\mathbf{E} & -l_{si}\mathbf{E} & \mathbf{I}_2 \end{bmatrix} \quad (4.34)$$

donde $l_{1pi}(\rho_i) = \rho_i + c_u - l_{ui}$ y $l_{2pi} = \|\mathbf{r}_{2pi}^i\|$, son las distancias entre los marcos de referencia, L_{1i} y L_{2i} , y la cabeza de la articulación prismática, respectivamente. La matriz \mathbf{I}_2 es una matriz identidad de dimensión 2. \mathbf{E} representa una rotación en contra de las manecillas del reloj de 90° grados:

$$\mathbf{E} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \quad (4.35)$$

Resolviendo (4.33) para cada pierna, las fuerzas radiales aplicadas en las articulaciones universal y esféricas pueden ser obtenidas. La inversa de cada matriz \mathbf{L}_i puede ser calculada concurrentemente en 6 hilos de ejecución.

Las ecuaciones para las fuerzas axiales son las siguientes:

$$-{}^a\mathbf{f}_{ui}^i - {}^a\mathbf{f}_{pi}^i = {}^a\mathbf{f}_{T1i}^i \quad (4.36)$$

y

$${}^a\mathbf{f}_{pi}^i - {}^a\mathbf{f}_{si}^i = {}^a\mathbf{f}_{T2i}^i \quad (4.37)$$

Las ecuaciones para los torques axiales son los siguientes:

$$-{}^a\mathbf{n}_{ui}^i - {}^a\mathbf{n}_{pi}^i = {}^a\mathbf{n}_{T1i}^i \quad (4.38)$$

y

$${}^a\mathbf{n}_{pi}^i = {}^a\mathbf{n}_{T2i}^i \quad (4.39)$$

Considérese la figura 4.4, se puede obtener la siguiente ecuación de movimiento:

$$\mathbf{f}_{t1}^{p1} + \sum {}^a\mathbf{f}_{ui}^{p1} + \sum {}^r\mathbf{f}_{ui}^{p1} = \mathbf{f}_{Tp1}^{p1} \quad (4.40)$$

y

$$\sum \mathbf{u}_i \times ({}^a\mathbf{f}_{ui}^{p1} + {}^r\mathbf{f}_{ui}^{p1}) + \sum \mathbf{n}_{ui}^{p1} = \mathbf{n}_{Tp1}^{p1} \quad (4.41)$$

Por otro lado para la plataforma 2 (Fig. 4.5), se pueden obtener las siguientes ecuaciones de movimiento:

$$\mathbf{f}_{t2}^{p2} + \sum {}^a\mathbf{f}_{si}^{p2} + \sum {}^r\mathbf{f}_{si}^{p2} = \mathbf{f}_{Tp2}^{p2} \quad (4.42)$$

y

$$\mathbf{n}_{CMG}^{p2} + \sum \mathbf{s}_i \times ({}^a\mathbf{f}_{si}^{p2} + {}^r\mathbf{f}_{si}^{p2}) = \mathbf{n}_{Tp2}^{p2} \quad (4.43)$$

Para nuestros propósitos, escribimos las ecuaciones (4.40) y (4.41) en términos de ${}^a\mathbf{f}_{si}$ utilizando las ecuaciones (4.36) y (4.85). Por lo tanto:

$$\mathbf{f}_{t1}^{p1} - \sum {}^a\mathbf{f}_{si}^{p1} = \mathbf{f}_{Tp1}^{p1} - \sum {}^r\mathbf{f}_{ui}^{p1} + \sum \mathbf{R}_i^{p1} ({}^a\mathbf{f}_{T1i}^i + {}^a\mathbf{f}_{T2i}^i) \quad (4.44)$$

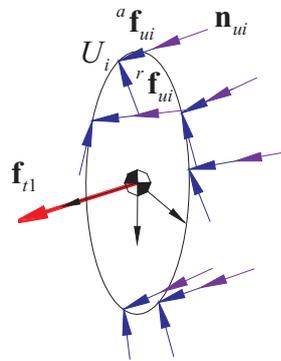


Figura 4.4: Diagrama de cuerpo libre de la Plataforma 1

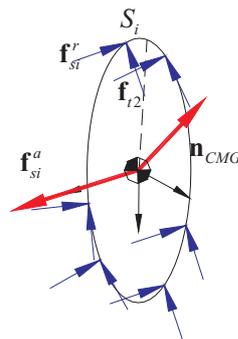


Figura 4.5: Diagrama de cuerpo libre de la Plataforma 2

y

$$\begin{aligned} \sum \mathbf{u}_i \times \mathbf{f}_{si}^{p1} = & -\mathbf{n}_{Tp1}^{p1} + \sum \mathbf{n}_{ui}^{p1} + \sum \mathbf{u}_i \times {}^r \mathbf{f}_{ui}^{p1} \\ & - \mathbf{u}_i \times \mathbf{R}_i^{p1} ({}^a \mathbf{f}_{T1i}^i + {}^a \mathbf{f}_{T2i}^i) \end{aligned} \quad (4.45)$$

Las ecuaciones anteriores representa un sistema sobre determinado de 12 ecuaciones con 11 incógnitas. Las ingonitas son las 6 fuerzas axiales en cada una de las piernas, la magnitud de las fuerzas en los impulsores, y el torque generado por el CMG. Para completar el sistema seria necesario tener un elemento generador de fuerza adicional como lo puede ser un impulsor. Estas ecuaciones pueden ser resueltas consideran el tipo de movimiento que el robot realiza y estableciendo algunas condiciones en algunas variables. Por ejemplo, considere el simple ejemplo del robot moviéndose en línea recta. Con el objetivo de distribuir la carga en ambos impulsores, se impone la condición de que \mathbf{f}_{t1} y \mathbf{f}_{t2} sean iguales. De esta manera, la ecuación (4.44) se substituye en (4.42). Con la ecuación resultante y (4.45), es posible formar un sistema de 6 ecuaciones con 6 incógnitas. Dado que ${}^a \mathbf{f}_{si}^{p1} = \hat{\mathbf{d}}_{si}^{p1} {}^a f_{si}$, $\mathbf{R}_{p1}^n = \mathbf{R}_{p2}^n$ y $\mathbf{R}_i^{p1} = \mathbf{R}_i^{p2}$, se tiene que :

$$\mathbf{C}_l {}^a \mathbf{F}_s = \mathbf{Q}_l \quad (4.46)$$

donde el vector ${}^a \mathbf{F}_s = [{}^a f_{s1} \quad {}^a f_{s2} \quad \dots \quad {}^a f_{s6}]^T$ contiene las fuerzas axiales aplicadas a las piernas. La matriz de coeficientes \mathbf{C}_s está dada por:

$$\mathbf{C}_l = \begin{bmatrix} 2\hat{\mathbf{d}}_{s1}^{p1} & 2\hat{\mathbf{d}}_{s2}^{p1} & \dots & 2\hat{\mathbf{d}}_{s6}^{p1} \\ \mathbf{u}_1 \times \hat{\mathbf{d}}_{s1}^{p1} & \mathbf{u}_2 \times \hat{\mathbf{d}}_{s2}^{p1} & \dots & \mathbf{u}_6 \times \hat{\mathbf{d}}_{s6}^{p1} \end{bmatrix} \quad (4.47)$$

y el vector \mathbf{Q}_l :

$$\mathbf{Q}_l = \begin{bmatrix} \mathbf{f}_{Tp2}^{p1} - \sum {}^r \mathbf{f}_{si}^{p1} - [\mathbf{f}_{Tp1}^{p1} - \sum {}^r \mathbf{f}_{ui}^{p1} + \sum \mathbf{R}_i^{p1} ({}^a \mathbf{f}_{T1i}^i + {}^a \mathbf{f}_{T2i}^i)] \\ -\mathbf{n}_{Tp1}^{p1} + \sum \mathbf{n}_{ui}^{p1} + \sum \mathbf{u}_i \times {}^r \mathbf{f}_{ui}^{p1} - \sum \mathbf{u}_i \times \mathbf{R}_i^{p1} ({}^a \mathbf{f}_{T1i}^i + {}^a \mathbf{f}_{T2i}^i) \end{bmatrix} \quad (4.48)$$

Con la resolución de la ecuación (4.46), Se substituyen los valores de ${}^a \mathbf{F}_s$ en las ecuaciones (4.42), (4.43) and (4.44) para determinar \mathbf{f}_{t2} , \mathbf{n}_{CMG} y \mathbf{f}_{t1} , respectivamente.

4.3. Parámetros del Robot

4.3.1. Consideraciones previas

En los estudios presentados en esta tesis consideramos que: los elementos del robot son simétricos en los tres planos, el origen del marco de referencia del cuerpo B coincide con el centro de masa, el centro de masa coincide con el centro de flotabilidad, y la fuerza de gravedad es igual a la fuerza de flotación (flotabilidad neutra). Esto será así a menos de que se indiquen otras consideraciones. Estas condiciones se pueden lograr mediante el adecuado diseño de los elementos mecánicos del robot.

Por lo tanto, las fuerzas inerciales, hidrodinámicas y de restitución se simplifican como se presentan a continuación.

Para las fuerzas inerciales tenemos que el vector \mathbf{r}_g^b es igual a cero, por lo tanto, la matriz de masa tiene la siguiente forma:

$$\mathbf{M}_R = \begin{bmatrix} m\mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_O \end{bmatrix} \quad (4.49)$$

Dado que $\mathbf{M}_{R12} = \mathbf{0}_3$ y $\mathbf{M}_{R21} = \mathbf{0}_3$, la matriz de fuerzas centrífugas de Coriolis tiene la siguiente forma:

$$\mathbf{C}_R(\boldsymbol{\nu}) = \begin{bmatrix} \mathbf{0}_3 & -\mathbb{S}(\mathbf{M}_{R11}\boldsymbol{\nu}_1) \\ -\mathbb{S}(\mathbf{M}_{R11}\boldsymbol{\nu}_1) & -\mathbb{S}(\mathbf{M}_{R22}\boldsymbol{\nu}_2) \end{bmatrix} \quad (4.50)$$

En el caso de las fuerzas hidrodinámicas, tenemos que la matriz de masa añadida tiene una forma diagonal, por tanto:

$$\mathbf{M}_A = \text{diag}\{X_{\dot{u}}, Y_{\dot{v}}, Z_{\dot{w}}, K_{\dot{p}}, M_{\dot{q}}, N_{\dot{r}}\} \quad (4.51)$$

Debido a lo anterior, la matriz de fuerzas centrífugas y de Coriolis de la masa añadida es la siguiente:

$$\mathbf{C}_A(\boldsymbol{\nu}) = \begin{bmatrix} \mathbf{0} & -\mathbb{S}(\mathbf{M}_{A11}\boldsymbol{\nu}_1) \\ -\mathbb{S}(\mathbf{M}_{A11}\boldsymbol{\nu}_1) & -\mathbb{S}(\mathbf{M}_{A22}\boldsymbol{\nu}_2) \end{bmatrix} \quad (4.52)$$

Como se comento anteriormente, la matriz de fuerzas viscosas tiene forma diagonal para cuerpos simétricos moviéndose a baja velocidad.

En el caso de las fuerzas de restitución, ya que $\mathbf{r}_g^b = \mathbf{r}_{by}^b = \mathbf{0}$ y $\mathbf{f}_g^b = \mathbf{f}_{by}^b$, se tiene que:

$$\mathbf{g}(\boldsymbol{\nu}) = \mathbf{0} \quad (4.53)$$

Para las simulaciones no consideramos otras fuerzas ambientales como lo pueden ser las corrientes marinas.

Dadas las expresiones anteriores, la fuerza total que se genera en cada uno de los elementos debido a los efectos inerciales e hidrodinámicos es la siguiente:

$$\mathbf{f}_T = (\mathbf{M}_{R11} + \mathbf{M}_{A11})\dot{\boldsymbol{\nu}}_1 - (\mathbb{S}(\mathbf{M}_{R11}\boldsymbol{\nu}_1) + \mathbb{S}(\mathbf{M}_{A11}\boldsymbol{\nu}_1))\boldsymbol{\nu}_2 + \mathbf{D}_1\boldsymbol{\nu}_1 \quad (4.54)$$

Debido a que $\mathbb{S}(\mathbf{M}_{R11}\boldsymbol{\nu}_1)\boldsymbol{\nu}_1 = \mathbf{0}$, el torque total es el siguiente:

$$\mathbf{n}_T = (\mathbf{M}_{R22} + \mathbf{M}_{A22})\dot{\boldsymbol{\nu}}_2 - \mathbb{S}(\mathbf{M}_{A11}\boldsymbol{\nu}_1)\boldsymbol{\nu}_1 - (\mathbb{S}(\mathbf{M}_{R22}\boldsymbol{\nu}_2) + \mathbb{S}(\mathbf{M}_{A22}\boldsymbol{\nu}_2))\boldsymbol{\nu}_2 + \mathbf{D}_2\boldsymbol{\nu}_2 \quad (4.55)$$

Escrito en términos de \mathbf{v}_o^b , $\boldsymbol{\omega}_o^b$ y sus derivadas con respecto al tiempo tenemos:

$$\mathbf{f}_T = (\mathbf{M}_{R11} + \mathbf{M}_{A11})\dot{\mathbf{v}}_o^b - (\mathbb{S}(\mathbf{M}_{R11}\mathbf{v}_o^b) + \mathbb{S}(\mathbf{M}_{A11}\mathbf{v}_o^b))\boldsymbol{\omega}_o^b + \mathbf{D}_1\mathbf{v}_o^b \quad (4.56)$$

y

$$\mathbf{n}_T = (\mathbf{M}_{R22} + \mathbf{M}_{A22})\dot{\boldsymbol{\omega}}_o^b - \mathbb{S}(\mathbf{M}_{A11}\mathbf{v}_o^b)\mathbf{v}_o^b - (\mathbb{S}(\mathbf{M}_{R22}\boldsymbol{\omega}_o^b) + \mathbb{S}(\mathbf{M}_{A22}\boldsymbol{\omega}_o^b))\boldsymbol{\omega}_o^b + \mathbf{D}_2\boldsymbol{\omega}_o^b \quad (4.57)$$

Tabla 4.1: Parámetros Geométricos del Robot

Parámetro		Valor
Longitud de carrera	l_{str}	243 mm
Longitud muerta	l_m	437 mm
Distancia entre U_i y L_{1i}	l_u	192 mm
Distancia entre S_i y L_{2i}	l_s	177 mm
Rad. ext. Cilindro	r_{ec}	15mm
Rad. int. Cilindro	r_{ic}	8 mm
Longitud del Cilindro	l_{ec}	320 mm
Profundidad int. del Cilindro	l_{ic}	315 mm
Radio Pistón	r_{ps}	8 mm
Longitud Pistón	l_{ps}	290 mm
Radio de la Esfera	r_s	110 mm
Radio del Cil. int. de la Esf.	r_{si}	25 mm
Radio de la plataforma	r_P	145 mm
Ángulo entre articulaciones	λ	30°

4.3.2. Parámetros Geométricos

A continuación, se presentan los parámetros de la estructura analizada en esta tesis. Los parámetros geométricos del robots se presentan en la siguiente tabla 4.1.

La distribución de las articulaciones universales y esféricas se describe en la figura RRRR. Esto se puede expresar mediante los siguientes vectores: $\mathbf{u}_i = r_P [\cos(\gamma_{1i}) \quad \sin(\gamma_{1i}) \quad 0]^T$ y $\mathbf{s}_i = r_P [\cos(\gamma_{2i}) \quad \sin(\gamma_{2i}) \quad 0]^T$, donde γ_1 and γ_2 estan definidos de la siguiente manera:

$$\gamma_1 = \begin{bmatrix} -\lambda/2 \\ \lambda/2 \\ 2\pi/3 - \lambda/2 \\ 2\pi/3 + \lambda/2 \\ 4\pi/3 - \lambda/2 \\ 4\pi/3 + \lambda/2 \end{bmatrix}$$

y

$$\gamma_2 = \begin{bmatrix} -\pi/3 + \lambda/2 \\ \pi/3 - \lambda/2 \\ \pi/3 + \lambda/2 \\ \pi - \lambda/2 \\ \pi + \lambda/2 \\ -\pi/3 - \lambda/2 \end{bmatrix}$$

4.3.3. Parámetros Inerciales e Hidrodinámicos

Independientemente del material utilizado para construcción del prototipo real del robot, para que se lleve a cabo la condición de flotabilidad neutra la fuerza de gravedad y la fuerza de flotabilidad deben ser iguales $B = W$, por lo tanto la masa total del elemento tiene que ser:

$$m = \rho_a V \quad (4.58)$$

Es decir que la densidad total de cada elemento es igual a la del agua, por lo tanto los parámetros inerciales e hidrodinámicos dependerán de la geometría del robot.

Plataformas

El radio de la esfera y el radio del cilindro interno están denotados por r_s y r_{si} , respectivamente. El radio de la plataforma es aquel que define la circunferencia sobre la cual están montadas las articulaciones universales y esféricas. Este radio está denotado por r_P .

Para simplificar las expresiones del volumen y los momentos de inercia consideraremos que en general r_{si} es pequeño con respecto a r_s . De esta forma el volumen de las plataformas está dado por la siguiente expresión

$$V_P = \frac{4}{3}\pi r_s^3 - 2\pi r_{si}^2 r_s \quad (4.59)$$

La masa por tanto es $m_P = \rho_a V_P$. Por otro lado, los elementos de la matriz de Inercia son los siguientes:

$$I_{xP} = I_{yP} = \frac{2}{5}r_s^2 \left(\frac{4}{3}\rho_a \pi r_s^3 \right) - \frac{1}{12} (3r_{si}^2 + (2r_s)^2) (2\pi \rho_a r_{si}^2 r_s) \quad (4.60)$$

$$I_{zP} = \frac{2}{5}r_s^2 \left(\frac{4}{3}\rho_a \pi r_s^3 \right) - \frac{1}{2}r_{si}^2 (2\pi \rho_a r_{si}^2 r_s) \quad (4.61)$$

Para la masa añadida consideramos que las plataformas son completamente esféricas. La masa añadida de una esfera de radio r_s está dado por la siguiente expresión:

$$X_{\dot{u}P} = Y_{\dot{v}P} = Z_{\dot{w}P} = \frac{2}{3}\pi \rho_a r_s \quad (4.62)$$

Por otro lado,

$$K_{\dot{p}P} = M_{\dot{q}P} = N_{\dot{r}P} = 0 \quad (4.63)$$

Por lo tanto el torque que se genera debido al movimiento de la esfera es:

$$\mathbf{n}_T = \mathbf{M}_{R22} \dot{\boldsymbol{\omega}}_o^b - \mathbb{S}(\mathbf{M}_{R22} \boldsymbol{\omega}_o^b) \boldsymbol{\omega}_o^b + \mathbf{D}_2 \boldsymbol{\omega}_o^b \quad (4.64)$$

ya que $\mathbf{M}_{A22} = \mathbf{0}_3$ y $\mathbf{M}_{A11} = m_a \mathbf{I}_3$ y por tanto $\mathbb{S}(\mathbf{M}_{A11} \mathbf{v}_o^b) \mathbf{v}_o^b = \mathbf{0}$. Esta consideración significa que no habrá efecto por la masa añadida en el torque.

Para determinar los coeficientes de las fuerzas de viscosas de arrastre, se tiene que las áreas laterales proyectadas son:

$$A_P = \pi r_s^2 \quad (4.65)$$

Para este elemento, y los demas consideraremos $C_D = 1,0$.

$$X_{|u|uP} = Y_{|v|vP} = Z_{|w|wP} = \frac{1}{2}\rho_a C_D A_P \quad (4.66)$$

No existen fuerzas de arrastre debido a al movimiento rotacional de una esfera. Sin embargo, existe fricción viscosa debido a la velocidad tangencial de la superficie de la esfera. Esta ultima fuerza se despreciara. Finalmente, tenemos que:

$$K_{|p|pP} = M_{|q|qP} = N_{|r|rP} = 0 \quad (4.67)$$

Piernas de la estructura

Las piernas de la estructura están compuestas de dos eslabones, el cilindro y el pistón. A continuación se derivan los parámetros de este par de elementos.

Cilindro. El radio externo y la longitud del cilindro están denotados por r_{ec} y l_{ec} , respectivamente. El radio y la longitud de la profundidad del orificio interior están denotados por r_{ic} y l_{ic} respectivamente.

El volumen de los cilindro es el siguiente:

$$V_C = \pi(r_{ec}^2 l_{ec} - r_{ic}^2 l_{ic}) \quad (4.68)$$

dado que en el caso de los parámetros del robot de estudio $l_{ec} \sim l_{ic}$, se calcula el volumen mediante $V_C = \pi l_{ec}(r_{ec}^2 - r_{ic}^2)$ y la masa por $m_C = \rho_a V_C$.

De igual manera considerando $l_{ec} \sim l_{ic}$ los momentos de inercia se calculan de la siguiente forma:

$$I_{xC} = I_{yC} = \frac{1}{12}\rho_a \pi l_{ec} (r_{ec}^2 (3r_{ec}^2 + l_{ec}^2) - r_{ic}^2 (3r_{ic}^2 + l_{ec}^2)) \quad (4.69)$$

$$I_{zC} = \frac{1}{2}\rho_a \pi l_{ec} (r_{ec}^4 - r_{ic}^4) \quad (4.70)$$

La masa añadida en las caras laterales del cilindro están dadas por la siguiente expresión:

$$X_{\dot{u}C} = Y_{\dot{v}C} = \pi \rho_a r_{ec}^2 l_{ec} \quad (4.71)$$

Para el cilindro y el pistón del robot de estudio se despreciaran las fuerzas de masa añadida y arrastre sobre el eje longitudinal, porque las dimensiones de las áreas transversales son pequeñas comparadas con las aéreas laterales de estos elementos y las respectivas aéreas de las plataformas. Por lo tanto, $Z_{\dot{w}P} \approx 0$.

Consideraremos que el momento de fuerza, debido a la masa añadida, y generado por una aceleración angular (α) es equivalente a un par de fuerzas (f_a) producidas por la aceleración lineal (a) de los centroides de dos cilindros adyacentes de la mitad de largo del cilindro original. Una de las fuerzas del par está dada por $f_a = \frac{1}{2}\pi \rho_a r_{ec}^2 l_{ec} a$. La aceleración lineal del centroide debida exclusivamente a una aceleracion angular es $a = \frac{l_{ec}}{4}\alpha$. El momento generado por el par de fuerzas es por tanto $\tau = 2\frac{l_{ec}}{4}\frac{1}{2}\pi \rho_a r_{ec}^2 l_{ec} \frac{l_{ec}}{4}\alpha$.

Utilizando esta logica, tenemos lo siguiente¹:

$$K_{\dot{p}C} = M_{\dot{q}C} = \frac{1}{16}\pi\rho_a r_{ec}^2 l_{ec}^3 \quad (4.72)$$

y $N_{\dot{r}C} = 0$.

Para determinar los coeficientes de las fuerzas de viscosas de arrastre, se tiene que las areas laterales proyectadas son:

$$A_C = 2r_{ec}l_{ec} \quad (4.73)$$

Los coeficientes de arrastre estan dados por:

$$X_{|u|uC} = Y_{|v|vC} = \frac{1}{2}\rho_a C_D A_C \quad (4.74)$$

y $Z_{|w|w} \approx 0$.

Para los parámetros de arrastre para movimientos rotacionales consideraremos que el momento de arrastre generado por una velocidad angular (ω) es equivalente a un par de fuerzas (f_c) de arrastre producidas por la velocidad lineal (la cual es generada por la velocidad angular citada anteriormente) (v_c) del centroide de la semi-área proyectada del cilindro. Una de las fuerzas del par está dada por $f_c = \frac{1}{2}\rho_a C_D \frac{A_C}{2} |v_c|v_c$. La velocidad lineal del centroide es $v_c = \frac{l_{ce}}{4}\omega$. El momento generado por el par de fuerzas es por tanto

$$\tau = 2\frac{l_{ce}}{4} \left(\frac{1}{2}\rho_a C_D \frac{A_C}{2} \frac{l_{ce}^2}{16} |\omega|\omega \right).$$

Lo anterior significa que los parámetros de arrastre son¹:

$$K_{\dot{p}C} = M_{\dot{q}C} = \frac{1}{128}\rho_a C_D A_C l_{ce}^3 \quad (4.75)$$

y $N_{\dot{r}C} = 0$.

Pistón. El radio y la longitud del pistón están denotados por r_{ps} y l_{ps} , respectivamente. El volumen del pistón viene dado por:

$$V_S = \pi r_{ps}^2 l_{ps} \quad (4.76)$$

La masa m_S se calcula como se ha descrito previamente. Los momentos de inercia son los siguientes:

$$I_{xS} = I_{yS} = \frac{1}{12}m_{ps}(3r_{ps}^2 + l_{ps}^2) \quad (4.77)$$

$$I_{zS} = \frac{1}{2}m_{ps}r_{ps}^2 \quad (4.78)$$

La masa añadida y los parámetros de arrastre se calculan de igual forma como se hizo en el caso del Cilindro.

¹Esta formulación carece de evidencia empírica que la valide. La hipótesis se basa en la lógica descrita en el párrafo precedente y su propósito es proveer de valores que se encuentre dentro del orden de magnitud del problema.

Tabla 4.2: Parámetros Inerciales

Parámetro	Plataforma	Cilindro	Pistón
$m(\text{kg})$	5.1433	0.1619	0.0583
$I_x(\text{kg}\cdot\text{m}^2)$	0.0252	0.0014	4.0957e-004
$I_y(\text{kg}\cdot\text{m}^2)$	0.0252	0.0014	4.0957e-004
$I_z(\text{kg}\cdot\text{m}^2)$	0.0268	2.3388e-005	1.8659e-006

Tabla 4.3: Coeficientes de la Masa Añadida

Parámetro	Plataforma	Cilindro	Pistón
$X_{\dot{u}}(\text{kg})$	2.7876	0.2262	0.0583
$Y_{\dot{v}}(\text{kg})$	2.7876	0.2262	0.0583
$Z_{\dot{w}}(\text{kg})$	2.7876	≈ 0.0	≈ 0.0
$K_{\dot{p}}(\text{kg}\cdot\text{m}^2)$	0.0	0.0014	3.0648e-004
$M_{\dot{q}}(\text{kg}\cdot\text{m}^2)$	0.0	0.0014	3.0648e-004
$N_{\dot{r}}(\text{kg}\cdot\text{m}^2)$	0.0	0.0	0.0

Tabla 4.4: Parámetros de Arrastre

Parámetro	Plataforma	Cilindro	Pistón
$X_{ u u}$	19.0066	4.8000	2.3200
$Y_{ v v}$	19.0066	4.8000	2.3200
$Z_{ w w}$	19.0066	≈ 0.0	≈ 0.0
$K_{ p p}$	≈ 0.0	0.0025	8.8410e-004
$M_{ q q}$	≈ 0.0	0.0025	8.8410e-004
$N_{ r r}$	≈ 0.0	≈ 0.0	≈ 0.0

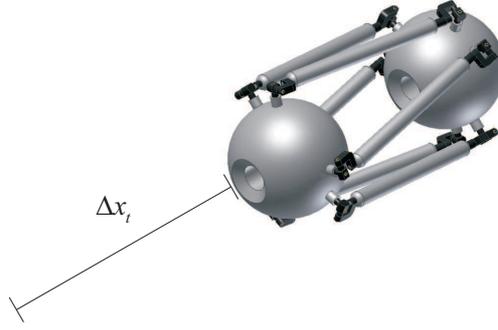


Figura 4.6: Trayectoria en línea recta.

Dados los parámetros geométricos del robot de estudio, se pueden calcular los parámetros inerciales e hidrodinámicos que se presentan en las tablas 4.2, 4.3 y 4.4.

Considerando esta información, el momento debido a las fuerzas hidrodinámicas se puede expresar de la siguiente manera:

$$\mathbf{n}_H = \mathbf{M}_{A22}\dot{\boldsymbol{\omega}}_o^b + \mathbf{D}_2\boldsymbol{\omega}_o^b - m_{11} \begin{bmatrix} v_{oz}^b v_{oy}^b \\ -v_{oz}^b v_{ox}^b \\ 0 \end{bmatrix} - m_{22} \begin{bmatrix} \omega_{oz}^b \omega_{oy}^b \\ -\omega_{oz}^b \omega_{ox}^b \\ 0 \end{bmatrix} \quad (4.79)$$

4.4. Simulación

En esta sección se presentan los resultados de la simulación del robot utilizando la formulación presentada anteriormente. Se presentan dos escenarios, uno en el que el robot se mueve en línea recta y otro en el que realiza una trayectoria circular. Para el último caso se considera que el robot posee un CMG como en el caso del Remo II.

4.4.1. Trayectoria en línea recta hacia el frente

Para esta simulación se considerará que el impulsor 1 es el que genera la fuerza para mover al robot en línea recta hacia adelante. En este caso la dinámica directa consiste en dada la trayectoria del robot, determinar la fuerza en el impulsor 1 y la fuerza en los actuadores prismáticos para realizarla.

De las ecuaciones de movimiento 4.42 y 4.43 se obtiene:

$$\sum {}^a \mathbf{f}_{si}^{p2} = \mathbf{f}_{Tp2}^{p2} - \sum {}^r \mathbf{f}_{si}^{p2} \quad (4.80)$$

y

$$\sum \mathbf{s}_i \times {}^a \mathbf{f}_{si}^{p2} = \mathbf{n}_{Tp2}^{p2} - \sum \mathbf{s}_i \times {}^r \mathbf{f}_{si}^{p2} \quad (4.81)$$

Se puede escribir lo anterior de manera matricial:

$${}^a \mathbf{F}_s = \mathbf{C}_{l2}^{-1} \mathbf{Q}_{l2} \quad (4.82)$$

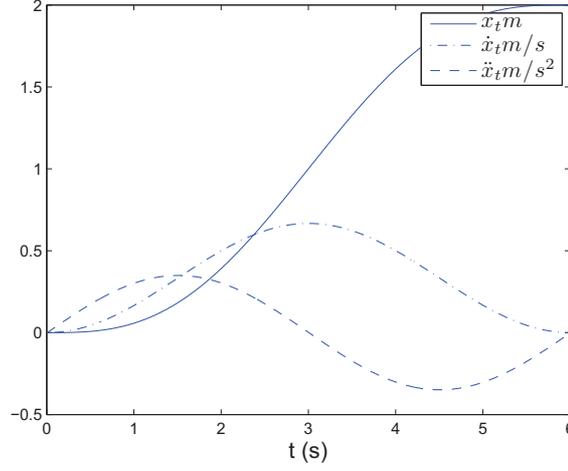


Figura 4.7: x_t , \dot{x}_t and \ddot{x}_t vs t

donde el vector ${}^a\mathbf{F}_s = [{}^a f_{s1} \quad {}^a f_{s2} \quad \dots \quad {}^a f_{s6}]^T$ contiene las fuerzas axiales aplicadas a las piernas. La matriz de coeficientes \mathbf{C}_s está dada por:

$$\mathbf{C}_{l2} = \begin{bmatrix} \hat{\mathbf{d}}_{s1}^{p1} & \hat{\mathbf{d}}_{s2}^{p1} & \dots & \hat{\mathbf{d}}_{s6}^{p1} \\ \mathbf{s}_1 \times \hat{\mathbf{d}}_{s1}^{p1} & \mathbf{s}_2 \times \hat{\mathbf{d}}_{s2}^{p1} & \dots & \mathbf{s}_6 \times \hat{\mathbf{d}}_{s6}^{p1} \end{bmatrix} \quad (4.83)$$

y el vector \mathbf{Q}_{l2} :

$$\mathbf{Q}_{l2} = \begin{bmatrix} \mathbf{f}_{Tp2}^{p2} - \sum r \mathbf{f}_{si}^{p2} \\ \mathbf{n}_{Tp2}^{p2} - \mathbf{s}_i \times r \mathbf{f}_{si}^{p2} \end{bmatrix} \quad (4.84)$$

Una vez conocidas las fuerzas ${}^a f_{si}$ que actúan sobre el bastago en el punto de la articulación esférica, se pueden obtener las fuerzas en el actuador prismático mediante la siguiente expresión:

$${}^a \mathbf{f}_{pi}^i = {}^a \mathbf{f}_{T2i}^i + {}^a \mathbf{f}_{si}^i \quad (4.85)$$

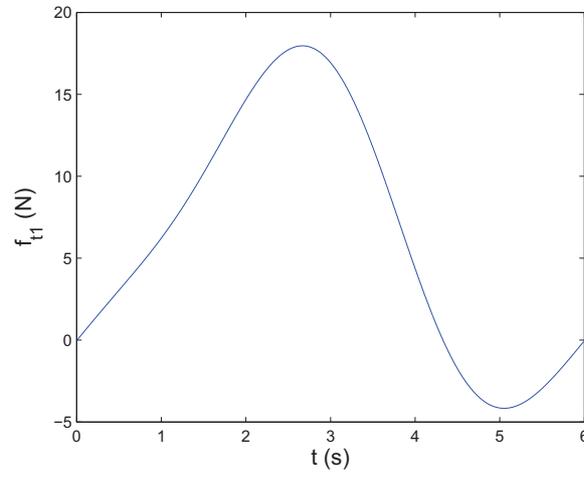
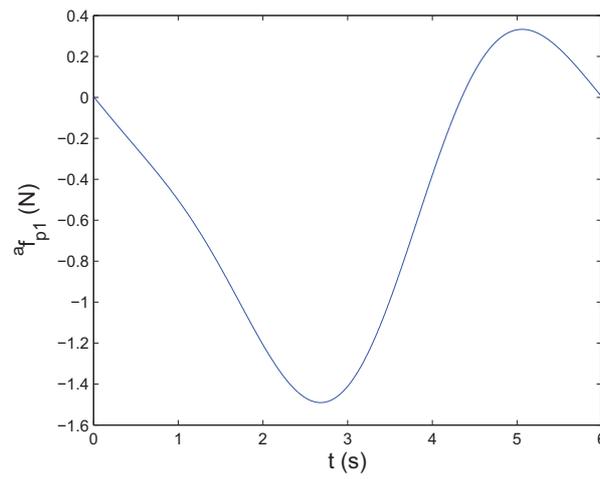
Por otro lado, con 4.44 la fuerza en el impulsor 1 se obtiene a través de la siguiente expresión:

$$\mathbf{f}_{t1}^{p1} = \mathbf{f}_{Tp1}^{p1} - \sum r \mathbf{f}_{ui}^{p1} + \sum \mathbf{R}_i^{p1} ({}^a \mathbf{f}_{T1i}^i + {}^a \mathbf{f}_{T2i}^i) + \sum {}^a \mathbf{f}_{si}^{p1} \quad (4.86)$$

Para esta simulación, la trayectoria rectilínea está parametrizada de la siguiente manera en función del tiempo:

$$x_t(t) = \frac{\Delta x_t t}{T} - \frac{\Delta x_t}{2\pi} \sin\left(\frac{2\pi t}{T}\right) \quad (4.87)$$

donde t es el tiempo y T es el periodo de tiempo de la trayectoria. Δx_t es la distancia que recorrerá el robot.

Figura 4.8: x_t , \dot{x}_t and \ddot{x}_t vs t Figura 4.9: x_t , \dot{x}_t and \ddot{x}_t vs t

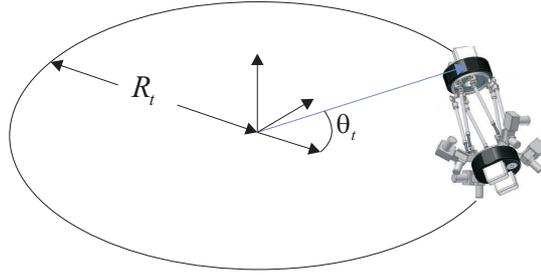


Figura 4.10: Trayectoria circular

Para la trayectoria consideramos que $\Delta x_t = 2m$ $T = 6s$. La figura 4.7 muestra la trayectoria de $x_t(t)$ y sus derivadas con respecto al tiempo. Las funciones cicloidales tiene la propiedad que las velocidades y aceleraciones son cero al inicio y final.

En la figura 4.8 se presenta la fuerza que debe generar el primer impulsor para realizar este movimiento. Se puede observar que la fuerza alcanza su máximo (17.96 N) un poco antes de los 3 s, para posteriormente reducirse hasta cambiar su sentido. El valor máximo en sentido contrario es -4.16 N. En la fase inicial el impulsor trabaja en contra de las fuerzas inerciales e hidrodinámicas. Posteriormente ayudado por las fuerzas de arrastre el impulsor hace que desacelere el robot. Gracias a las fuerzas de arrastre el impulsor es exigido en menor medida en la segunda parte de la trayectoria.

Por otro lado, en la fig. 4.9 se presenta la fuerza que debe generar el actuador lineal de la primera pierna. Se observo que para todos actuadores lineales la fuerza que deben de generar es la misma. Esto se debe a la simetría del robot y a la configuración en la que se realizo la tarea. La máxima fuerza que debe realizar el actuador es 1.49 N a tensión.

4.4.2. Trayectoria Circular utilizando un CMG

En este caso se considero una trayectoria circular parametrizada por el ángulo ($\theta_t(t)$) (es el angulo entre el vector de posición de la plataforma con respecto al marco de referencia N en el plano $x - y$), véase la figura 4.10. Este angulo esta definido por una función cicloidal:

$$\theta_t(t) = \frac{\Delta\theta_t t}{T} - \frac{\Delta\theta_t}{2\pi} \sin\left(\frac{2\pi t}{T}\right) + \theta_t^{ini} \quad (4.88)$$

El ángulo θ_t^{ini} es el valor inicial de $\theta_t(t)$ y $\Delta\theta_t$ es la diferencia entre el valor inicial y final de $\theta_t(t)$. Las derivadas temporales de 4.88 son:

$$\dot{\theta}_t(t) = \frac{\Delta\theta_t}{T} - \frac{\Delta\theta_t}{T} \cos\left(\frac{2\pi t}{T}\right) \quad (4.89)$$

y

$$\ddot{\theta}_t(t) = \frac{2\pi\Delta\theta_t}{T^2} \sin\left(\frac{2\pi t}{T}\right) \quad (4.90)$$

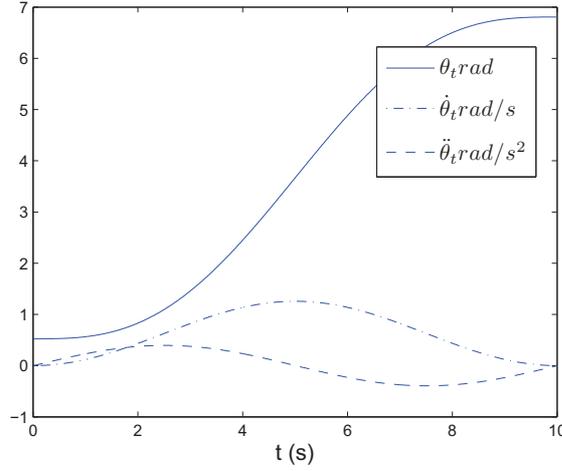


Figura 4.11: θ_t , $\dot{\theta}_t$ and $\ddot{\theta}_t$ vs t

La trayectoria parametrizada en función $\theta_t(t)$ de esta dada de la siguiente manera:

$$\eta_p = \begin{bmatrix} R_t \cos(\theta_t) \\ R_t \sin(\theta_t) \\ 0 \\ \pi/2 \\ \pi/2 \\ -\theta_t \end{bmatrix} \quad (4.91)$$

donde R_t es el radio de la trayectoria.

Para la trayectoria consideramos que $\theta_t^{ini} = \pi/6$, $\Delta\theta_t = 2\pi$ and $T = 10s$. La figura 4.11 muestra la trayectoria de $\theta_t(t)$ y sus derivadas con respecto al tiempo. Las funciones cicloidales tiene la propiedad que las velocidades y aceleraciones son cero al inicio y final de la trayectoria, y este hecho las hace adecuadas para la planificación de trayectorias.

En la figura 4.12, es posible observar la posición de los marcos de referencia unidos a los eslabones de la pierna, estos son L_{11} y L_{21} . Debido a que la trayectoria seguida por el robot esta en el plano xy la posición de los cuerpos en z es cero (i.e. $r_{11z}^n = r_{21z}^n = 0$).

La figura 4.13 muestra las velocidades de los marcos de referencia, L_{11} and L_{21} . Se puede observa que la velocidades al principio y al final son cero, esto se debe al uso de la función cicloidal

La aceleración de los eslabones de la pierna 1 se muestran en la figura 4.14. La aceleración tiene su máximo en la mitad de la trayectoria y es cero al principio y al final, de la misma manera que las velocidades.

Con el objetivo de determinar la dinámica inversa del sistema, las ecuaciones (4.42) y (4.44) se suman ambas con respecto al marco de referencia N , y se obtiene la siguiente expresión:

$$\mathbf{f}_{t1}^n + \mathbf{f}_{t2}^n = \mathbf{Q}_{t12}^n \quad (4.92)$$

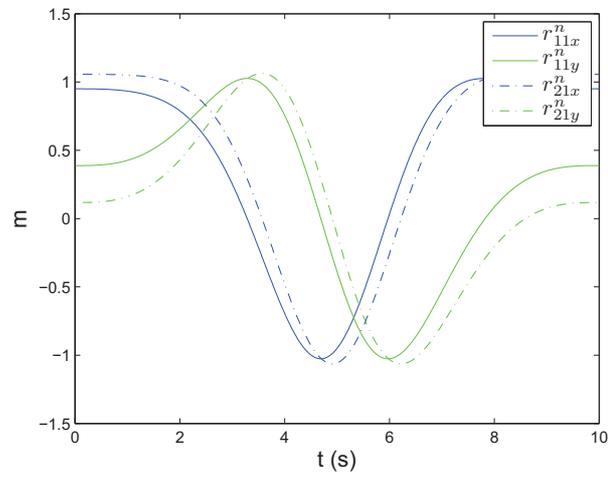


Figura 4.12: Position of leg 1

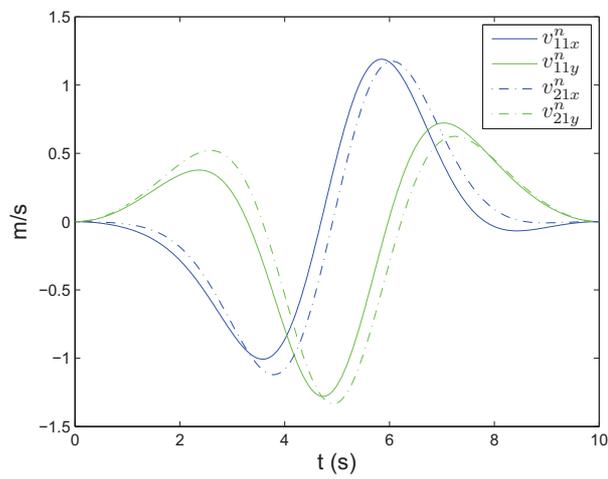


Figura 4.13: Velocidades de la pierna 1

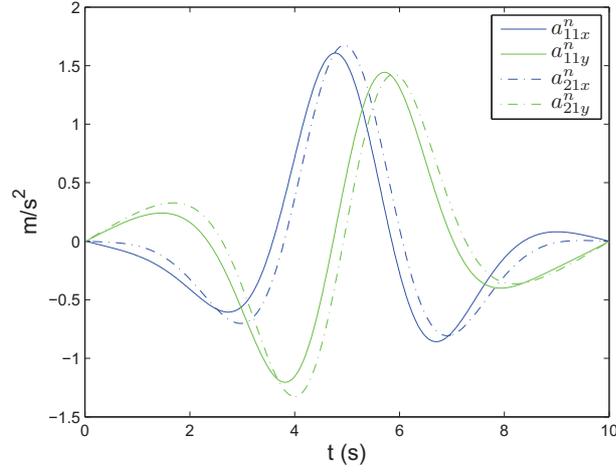


Figura 4.14: Aceleraciones de la pierna 1

donde

$$\begin{aligned} \mathbf{Q}_{t12}^n = & \mathbf{R}_{p1}^n [\mathbf{f}_{Tp1}^{p1} - \sum r \mathbf{f}_{ui}^{p1} + \sum \mathbf{R}_i^{p1} ({}^a \mathbf{f}_{T1i}^i + {}^a \mathbf{f}_{T2i}^i)] \\ & + \mathbf{R}_{p2}^n [\mathbf{f}_{Tp2}^{p1} - \sum r \mathbf{f}_{si}^{p1}] \end{aligned} \quad (4.93)$$

Las fuerzas generadas por los impulsores pueden ser expresadas de la siguiente manera, $\mathbf{f}_{t1}^n = \hat{\mathbf{z}}_{p1}^n f_{t1}$ y $\mathbf{f}_{t2}^n = \hat{\mathbf{z}}_{p2}^n f_{t2}$. Debido a que el movimiento del robot es en el plano, tenemos que $\hat{\mathbf{z}}_{p1}^n = [x \hat{z}_{p1}^n \quad y \hat{z}_{p1}^n \quad 0]^T$ y $\hat{\mathbf{z}}_{p2}^n = [x \hat{z}_{p2}^n \quad y \hat{z}_{p2}^n \quad 0]^T$. Por lo tanto tenemos la siguiente ecuación:

$$\mathbf{C}_{t12} \begin{bmatrix} f_{t1} \\ f_{t2} \end{bmatrix} = \bar{\mathbf{Q}}_{t12}^n \quad (4.94)$$

donde

$$\mathbf{C}_{t12} = \begin{bmatrix} x \hat{z}_{p1}^n & x \hat{z}_{p2}^n \\ y \hat{z}_{p1}^n & y \hat{z}_{p2}^n \end{bmatrix} \quad (4.95)$$

y $\bar{\mathbf{Q}}_{t12}^n$ es un subvector de \mathbf{Q}_{t12}^n que contiene solo las primeras dos entradas. Por lo tanto utilizando las ecuaciones (4.44) y (4.45), es posible resolver el sistema de tal manera que se obtienen las fuerzas en los actuadores prismáticos:

$$\mathbf{C}_c {}^a \mathbf{F}_s = \mathbf{Q}_c \quad (4.96)$$

donde:

$$\mathbf{C}_c = \begin{bmatrix} \hat{\mathbf{d}}_{s1}^{p1} & \hat{\mathbf{d}}_{s2}^{p1} & \cdots & \hat{\mathbf{d}}_{s6}^{p1} \\ \mathbf{u}_1 \times \hat{\mathbf{d}}_{s1}^{p1} & \mathbf{u}_2 \times \hat{\mathbf{d}}_{s2}^{p1} & \cdots & \mathbf{u}_6 \times \hat{\mathbf{d}}_{s6}^{p1} \end{bmatrix} \quad (4.97)$$

y

$$\mathbf{Q}_l = \begin{bmatrix} \mathbf{f}_{t1}^{p1} - \mathbf{f}_{Tp1}^{p1} + \sum r \mathbf{f}_{ui}^{p1} - \sum \mathbf{R}_i^{p1} ({}^a \mathbf{f}_{T1i}^i + {}^a \mathbf{f}_{T2i}^i) \\ -\mathbf{n}_{Tp1}^{p1} + \sum \mathbf{n}_{ui}^{p1} + \sum \mathbf{u}_i \times r \mathbf{f}_{ui}^{p1} - \sum \mathbf{u}_i \times \mathbf{R}_i^{p1} ({}^a \mathbf{f}_{T1i}^i + {}^a \mathbf{f}_{T2i}^i) \end{bmatrix} \quad (4.98)$$

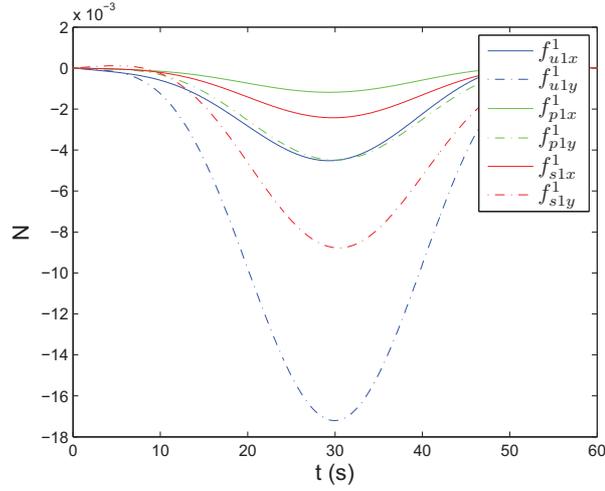


Figura 4.15: Fuerzas radiales en las articulaciones de la pierna 1

Finalmente se obtiene \mathbf{n}_{CMG} resolviendo la ecuación (4.43).

La figura 4.15 muestra las fuerzas radiales en las articulaciones universal, prismática y esférica de la pierna 1. Por otro lado, la figura 4.16 muestra el torque radial en la articulación prismática. Estas fuerzas y torque están referidos al marco de referencia de la pierna.

La figura 4.17 muestra las fueras requeridas en los impulsores. La dirección de la fuerza del impulsor 1 es positive, mientras que la del impulsor 2 es generalmente negativa. Ambas fuerzas tienen direcciones opuestas para compensar la dirección de la fuerza resultante de tal manera que el robot puede seguir una trayectoria circular.

La figura 4.18 muestra las fuerzas en los actuadores. Cuatro actuadores están a compresión y dos están a tensión. Las fuerzas en los actuadores son mayores en la mitad.

La figura 4.19 muestr el torque requerido en el CMG. Como era de esperarse, el torque en el eje z del marco N es remarcablemente dominante.

Computación paralela de la cinemática y dinámica inversa

Las formulaciones de la cinemática y la dinámica inversa son fácilmente paralelizables y pueden ser implementadas en varios procesadores. La figura 4.20 muestra un esquema de la implementación.

En primer lugar, en seis diferentes hilos de ejecución se realiza el cálculo de cada vector \mathbf{d}_i^n y la matriz \mathbf{R}_i^n , y sus derivadas temporales ($\dot{\mathbf{d}}_i^n$, $\ddot{\mathbf{d}}_i^n$, ω_{di}^n y $\dot{\omega}_{di}^n$). Posteriormente, en los mismos hilos se calcula las posición, velocidad y aceleración de las variables articulares i.e. ρ_i , $\dot{\rho}_i$ and $\ddot{\rho}_i$. Para calcular las posiciones, velocidades y aceleraciones de cada eslabón de cada pierna, leg (\mathbf{r}_{ji}^n , \mathbf{v}_{ji}^n y \mathbf{a}_{ji}^n , for $j \in [1, 2]$), se crean dos hilos hijo. Para las seis piernas, se hacen 12 threads concurrentes. Este esquema reduce considerablemente el tiempo de calculo de la cinemática inversa.

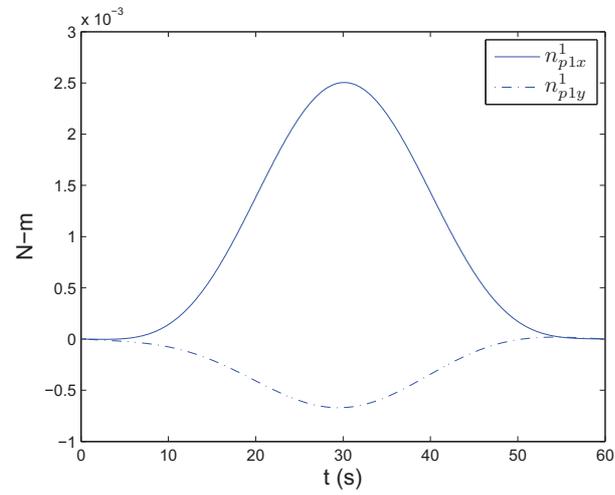


Figura 4.16: Torque radial en la articulación prismática de la pierna 1

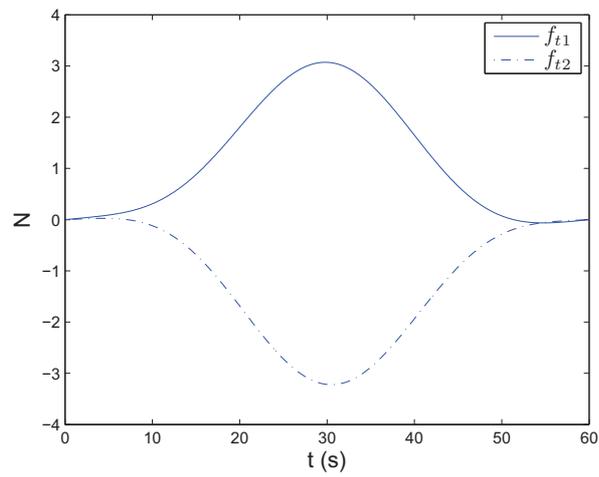


Figura 4.17: Fuerzas en los impulsores 1 y 2

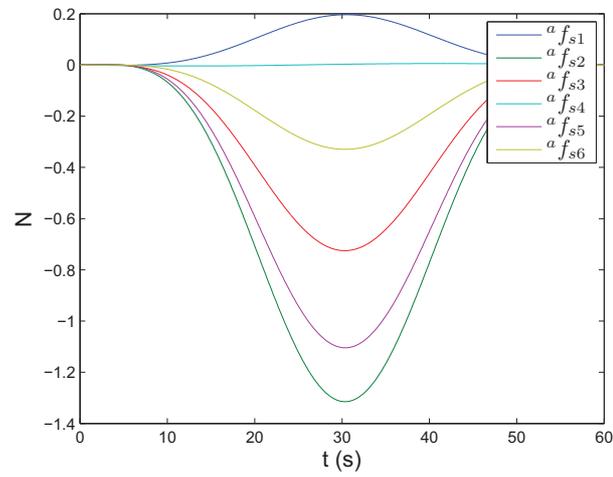


Figura 4.18: Fuerzas axiales

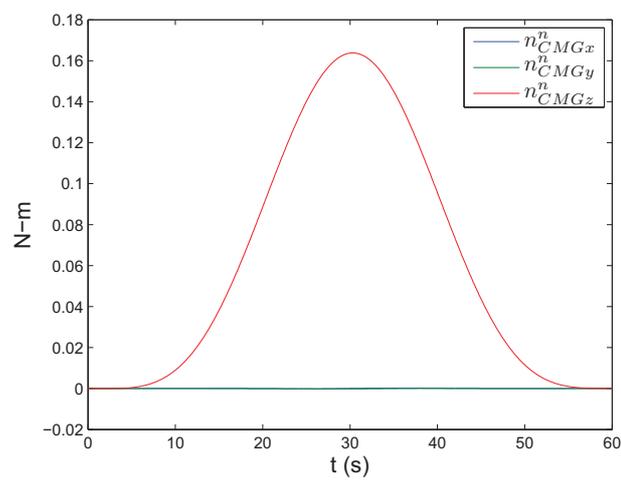


Figura 4.19: Torque generado por el CMG

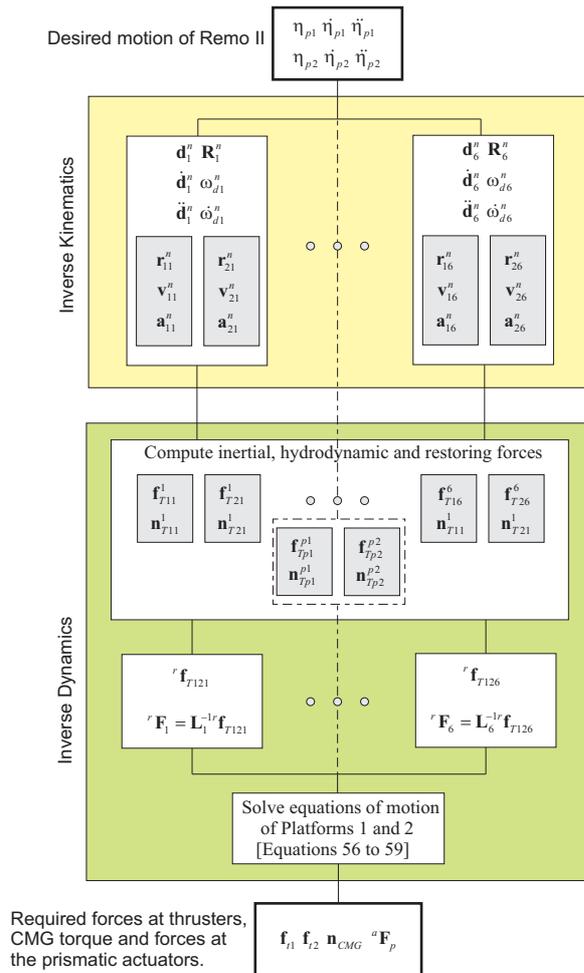


Figura 4.20: Procesos concurrente para calcular la cinemática y dinámica inversa

Posteriormente, se calculan las fuerzas inerciales, hidrodinámicas y de restitución que actúan sobre el cuerpo. Esto puede hacerse con 14 hilos (12 para cada eslabón de cada pierna, y 2 para las plataformas. Después de eso, se calculan las componentes radiales de la fuerzas totales. Resolviendo eq. (4.33), se puede obtener las fuerzas radiales de cada eslabón de cada pierna. Este paso puede hacerse concurrentemente para cada pierna. Finalmente, con los resultados obtenidos en el paso previo, las ecuaciones de movimiento de ambas plataformas se resuelven (i.e. las ecuaciones de la (4.42) a ala (4.45)), y se obtienen las fuerzas en los actuadores prismáticos, la fuerza en los impulsores y el torque requerido en el CMG.

4.5. Resumen

En esta sección se derivaron las ecuaciones de la cinemática y dinámica inversa de una estructura paralela donde ninguna de las plataformas esta unida al marco inercial. Para el autor esta es la primera vez que se presenta un trabajo como este. La formulación presentada es fácilmente paralelizable lo cual puede ser ventajoso si hay múltiples procesadores disponibles. El modelo presentado puede ser útil en esquemas de control basados en la planta.

Capítulo 5

Control de Robots Submarinos con IVEP

5.1. Introducción

El control de un robot submarino es en general un problema complicado, esto se debe en parte a que es un sistema no lineal y variante en el tiempo. Por otro lado, en ocasiones existen incertidumbres en los parámetros del robot además de las perturbaciones debido a las corrientes marinas. Cuando el robot posee un brazo robótico, o como en el caso de Remo 2, si su estructura está compuesta de varios cuerpos, el problema de control se complica aun más.

En este capítulo se presenta una exploración sobre el desempeño de algunas estrategias de control para robot submarino Remo 2. Por un lado se analiza el comportamiento del robot empleando diversas leyes de control y por otro lado mediante la simulación del robot es posible observar su dinámica en el ambiente submarino.

Para controlar robots submarinos se pueden emplear las distintas técnicas de control que previamente se han propuesto para otros sistemas. Este es el caso de los controladores en lazo cerrado que realimentan las señales medidas por los sensores del robot. En el caso de un controlador PID, la ley de control se expresa de la siguiente manera:

$$\boldsymbol{\tau}_{PID} = \mathbf{J}(\boldsymbol{\nu})^T \left[\mathbf{K}_p \tilde{\boldsymbol{\eta}} + \mathbf{K}_d \dot{\tilde{\boldsymbol{\eta}}} + \mathbf{K}_i \int \tilde{\boldsymbol{\eta}}(\sigma) d\sigma \right] \quad (5.1)$$

donde $\tilde{\boldsymbol{\eta}} = \boldsymbol{\eta}_d - \boldsymbol{\eta}$ es el error entre la posición real y la deseada, \mathbf{K}_p es la matriz de ganancias proporcionales, \mathbf{K}_d ganancias derivativas, y \mathbf{K}_i es la matriz de ganancias integrales. Por otro lado se ha propuesto también el controlador PID con realimentación de aceleración.

$$\boldsymbol{\tau}_{PIDA} = \mathbf{J}(\boldsymbol{\nu})^T \left[\mathbf{K}_p \tilde{\boldsymbol{\eta}} + \mathbf{K}_d \dot{\tilde{\boldsymbol{\eta}}} + \mathbf{K}_i \int \tilde{\boldsymbol{\eta}}(\sigma) d\sigma \right] - \mathbf{H} \dot{\boldsymbol{\nu}} \quad (5.2)$$

Donde la matriz \mathbf{H} puede ser constante o dependiente de la frecuencia. El propósito de la realimentación de aceleración es que el sistema sea menos sensible a perturbaciones externas como lo pueden ser las corrientes marinas. En (Fossen, 2002) se explica este concepto.

Dada la estructura del modelo presentado en la ec. 3.20 se podrían extrapolar las distintas leyes de control que se han propuesto para robot manipuladores.

En (Yoerger and Slotine, 1985) se presentó un controlador en modos deslizantes para robots submarinos. Las ventajas de este tipo de controlador es que trabaja directamente con los términos no lineales, es altamente robusto a modelos imprecisos y a la presencia de perturbaciones que son difíciles de medir o estimar. En (Healey and Lienard, 1993) se diseñó un autopiloto basado en modos deslizantes para controlar la dirección, velocidad y profundidad de un vehículo submarino. En (Innocenti and Campa, 1999) se presenta la síntesis de dos controladores robustos (un basado en modos deslizantes y otro basado en técnicas de desigualdades matriciales lineales (LMI)) para robots submarinos.

Por otro lado, diversos autores han propuesto controladores adaptables, ya que pueden ser una solución al problema de determinar los parámetros hidrodinámicos del robot (es difícil obtener estos parámetros mediante formulaciones matemáticas, y por otro lado, determinarlos experimentalmente conlleva a un costo económico y de tiempo). En (Goheen and Jeffery, 1990) se presenta un par de autopilotos auto-sintonizables de múltiples entradas y múltiples salidas. En ese trabajo se presenta un controlador lineal cuadrático auto-sintonizable en línea, y por otro lado una ley de control que se basa en una aproximación de primer orden de la dinámica del sistema en lazo abierto con identificación recursiva en línea. En (Fossen and Sagatun, 1991) se presentan dos controladores para compensar las incertidumbres del modelo de un ROV, estos son: un controlador adaptable basado en pasividad, y un controlador híbrido (adaptativo y por modos deslizantes). En (Nie et al., 1998) se presentó un control adaptable para un robot que opera en aguas poco profundas.

En (Yuh, 1990) y (Yuh, 1994) se presentaron controladores basados en redes neuronales para robots submarinos. En (Yuh, 1994) el sistema de control consistía de una red neuronal y un algoritmo de aprendizaje. La red neuronal está compuesta por múltiples capas de neuronas. La capa de entrada recibe la posición y velocidad deseada, la posición y velocidad actual, y señales provenientes del algoritmo de aprendizaje. La salida de la red neuronal es la que controla al vehículo. El algoritmo de aprendizaje ajusta los valores de los parámetros de ponderación de las neuronas, en función del error existente entre el estado deseado y el real. En (van de Ven et al., 2005) se hace una revisión sobre diferentes trabajos en los cuales se ha aplicado redes neuronales al control de robots submarinos y se realiza una clasificación de las estrategias presentadas.

En (DeBitetto, 1995) se presenta el diseño de un controlador de lógica difusa para el control de profundidad de robots submarinos. Este tipo de controlador es poco sensible a la configuración del vehículo y su dinámica. Según los autores de ese trabajo, los beneficios del uso de este tipo de controladores son: simplicidad, debido que no se requiere un modelo dinámico del sistema; la estrategia de control encaja mejor con los objetivos de control y las limitaciones; mediante el uso de reglas lingüísticas se puede tener una fácil comprensión y manipulación del controlador. En (Guo and Huang, 1996) se aplicaron algoritmos genéticos para sintonizar de reglas para un controlador de lógica difusa. Se realizaron pruebas experimentales y se mostró como el controlador ajustado por el algoritmo genético mejoraba su desempeño. Por otro lado, en (Guo et al., 2003)

se presentó la aplicación de un controlador en modos deslizantes con lógica difusa.

En el caso de los robots de estructura paralela, existen algunos trabajos sobre control en la literatura. Principalmente se han empleado controladores tipo P (Li and Salcudean, 1997), PD (Williams, 1999) y PID (Staicu et al., 2006) para el control de cada articulación. En (Yang et al., 2010) se presentó un controlador con compensación de gravedad. Otros controladores más sofisticados se han presentado en (Zhu et al., 2005) (Yen and Lai, 2009) (Callegari et al., 2006). Estos controladores están basados en el modelo del robot. Se han propuesto dos estrategias, control precompensación y control par calculado.

Simulación de los Controladores

En esta capítulo utilizamos ADAMS para simular la dinámica del robot y los controladores considerados. ADAMS es un software para simular sistemas multicuerpo. Un sistema multicuerpo es cualquier sistema mecánico compuesto de varios cuerpos unidos a través articulaciones cinemáticas. El software de simulación es útil para estudiar la dinámica del sistema, los movimientos de las piezas de una máquina o mecanismo, determinar las fuerzas internas y externas a las que están sometidos los cuerpos, etc. Esto tiene como objetivo ayudar en el diseño de cualquier dispositivo mecánico.

ADAMS se basa en la formulación de las ecuaciones de movimiento Newton-Euler para sistemas Multicuerpo. Es lo que se llama Multibody Dynamics. Esta formulación permite modelar cualquier sistema mecánico por más complicado que este sea. Mediante métodos numéricos, como lo son los integradores de Ecuaciones Diferenciales Algebraicas, se determina la dinámica directa, es decir, el movimiento de cada uno de los cuerpos en cada tiempo dadas las fuerzas externas a la que está sometido el sistema.

La principal aportación de este tipo de software es que se pueden crear prototipos virtuales, los cuales pueden ser sometidos a pruebas para determinar el funcionamiento de los mismos. Esto ahorra una considerable cantidad de tiempo y dinero cuando los prototipos son de gran complejidad.

Adams ha sido utilizado por diferentes compañías para realizar análisis de sistemas mecánicos. Este tipo de herramientas es muy popular en el diseño y análisis de automóviles, ya que estos son sistemas compuestos de varios elementos interconectados y además trabajan sometidos a continuos cambios de velocidad, aceleración y diferentes fuerzas externas.

Por otro lado, este software permite simulación de sistemas multicuerpo en ambientes especiales. Un ejemplo de esto fueron las simulaciones que realizaron el Laboratorio de Propulsión a Chorro (JPL, por sus siglas en inglés) para realizar pruebas sobre el robot del proyecto Curiosity. Durante la separación del rover y el sistema de vuelo, se tenía que evitar que el sistema de vuelo (el skycrane) chocara o entrara en contacto con el rover durante la "fase dos cuerpos", es decir cuando ambos sistemas estaban separados y el rover era bajado. Usando Adams, los ingenieros fueron capaces

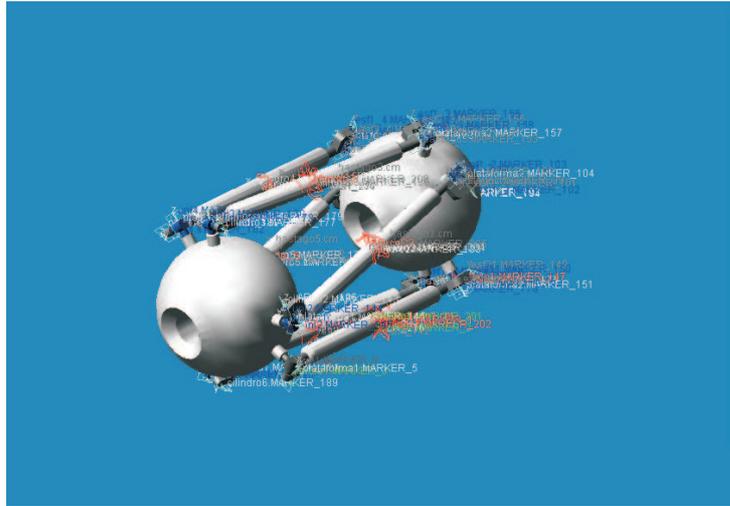


Figura 5.1: Modelo en ADAMS.

de modelar y simular este evento.

Por otro lado el equipo del JPL se basó en simulaciones por ordenador para estudiar las potenciales condiciones extremas que el Curiosity tendría que afrontar durante el descenso final, la separación y el aterrizaje. Muchas condiciones no podían ser probadas físicamente en la tierra. Para desarrollar y verificar el diseño del sistema tenían que lidiar con las complejidades que implican la gravedad marciana, la atmósfera, diferentes pendientes de la superficie y las velocidades de aterrizaje, las cuales no eran posibles reproducir exactamente aquí en la Tierra.

Por este clase de prestaciones, se eligió utilizar este software para simular la dinámica directa de Remo 2 en un ambiente submarino.

En la fig. 5.1 se presenta una imagen del simulador de Remo 2. Este es un modelo detallado que considera los parámetros inerciales presentados en capítulos anteriores, incluye todas las restricciones cinemáticas entre cada uno de los cuerpos y las fuerzas externas debido a la hidrodinámica del robot.

5.2. Control del Robot considerado como una estructura rígida

En esta sección se presenta un estudio sobre el desempeño de diversos controladores del robot durante la navegación. Este estudio se enfoca en la dinámica del robot considerándolo como una estructura rígida. Para esto, se estudiará el desempeño de distintos controladores en los impulsores del robot.

Aquí no se implementa ningún controlador sobre los actuadores prismáticos del robot. Se considera que la articulación prismática está bloqueada. El propósito es estudiar el desempeño del robot bajo la acción únicamente de los impulsores.

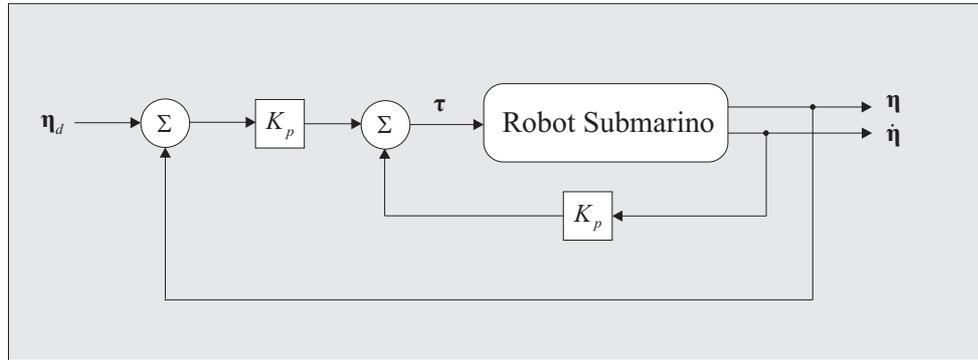


Figura 5.2: Controlador P con retroalimentación de velocidad.

5.2.1. Controlador P con retroalimentación de velocidad

La ecuación del control proporcional con retroalimentación viene dada por:

$$\tau = K_p \tilde{\eta} - K_v \dot{\eta} \quad (5.3)$$

donde $\tilde{\eta} = \eta_d - \eta$ es el error entre la posición real y la deseada, \mathbf{K}_p es la matriz de ganancias proporcionales y \mathbf{K}_d ganancias derivativas. En la figura 5.2 se presenta un diagrama de bloques de un sistema de control P con retroalimentación de velocidad para robots submarinos.

A continuación se explorara el funcionamiento del robot para realizar diferentes tipos de maniobras bajo este tipo de controlador.

5.2.1.1. Línea Recta

Un movimiento común del robot será el movimiento en línea recta. Una vez que se determine la posición deseada del robot este podrá orientar su estructura de tal manera que el eje de los impulsores sea paralelo a la línea que une el punto de origen y el punto de destino. En la figura 4.6 se describe este tipo de movimiento.

Se realizaron una serie de experimentos para llevar al robot desde la posición inicial hasta una distancia de 1 m. Durante estos experimentos fue posible observar como las fuerzas en las piernas se anulaban unas a otras y mantenían al sistema balanceado. Eso se puede observar en la figura 5.3 del robot .

En la figura 5.4 se presenta las diferentes respuestas en el tiempo para diversos valores de la ganancia K_p .

Se puede observar que el sistema no requiere de una ganancia K_v para estabilizarse. Esto se debe a que el sistema es afectado por fuerzas de arraste, que hacen que la energía generada por la componente proporcional del controlador se disipe con el tiempo.

Por otro lado, la figura 5.5 se presenta las diferentes respuestas en el tiempo para diversos valores de la ganancia K_v , manteniendo $K_p = 50$.

Se puede observar que el sistema tiene un comportamiento críticamente amortiguado cuando $K_v = 30$. En este caso el sistema logra llegar la referencia cerca de los 4 segundos.

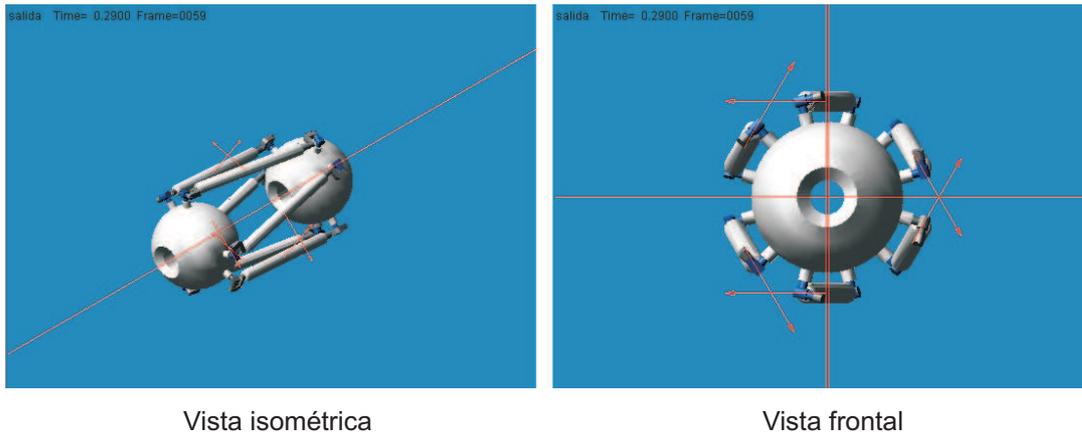


Figura 5.3: Fuerzas por masa añadida y por arrastre.

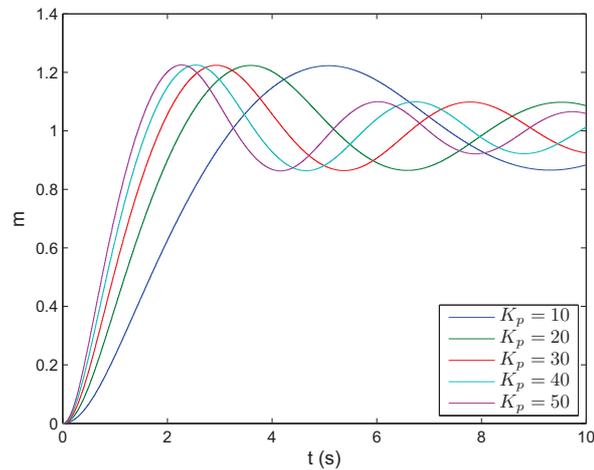


Figura 5.4: Comportamiento del robot empleando un controlador tipo proporcional.

Por otro lado, en la fig. 5.6 se presenta la fuerza en el impulsor para unas ganancias de $K_p = 50$ y $K_v = 30$. En este caso la fuerza del impulsor alcanza 50 N al inicio de la maniobra. Dado que ambos impulsores están alineados, en el caso de que ambos trabajaran en conjunto la fuerza se distribuiría proporcionalmente.

5.2.1.2. Traslación Pura Lateral

Una maniobra interesante que un robot tipo Remo II puede ser capaz de realizar es la traslación lateral pura mediante la reconfiguración del robot. En figura 5.7 se muestra una configuración del robot para realizar un movimiento lateral. En esta configuración las plataformas están giradas un ángulo α con respecto al *eje central* de la estructura. El eje central está alineado con el vector que une los centros de ambas plataformas.

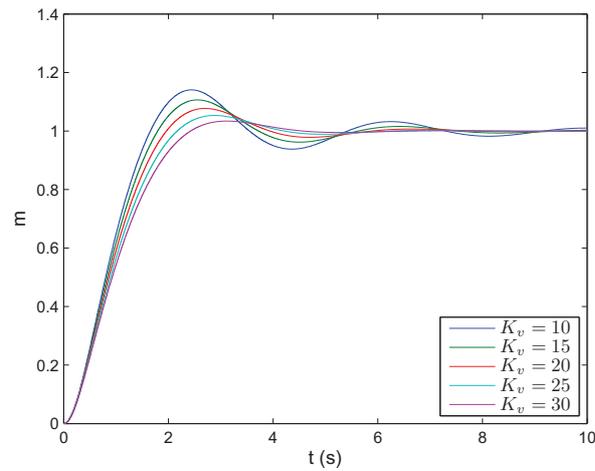


Figura 5.5: Comportamiento del robot empleando un controlador P con retroalimentación de velocidad, $K_p = 50$.

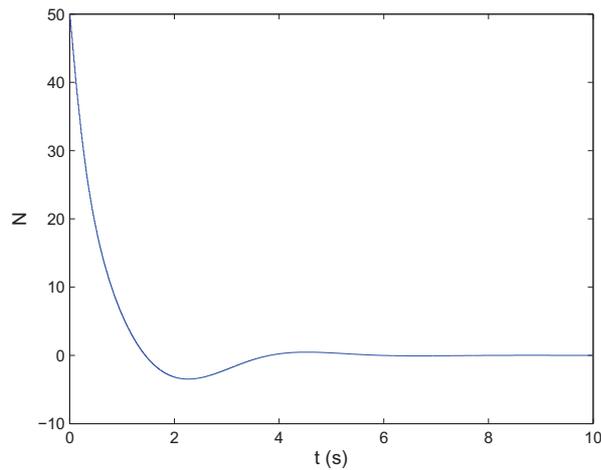


Figura 5.6: Fuerza en el primer impulsor, para $K_p=50$ y $K_v= 30$.

Cuando los impulsores que hay en cada plataforma se activan con la misma fuerza, las componentes de las fuerzas que son paralelas al eje central se anulan, y las componentes perpendiculares se suman. Al tener la misma dirección y colocarse casi a la misma distancia del centro de masa del robot, estas fuerzas generan un movimiento de traslación lateral.

La maniobra consistiría de dos fases, la primera de reconfiguración (que consiste en que el robot consiga la configuración que se muestra en la figura 5.7) y la segunda de traslación en la que los impulsores se activan para llevar a cabo el movimiento del robot.

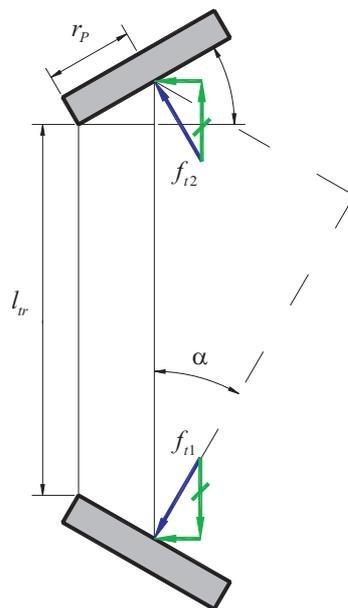


Figura 5.7: Configuración para traslación pura lateral.

Durante la simulación, los movimientos de los actuadores fueron realizados siguiendo una trayectoria tipo cicloidal. El movimiento de reconfiguración se fijó en 2 segundos. En el tiempo igual a 2.5 segundos se comienza a ejecutar la fase de traslación.

La primera estrategia que se exploró fue controlar la posición de un punto del robot. En este caso se eligió que el punto de referencia fuese el centro de la plataforma 1. El esquema de control fue un P con retroalimentación de velocidad. Para el controlador se considera que las fuerzas en los impulsores serán iguales en magnitud y diferentes en sentido, por lo tanto la fuerza aplicada en ambos impulsores es proporcional al error de posición y a la velocidad del centro de la plataforma 1. Haciendo una prueba para posicionar el vehículo a 0.20 m de su posición original el robot fue capaz de alcanzar la referencia como era de esperarse.

En otra prueba para posicionar el vehículo a 1 m de su posición original se obtuvo la respuesta presentada en la figura 5.8. Aunque el punto de control llega a la referencia, cuando se observa la postura del robot al final la simulación se puede observar que el robot ha perdido la orientación que tenía originalmente, es decir la orientación del robot es incontrolable bajo este esquema.

A pesar de que la configuración geométrica de la estructura del robot es simétrica, existe un desbalanceo de fuerzas, ya que los cilindros de las piernas poseen mayor masa que los pistones y además los parámetros hidrodinámicos son igualmente mayores en los primeros. Por lo tanto, el centro de masa se encuentra a diferentes distancias de las fuerzas de impulsión. Para compensar esto, se realizaron pruebas con diferentes factores de corrección sobre la fuerza del impulsor 2. Aunque, para ciertos valores se alcanzaba

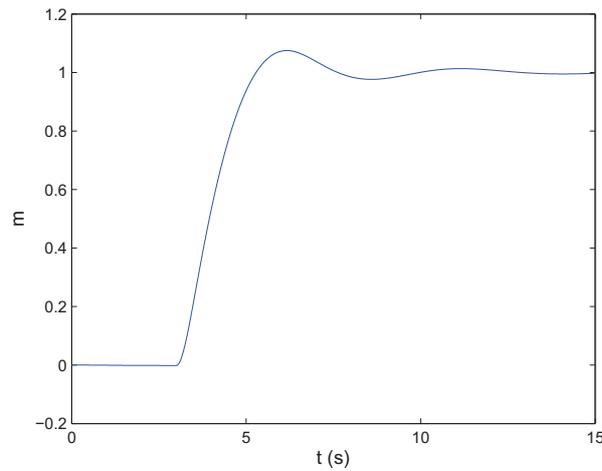


Figura 5.8: Respuesta del sistema para una referencia de 1 m, con ganancias $K_p = 50$ $K_v = 30$.

una postura paralela a la inicial, este método no se considera consistente.

Una estrategia diferente fue utilizar como variables de estado la posición y velocidad de los centros de las plataformas. Esta información es realimentada a controladores independientes de cada impulsor. El robot fue capaz de alcanzar la referencia deseada. Bajo esta estrategia fue posible comprobar que el robot mantenía el paralelismo con respecto a la postura inicial. En la fig. 5.9 se presenta las posturas finales para los esquemas de control en los que se utilizan como variables de estado las posiciones y velocidades de una o de las dos plataformas.

5.2.1.3. Rotación Pura

La rotación pura es una maniobra interesante que un robot tipo Remo II puede ser capaz de realizar. Para realizar esta maniobra consideramos la configuración mostrada en la fig. 5.10. En esta configuración ambas plataformas están giradas un ángulo α con respecto al eje central de la estructura.

En esta configuración, cada impulsor genera una fuerza en dirección contraria. De estas fuerzas las componentes en dirección del eje central se anulan, y las fuerzas perpendiculares generan un par de fuerzas, lo cual es a su vez un momento que hace girar al robot sobre un eje perpendicular al plano que contiene a estas fuerzas.

Se realizó una exploración para conocer el comportamiento del robot ante un controlador tipo proporcional. El interés por dicha exploración es conocer si es posible que el robot se estabilice por efecto de las fuerzas de arrastre del robot contra el agua. Estas fuerzas son disipativas de energía y combinadas con las fuerzas del controlador proporcional pueden tener el efecto de un controlador tipo PD.

En la figura 5.11 se presenta el comportamiento del robot para valores de K_p entre 0.5 y 3.0. Como es posible apreciar el comportamiento del sistema es subamortiguado y

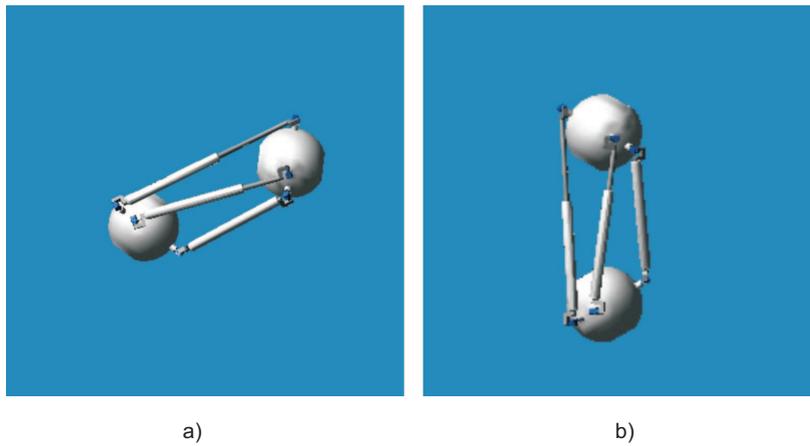


Figura 5.9: Postura final del robot a) cuando se controla únicamente la posición de la plataforma 1, b) cuando se controla la posición de ambas plataformas.

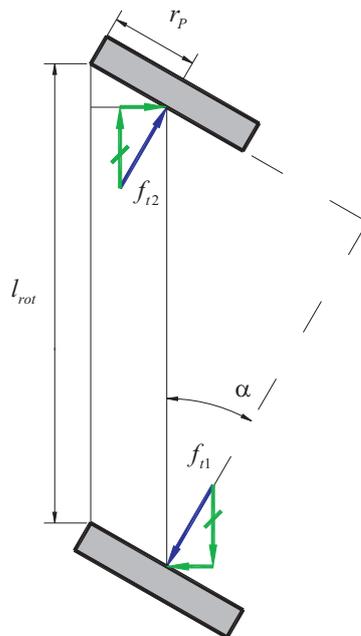


Figura 5.10: Configuración para rotación pura.

no llega a lograr la estabilidad hasta los 10 s (7.5 s después de que actúa el controlador). Se puede apreciar que al aumentar la ganancia K_p el sobre impulso es mayor y la frecuencia del sistema aumenta.

Para visualizar el desempeño del controlador en un rango más extenso de valores en las figuras 5.12 y 5.13 se presentan las salidas de controlador para $K_p = 0,05$ y $K_p = 10,0$. En el caso de $K_p = 0,05$ se muestra que el primer sobreimpulso se extiende

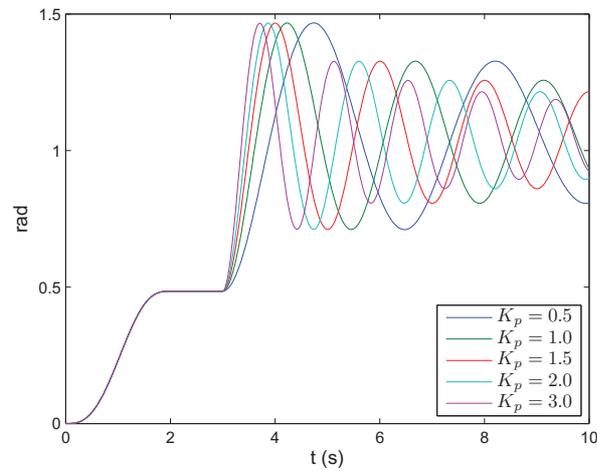


Figura 5.11: Comportamiento del robot empleando un controlador tipo proporcional.

ampliamente en el tiempo. Por otro lado, para $K_p = 10,0$ el comportamiento es similar al mostrado anteriormente en la figura 5.11.

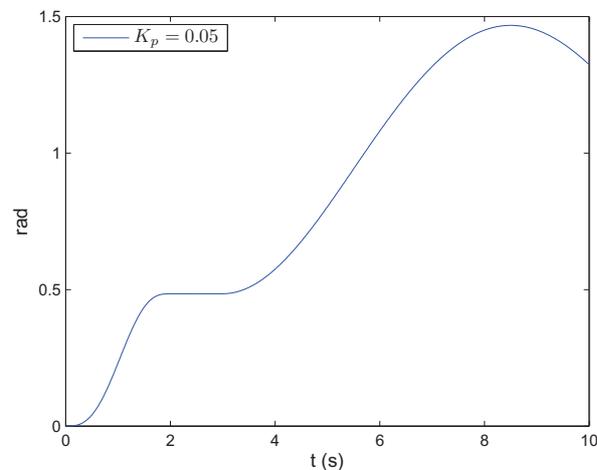


Figura 5.12: Comportamiento del robot para $K_p = 0,05$.

En este caso, consideramos que un controlador proporcional es insuficiente y no es práctico para lograr el desempeño deseado del robot. Por tal motivo se descarta completamente el empleo de este tipo de controlador.

En la figura 5.14 se presenta el comportamiento del robot cuando es controlado mediante un esquema P con retroalimentación de velocidad. En este caso, se aplica una ganancia $K_v = 5,0$ y se hace variar el valor de la ganancia proporcional K_p .

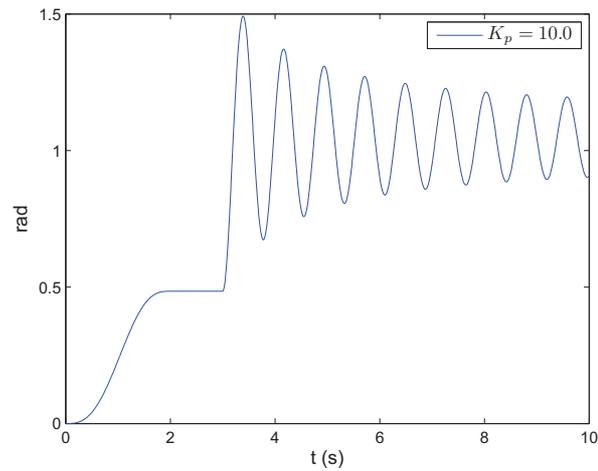


Figura 5.13: Comportamiento del robot para $K_p = 10,0$.

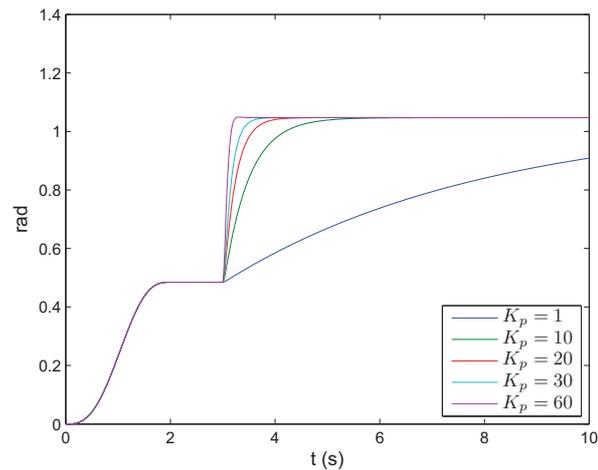


Figura 5.14: Comportamiento del robot para $K_v = 5,0$ y diversos valores de K_p .

Los valores de K_p varían entre 1 y 60. En cada caso se puede apreciar un comportamiento críticamente amortiguado. A medida que la ganancia K_p se incrementa la planta llega más rápido a referencia.

5.2.2. Controlador PID

Uno de los esquemas de control ampliamente utilizado en diversas aplicaciones es el PID. Este controlador contiene tres elementos que, el proporcional, el derivativo y el integral al error de las variables de estado. El controlador PID se expresa como en la siguiente ecuación:

$$\tau_{PID} = \mathbf{J}(\boldsymbol{\nu})^T \left[\mathbf{K}_p \tilde{\boldsymbol{\eta}} + \mathbf{K}_d \dot{\tilde{\boldsymbol{\eta}}} + \mathbf{K}_i \int \tilde{\boldsymbol{\eta}}(\sigma) d\sigma \right] \quad (5.4)$$

donde $\tilde{\boldsymbol{\eta}} = \boldsymbol{\eta}_d - \boldsymbol{\eta}$ es el error entre la posición real y la deseada, \mathbf{K}_p es la matriz de ganancias proporcionales, \mathbf{K}_d ganancias derivativas, y \mathbf{K}_i es la matriz de ganancias integrales. En la figura 5.15 se presenta un diagrama de bloques de un sistema de control PID para robots submarinos.

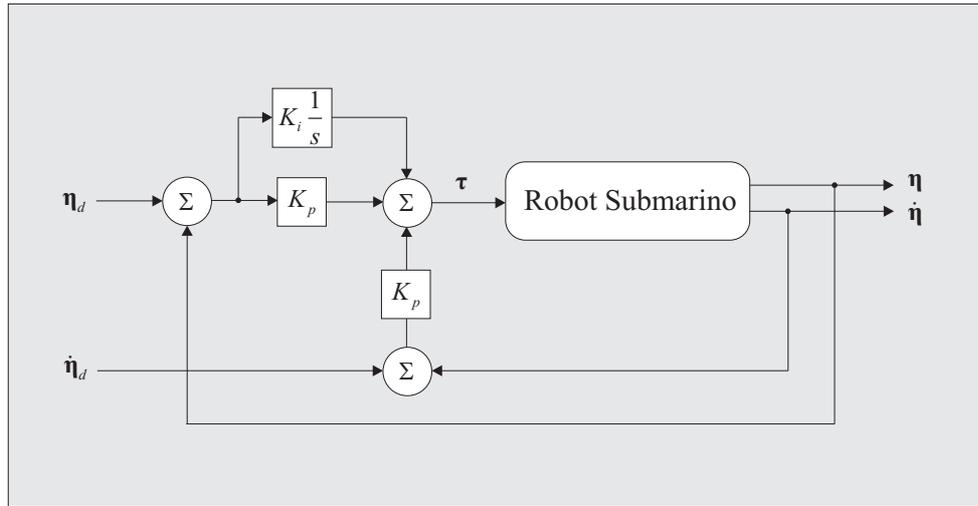


Figura 5.15: Controlador PID.

En robots manipuladores, cuando los eslabones del robot están sujetos a fuerzas gravitacionales es necesario introducir la componente integral. El controlador PID aparece como una alternativa a los controladores PD con compensación de gravedad y al PD con compensación de precalculada de gravedad.

Al igual que en el caso del controlador P con retroalimentación de velocidad de explorará el desempeño del robot cuando se controla bajo un esquema tipo PID. Principalmente se estudiará el efecto que tiene la componente integrar sobre el sistema.

Al igual que en el caso del controlador P con retroalimentación de velocidad, se analiza el desempeño del robot para tres maniobras. En primer lugar se estudia el caso de un movimiento hacia adelante. En la figura 5.16 se presenta la respuesta del sistema para diferentes valores de la ganancia integral K_i . Las ganancias proporcional y derivativa se establecieron como $K_p = 50$ y $K_v = 30$. Estas ganancias proporcionaron un buen desempeño (críticamente amortiguado) para el controlador P con retroalimentación de velocidad

De la exploración de diversos valores de K_i se pudo observar que la componente integral del controlador produce en general una respuesta más lenta del sistema. Como se puede observar en la figura 5.16 en ningún caso de los mostrados el controlador logra llegar a referencia después de 10 s. A medida que se incrementa el valor de K_i el comportamiento del controlador empeora. Sin embargo, después de cierto tiempo el robot logra llegar a la referencia

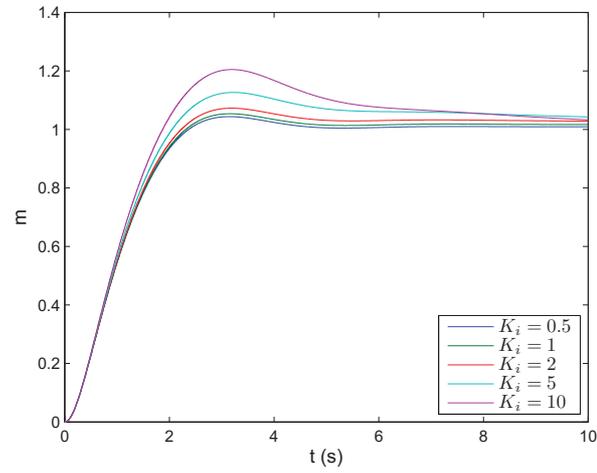


Figura 5.16: Comportamiento del robot empleando un controlador PID, $K_p = 50$ y $K_v = 30$.

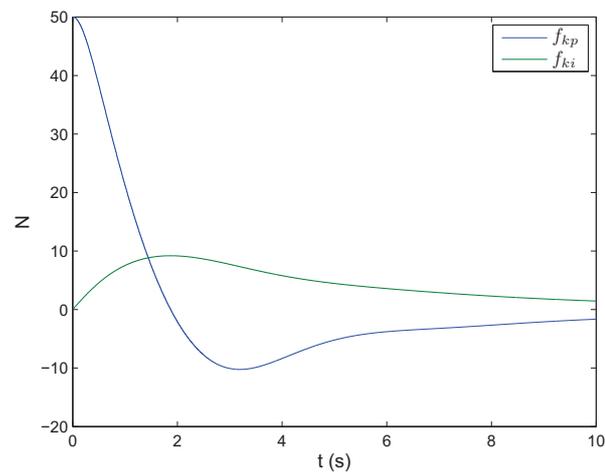


Figura 5.17: Fuerzas debido a la componente proporcional (f_{kp}) y a la componente integral (f_{ki}).

Con el motivo de esclarecer el comportamiento del controlador, en la figura 5.17 se presenta las fuerzas producto de las componentes proporcional (f_{kp}) e integral del controlador (f_{ki}). Como es de esperar la fuerza proporcional tiene su valor más alto al principio ya que el error de posición es el máximo. A medida que el robot alcanza la referencia el valor de esta fuerza disminuye. Por otro lado, f_{ki} es cero al principio pero a medida que transcurre el tiempo la integral del error va incrementándose, hasta alcanzar su máximo cerca de los 2 s. Después reduce lentamente su valor. Se puede observar que esta fuerza es casi igual pero de signo contrario a f_{kp} después de cierto tiempo (aprox. después de 6 s). De alguna manera este par de se anula lo cual no permite que el controlador tenga como salida la fuerza necesaria para reducir el error.

En la figura 5.18 se presenta el desempeño del robot al realizar un movimiento traslacional lateral bajo el esquema de control PID. Se aplicó la estrategia en la que las variables a controlar son la posición y velocidad de los centro de ambas plataformas. Se observó que de igual manera que en caso anterior, la llegada a la referencia fue más lenta que en el caso del controlador P con retroalimentación de velocidad.

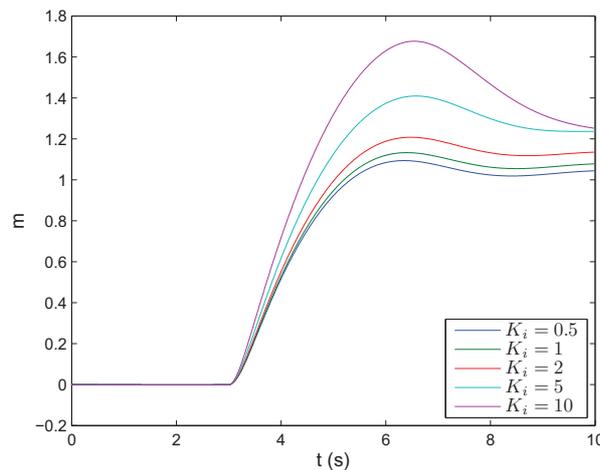


Figura 5.18: Movimiento lateral empleando un PID.

Por otro lado, en la figura 5.19 se presenta la respuesta del sistema para un movimiento rotacional cuando el robot es controlado por un PID. Se observó que en este esquema, una incorrecta selección de la ganancia integral puede provocar una pérdida total del control del sistema (lo mismo sucede para la ganancia proporcional), véase la grafica para $K_i = 4$. El desempeño del controlador se empeora a medida que la ganancia proporcional se incrementa.

En términos generales se observó que la componente integral hizo que el sistema fuera más lento para llegar a la referencia deseada. En ocasiones la componente integral es empleada para eliminar aquel error que un PD por sí solo no puede. Esto puede suceder cuando existen fuerzas externas persistentes aplicándose sobre el robot. Tal es el caso de un robot sin flotabilidad neutra. El robot se encuentra sujeto a una fuerza

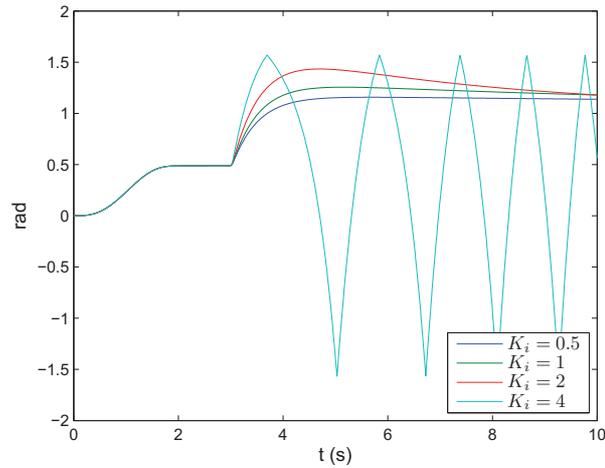


Figura 5.19: Movimiento rotacional empleando un PID.

que continuamente se aplica hacia arriba o hacia abajo y un PD es insuficiente para lograr el objetivo de control. Un PID es un controlador aplicable en condiciones más generales que un PD. Sin embargo, en el caso del robot de estudio, en el que se considera flotabilidad neutra, un controlador P con retroalimentación de velocidad es más eficiente que un PID.

5.3. Control de la Estructura Paralela durante la navegación

En esta sección se revisará el desempeño de la estructura del robot durante la navegación. En la anterior sección se consideró que la estructura del robot era rígida. Esta condición se puede lograr a través de frenos eléctricos, o mediante algún dispositivo mecánico que cause la inmovilidad de la articulación actuada. Ahora se considerara que el robot tiene actuadores que continuamente controlan la configuración de la estructura.

Cada actuador tiene su propio controlador. Una vez establecidas la posición y orientación de una plataforma con respecto a la otra, a través de la cinemática inversa se calcula el vector de coordenadas articulares deseado. Estos valores son los que se utilizan como referencias para el sistema de control. Adicionalmente se puede incluir un algoritmo de planificación de trayectorias del que se obtenga información de la posición y velocidad articular en cada instante de tiempo.

En base a los resultados del capítulo anterior en los que se mostró que un controlador P con retroalimentación de velocidad proveía un desempeño más eficiente (dadas las condiciones de los escenarios de prueba), se utilizará este esquema en las pruebas para el control de la estructura.

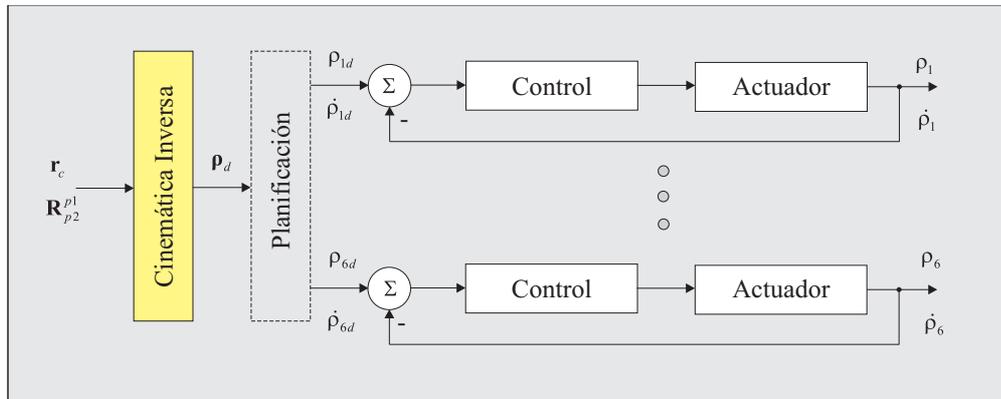


Figura 5.20: Control articular de la estructura paralela.

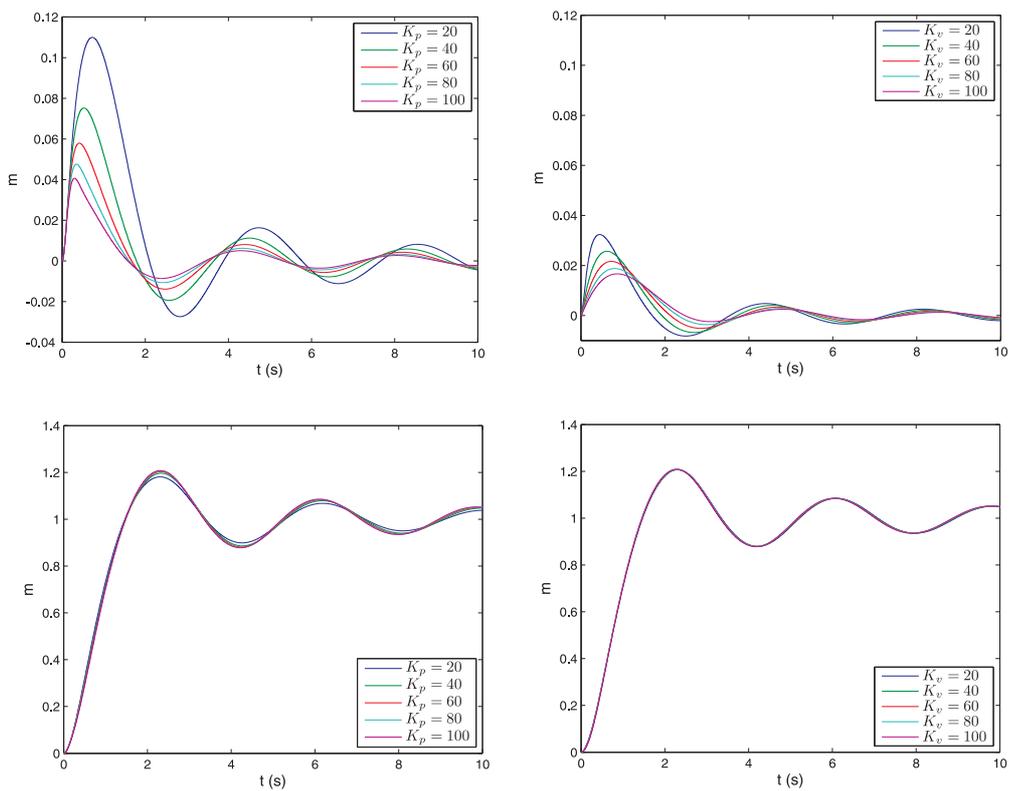


Figura 5.21: Control articular P con retroalimentación de velocidad. a) Comportamiento de las variables articulares para diferentes ganancias proporcionales y $K_v = 10$; b) comportamiento de las variables articulares para diferentes ganancias derivativas y $K_p = 100$; c) movimiento global del robot cuando se varían las ganancias proporcionales; d) movimiento global del robot cuando se varían las ganancias integrales.

5.3.1. Controlador articular P con retroalimentación de velocidad

5.3.1.1. Línea Recta

Para estudiar el desempeño de este tipo de controlador el movimiento que realiza el robot es en línea recta y hacia adelante. La posición deseada del robot es un 1m hacia el frente. La estructura debe mantenerse estática en este movimiento. Por lo tanto el vector de coordenadas articulares deseadas es $\rho = \mathbf{0}$.

Un aspecto interesante de revisar es como puede afectar el desempeño del esquema de control aplicado en los impulsores en el control de las piernas de la estructura y viceversa. Para se considero una prueba en la que el control aplicado en los impulsores generara una respuesta subamortiguada en el sistema. De esta manera se pueda apreciar el desempeño de los controladores aplicados en los actuadores de las piernas del robot.

En la figura 5.21 se presentan los resultados del movimiento rectilíneo, con ganancias $K_p = 50$ y $K_v = 1$ para el control de los impulsores. En la fig. 5.21a se presentan los valores de las variables articulares para $K_{v\rho} = 10$ y distintos valores de $K_{p\rho}$. Se observa que a mayores valores de $K_{p\rho}$ el sobre impulso es menor. Para $K_{p\rho}=100$ el sobre impulso es de 4 cm. Por otro lado en 5.21b se presenta el comportamiento de las articulaciones para $K_{p\rho} = 100$ y diferentes valores de $K_{v\rho}$. A medida que aumenta la ganancia derivativa el sobre impulso es menor $K_{v\rho} = 100$ el sobre impulso es inferior a 2 cm. En las graficas 5.21c-d se presenta el movimiento del robot al llegar a la referencia. Se observa el comportamiento subamortiguado del sistema y que las variaciones en las ganancias de los controladores de los actuadores lineales no afecta de manera considerable para el rango de ganancias consideradas.

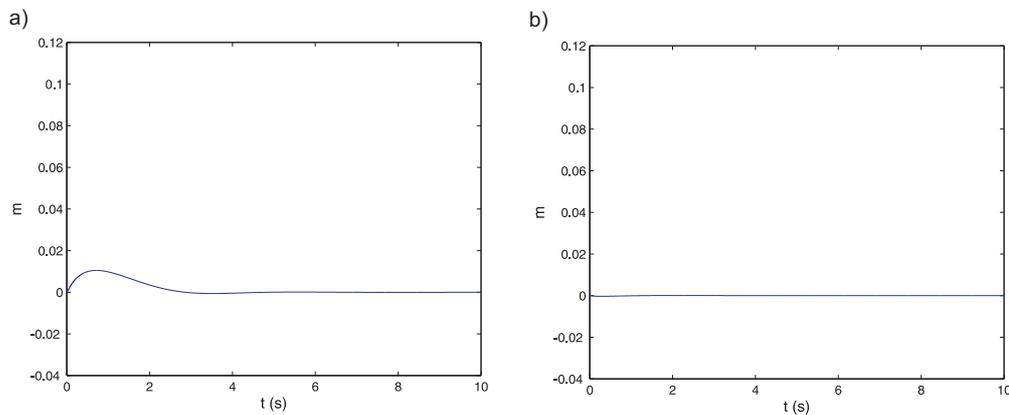


Figura 5.22: Controlador articular P con retroalimentación de velocidad, $K_p = 50$ y $K_v = 30$. a) Un impulsor y b) Dos impulsores.

Por otro lado, se realizó una prueba empleando ganancias en el sistema de control de los impulsores que generan una respuesta críticamente amortiguada, en este caso $K_p = 50$ y $K_v = 30$ (ver fig. 5.22a). Se encontró que el primer sobreimpulso que sufre la articulación prismática fue cercano a 1 cm, z y que regresa a su posición de

referencia alrededor de los 3 s. Esto muestra que un buen desempeño en el control del sistema de navegación tiene influencia en el desempeño de los controladores articulares, principalmente cuando se desea que el robot llegue a una posición de referencia.

Además de la anterior prueba se realizó una en la que ambos impulsores trabajan al mismo tiempo. En este caso fue posible observar que el error de posición en las articulaciones fue prácticamente nulo durante el movimiento que realizó el robot, véase fig. 5.22b.

5.3.1.2. Rotación Pura

Para revisar el desempeño de la estructura en un escenario más general se elige la maniobra de rotación pura. En esta maniobra los eslabones están sujetos a fuerzas y momentos por masa añadida y arrastre. Además, en este movimiento se generan diversas fuerzas inerciales como las fuerzas centrífugas.

La fig. 5.23 presenta los valores de las variables articulares durante la maniobra. Se observa que cada articulación alcanza la posición deseada antes de llegar a los 10 s. Como se explicó anteriormente, esta maniobra está compuesta de dos fases, la primera en la que el robot reconfigura su estructura, y la segunda en la que los impulsores trabajan para llegar a la orientación deseada. En la prueba, se estableció que la segunda fase comenzara a los 3 s. Se nota en la fig. 5.23 que en este punto existe un pequeño sobreimpulso. Esto se debe a que los impulsores han comenzado a funcionar y los actuadores están sometidos a nuevas fuerzas. No obstante el controlador es capaz de soportar tal perturbación. Cuando se revisa el desempeño del robot a nivel global se observa que el robot ha alcanzado la orientación deseada de $\frac{2}{3}\pi$.

5.3.2. Controlador articular PID

En este apartado se realiza una exploración sobre el desempeño de la estructura cuando se emplea un esquema PID para controlar las articulaciones. Se realizó una prueba para determinar cómo afecta la ganancia integral cuando el robot es comandado para posicionarse a una distancia de 1 m. Se emplean las mismas ganancias en el controlador de los impulsores que las empleadas en la prueba inicial de esta sección (i.e. $Kp = 50$ y $Kv = 1$). La fig. 5.24 muestra el comportamiento de la primera variable articular para diversos valores de $K_{i\rho}$. Se aprecia una mejoría en el desempeño del controlador con respecto a la magnitud del primer sobreimpulso. A medida que se incrementa la ganancia se reduce el primer sobreimpulso.

Por otro lado, un escenario en el que la estructura del robot está sometido a fuerzas persistentes, es aquel en el que se mueve a una velocidad constante. En este caso las fuerzas de arrastre son continuas. Ante esta situación con control P con retroalimentación de velocidad puede ser insuficiente. Para demostrar esto, se realiza una prueba en la que el primer impulsor genera una fuerza constante ($f_{t1} = 20N$). En la figura 5.25a se muestra el comportamiento de las variables articulares cuando se emplea un controlador P con retroalimentación de velocidad. Como se observa, existe un error con respecto a la referencia comandada ($\rho_i = 0m$). Este error es de 1.7 cm.

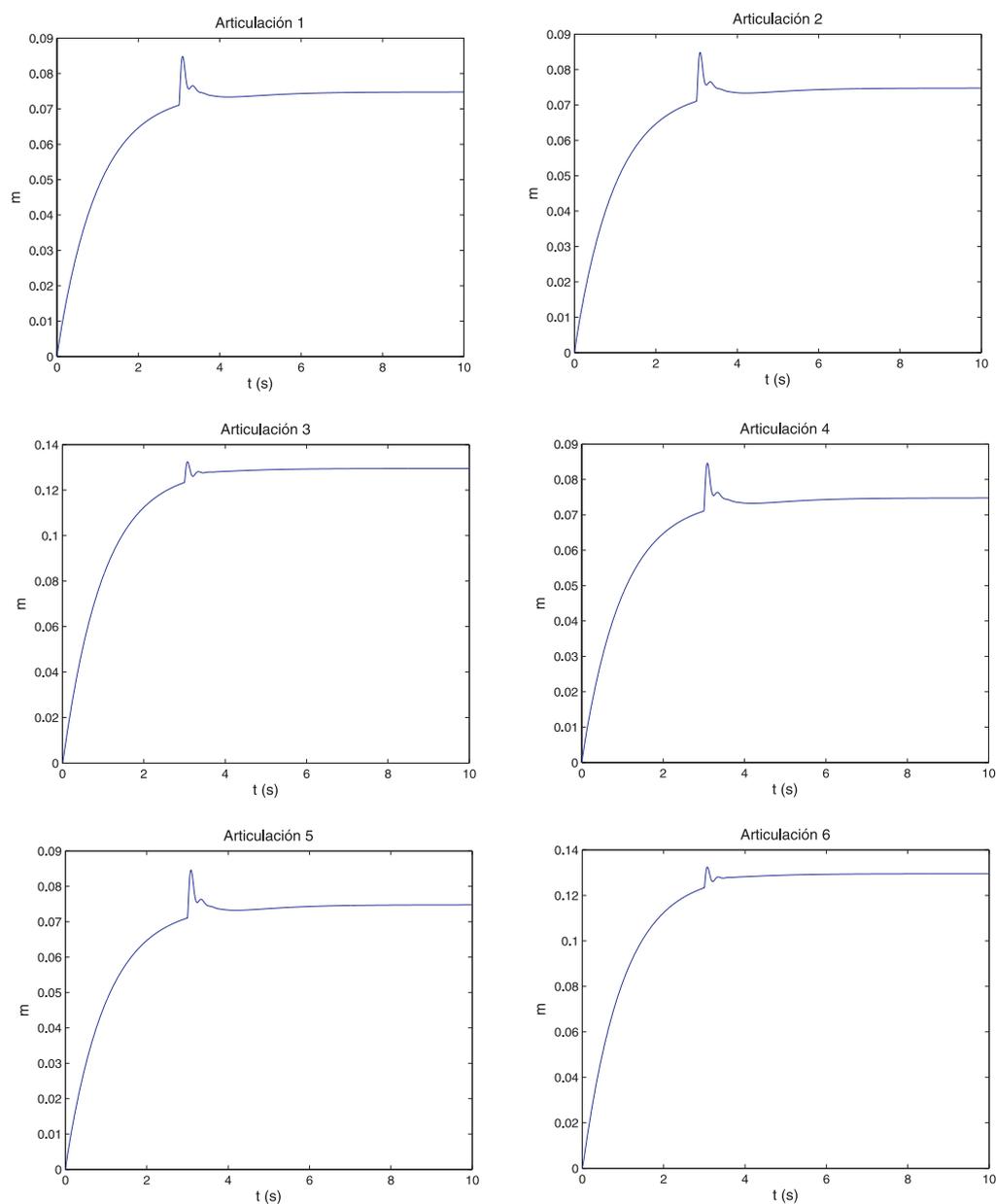


Figura 5.23: Comportamiento de las variables articulares para el movimiento rotacional del robot. Se muestra como cada articulación llega a la posición deseada, la cual corresponde a la configuración del robot que permite un movimiento rotacional.

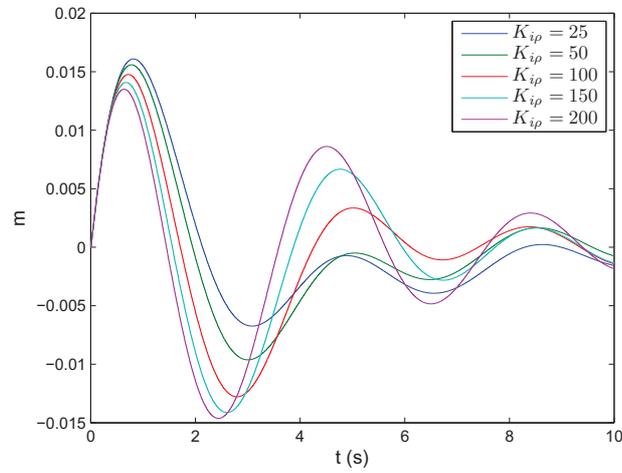


Figura 5.24: Controlador articular PID, $K_p = 50$, $K_v = 1$, $K_{pp} = 100$ y $K_{vp} = 100$.

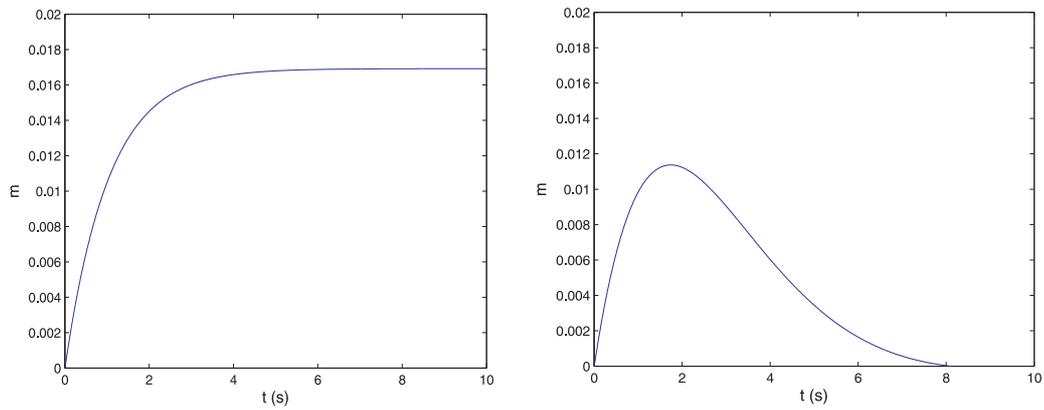


Figura 5.25: Comportamiento de las variables articulares cuando el robot se mueve a velocidad constante. a) Empleando con controlador P con retroalimentación de velocidad; b) empleando un controlador PID. Ganancias $K_{pp} = 100$ y $K_{vp} = 100$ y $K_{ip} = 50$.

Aunque las ganancias del controlador permiten que el error sea relativamente pequeño, este error puede agrandarse a medida de que las fuerzas de arrastre sean mayores. Por otro lado, en la fig. 5.25b se muestran los valores de las variables articulares cuando se controla al actuador a través de un PID. Como se muestra, este controlador si es capaz de reducir el error hasta alcanzar la referencia deseada. En este escenario se muestra la eficacia de esta clase de controlador.

5.4. Controlador PD con Precompensación

En esta sección se presenta un estudio sobre un controlador basado en el modelo del robot. Este es el controlador PD con Precompensación. El controlador emplea el modelo dinámico inverso del robot, el cual se explico en el Capítulo 4. Esta clase controlador tiene aplicación en el seguimiento de una trayectoria deseada. Para esto, la trayectoria debe estar definida en posición velocidad y aceleración en cada instante de tiempo.

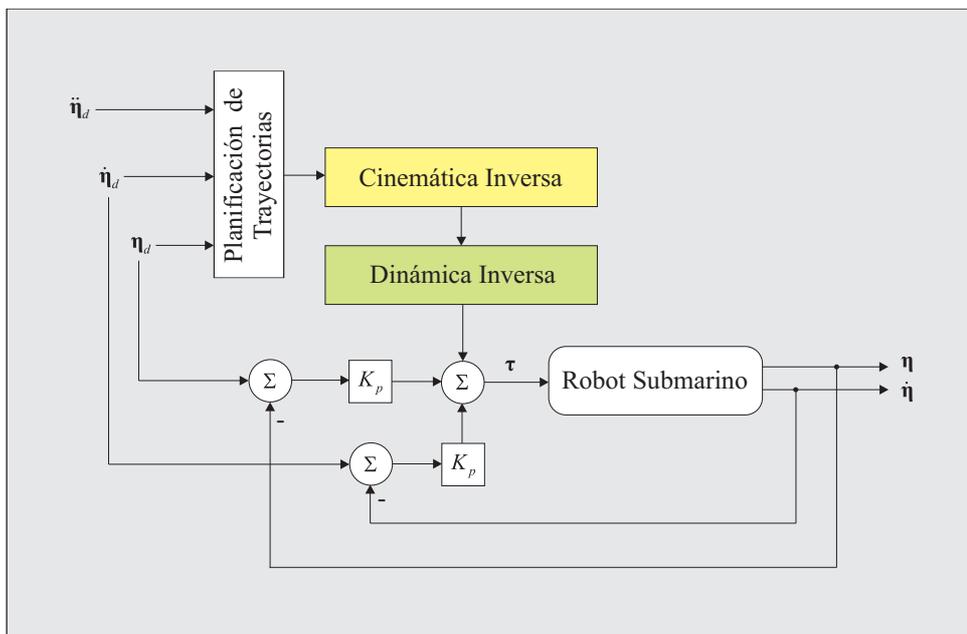


Figura 5.26: Controlador PD con Precompensación.

La ley de control tiene tres componentes, una proporcional al error de posición, otra proporcional al error de velocidad, y las fuerzas calculadas a partir del modelo dinámico inverso. Estas últimas fuerzas se calculan previamente a partir de la trayectoria deseada.

Para la prueba de este control se considera el movimiento cicloidal en línea recta que se presentó previamente en el Capítulo 4. En las fig. 5.27 se presenta el perfil de posiciones y velocidades deseadas, y en las fig. 5.27 se presentan los resultados de la dinámica inversa, que son las fuerzas que debe generar el impulsor 1 y los actuadores lineales de las piernas.

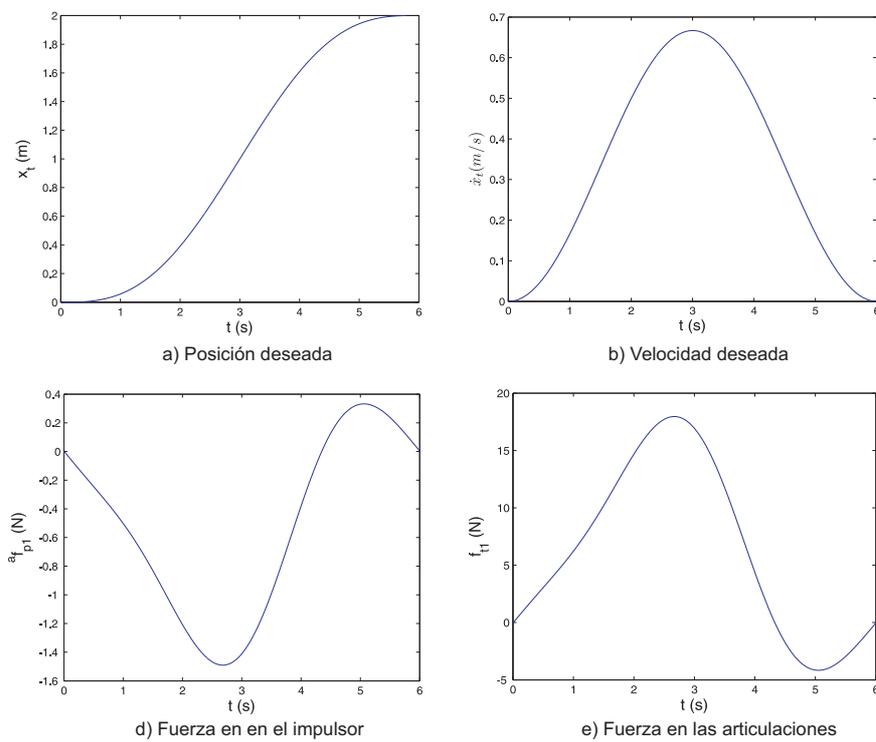


Figura 5.27: Trayectoria deseada. a) Posición y b) velocidad deseada de la plataforma 1 del robot; c) fuerza del impulsor 1 y d) fuerza en los actuadores lineales obtenidas a partir de la dinámica inversa.

Antes de revisar los resultados del empleo de este controlador se analizan los resultados de un controlador PD. En la figura 5.28 se presentan los errores de posición y velocidad que se dan durante el seguimiento de la trayectoria. Se observa que el error de posición en cierto punto llega a ser mayor que 25cm . Por otro lado, el error de velocidad máximo es cercano a $0,2\text{m/s}^2$.

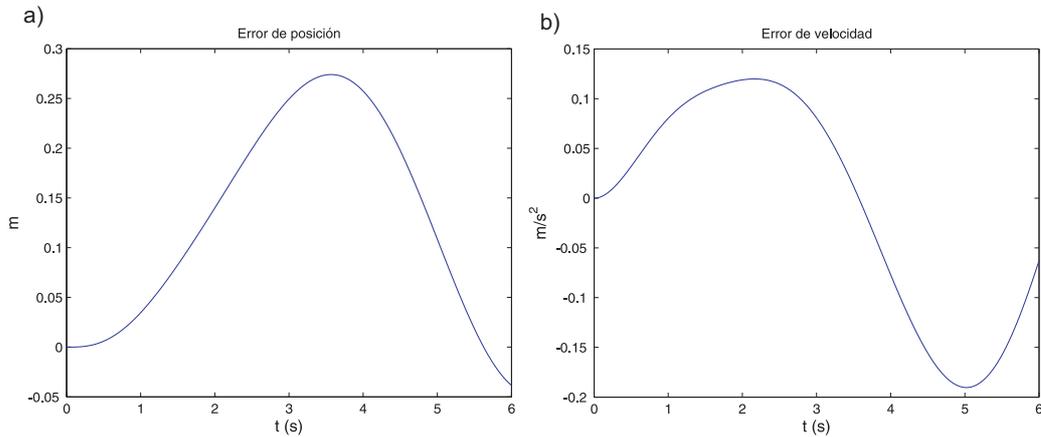


Figura 5.28: Errores de posición y velocidad empleando un controlador PD.

En la fig. 5.29 se presentan los errores de posición y velocidad empleando el esquema de control PD con Precompensación. Se emplearon las mismas escalas de los ejes que en la fig. 5.28. Como se observa, los errores de posición y velocidad son prácticamente nulos. Esto demuestra que bajo las condiciones del experimento (sin perturbaciones y con parámetros conocidos del sistema) el empleo de la dinámica inversa es notablemente superior. Los resultados también verifican la validez del modelo presentado en el Capítulo 4.

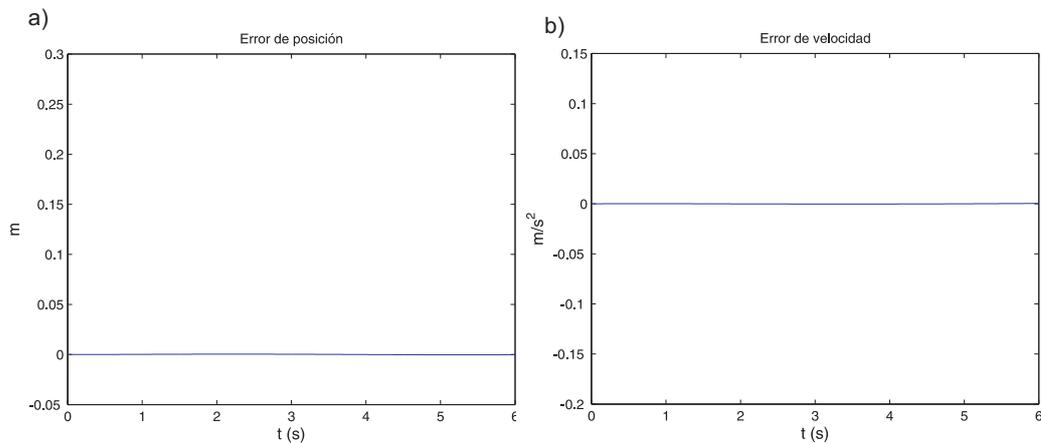


Figura 5.29: Errores de posición y velocidad empleando un controlador PD con Precompensación.

Aunque las condiciones ideales en las que se realiza esta prueba son difíciles de encontrar en ambientes submarinos, los resultados presentados muestran lo prometedores que pueden ser otros controladores basados en el modelo (o en su estructura), como lo sería el controlador par calculado o un controlador adaptable. Lo anterior es una conclusión importante de este trabajo.

5.5. Resumen

En este capítulo se presentó una exploración sobre el desempeño de algunas estrategias de control para el robot submarino Remo 2. Por un lado se analizó el comportamiento del robot empleando diversas leyes de control, y por otro lado mediante la simulación del robot fue posible observar aspectos de la dinámica del robot en ambientes submarinos.

Se observó que la simetría de la arquitectura del robot facilitaba la realización de maniobras. Por ejemplo, cuando el robot se mueve en línea recta hacia adelante, las fuerzas de arrastre que se generan en los eslabones de las piernas, se anulaban perfectamente unas a otras. Se realizaron pruebas en las que la distribución de las fuerzas de arrastre en piernas no era simétrica y encontró que se generaba un desbalanceo que hacía que se perdiera el equilibrio de fuerzas y a su vez se provocaran movimientos indeseables.

Los esquemas de control que se estudiaron fueron P con retroalimentación de velocidad, PID, PD con Precompensación, tanto para controlar a los impulsores, como para controlar los actuadores lineales de las piernas. Se revisó el desempeño de estos controladores con tres maniobras de prueba. Línea recta hacia delante, movimiento rotacional puro, y movimiento lateral.

Se observó que para controlar la posición de un robot con las características de Remo 2, un controlador P con retroalimentación de velocidad es más eficiente que un controlador PID. La componente integral hacia más lenta, en general, la llegada del sistema a la posición de referencia. Esta componente es en general útil cuando se desea eliminar el error que la componente proporcional no es capaz de eliminar. Sin embargo en las condiciones del análisis no se da esta situación. No obstante, si el robot estuviera sujeto a las fuerzas de una corriente de agua continuamente un controlador PID puede ser más eficaz.

En las pruebas que se realizaron para evaluar el control de la estructura paralela del robot, se encontró que el controlador PID fue más eficaz que un controlador P con retroalimentación de velocidad. En un escenario en el que la estructura del robot está sometida a fuerzas persistentes, como cuando se mueve a una velocidad constante, se mostro que el PID si era capaz de reducir el error hasta alcanzar la referencia deseada.

Posteriormente se presentó un estudio sobre un controlador basado en el modelo del robot. Este es el controlador PD con Precompensación. El controlador emplea el modelo dinámico inverso del robot, el cual se explico en el Capítulo 4. Esta clase controlador tiene aplicación en el seguimiento de una trayectoria deseada. Para esto, la trayectoria debe estar definida en posición velocidad y aceleración en cada instante de tiempo.

Se observó que en el caso del controlador PD con Precompensación, los errores de posición y velocidad eran prácticamente nulos. Esto demostró que bajo las condiciones del experimento (sin perturbaciones y con parámetros conocidos del sistema) el empleo de la dinámica inversa es notablemente superior. Los resultados también verifican la validez del modelo presentado en el Capítulo 4.

Aunque las condiciones ideales en las que se realizó esta prueba son difíciles de encontrar en ambientes submarinos, los resultados presentados muestran lo prometedor que pueden ser otros controladores basados en el modelo (o en su estructura), como lo sería el controlador par calculado o un controlador adaptable. Lo anterior es una conclusión importante de este trabajo.

Capítulo 6

Diseño de Robots Submarinos con IVEP

6.1. Introducción

Un aspecto importante en el desempeño del robot Remo 2 es la capacidad de su estructura para reconfigurarse y lograr que sus plataformas obtengan diversas posiciones (y orientaciones) relativas entre ellas. El diseño de la estructura, tanto su topología y dimensiones, tienen un efecto determinante en este sentido. Para conocer la capacidad de este robot para ejecutar diferentes reconfiguraciones es importante realizar un estudio de su espacio de trabajo. Este estudio expondrá las acotaciones de movimiento que la estructura del robot posee debido principalmente a su geometría.

Además, es importante identificar las configuraciones o regiones del espacio de trabajo en las que la estructura del robot tenga una mejor capacidad para realizar su movimiento. Para ello es importante contar con índices de desempeño que evalúen la capacidad de una postura para realizar una reconfiguración. En esta capítulo se analizará el espacio de trabajo empleando diversos índices de desempeño cinetostáticos.

Finalmente, se presenta en este capítulo un trabajo de optimización de la geometría del robot Remo 2. Para ello se introducen algunos índices de desempeño globales, se definen los parámetros de diseño, y se hace una revisión del espacio de los parámetros con el fin de identificar el diseño óptimo de la estructura del robot.

6.2. Análisis del Desempeño Cinetostático

En esta sección se realizará un análisis sobre el desempeño cinetostático del robot Remo 2. Para ello se emplea el número de condición de la matriz Jacobiana el cual permite evaluar la capacidad de la estructura del robot para transformar las velocidades o fuerzas en los actuadores lineales, en velocidades o fuerzas relativas entre ambas plataformas. Antes de definir este índice de desempeño se realiza una revisión de la matriz Jacobiana, el modelo cinemático de velocidad y del modelo estático del robot.

El modelado cinemático de un robot consiste en la determinación de las relaciones existentes entre las coordenadas operacionales del robot y sus variables articulares; las

coordenadas operacionales describen la situación (posición y orientación) de una de las plataformas con respecto a la otra; y las variables articulares son las que definen la posición relativa de un eslabón de la cadena cinemática con respecto al precedente.

El modelado cinemático de velocidad relaciona las velocidades articulares y las velocidades operacionales del robot; en robots paralelos dichas relación generalmente se representa de la siguiente manera:

$$\dot{\mathbf{x}}\mathbf{J}_x = \mathbf{J}_\rho\dot{\boldsymbol{\rho}} \quad (6.1)$$

En la anterior ecuación \mathbf{J}_x y \mathbf{J}_ρ son, respectivamente, las matrices Jacobianas paralela y serial (Gosselin y Angeles, 1999); $\dot{\mathbf{x}}$ es el vector de velocidades operacionales del órgano terminal del robot, y $\dot{\boldsymbol{\rho}}$ es el vector de velocidades articulares del robot.

Alternativamente se puede escribir el modelo de velocidad de la siguiente manera:

$$\dot{\mathbf{x}} = \mathbf{J}\dot{\boldsymbol{\rho}} \quad (6.2)$$

donde $\mathbf{J} = \mathbf{J}_x^{-1}\mathbf{J}_\rho$.

Para la plataforma de Stewart-Gough, las matrices Jacobianas pueden tener la siguiente forma:

$$\mathbf{J}_x = \begin{bmatrix} \mathbf{s}_1^T & (\mathbf{b}_1 \times \mathbf{s}_1)^T \\ \mathbf{s}_2^T & (\mathbf{b}_2 \times \mathbf{s}_2)^T \\ \vdots & \vdots \\ \mathbf{s}_6^T & (\mathbf{b}_6 \times \mathbf{s}_6)^T \end{bmatrix} \text{ y } \mathbf{J}_\rho = \mathbf{I}_6$$

donde \mathbf{s}_i son los vectores unitarios a lo largo de las i piernas del manipulador, y \mathbf{b}_i son los vectores que definen la posición de las articulaciones universales (o esféricas) sobre la plataforma móvil.

La matriz \mathbf{J}_x es la misma que \mathbf{J}_{p1} o \mathbf{J}_{p2} derivadas en el capítulo 4 cuando la velocidad y aceleración de una de las plataformas es cero.

Por otro lado, el modelado estático consiste en determinar las relaciones que existen entre las fuerzas que generan los actuadores y la fuerza (y momento) que se genera en el efector final cuando el robot está en una posición fija. Este análisis se puede realizar estudiando los diagramas de cuerpo libre de cada eslabón y resolviendo las ecuaciones de fuerzas y momentos para una postura estática del robot.

Es posible comprobar que la fuerza en el efector final y la fuerza en los actuadores se relacionan a través de la transpuesta de la Jacobiana:

$$\boldsymbol{\tau} = \mathbf{J}^T \mathbf{f} \quad (6.3)$$

donde $\boldsymbol{\tau}$ y \mathbf{f} representan el vector de fuerzas de los actuadores y el vector de fuerzas y momentos externos en el efector final, respectivamente. Se debe observar que esta ecuación no considera las internas externas que actúan sobre los eslabones de las piernas.

Como se puede notar, el modelo cinemático de velocidad y el modelo estático se basan en la obtención de una matriz Jacobiana. A continuación se describirán algunos de índices de desempeño que están definidos en función de la matriz Jacobiana del robot.

6.2.1. Número de Condición

Los índices de desempeño cinetostático permiten evaluar la capacidad del manipulador para transformar las velocidades o fuerzas en los actuadores, en velocidades o fuerzas en el efector final. La mayoría de estos índices están definidos en función de la matriz Jacobiana del manipulador, la cual establece las relaciones de velocidad y fuerza entre las articulaciones y el efector final. Los elementos de la matriz Jacobiana dependen únicamente de la geometría del manipulador y su configuración. El número de condición, un índice de desempeño cinetostático, puede ser utilizado en la planificación de trayectorias de tal manera que el robot acoja configuraciones en las cuales su capacidad para generar velocidades y fuerzas, ó la exactitud en el control, sea optimizada.

Un concepto que ayuda a visualizar la definición de este índice es el elipsoide de velocidad. El elipsoide de velocidad fue propuesto inicialmente por Yoshikawa (1985) para la definición del índice de la manipulabilidad, sin embargo también resulta útil para la definición de otros índices de desempeño cinemático.

Para comenzar considérese el modelo de velocidad de un robot manipulador:

$$\dot{\mathbf{x}} = \mathbf{J}\dot{\mathbf{q}} \quad (6.4)$$

donde $\dot{\mathbf{x}}$ es el vector de velocidades del efector final, $\dot{\mathbf{q}}$ es el vector de velocidades articulares, y \mathbf{J} es la matriz Jacobiana.

El elipsoide de velocidad esta dado por el conjunto de todas la velocidades del efector final que son realizables para las velocidades articulares que satisfacen que $\|\dot{\mathbf{q}}\| \leq 1$.

Dado que

$$\|\dot{\mathbf{q}}\|^2 = \dot{\mathbf{q}}^T \dot{\mathbf{q}} = \dot{\mathbf{x}}^T \mathbf{J}^T \mathbf{J} \dot{\mathbf{x}},$$

se tiene que las velocidades que forman parte del elipsoide de velocidad son aquellas que satisfacen la siguiente condición:

$$\dot{\mathbf{x}}^T \mathbf{J}^T \mathbf{J} \dot{\mathbf{x}} \leq 1.$$

Del elipsoide de velocidad se puede observar que el robot tiene la capacidad de moverse a mayor velocidad cuanto mayor sea el volumen del elipsoide. Para las velocidades en el espacio articular que cumplan con $\|\dot{\mathbf{q}}\| \leq 1$, las mayores velocidades que efector final puede alcanzar, se obtienen sobre el semieje mayor del elipsoide. Por el contrario, el efector final del robot se moverá con a baja velocidad en dirección del semieje menor del elipsoide. Si la forma del elipsoide se aproxima a la esfera, entonces el efector final se puede mover uniformemente en todas direcciones.

Mediante la descomposición de la matriz Jacobiana en valores singulares se puede demostrar que los valores singulares $\sigma_1, \sigma_2, \dots, \sigma_m$, corresponden a las magnitudes de los ejes principales del elipsoide de velocidad. Los valores singulares, corresponden también a las raíces cuadradas de los valores propios (o eigenvalores) de la matriz $\mathbf{J}^T \mathbf{J}$.

Cuando la matriz Jacobiana es singular, al menos uno de los valores singulares es igual a cero. Aquellas configuraciones en las cuales la matriz \mathbf{J} es singular, se

llaman configuraciones singulares. En el caso de robot paralelos, existen dos matrices Jacobianas, y cuando alguna de estas es singular el robot pierde o gana grados de libertad. Por lo anterior, estas configuraciones deben ser evitadas durante la ejecución de una tarea.

El empleo del número de condición como índice de desempeño fue propuesto por Salisbury (1982) para el diseño de los dedos de una mano robótica. El número de condición κ de una matriz Jacobiana \mathbf{J} está definido de la siguiente manera:

$$\kappa(\mathbf{J}) = \|\mathbf{J}\| \times \|\mathbf{J}^{-1}\| \quad (6.5)$$

donde $\|\cdot\|$ es la norma de la matriz. Esta norma puede ser cualquiera, como la norma de Frobenius

$$\|\mathbf{J}\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |J_{ij}|^2},$$

donde J_{ij} representa el elemento de la i -ésima fila y j -ésima columna de la matriz \mathbf{J} . Por otro lado, el número de condición puede ser calculado en función de los valores singulares de la matriz Jacobiana, $\kappa = \frac{\sigma_M}{\sigma_m}$, donde σ_M es el máximo valor singular y σ_m es el mínimo valor singular.

En general, el número de condición de una matriz expresa la sensibilidad de la solución de un sistema matricial a los errores en los datos conocidos y/o de la matriz misma.

Cuando el número de condición alcanza su mínimo valor (la unidad), se dice que la matriz Jacobiana es isotrópica. En este caso el elipsoide de velocidad se vuelve una esfera. Por otra parte, el número de condición se vuelve infinito si la matriz jacobiana es singular.

En el contexto de la robótica el número de condición de la matriz Jacobiana es un índice interesante en la evaluación del desempeño cinemático del manipulador. Este índice mide la precisión del control del manipulador (Salisbury (1982)) y la uniformidad de la distribución de velocidades y fuerzas operacionales; esta segunda característica es de principal importancia para las tareas de proceso como el maquinado (Chablat (2003); Huang (1998)).

El número de condición ha sido frecuentemente utilizado como criterio para el diseño de manipuladores isotrópicos (Zanganeh (1997); Baron L. y Bernier (2001); Huang (2004); Moreno (2011)). Por otra parte, este número ha sido aplicado en el contexto de la inversión cinemática y planificación de movimientos (Chablat (1998a); Gosselin (1990b); Moreno (2006)). Para manipuladores que tienen múltiples soluciones del modelo cinemático inverso de posición, el número de condición puede ser utilizado para elegir el mejor modo de trabajo del manipulador (Chablat (1998b); Chablat (2002)).

Mínimo Valor Singular

El mínimo valor singular fue propuesto por Klein (1987) como índice de desempeño. Naturalmente este valor permite medir la cercanía de la configuración del robot a una

configuración singular, ya que directamente indica si la magnitud de alguno de los ejes principales del elipsoide esta apunto de anularse. En el caso del número de condición y la manipulabilidad, esta situación (la cercanía a una singularidad) en algunos casos puede ser imprecisa ya que su valor también depende de otros valores singulares y la disminución en un valor singular puede ser compensada por un incremento o decremento en otro valor singular. El volumen del elipsoide de velocidad será mayor cuando el valor de este índice sea mayor.

6.2.2. Homogeneización de la Matriz Jacobiana

Los índices de desempeño cinetostático en general, y algunos de los índices de desempeño dinámico, están definidos en función de la matriz Jacobiana y/o la matriz de inercia. Cuando el manipulador es de mas de tres grados de libertad o sus articulaciones no son del mismo tipo, las unidades de los elementos de estas matrices no son homogéneas, y esta situación dificulta el uso de los índices de desempeño que están definidos en función de ellas.

Para que los índices de desempeño basados en la matriz Jacobiana puedan ser evaluados de manera consistente, los elementos de la matriz Jacobiana deben ser dimensionalmente homogéneos (Lipkin (1989)). En la literatura científica se ha advertido (Doty (1993)) que el uso de índices de desempeño de una matriz Jacobiana cuyos elementos no sean dimensionalmente homogéneos puede causar problemas importantes en el diseño y el control de un manipulador.

Algunos autores han sugerido que se divida el estudio de los índices de desempeño para manipuladores con Jacobianas homogéneas y no homogéneas (Di Gregorio (2002); Di Gregorio (2006); Angeles (2007)). Para los manipuladores cuya Jacobiana es homogénea, es decir, aquellos manipuladores de posicionamiento ú orientación de 2 o 3 grados de libertad cuyas articulaciones son del mismo tipo, los índices presentados anteriormente tiene aplicación directa y ofrecen información a nivel teórico y práctico. Por otro lado, en el caso de manipuladores cuya matriz no es homogénea, es decir aquellos manipuladores de posicionamiento y orientación de 3 o mas grados de libertad, se han propuesto diversas formas de solucionar este problema. Algunas de estas estrategias se presenta a continuación.

Una manera de solucionar este problema se basa en la aplicación de la *longitud característica* (Angeles (2007)). Para homogeneizar la matriz Jacobiana se dividen los elementos de esta matriz asociados a las componentes de la velocidad angular por la longitud característica. La selección de esta longitud característica es arbitraria, sin embargo varios autores proponen que sea aquella que minimice el número de condición de la matriz Jacobiana normalizada (Ma and Angeles (1991); Alba (2005)). Para la plataforma de Stewart-Gough Ma and Angeles (1991) encontraron la longitud característica que optimiza el número de condición de la matriz Jacobiana. En ese trabajo se encontró que el mínimo número de condición de la Jacobiana se obtiene en las llamadas *posturas neutrales* (estas posturas son tales que las dos plataformas del manipulador son paralelas entre sí y al mismo tiempo están centradas). Para calcular el valor de la longitud característica en (Moreno (2010)), se planteó el problema como

uno de optimización siendo la función objetivo el número de condición de la matriz Jacobiana normalizada $\kappa(\bar{\mathbf{A}}_x)$ y como variables de diseño la postura definida por la coordenada en el eje vertical z_p y la longitud característica L_c . La matriz normalizada esta dada por

$$\bar{\mathbf{J}}_x = \begin{bmatrix} \mathbf{s}_1^T & (\mathbf{b}_1 \times \mathbf{s}_1)^T / L_c \\ \mathbf{s}_2^T & (\mathbf{b}_2 \times \mathbf{s}_2)^T / L_c \\ \vdots & \vdots \\ \mathbf{s}_6^T & (\mathbf{b}_6 \times \mathbf{s}_6)^T / L_c \end{bmatrix},$$

donde \mathbf{s}_i son los vectores unitarios a lo largo de las piernas del manipulador, y \mathbf{b}_i son los vectores que definen la posición de las articulaciones universales sobre la plataforma móvil.

En el caso de algunos robots paralelos, otra longitud característica que se ha propuesto es el radio de la plataforma móvil. De esta forma, en el caso de la plataforma Stewart-Gough, el vector $\mathbf{b}_i \times \mathbf{s}_i$ se vuelve unitario, al igual que el vector \mathbf{s}_i , y por tanto los valores de todos los elementos de la matriz Jacobiana normalizada se encuentran en el rango de 0 y 1.

No obstante, esta estrategia ha recibido criticas debido al hecho de que la selección de la longitud característica es en cierta forma arbitraria y su significado físico es difícil de determinar.

Otra estrategia que se ha propuesto es encontrar una matriz Jacobiana homogénea basada en un modelo cinemático diferente (Gosselin (1990a); Kim (2003)). En (Kim (2003)) se reportó la formulación de una matriz Jacobiana homogénea a partir del análisis de tres puntos del órgano terminal. Tal formulación consiste en la determinación de un sistema de ecuaciones que relaciona las velocidades articulares con las velocidades de tres puntos unidos a la plataforma móvil; de este sistema se extrae una matriz jacobiana homogénea. En ese trabajo se utilizó el número de condición de esta Jacobiana homogénea para obtener el diseño óptimo de una plataforma de Stewart-Gough.

6.2.3. Singularidades de la plataforma de Stewart-Gough

Las configuraciones singulares de un manipulador paralelo ocurren cuando al menos una de las dos matrices Jacobianas es singular, es decir, que tengan su determinante nulo. En un robot paralelo se presentan tres tipos de singularidades: cuando \mathbf{J}_x es singular, cuando B es singular, cuando ambas son singulares, (Gosselin y Angeles, 1990-b).

Si \mathbf{J}_x es singular se dice que el manipulador está bajo una configuración singular paralela; en ella es posible tener un vector de velocidades operacionales diferente de cero correspondiente a velocidades articulares nulas. En la vecindad de tal configuración, el robot es capaz de moverse infinitesimalmente sin ningún cambio en sus variables articulares, por lo que se pierde el control del movimiento.

Si \mathbf{J}_p es singular se dice que el manipulador está bajo una configuración singular serial; en una configuración de este tipo existe un vector de velocidades articulares

diferente de cero que no produce movimiento alguno de la plataforma móvil. Esta singularidad generalmente corresponde a una postura ubicada en los límites del espacio de trabajo.

Si \mathbf{J}_x y \mathbf{J}_ρ son singulares se tiene una configuración singular bajo la cual el órgano terminal puede ser movido mientras los actuadores están bloqueados, y viceversa.

Las singularidades seriales no existen dentro del espacio de trabajo de este manipulador dado que \mathbf{J}_ρ es una matriz identidad.

Las singularidades paralelas son generalmente difíciles de encontrar (Tsai, 1999). Aunque es imposible encontrar todas las singularidades, algunas de ellas pueden ser identificadas. Por ejemplo cuando la plataforma móvil está sobre la plataforma fija, se tiene que $\det(\mathbf{J}_x) = 0$. La plataforma móvil puede hacer una rotación infinitesimal alrededor de cualquier eje en el plano $x - y$ sin que los actuadores se desplacen. Asimismo, se puede realizar un movimiento translacional a lo largo del eje z sin que trabajen los actuadores.

Otra singularidad paralela ocurre cuando la geometría del manipulador satisface las siguientes condiciones:

1. La geometría de la plataforma móvil es idéntica a la de la plataforma fija.
2. El manipulador asume una posición en la cual la longitud de las piernas es la misma
3. Todas las piernas son paralelas entre ellas.

Cuando se tienen las condiciones anteriores, estando bloqueados los actuadores, el punto de interés de la plataforma móvil se puede mover sobre una superficie esférica de radio igual a la longitud de de las pierna y centrada en el centro geométrico de la plataforma fija, La orientación de la plataforma móvil se mantiene fija en cualquier punto. Por otro lado, también existe la llamada singularidad de Fichter, la cual ocurre cuando la plataforma móvil se encuentra rotada 90° alrededor del eje z con respecto a la plataforma fija.

6.2.4. Espacio de Trabajo

El espacio de trabajo de un robot es la zona del espacio físico que puede ser alcanzada por un punto de su órgano terminal. Lo anterior incluye el espacio cartesiano y las diferentes formas de expresar la orientación (i.e. ángulos de Euler, cuaterniones, etc.).

Para determinar el espacio de trabajo se deben considerar las dimensiones de los eslabones, los límites en las articulaciones, y los diferentes tipos de configuraciones singulares. Además, se deben considerar las colisiones entre los eslabones del mismo manipulador.

La dificultad de calcular el espacio de trabajo depende del tipo de estructura del manipulador. En general es más complicado calcular el espacio de trabajo de manipuladores paralelos que de manipuladores seriales. Se han propuesto diferentes

métodos para determinar el espacio de trabajo. De ellos, presentamos dos, uno consiste en el estudio de la geometría de los eslabones y de los movimientos que las articulaciones permiten, y el otro consiste en la discretización del espacio de trabajo.

El *método geométrico* consiste en analizar la geometría de los eslabones y el movimiento relativo que las articulaciones permiten realizar. Utilizando conceptos de geometría se pueden establecer las condiciones que permiten determinar si un punto pertenece al espacio de trabajo. Además, podemos encontrar expresiones para el cálculo del área o el volumen del espacio de trabajo. Estas expresiones permiten que el cálculo sea rápido y exacto. Sin embargo, este método no es generalizado y se debe adaptar a cada mecanismo.

La *discretización del espacio de trabajo* consiste en definir una malla de nodos para ser explorada. Mediante un proceso iterativo se explora cada nodo para determinar si pertenece al espacio de trabajo o no. A través de este método se pueden considerar todas las restricciones posibles, como lo son los límites articulares, la colisión entre eslabones, y los diferentes tipos de singularidades. Por ejemplo, si deseamos explorar el espacio de trabajo de posición de un manipulador en el espacio cartesiano el algoritmo de exploración consistiría de tres bucles anidados, en cada uno se variaría el valor de cada coordenada cartesiana. En cada nodo se revisarían cada una de las restricciones de interés.

La precisión del espacio de trabajo depende del paso de exploración. A medida que se disminuye el paso de exploración el tiempo de cómputo crece exponencialmente. Esto no es un problema cuando se desea caracterizar el espacio de trabajo un manipulador de dimensiones dadas, ya que el cálculo se puede realizar fuera de línea en un ordenador. No obstante, en la etapa de diseño el proceso puede ser relativamente lento debido a que se deben ejecutar los bucles de exploración en cada variación de parámetros.

A continuación se enlistan algunas de las medidas del espacio de trabajo definidas en la literatura Merlet (2006). Estos valores se pueden utilizar como medidas para caracterizar a un manipulador, pero también pueden ser utilizadas como criterios de desempeño en la fase de diseño de un manipulador.

Espacio de trabajo máximo (o alcanzable) W_M . Está compuesto por todos los puntos en el espacio cartesiano que el efector final puede alcanzar con al menos una orientación (Kumar (1985)).

Espacio de trabajo de traslación W_t . Está compuesto por todos los posibles puntos en el espacio cartesiano que el efector final puede alcanzar en una orientación dada.

Espacio de trabajo de orientación W_o . Se compone de todas las posibles orientaciones que se pueden alcanzar para una posición fija del efector final (Bonev (2001)).

Espacio de trabajo diestro W_d . Esta compuesto por todos los puntos en el espacio cartesiano que pueden ser alcanzados y en los cuales se puede alcanzar todas las orientaciones.

Espacio de trabajo diestro y controlable W_{dc} . Es un subconjunto del espacio de trabajo diestro que no contiene configuraciones singulares.

Espacio de trabajo de orientación total W_{ot} . Todos los puntos en el espacio cartesiano que pueden ser alcanzadas con todas las orientaciones entre un rango de valores.

Espacio de trabajo inclusivo W_i . Todos los puntos que se pueden alcanzar con al menos una orientación dentro de cierto rango de valores

6.2.5. Análisis del espacio de trabajo del robot Remo 2

A continuación se explorará el espacio de trabajo de la estructura del robot Remo 2. Por otra parte, se presentan también diversas gráficas que muestran el comportamiento de la inversa del número de condicionamiento (κ^{-1}) de la matriz jacobiana \mathbf{J} en el interior de los correspondientes espacios de trabajo. Dichas gráficas permiten conocer globalmente los niveles de calidad del desempeño del robot, que podrían esperarse en los problemas de planificación de movimientos. En (Alba, 2005) se mostró que el número de condicionamiento de \mathbf{J} resulta una herramienta eficaz para la evaluación de la calidad de las configuraciones de un manipulador paralelo 3-RRR, en términos de su lejanía de las singularidades paralelas, con un nivel razonable de repulsividad de las singularidades seriales. Teniendo en cuenta los resultados de esa referencia, en este trabajo de tesis se decidió aplicar la inversa del número de condicionamiento de la matriz jacobiana para la evaluación de la calidad de las configuraciones correspondientes a los puntos del espacio de trabajo del robot considerado.

6.2.5.1. Espacio de Trabajo de Traslación

En la figura 6.1 se presenta el espacio de trabajo de traslación del robot para cuando los marcos de referencia de ambas plataformas son paralelas. Se observa que la forma del espacio de trabajo es una especie de casco, cuyo eje de simetría es el eje x del marco de referencia de la plataforma fija. El espacio de trabajo de traslación en este caso no posee singularidades, excepto cuando las plataformas están contenidas en el mismo plano, situación que no es práctica.

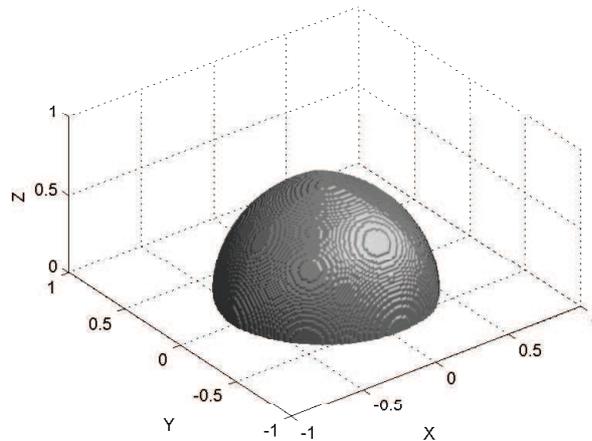


Figura 6.1: Espacio de trabajo de traslación cuando ambas plataformas son paralelas.

Por otro lado, en la figura 6.2 se muestra el comportamiento del número condición en diferentes puntos del espacio cartesiano. Se observa que los mejores valores del número de condición se encuentran en el centro de la gráfica, sobre el eje z alrededor de una altura de 0.15 m. Por otra parte, se observa el casco del espacio de trabajo (de configuraciones admisibles por los límites articulares), no posee las configuraciones con los mejores valores del número de condición.

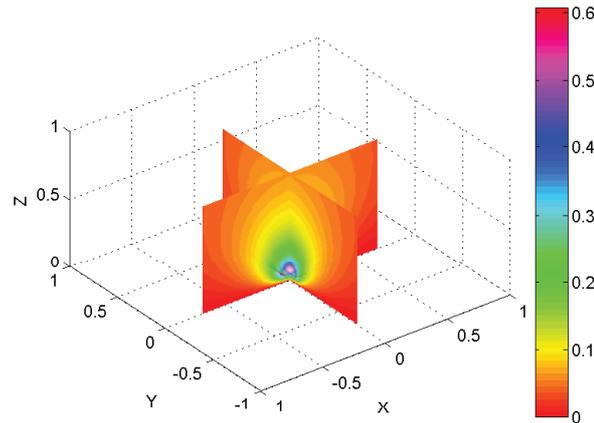


Figura 6.2: Comportamiento de κ^{-1} cuando ambas plataformas son paralelas.

Otra configuración interesante es aquella en la que los ejes centrales de ambas plataformas son ortogonales. En esta configuración de la estructura representa la máxima diferencia entre ángulos. Por otro lado, como se ha comentado anteriormente, una de las ventajas de la estructura del robot es que puede ser utilizada para deslizarse, trepar, o apoyar en las tareas de manipulación, como es el caso del robot trepador presentado en (Moreno, 2010). En esta misma referencia se realiza un estudio sobre la capacidad del robot para trepar sobre nodos estructurales, en este caso la capacidad del robot para alcanzar ángulos de 90 es importante.

Se realiza una exploración del espacio de trabajo del robot para una orientación fija de la plataforma móvil correspondiente a un giro de 90° sobre el eje y . En esta configuración la matriz de rotación de la plataforma móvil con respecto a la fija es la siguiente:

$$\mathbf{R} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix}$$

La forma del espacio de trabajo es simétrica (fig. 6.3), con el eje x como eje de simetría. Se observa que el espacio de trabajo está dividido en tres regiones que no están conectadas. La mayor parte de los puntos del espacio de trabajo están ubicados detrás

del plano $y - z$, es decir, las coordenadas de estos puntos en el eje x tienen valores negativos.

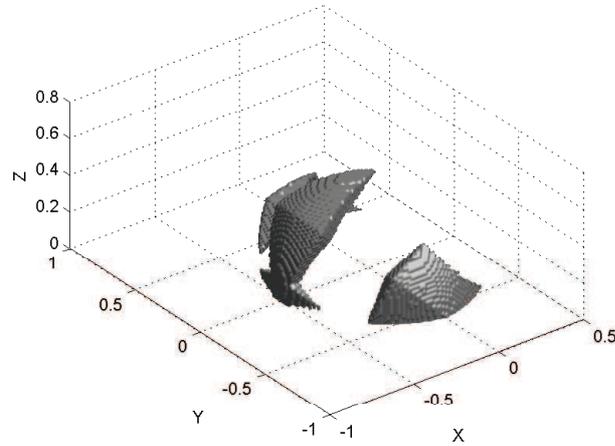


Figura 6.3: Espacio de trabajo para una orientación equivalente a un giro de 90° sobre el eje y .

En la figura 6.4 se muestra el comportamiento del número condición en diferentes puntos del espacio cartesiano. Se observa que los mejores valores del número de condición se encuentran al frente del plano $y - z$. También es posible notar que las configuraciones correspondientes a los puntos detrás del plano $y - z$ poseen un desempeño cinemático pobre. Afortunadamente, estos son los puntos que son admisibles por los límites articulares del robot. La gráfica muestra que el diseño del robot es susceptible a optimizarse en caso de que tareas que requieran una orientación de este tipo entre las plataformas sea necesaria.

Por otro lado, en la figura 6.5 se presenta el espacio de trabajo para una orientación correspondiente a un giro de -90° sobre el eje x . En este caso la matriz de rotación de la plataforma móvil con respecto a la fija es la siguiente:

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}$$

Se puede notar que el espacio de trabajo consiste de 5 regiones separadas y no es simétrico como en el caso anterior. En la figura 6.6 se muestra que los puntos con mejores valores de son aquellos cuya coordenada y es positiva.

6.2.5.2. Espacio de Trabajo de Orientación

Espacio de trabajo de orientación se compone de todas las posibles orientaciones que se pueden alcanzar para una posición fija del efector final. En este apartado se

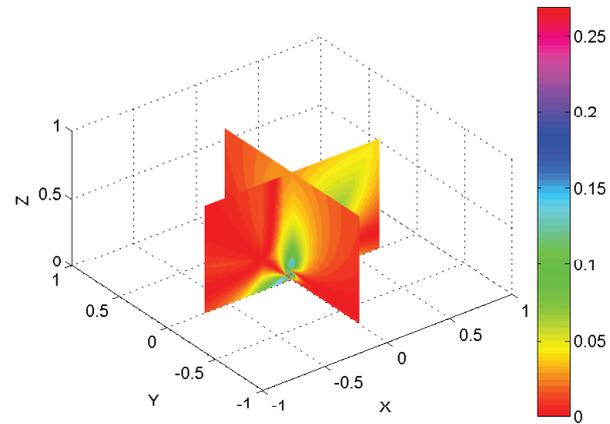


Figura 6.4: Comportamiento de κ^{-1} para una orientación equivalente a un giro de 90° sobre el eje y .

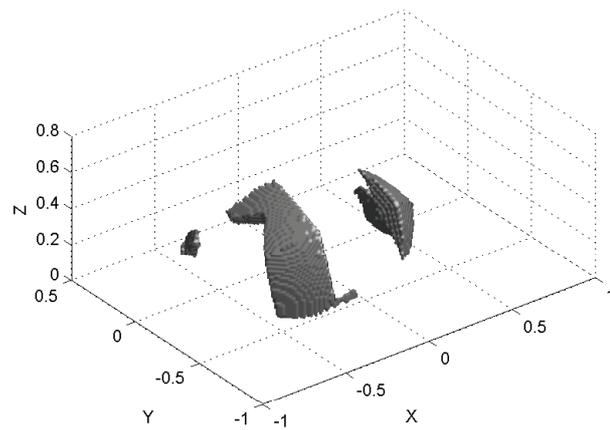


Figura 6.5: Espacio de trabajo para una orientación equivalente a un giro de 90° sobre el eje x .

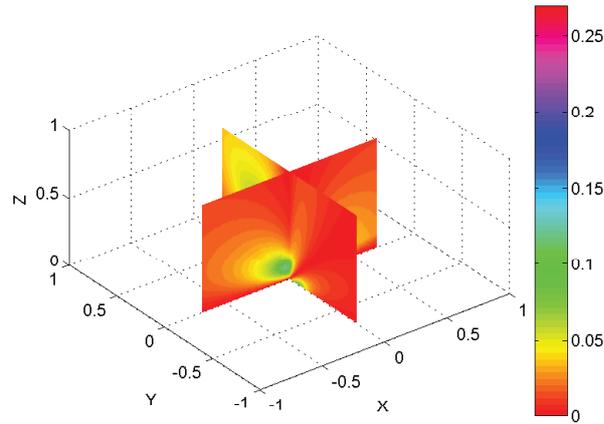


Figura 6.6: Comportamiento de κ^{-1} para una orientación equivalente a un giro de 90° sobre el eje x .

analizará el espacio de trabajo de orientación cuando el centro de la plataforma móvil está situado de tal manera que sus coordenadas x e y son cero. Se estudiará el volumen del espacio de trabajo a diversas alturas del eje z . Se considera interesante determinar el mayor espacio de trabajo dado que en estas configuraciones la plataforma móvil no realiza ninguna traslación durante un cambio de orientación. Lo anterior puede permitir que dada una orientación deseada se pueda llegar a ella evitando movimientos innecesarios.

Para establecer la orientación de la plataforma móvil con respecto a la fija consideramos los siguientes puntos:

1. Controlar la orientación en el eje z es irrelevante, pues este grado de libertad es redundante con respecto a la tarea. Por lo tanto, la orientación de la plataforma puede estar definida únicamente en función de dos parámetros, como lo son los ángulos de azimut y elevación.
2. Dada la simetría de las plataformas con respecto al eje x , se puede establecer como restricción que el eje x de la plataforma móvil se mantenga en el plano $x-z$ para evitar giros innecesarios o configuraciones complicadas.

Antes de establecer las relaciones necesarias para parametrizar la orientación, considérese que la matriz de rotación de la plataforma móvil con respecto a la fija es la siguiente:

$$\mathbf{R} = \begin{bmatrix} x_x & y_x & z_x \\ x_y & y_y & z_y \\ x_z & y_z & z_z \end{bmatrix} \quad (6.6)$$

La orientación de la plataforma, estará dada por la orientación del eje z de la plataforma móvil. Para que el eje x de la plataforma móvil este contenida dentro del

plano $x - z$, es suficiente con que $x_y = 0$. Por otro lado, dado se tienen las siguientes ecuaciones sobre el vector unitario y la ortogonalidad con el eje z :

$$x_x^2 + x_y^2 + x_z^2 = 1 \quad (6.7)$$

$$x_x z_x + x_y z_y + x_z z_z = 0 \quad (6.8)$$

Dados los valores de z_x y z_z , se puede resolver las expresiones anteriores para encontrar los valores de x_x y x_y . De la ecuación 6.8 se tiene que:

$$x_z = -\frac{z_x}{z_z} x_x \quad (6.9)$$

y por otro lado de 6.7:

$$x_x^2 + \left(\frac{z_x}{z_z} x_x\right)^2 = 1 \quad (6.10)$$

La solución de x_x es:

$$x_x = \sqrt{\frac{1}{1 + \left(\frac{z_x}{z_z}\right)^2}} \quad (6.11)$$

Para este trabajo se elige la solución positiva de la raíz cuadrada. El vector unitario correspondiente al eje y se calcula a través del producto cruz de los vectores correspondientes al eje x y z . Como se indico anteriormente, las componentes del eje z están dadas en función de un par de parámetros, el ángulo de azimut (α) y el de elevación (β):

$$\begin{bmatrix} z_x \\ z_y \\ z_z \end{bmatrix} = \begin{bmatrix} \cos(\alpha) \cos(\beta) \\ \sin(\alpha) \cos(\beta) \\ \sin(\beta) \end{bmatrix} \quad (6.12)$$

Se realizó una exploración del espacio de trabajo para diferentes alturas de la plataforma móvil, desde 0.43 m hasta 0.68 m. En la figura 6.7 se presenta el volumen del espacio de trabajo para diversas alturas de la plataforma. Se observó que el máximo espacio de trabajo se obtuvo para $z = 0,55m$. Por otro lado en la figura 6.8 se muestra cómo se comporta el valor del GCI. Se observa que el valor de este índice disminuye con la altura. Es decir, que la calidad de las configuraciones en los espacios de trabajo a diversas alturas disminuye a medida que se alejan las plataformas.

En la figura 6.9 se muestra el espacio de trabajo para $z = 0,55m$. La región blanca representa las orientaciones que puede alcanzar la plataforma móvil. Se observa que a partir de $\beta > 35^\circ$ casi cualquier orientación se puede alcanzar. Por otro lado, en la figura 6.10 se muestra el comportamiento del número de condición. A partir de esa figura se puede notar que las configuraciones admisibles tienen buenos valores del número de condición. En la región correspondiente a $\beta < 20^\circ$ se pueden apreciar algunas singularidades, sin embargo no tienen importancia ya que el robot es incapaz de lograr tales configuraciones por los límites de sus articulaciones.

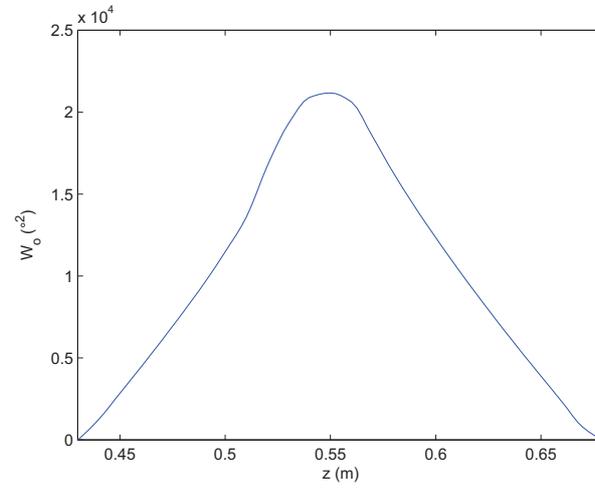


Figura 6.7: Volumen del espacio de trabajo para diversas alturas de la plataforma.

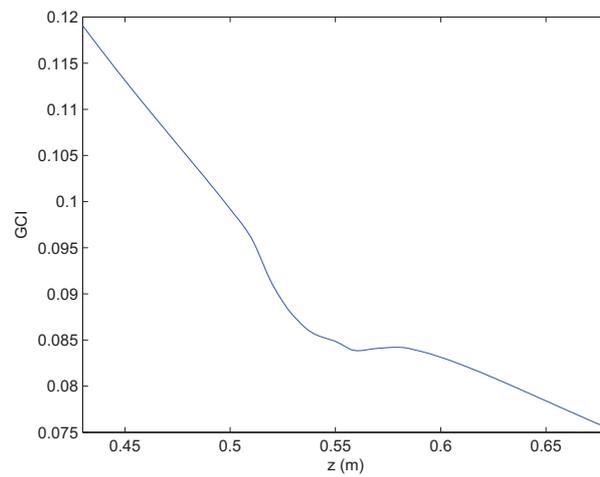


Figura 6.8: Comportamiento del GCI para diversas alturas de la plataforma.

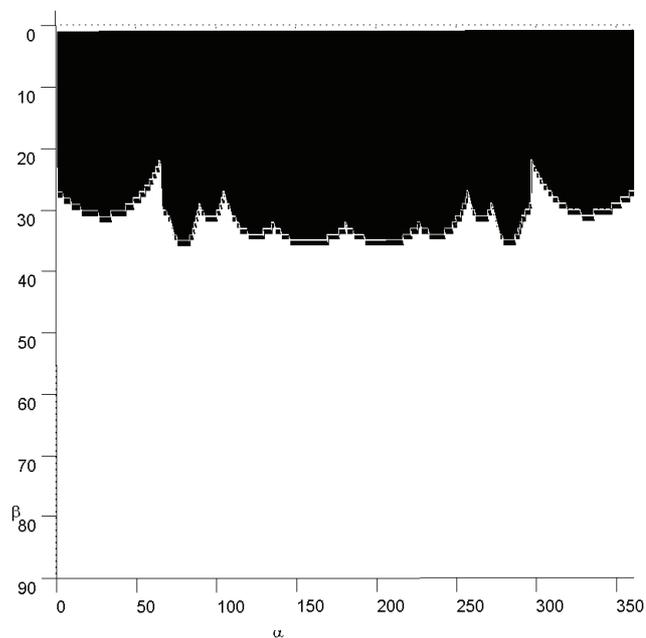


Figura 6.9: Espacio de trabajo de orientación para $z = 0,55m$.

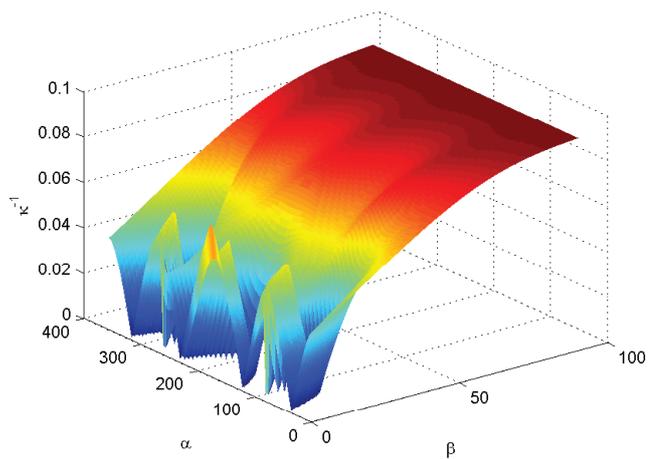


Figura 6.10: Valores de κ para $z = 0,55m$.

6.3. Diseño de la Estructura del Robot

En el diseño de un robot submarino de estructura paralela (o un impulsor vectorizado) es importante determinar la mejor estructura mecánica dados los requerimientos de las tareas a realizar (espacio de trabajo, fuerzas a soportar, rigidez, etc.). El proceso de diseño mecánico consistiría de dos puntos:

1. La síntesis estructural. Aquí se realizaría la selección de la estructura desde el punto de vista de su topología, es decir el número, el tipo y disposición de sus articulaciones.
2. La síntesis dimensional. Esta fase, una vez seleccionada la estructura, consiste en la determinación de las dimensiones de los eslabones y la ubicación de las articulaciones.

En primer lugar es necesario definir los grados de libertad de la estructura dada las características de la tarea. Una estructura paralela puede tener entre 2 y 6 grados de libertad. En este sentido, en un impulsor vectorizado se pueden controlar a diferentes niveles la posición y orientación relativa de la fuerza de empuje con respecto al centro de masa del robot. Generalmente, un AUV posee un mecanismo que permite cambiar la orientación de fuerza de empuje en dos ejes. Por otro lado, en el Remo I se demostró que cambiar la posición relativa puede ser útil, ya que esto permite variar el brazo de palanca de la fuerza del impulsor con respecto a la cabina del robot.

Por otro lado, un mecanismo para un impulsor vectorial pudiera combinar diferentes tipos de movimiento o grados de libertad de diferente naturaleza. Por ejemplo, se podría tener un mecanismo paralelo de 4 grados de libertad, que controle la orientación en dos ejes y la posición en un plano de la fuerza de impulsión con respecto al centro de masa. Diferentes combinaciones pudieran obtenerse dependiendo del tipo de movimientos deseados.

Al realizar este trabajo se identificó que una arquitectura interesante de robot son los de estructura con cinemática desacoplada. Los robots paralelos con cinemática desacoplada son aquellos en los cuales algunos actuadores controlan algunas variables operacionales (por ejemplo la posición de la plataforma) mientras que los actuadores restantes controlan el resto de las variables operacionales (por ejemplo la orientación). Algunos robots desacoplados han sido propuestos por Innocenti and Parenti-Castelli (1992), D. Bernier and Li (1995), P.Wenger and Chablat (2000), Zabalza et al. (2002), Y. Takeda and Sugimoto (2005), y Briota et al. (2009).

En la figura 6.11 se presenta el diseño de un robot de 6 grados de libertad con cinemática desacoplada. Esta arquitectura sería interesante para aplicarse en un robot tipo Remo 2, ya que durante la navegación se mantendrían activas únicamente las articulaciones responsables de la orientación. Por otro lado, en la etapa de intervención se haría uso de todos los grados de libertad, tanto los de orientación, como los de traslación. Este robot fue propuestos por primera vez en (Yime et al., 2011) siendo el autor de esta tesis, el coautor de esta estructura. En el apéndice se presenta un análisis detallado de la cinemática del robot.

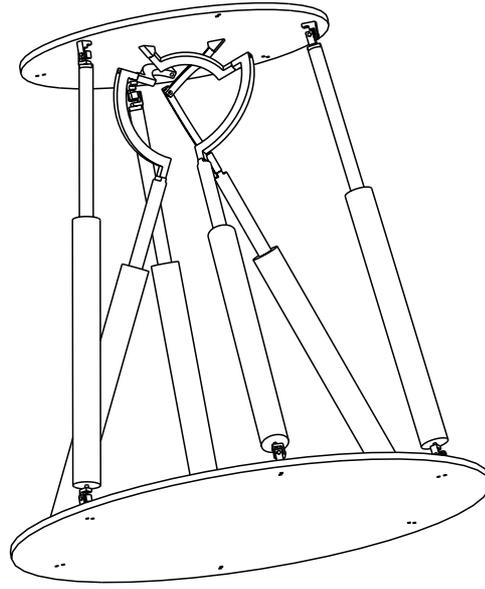


Figura 6.11: Robot de 6 grados de libertad con cinemática desacoplada.

En lo que resta de este documento, la investigación se centrará en la síntesis dimensional del robot Remo 2. Primero se describirán algunos índices de desempeño para usarse como criterios de diseño. Algunos de estos índices son propuestos por primera vez aquí. Posteriormente se realizará la formulación del problema de diseño y finalmente se discutirán los resultados obtenidos.

6.3.1. Índices de desempeño globales

Índice Global de Condicionamiento (GCI)

El Índice Global de Condicionamiento (Gosselin (1991)) es un índice independiente de la postura que permite evaluar globalmente el desempeño del manipulador en el espacio de trabajo. El GCI está definido de la siguiente manera:

$$GCI = \frac{\int_W \kappa^{-1} dW}{\int_W dW} \quad (6.13)$$

Si el GCI se acerca a cero, se dice que el espacio de trabajo del manipulador está mal condicionado. Si el GCI se acerca a 1 se dice que dicho espacio está bien condicionado.

Debido a la dificultad de obtener una solución exacta de la integral, una versión discreta de GCI puede ser expresada como:

$$GCI = \frac{\sum_{j=1}^{n_W} \kappa_j^{-1}}{n_W} \quad (6.14)$$

donde n_W es el número de nodos de una malla del espacio de trabajo discretizado; y el numerador es la suma de la inversa del número de condicionamiento de los distintos

nodos que componen dicha malla. El GCI ha sido utilizado frecuentemente en el diseño y evaluación del espacio de trabajo de robots paralelos (Saltaren et al. (2007b)). En (Merlet (2007)) se sugiere que la integración numérica de este índice se realice mediante un método de Montecarlo.

Índice de Condicionamiento Cinemático

El índice de condicionamiento cinemático (KCI, por sus siglas en inglés) fue definido por Angeles y Lopez. Este índice independiente de la postura fue definido de la siguiente manera

$$KCI = \frac{1}{\kappa_m} \times 100 \quad (6.15)$$

donde κ_m es el mínimo valor del número de condición dentro del espacio de trabajo. Los manipuladores con un KCI igual al 100 % son los llamados isotrópicos.

Índice Global de Nivel y Distribución

Cuando se desea optimizar el diseño de un manipulador, para que las configuraciones que este alcance, obtengan buenos valores de un índice dependiente de la postura, diversos autores han propuesto optimizar la integral de este índice sobre el espacio de trabajo (Khatib (1987); Gosselin (1991)). Para manipuladores sencillos, es posible solucionar analíticamente esa integral y determinar que parámetros del diseño optimizan su valor. No obstante, en la mayoría de los casos se tiene que utilizar una versión discreta de la integrar, la cual corresponde a su vez en el promedio del índice sobre el espacio de trabajo. Este es el caso del GCI, (Merlet (2007)).

El promedio naturalmente refleja el nivel de los valores del índice, no obstante no refleja la distribución de estos valores en las diferentes regiones del espacio de trabajo. Algunos autores han sugerido el uso de la desviación estándar para medir la uniformidad del valores que alcanza el índice en los diversos puntos o regiones (Pamanes (1991); Li (2005); Li (2010)).

En base a lo anterior, se podría definir de manera general un Índice Global de Nivel y Distribución de un índice de desempeño dependiente de la postura, μ . Este índice estaría dado de la siguiente manera:

$$G_W(\mu) = \alpha_1 n(\mu) \pm \alpha_2 d(\mu) \quad (6.16)$$

donde:

$$n(\mu) = \frac{1}{n_W} \sum_{j=1}^n \mu_j, \quad (6.17)$$

$$d(\mu) = \sqrt{\frac{1}{n_W} \sum_{j=1}^n (\mu_j - n)^2} \quad (6.18)$$

y α_1 y α_2 son factores de ponderación. Los términos $n(\mu)$ y $d(\mu)$ corresponden al promedio y la desviación estándar de los valores del índice μ en el espacio de trabajo del manipulador. El signo + se utiliza si lo que se desea es minimizar los valores del

índice μ , de esta manera se penaliza el valor de la media con el valor de la desviación estándar. Contrariamente, si lo que se desea es maximizar los valores de μ , entonces se utiliza el signo $-$.

El propósito de este índice global es determinar el nivel medio del valor de μ , y la mismo tiempo la distancia media que existe entre este valor medio y los valores del índice μ en las diversas configuraciones. De esta manera se puede determinar diseños en los cuales el nivel general de los valores del índice sea bueno. Por ejemplo, considere que los mapeos de μ , presentados en la siguiente figura, corresponden a 2 diseños diferentes de un manipulador (es decir, que los eslabones tiene diferentes dimensiones) y suponga que el promedio de ambos mapeos es idéntico.

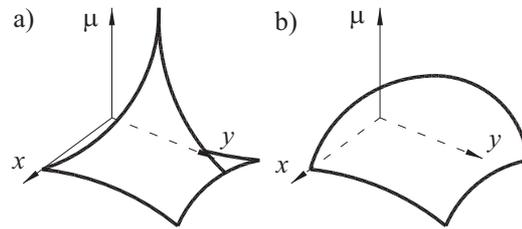


Figura 6.12: Mapeo del índice μ para dos diseños diferentes de un manipulador.

Si μ fuera la inversa del número de condición, el GCI para ambos diseños sería el mismo. Se puede observar que en la figura a) solo algunas configuraciones logran valores muy altos de μ , mientras que los alrededores alcanzan valores muy pobres. Por otro lado, en el caso de la figura b) se puede observar que el índice μ no alcanza valores tan altos como en el caso de la figura a), sin embargo la mayoría de las configuraciones alcanzan valores cercanos a la media. El índice $G_W(\mu)$ valoraría mejor este segundo caso. De esta forma el índice $G_W(\mu)$ no solo valora los niveles de μ en el espacio de trabajo, sino también la adecuada distribución de este valor en todas las configuraciones. Para el caso número de condición este índice fue definido recientemente por Li (2010).

Otro índice global se podría expresar mediante la división del promedio entre la desviación estándar: $G_W(\mu) = \frac{n(\mu)}{d(\mu)}$. Este índice fue definido por Li (2005) para el caso particular de la inversa del número de condición. Sin embargo, este índice puede ocasionar problemas cuando el valor de la desviación estándar es menor que 1. Por lo tanto, definimos aquí otro índice, llamado *relación de nivel y distribución* de un índice μ dependiente de la postura:

$$R(\mu) = \frac{\alpha_1 n(\mu)}{\alpha_2 d(\mu) + 1} \quad (6.19)$$

De esta manera, este índice sería directamente proporcional al promedio de μ , e inversamente proporcional a la desviación estándar. Si lo que se desea es minimizar los valores μ en el espacio de trabajo, entonces este índice se adecuaría mejor mediante el producto, $R(\mu) = \alpha_1 n(\mu)(\alpha_2 d(\mu) + 1)$.

Por otro lado, podríamos definir otro índice a partir de la desviación estándar,

seleccionando un valor medio deseado de un índice para el espacio de trabajo, y midiendo la lejanía de los valores (en cada configuración) a este valor mediante la desviación estándar. Minimizando la desviación estándar encontraríamos el diseño cuyas configuraciones en general estén más cerca del valor medio deseado. Finalmente, las expresiones $n(\mu)$ y $b(\mu)$, pudieran expresarse también en forma de integrales.

6.3.2. Optimización del diseño

En esta sección se realiza un estudio sobre el diseño de la estructura del robot Remo 2. Para ello se realiza una exploración del *espacio de los parámetros*. El espacio de los parámetros es un espacio de n dimensiones en el cual cada una de estas dimensiones corresponde a un parámetro de diseño. Por lo tanto un punto del espacio de los parámetros corresponde un diseño en particular.

Los propósitos de esta sección son dos: por un lado identificar los diseños óptimos de la estructura del Remo 2 para cada uno de los índices de desempeño presentados anteriormente, y por otro lado discutir las diferencias entre los resultados obtenidos empleando cada uno de estos criterios.

Antes de realizar la exploración del espacio de los parámetros, hay que hacer algunas consideraciones sobre el diseño del Remo 2. Para que la arquitectura del robot sea útil para los propósitos explicados en la introducción de este trabajo, una característica importante la estructura es que esta sea simétrica. Para ello, el diámetro de ambas plataformas debe ser igual. Por otro lado, la distribución de las posiciones de articulaciones esféricas y universales debe ser simétrica en ambas plataformas.

El número de parámetros de diseño se reduce tomando en cuenta las consideraciones anteriores. Por otro lado, se define como dimensión fija (o base) a longitud de los actuadores prismáticos. La longitud muerta de los cilindros es unitaria. Por otro lado, la carrera es el 75% de la longitud muerta. Por lo tanto $L_{min} = 1$ y $L_{max} = 0,75$. Finalmente, los parámetros de diseño son:

- El radio de ambas plataformas, r_P .
- El ángulo entre las articulaciones, λ .

En la figura 6.13 se presenta el volumen del espacio de trabajo para diversos valores de los parámetros de diseño. Se observa que el volumen del espacio de trabajo disminuye a medida que aumenta el diámetro de las plataformas. Por otro lado, el espacio de trabajo aumenta a medida que el ángulo entre las articulaciones aumenta, teniendo su máximo en $\lambda = 60^\circ$. A partir de valores cercanos a $\lambda = 50^\circ$ el espacio de trabajo tiene un crecimiento rápido.

En la figura 6.14 se presenta los valores máximos de la inversa del número de condición dentro del espacio de trabajo para diversos diseños del robot. Los mayores máximos se encuentran para ángulos menores de $\lambda = 60^\circ$. Por otro lado en la figura 6.15 se presenta los valores mínimos de la inversa del número de condición dentro del espacio de trabajo. A partir de esta grafica es posible identificar aquellos diseños sin configuraciones innacesibles.

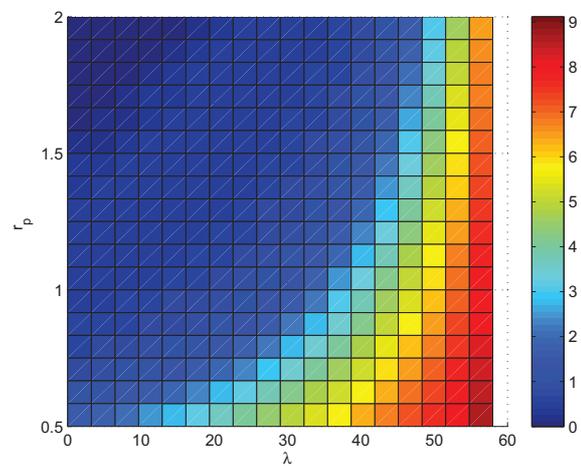


Figura 6.13: Volumen del espacio de trabajo para diversos valores de los parámetros de diseño.

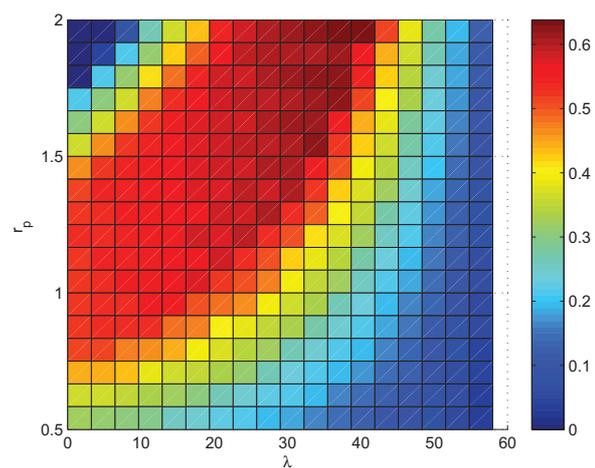


Figura 6.14: Máximo de κ^{-1} en el espacio de trabajo de diversos diseños.

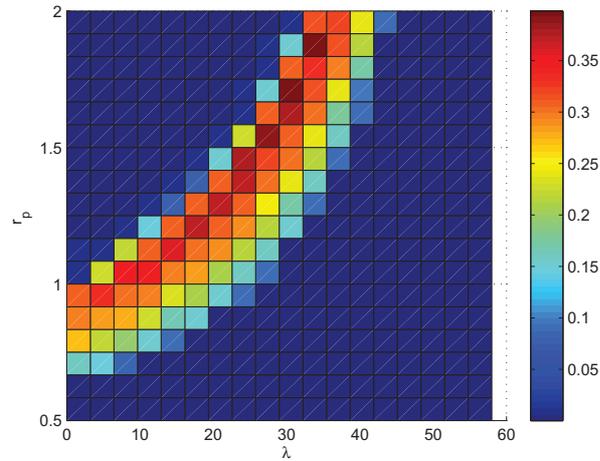


Figura 6.15: Mínimo de κ^{-1} en el espacio de trabajo de diversos diseños.

En la figura 6.16 se muestra el GCI para los diversos diseños. A partir de esta gráfica es posible identificar aquellos diseños con un mejor desempeño cinetostático. Al contrario del comportamiento del volumen del espacio de trabajo, el GCI disminuye al aumentar el ángulo entre las articulaciones. De esta gráfica es posible determinar el radio de la plataforma correspondiente a cada ángulo que permita obtener el mejor valor del GCI.

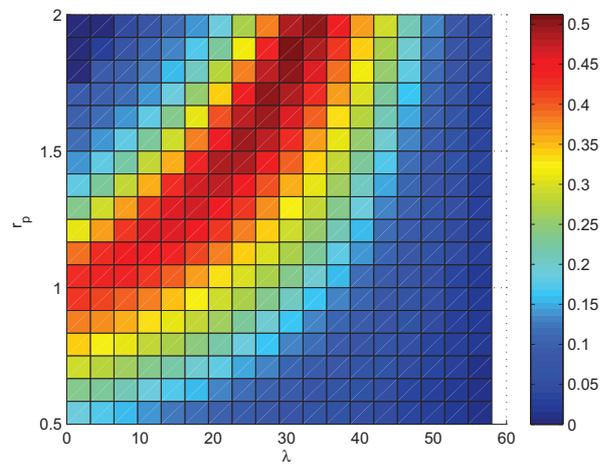


Figura 6.16: GCI para diversos diseños.

En el apartado anterior se presentó la definición del índice llamado relación de nivel y distribución. El propósito de este índice es identificar aquellos diseños que no solo tengan un buen valor de κ^{-1} , sino que además este homogenizado su valor. En la definición de este índice está incluido el valor de la desviación estándar de κ^{-1} , en la

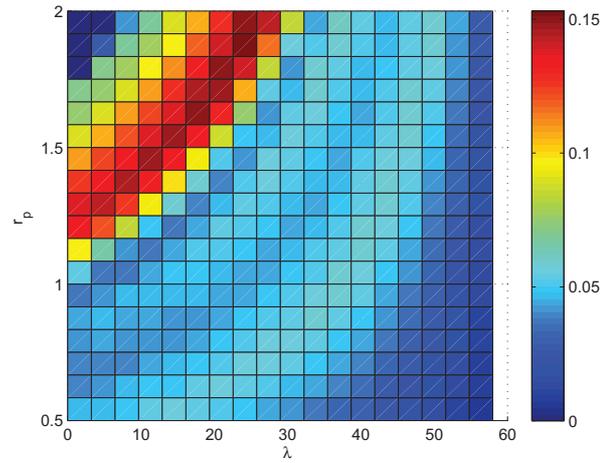


Figura 6.17: Valor de $d(\kappa^{-1})$ para diversos diseños.

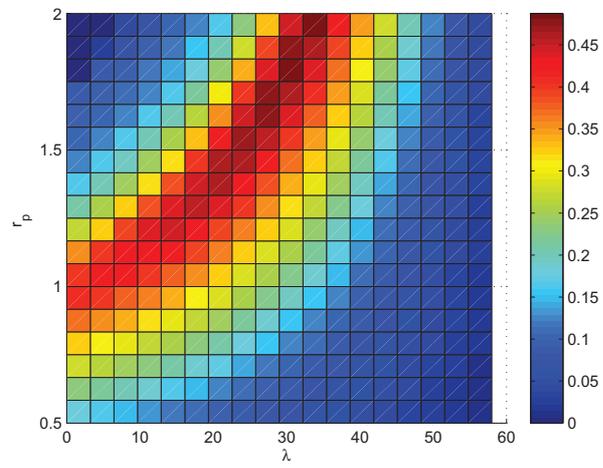
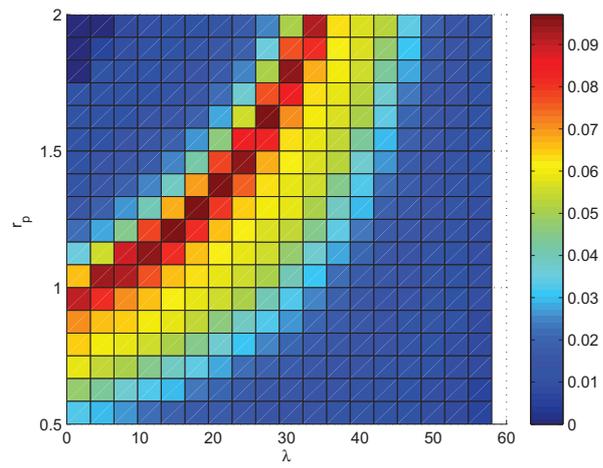
figura 6.17 se presentan los valores de este término para diversos diseños. Se observa que aquellos diseños con una desviación estándar grande (mala distribución de los valores de κ^{-1}) se encuentran en una región cercana, pero no coincidente, a los que tienen un alto GCI. Fuera de esta región los valores de la desviación estándar son relativamente bajos.

En la figura 6.18 se presenta el valor de la relación de nivel y distribución de κ^{-1} . Se observa que esta gráfica es muy parecida a la del GCI. Esto se debe a que para este robot, los diseños que tienen un buen promedio de κ^{-1} también poseen buenos valores de su desviación estándar. Por tal motivo, las ventajas de este índice no son evidentes en este caso. En la figura 6.19 se presentan los valores de $R(\kappa^{-1})$ para un valor del coeficiente de ponderación de $\alpha_2 = 100$. Se observa que ahora la región con valores más altos se aleja de aquella con los valores más altos de desviación estándar.

Finalmente en la figura 6.20 se presenta la suma del número de condición de los espacios de trabajo de los diversos diseños. Se observa que la región con valores más altos de este índice es la que se encuentra entre la que provee los diseños con mejor GCI y la que tiene un espacio de trabajo más extenso.

Tabla 6.1: Parámetros óptimos

Indice	Parametros		Valor	W_t	n
	r_P	λ			
KCI	1.91	38.66	0.6382	0.7800	0.3652
GCI	2	32.22	0.5120	0.4850	0.5120
$R(\kappa^{-1})$	1.25	16.11	0.0971	0.5170	0.4704
$S(\kappa^{-1})$	1.75	48.33	4.0069×10^2	3.53	0.1133

Figura 6.18: Valor de $R(\kappa^{-1})$ para diversos diseños.Figura 6.19: Valor de $R(\kappa^{-1})$ para diversos diseños con $\alpha_2 = 100$.

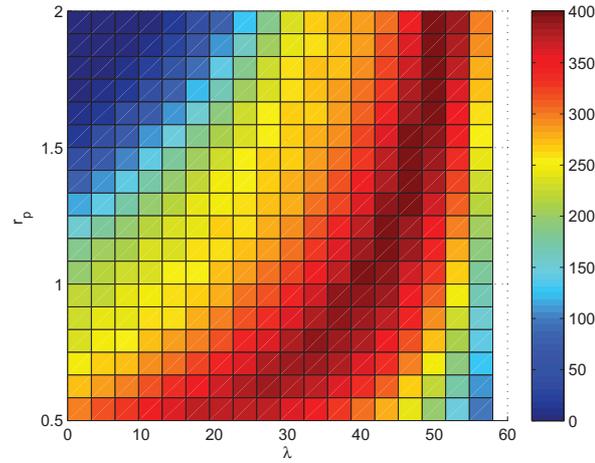


Figura 6.20: Sumatoria de κ^{-1} en el espacio de trabajo de diversos diseños.

La tabla 6.1 resume los parámetros óptimos desde el punto de vista de cada índice de desempeño. En esta tabla se presenta además, el volumen del espacio de trabajo y el valor promedio de la inversa del número de condición para cada diseño. Se observa que el índice S provee el diseño con el mayor espacio de trabajo, no obstante el promedio del número de condición es bajo. Por otro lado, como es claro, el diseño con mejor GCI tiene el mejor promedio, sin embargo el espacio de trabajo es el menor de todos casos. El índice R tiene el segundo lugar con respecto al promedio de la inversa del número de condición y el tercer lugar con respecto al volumen del espacio de trabajo. Para el KCI, la situación es la contraria.

6.4. Resumen

En este capítulo se realizó una serie de análisis con el fin de determinar la capacidad del robot Remo 2 para ejecutar diferentes reconfiguraciones. Este estudio expuso las acotaciones de movimiento que la estructura del robot posee debido principalmente a su geometría.

Además, se identificaron las configuraciones o regiones del espacio de trabajo en las que la estructura del robot tiene una mejor capacidad para realizar su movimiento mediante el uso de índices de desempeño cinetostáticos. Estos índices permiten evaluar la capacidad de la estructura del robot para transformar las velocidades o fuerzas en los actuadores lineales, en velocidades o fuerzas relativas entre ambas plataformas.

Se realizó un análisis del espacio de trabajo de traslación del robot para cuando los marcos de referencia de ambas plataformas son paralelas. Se observó que la forma del espacio de trabajo es una especie de casco, cuyo eje de simetría es el eje x del marco de referencia de la plataforma fija. El espacio de trabajo de traslación en este caso no posee singularidades, excepto cuando las plataformas están contenidas en el mismo plano, situación que no es práctica.

Por otro lado se analizó el espacio de trabajo de orientación cuando el centro de la plataforma móvil está situado de tal manera que sus coordenadas x e y son cero. Se estudió el volumen del espacio de trabajo a diversas alturas del eje z . Se consideró interesante determinar el mayor espacio de trabajo dado que en estas configuraciones la plataforma móvil no realiza ninguna traslación durante un cambio de orientación. Lo anterior puede permitir que dada una orientación deseada se pueda llegar a ella evitando movimientos innecesarios. Se presentó además una parametrización de la orientación de la plataforma móvil empleando los ángulos de azimut y elevación.

Se realizó un estudio sobre el diseño de la estructura del robot Remo 2. Para ello se realizó una exploración del espacio de los parámetros. Los propósitos de este estudio fueron dos: por un lado identificar los diseños óptimos de la estructura del Remo 2 para cada uno de los índices de desempeño globales presentados, y por otro lado discutir las diferencias entre los resultados obtenidos empleando cada uno de estos criterios.

Se definieron una serie de índices de desempeño como el llamado relación de nivel y distribución. El propósito de este índice global es determinar el nivel medio del valor de κ^{-1} , y al mismo tiempo la distancia media que existe entre este valor medio y los valores del índice en las diversas configuraciones. De esta manera se puede determinar diseños en los cuales el nivel general de los valores del índice sea bueno.

Capítulo 7

Análisis y Simulación de Robots Submarinos con Múltiples Impulsores Vectorizados

7.1. Introducción

En este capítulo se presentan un par de trabajos complementarios a los objetivos planteados en esta tesis. En primer lugar se presentan una serie de análisis sobre robots submarinos con múltiples impulsores vectorizados. A pesar de existir antecedentes de robots con más de un impulsor vectorizado, esta estrategia de propulsión no ha sido explorada extensamente. Consideramos que un sistema de propulsión de este tipo es ventajoso en algunos aspectos. Por otro lado, se presenta el diseño de un simulador para robots submarinos. Este simulador se desarrolló en un lenguaje orientado a objetos, lo cual hace que esta herramienta posea varias prestaciones interesantes. Ambos trabajos se relacionan al final de este capítulo al presentarse varias simulaciones de un robot submarino con impulsores vectorizados empleando la librería desarrollada.

7.2. Múltiples impulsores vectorizados

Una idea interesante que surgió durante el desarrollo de este trabajo fue la de un sistema de propulsión basado en múltiples impulsores vectorizados de 2 g.d.l. Como se ha tratado de hacer ver en esta tesis, un impulsor vectorizado trataría de sustituir el efecto que puede generar tres impulsores de hélice cuyos ejes sean ortogonales, por uno solo impulsor de hélice con un dos actuadores para controlar su orientación.

Existen un par de antecedentes sobre robots submarinos de este tipo. En (Walker, 2005) se presentó un estudio sobre el diseño de un sistema de propulsión de un robot submarino con dos impulsores vectorizados para un robot con estabilidad en alabeo y cabeceo. Por otro lado, en (Lin and Guo, 2012) se presenta el diseño de un mini-AUV de cabina esférica. El sistema de propulsión consiste de tres impulsores vectorizados de dos grados de libertad, ubicados en un mismo plano y en los vértices de un triángulo equilátero. Los impulsores son de inyección de agua. Estos impulsores previenen el daño

a animales marinos que otros impulsores con partes móviles expuestas puede provocar. En ese trabajo se presentan diferentes movimientos que el prototipo realiza en los tres ejes, y rotaciones alrededor de eje z .

En ambos trabajos se emplearon impulsores de 2 g.d.l. Un impulsor vectorizado de este tipo puede una fuerza en tres dimensiones mediante un mecanismo que permita controlar la orientación del impulsor de hélice en dos ejes. Este mecanismo puede ser serial o paralelo. Para evitar entrar en detalles de la implementación del impulsor vectorizado, es suficiente decir que este se puede implementar mediante cualquier mecanismo paralelo de 2 g.d.l. , como es el caso del *Agile Eye*, presentado en la fig. 7.1.

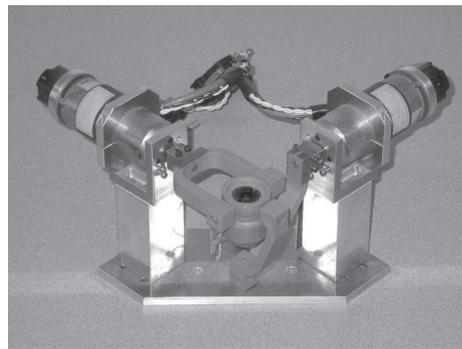


Figura 7.1: Mecanismo paralelo Agile Eye.

En esta sección se realiza el análisis de robots submarinos con 2, 3 y 4 impulsores vectorizados. El propósito es explorar las capacidades que tendría un robot con un sistema de propulsión de este tipo, y esclarecer las diferencias entre sistemas de propulsión consistentes de un diferente número de impulsores vectorizados.

Dos impulsores

En la figura 7.2 se presenta un esquema de un robot de cabina esférica con un par de impulsores vectorizados a los lados (en este caso el frente del robot es aquel hacia donde esta dirigido el eje x de marco de referencia del cuerpo). Para que la configuración de los impulsores sea simétrica, los impulsores están ubicados de tal manera que la línea que une el punto central del mecanismo de orientación de cada uno pasa por el centro de masa del robot.

Para realizar el análisis de las fuerzas y momentos que este sistema de propulsión puede ejercer sobre el robot considérese el esquema de la fig. 7.3. Para especificar la orientación del vector de fuerza \mathbf{f}_i que genera el impulsor i se emplean los ángulos α_i y β_i . El ángulo α_i es el ángulo entre el eje y y la proyecto del vector de fuerza en el plano $x - y$. El ángulo β_i es el angulo entre el vector de fuerza y el plano $x - y$

Si denotamos por t_i la magnitud de fuerza que el impulsor genera, el vector de

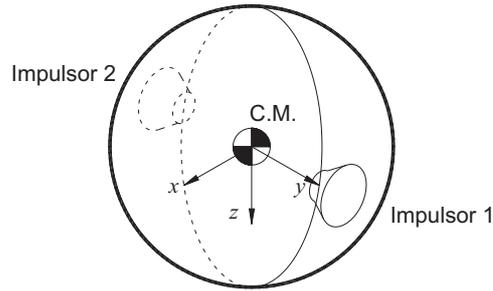


Figura 7.2: Esquema de robot con dos impulsores vectorizados.

fuerza se puede escribir de la siguiente manera:

$$\mathbf{f}_{ti} = \begin{bmatrix} t_i \cos \beta_i \sin \alpha_i \\ t_i \cos \beta_i \cos \alpha_i \\ t_i \sin \beta_i \end{bmatrix} \quad (7.1)$$

Por otro lado, el momento que este genera con respecto al centro de masa es el siguiente:

$$\mathbf{n}_{ti} = \mathbf{r}_{ti} \times \mathbf{f}_{ti} \quad (7.2)$$

Para robot con n impulsores vectoriales, el vector total de fuerzas por el sistema de propulsión ($\boldsymbol{\tau}$) se puede escribir de la siguiente manera:

$$\boldsymbol{\tau} = \mathbf{B}_v \mathbf{u}_v \quad (7.3)$$

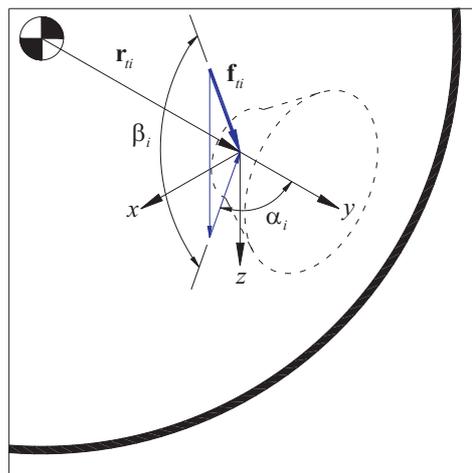


Figura 7.3: Notación del impulsor vectorizado.

donde $\mathbf{u}_v = [t_1 \ t_2 \ \dots \ t_n]^T$ es el vector de fuerzas lineales que generan los impulsores y por otro lado

$$\mathbf{B}_v = \begin{bmatrix} \mathbf{s}_{t1} & \mathbf{s}_{t1} & \dots & \mathbf{s}_{tn} \\ \mathbf{r}_{t1} \times \mathbf{s}_{t1} & \mathbf{r}_{t2} \times \mathbf{s}_{t2} & \dots & \mathbf{r}_{tn} \times \mathbf{s}_{tn} \end{bmatrix}$$

siendo \mathbf{s}_{ti} un vector unitario con la dirección de la fuerza \mathbf{f}_{ti} .

Si se expande la ecuación fuerzas para un robot con dos impulsores con la configuración presentada en la fig. 7.2 (siendo $\mathbf{r}_{t1} = [0 \ y_t \ 0]^T$ y $\mathbf{r}_{t2} = [0 \ -y_t \ 0]^T$), se puede llegar a una primera conclusión,

$$\boldsymbol{\tau} = \begin{bmatrix} t_1 \cos \beta_1 \sin \alpha_1 + t_2 \cos \beta_2 \sin \alpha_2 \\ t_1 \cos \beta_1 \cos \alpha_1 + t_2 \cos \beta_2 \cos \alpha_2 \\ t_1 \sin \beta_1 + t_2 \sin \beta_2 \\ y_t(t_1 \sin \beta_1 - t_2 \sin \beta_2) \\ 0 \\ -y_t(t_1 \cos \beta_1 \sin \alpha_1 - t_2 \cos \beta_2 \sin \alpha_2) \end{bmatrix} \quad (7.4)$$

Como es posible ver en esta ecuación, este sistema no puede generar un momento sobre el eje y , es decir que el robot sería incapaz de controlar el cabeceo. En tal caso es necesario un elemento adicional que genere un momento en este eje, como puede ser que el centro de flotabilidad no coincida con el centro de masa (), o un actuador activo como lo es un giróscopo.

En el caso de la línea entre ambos impulsores no cruce el centro de masa, el movimiento de traslación pura en el eje z (viraje) no podrá realizarse sin que ocurra un cambio de orientación en el vehículo.

Como es posible apreciar en la ecuación 7.4, si $t_1 = t_2$, y además $\alpha_1 = \alpha_2$ y $\beta_1 = \beta_2$, el sistema de propulsión genera únicamente una fuerza lineal y no se genera ningún momento. Si el robot tiene una geometría simétrica, como lo puede ser una esfera, esto permitirá que el robot pueda realizar traslaciones puras.

Por otro lado si $\alpha_2 = \alpha_1 + \pi$ con $\beta_1 = \beta_2 = 0$ el sistema genera un momento puro alrededor del eje z . De manera similar si $\beta_2 = \beta_1 + \pi$ con $\alpha_1 = \alpha_2 = 0$ genera un momento puro alrededor del eje x .

Sería interesante que un robot con este sistema de propulsión pudiera realizar movimientos en un plano controlando los tres grados de libertad (posiciones en dos ejes ortogonales y la orientación con respecto a un eje). Para esto, considerando un robot de geometría simétrica y especificados los movimientos deseados el problema consiste en resolver la dinámica inversa. Para el caso del esquema de la fig. 7.2 y considerando el plano de movimiento como el $x - y$, las fuerzas y el momento a generar serían X , Y y N . Estos están relacionados con las fuerzas en los actuadores de la siguiente manera:

$$\begin{bmatrix} X \\ Y \\ N \end{bmatrix} = \begin{bmatrix} t_1 \cos \beta_1 \sin \alpha_1 + t_2 \cos \beta_2 \sin \alpha_2 \\ t_1 \cos \beta_1 \cos \alpha_1 + t_2 \cos \beta_2 \cos \alpha_2 \\ -y_t(t_1 \cos \beta_1 \sin \alpha_1 - t_2 \cos \beta_2 \sin \alpha_2) \end{bmatrix} \quad (7.5)$$

Se tiene un número mayor de incógnitas que de ecuaciones, una forma de resolver el sistema es poner tres incógnitas en función de una. Por ejemplo si se toma el ángulo α_1 , se puede despejar t_1 empleando el primer y tercer renglón de 7.5:

$$t_1 = \frac{Xy_t - N}{2y_t \sin \alpha_1} \quad (7.6)$$

Utilizando el primer y segundo renglón de 7.5 se puede obtener t_2 a través de identidades trigonométricas:

$$t_2 = \sqrt{\left(\frac{X}{2} + \frac{N}{y_t}\right)^2 + \left(Y - \frac{(Xy_t - N) \cos \alpha_1}{2y_t \sin \alpha_1}\right)^2} \quad (7.7)$$

Por su parte α_2 se obtiene empleando cualquiera de las ecuaciones y sustituyendo t_1 y t_2 .

Al tener un grado de redundancia este sistema se podría emplear un algoritmo de optimización y definir como función objetivo el consumo de energía.

Tres impulsores

Un robot con tres impulsores vectorizados (como el esquematizado en la fig. 7.4) es obviamente un sistema redundante.

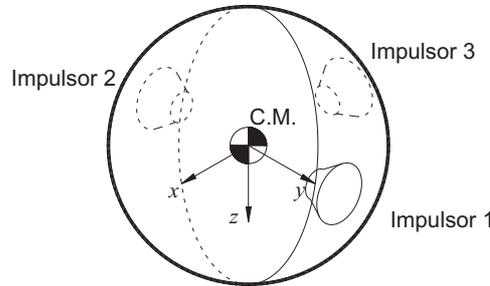


Figura 7.4: Esquema de robot con tres impulsores vectorizados.

Considerando la posición de un tercer impulsor se encuentre sobre el eje x , es decir $\mathbf{r}_{t1} = [x_t \ 0 \ 0]^T$, se puede escribir el vector de fuerzas de la siguiente manera:

$$\boldsymbol{\tau} = \begin{bmatrix} t_1 \cos \beta_1 \sin \alpha_1 + t_2 \cos \beta_2 \sin \alpha_2 + t_3 \cos \beta_3 \sin \alpha_3 \\ t_1 \cos \beta_1 \cos \alpha_1 + t_2 \cos \beta_2 \cos \alpha_2 + t_3 \cos \beta_3 \cos \alpha_3 \\ t_1 \sin \beta_1 + t_2 \sin \beta_2 + t_3 \sin \beta_3 \\ y_t(t_1 \sin \beta_1 - t_2 \sin \beta_2) \\ x_t t_3 \sin \beta_3 \\ -y_t(t_1 \cos \beta_1 \sin \alpha_1 - t_2 \cos \beta_2 \sin \alpha_2) - x_t t_3 \cos \beta_3 \cos \alpha_3 \end{bmatrix} \quad (7.8)$$

Es posible apreciar que el cabeceo es controlable ahora. El momento en el eje y viene dado por la expresión $x_t t_3 \sin \beta_3$. El cabeceo estará en función de t_3 y el ángulo β_3 .

Para un robot con esta configuración de impulsores no es posible realizar traslaciones puras, distribuyendo la fuerza necesaria para esto en cada uno de los actuadores, i.e. si se define la misma orientación con $t_1 = t_2 = t_3$ no existe un balanceo en los momentos generados por los impulsores.

Considérese un robot en el que los centros de los actuadores vectorizados se encuentren en los vértices de un triángulo equilátero, cuyo centro es el centro de masa del robot. Siendo esto así, los vectores de posición de los impulsores 2 y 3 están dados de la siguiente manera:

$$\mathbf{r}_{t2} = \begin{bmatrix} \sqrt{\frac{3}{4}}c_t \\ -\frac{1}{2}c_t \\ 0 \end{bmatrix} \text{ y } \mathbf{r}_{t3} = \begin{bmatrix} -\sqrt{\frac{3}{4}}c_t \\ -\frac{1}{2}c_t \\ 0 \end{bmatrix}$$

el parámetro c_t es el radio del círculo que circunscribe al triángulo.

Para este robot se tiene que los momentos que puede generar el sistema de impulsión están dados de la siguiente manera:

$$\boldsymbol{\tau} = \begin{bmatrix} c_t t_1 \sin \beta_1 - \frac{1}{2}c_t(t_3 \sin \beta_3 + t_2 \sin \beta_2) \\ \sqrt{\frac{3}{4}}c_t(t_3 \sin \beta_3 - t_2 \sin \beta_2) \\ -c_t t_1 \cos \beta_1 \sin \alpha_1 + \frac{1}{2}c_t(t_2 \cos \beta_2 \sin \alpha_2 + t_3 \cos \beta_3 \sin \alpha_3) + \sqrt{\frac{3}{4}}c_t(t_2 \cos \beta_2 \cos \alpha_2 - t_3 \cos \beta_3 \cos \alpha_3) \end{bmatrix} \quad (7.9)$$

De esta expresión se puede observar que cuando todos los impulsores tienen la misma dirección y generan la misma fuerza de impulsión los momentos son nulos. Esta es una característica deseable en algunas ocasiones, ya que en un movimiento de traslación la fuerza se distribuye uniformemente en los tres actuadores o de visto desde otro punto de vista se puede utilizar la máxima potencia de los impulsores para generar un movimiento en determinado sentido. Como se puede observar la distribución de los impulsores tiene efectos en el desempeño del sistema de propulsión.

Cuatro impulsores

Un robot con 4 impulsores vectorizados es un sistema de que propulsión redundante para generar fuerzas y momentos controlables sobre el robot. Encontramos que 5 variables son redundantes. Considerando un robot con una distribución simétrica de la localización de los impulsores puede generar el siguiente vector de fuerzas:

$$\boldsymbol{\tau} = \begin{bmatrix} t_1 \cos \beta_1 \sin \alpha_1 + t_2 \cos \beta_2 \sin \alpha_2 + t_3 \cos \beta_3 \sin \alpha_3 + t_4 \cos \beta_4 \sin \alpha_4 \\ t_1 \cos \beta_1 \cos \alpha_1 + t_2 \cos \beta_2 \cos \alpha_2 + t_3 \cos \beta_3 \cos \alpha_3 \\ t_1 \sin \beta_1 + t_2 \sin \beta_2 + t_3 \sin \beta_3 \\ y_t(t_1 \sin \beta_1 - t_2 \sin \beta_2) \\ x_t(-t_4 \sin \beta_4 + t_3 \sin \beta_3) \\ -y_t(t_1 \cos \beta_1 \sin \alpha_1 - t_2 \cos \beta_2 \sin \alpha_2) + x_t(t_4 \cos \beta_4 \cos \alpha_4 - t_3 \cos \beta_3 \cos \alpha_3) \end{bmatrix} \quad (7.10)$$

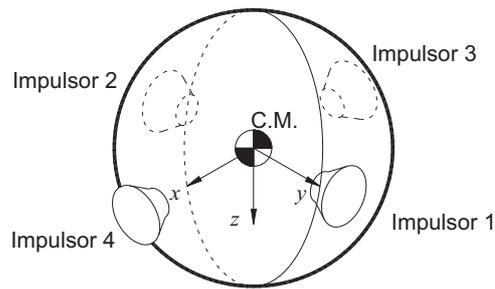


Figura 7.5: Esquema de robot con cuatro impulsores vectorizados.

Este sistema de propulsión puede distribuir las fuerzas proporcionalmente en cada impulsor. Por otro lado, los momentos para generar el alabeo y el cabeceo dependen por un lado de t_1 y t_2 , y por otro de t_3 y t_4 , respectivamente.

7.3. Simulación

7.3.1. Simulador de robots submarinos

Como se indicó en la introducción uno de los objetivos de esta tesis es el desarrollo de herramientas de simulación para robots submarinos. Como parte del seguimiento de este objetivo se realizó el diseño e implementación de un simulador dinámico para robots submarinos. Este simulador consiste de una librería de software desarrollada en el lenguaje de programación orientado a objetos C++.

El interés en el desarrollo de esta librería de simulación, es desarrollar una herramienta que permita una fácil definición de la arquitectura del robot, y que permita visualizar su desempeño. Como se ha ejemplificado anteriormente, la simulación es una herramienta que permite obtener información de manera económica sobre el funcionamiento de un robot en la realización de una maniobra o una misión.

Los trabajos en simulación de robots submarinos se han presentado desde finales de la década de 1980. Uno de los primeros trabajos en simulación de robots submarinos se presentó en (Hillson and Jones, 1989). El software, en el que se recreaba un ambiente virtual, fue utilizado para probar una arquitectura de control inteligente para robots submarinos. En la Escuela Naval de Posgraduados de la Marina de Estados Unidos (NPS, por sus siglas en inglés) se desarrolló un simulador que permitía integrar Hardware en el bucle de simulación. Este simulador fue utilizado como cama de pruebas para el desarrollo y evaluación de controladores no-lineales (Zyda et al., 1990). En la Universidad Heriott-Watt se desarrolló un sistema de simulación distribuido para robots submarinos. Este sistema, llamado Core Simulation System, poseía una interface de usuario para entrenar a operadores, determinar la factibilidad de una tarea y reproducir una misión (Lane et al., 1998). Otro desarrollo interesante fue el proyecto SAMON (Phoha et al., 2001). El objetivo de SAMON (Ocean-Sampling Mobile Network) es soportar la integración de simuladores diversos de AUVs mediante internet, con el

objetivo de formular y evaluar estrategias de control inteligente y soportar el desarrollo de un lenguaje de control forma compartido entre los AUVs.

La librería de simulación desarrollada durante este trabajo tiene como principal característica que está desarrollada en un lenguaje orientado a objetos. A través de diferentes clases se modela la arquitectura física del robot.

Por otro lado, esta librería incluye las funciones de determinación de distancias entre objetos, la detección de colisiones y el cálculo de fuerzas de contacto entre cuerpos. Estas funciones pueden ser útiles en ambientes donde múltiples robots submarinos colaboren en la realización de una tarea y se encuentren próximos entre sí.

Los ficheros que componen la librería del simulador se han separado en diferentes paquetes dependiendo de su funcionalidad. Una colección de paquetes es la que integra librería encargada de realizar la simulación dinámica de cuerpos rígidos. Se desarrollo esta librería de manera independiente con el objetivo de que fuera útil para otros propósitos. Esta librería se puede emplear para simular tanto cuerpos que se encuentran en el aire como los que encuentren en un ambiente submarino. En la fig. 7.6 se presenta el diagram de clases de esta librería, llamada MinPBS (acrónimo de Minimal Library for Penalty Based Simulation).

La librería MinPBS está compuesta de tres paquetes principalmente, el de detección de colisiones, el de fuerzas de contacto y el de simulación dinámica. En la figura se presentan las clases que componen la librería y las relaciones entre estas. También se muestran las diferentes estructuras de datos que permiten que se pueda trabajar con tantos cuerpos como sea necesario. En el primer apéndice de este documento se describe en detalle cada una de las clases (atributos y métodos) de esta librería.

En la figura 7.8, se presenta el diagrama de clases de la librería de simulación de robots submarinos. Existen 4 tipos de clases principales: `uRobot`, `uRigidBody`, `Actuator` y `Sensor`. Estas se encargan de modelar la dinámica del cuerpo rígido en el ambiente submarino, los actuadores y los sensores.

La clase `uRigidBody` se deriva de la clase `RigidBody`. Esta clase se encarga de modelar la dinámica de un cuerpo rígido en un ambiente submarino. Además de los atributos y métodos de la clase `RigidBody` (esta se explicará en detalle en el primer apéndice), esta incluye los parámetros correspondientes a la masa añadida, los coeficientes de arrastre, la flotabilidad y la posición del centro de flotación. La clase `uRigidBody` tiene su propio método para realizar la integración numérica de las ecuaciones de movimiento, que son diferentes a las de un cuerpo rígido en el aire, principalmente por las fuerzas hidrodinámicas e hidrostáticas a las que está sometido dentro del agua.

La clase `Actuator` encapsula los atributos y métodos necesarios para modelar un actuador. De esta clase se derivan las clases correspondientes a los distintos tipos de actuadores para robots submarinos, como lo pueden ser impulsores de hélice (y los distintos modelos matemáticos que existen para modelarlos), sistema de lastre, aletas, y cualquier otro tipo de actuador existente como los que se describieron en la primera parte de esta tesis.

La clase `ActList` es una clase que permite crear y gestionar una lista de actuadores.

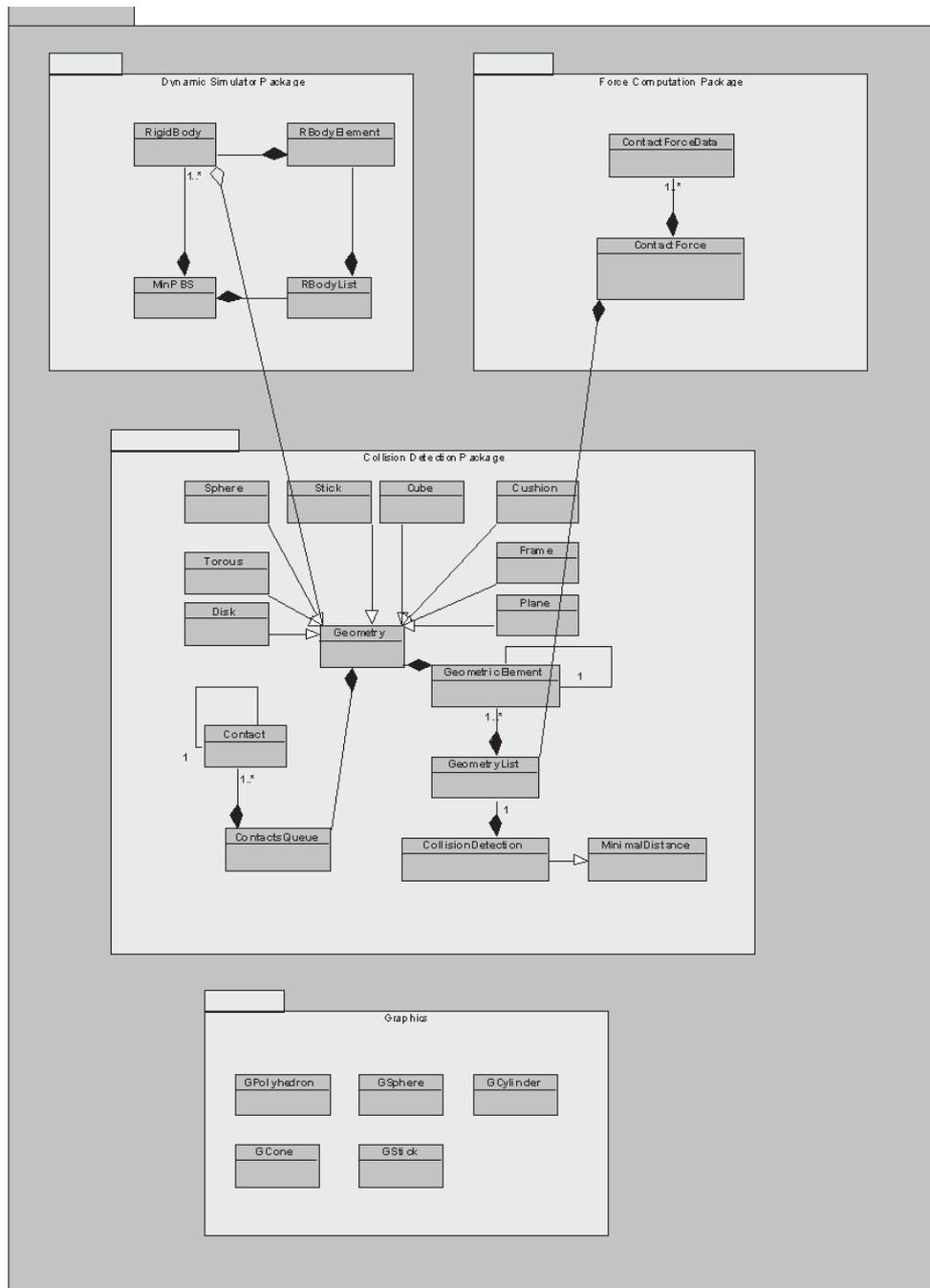


Figura 7.6: Diagrama de clases del simulador MinPBS.

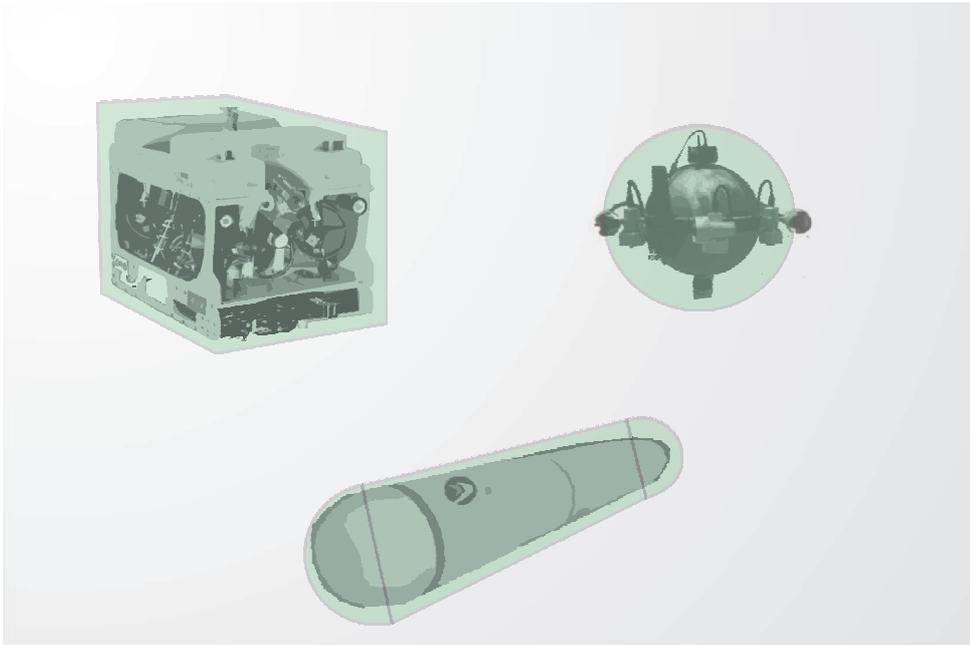


Figura 7.7: Primitivas geométricas del simulador empleadas para representar la geometrías de robots submarinos, (*cube*, *sphere* y *stick*).

Esta clase se utiliza principalmente para calcular de manera automática la fuerza y momento total que generan los actuadores del robot. Esta lista permite acceder a los métodos de las diferentes instancias de actuadores que hayan sido creadas para constituir el sistema de propulsión de un robot. De esta manera, en el caso de robots con impulsores vectoriales, se puede acceder a cierto actuador para modificar su posición y orientación.

Por otro lado, la clase **Sensor** encapsula los atributos y métodos necesarios para modelar un sensor propioceptivo. De esta clase se derivan las clases correspondientes a los distintos tipos de sensores propioceptivos, los cuales aportan información sobre la posición y velocidad del robot, como lo son los GPS, los sistemas de posicionamiento acústico, sensores de profundidad, etc. De igual manera que en la clase **Actuador**, la clase **Sensor** tiene una clase para gestionar los diferentes sensores que el robot pudiera tener, esta clase es llamada **SenList**.

La clase **uRobot** se encarga principalmente de contener los elementos que integran al robot y gestionarlos. De esta se derivara la clase que implementa el método en el que está definido el algoritmo de control. La clase **uRobot** tiene tres atributos principales que son apuntadores a las listas de actuadores y sensores, y al objeto que modela la dinámica del vehículo.

La simulación dinámica de un robot submarino consiste en determinar el movimiento del robot dadas las fuerzas en los impulsores, por lo tanto, el vector de aceleraciones se despeja de la ecuación (3.20):

$$\dot{\nu} = \mathbf{M}^{-1}[\boldsymbol{\tau} + \mathbf{g}_o - (\mathbf{C}(\nu)\nu + \mathbf{D}(\nu)\nu + \mathbf{g}(\eta))] \quad (7.11)$$

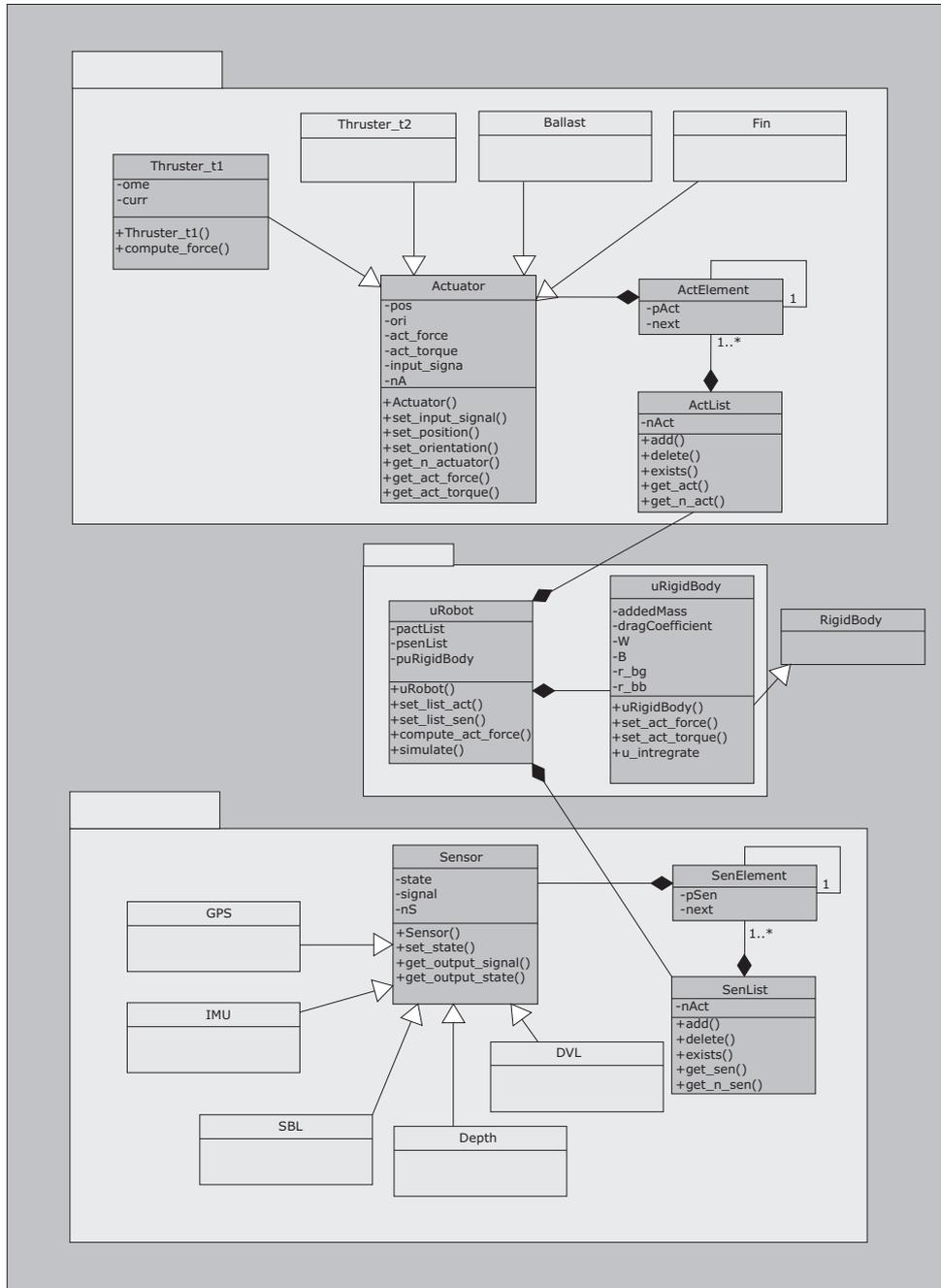


Figura 7.8: Diagrama de clases del simulador MinPBS.

Tabla 7.1: Parámetros del robot simulado

Parámetro	Valor
m	5.5752
$I_x = I_y = I_z$	0.0269
$X_{\dot{u}} = Y_{\dot{v}} = Z_{\dot{w}}$	2.7876
$X_{ u u} = Y_{ v v} = Z_{ w w}$	19.0066

Esta ecuación debe ser integrada para poder determinar la posición y velocidad del robot en diferentes puntos del tiempo. Para esto, se emplea el método de integración Runge-Kutta de cuarto orden.

El método de integración Runge-Kutta consiste en que dada una ecuación :

$$\dot{\mathbf{x}}_n = \mathbf{f}(\mathbf{x}_n, \mathbf{u}_n) \quad (7.12)$$

se obtenga el promedio de las pendientes en cuatro puntos dentro del paso de integración h . De esta manera se calculan las siguientes variables:

$$\begin{aligned} \mathbf{k}_1 &= h\mathbf{f}(\mathbf{x}(k), t_k) \\ \mathbf{k}_2 &= h\mathbf{f}(\mathbf{x}(k) + \frac{h}{2}\mathbf{k}_1, t_k + h/2) \\ \mathbf{k}_3 &= h\mathbf{f}(\mathbf{x}(k) + \mathbf{k}_2/2, t_k + h/2) \\ \mathbf{k}_4 &= h\mathbf{f}(\mathbf{x}(k) + \mathbf{k}_3/2, t_k + h) \end{aligned}$$

Utilizando estas variables, se puede calcular el valor de la variable después del paso de integración:

$$\mathbf{x}_{n+1} = \mathbf{x}(k) + \frac{1}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4) \quad (7.13)$$

A continuación se presentara una serie de simulaciones utilizando la librería desarrollada.

7.3.2. Simulación del Robot

Empleando la librería previamente descrita se simulara el movimiento de un robot con múltiples impulsores vectorizados. Se presentarán los movimientos del robot ante diferentes condiciones de los impulsores.

El robot considerado tiene una geometría esférica y tiene como radio, $r_s = 110$ mm. Los parámetros inerciales e hidrodinámicos se enlistan en la tabla ???. Se considera que el robot posee flotabilidad neutra y que el centro de flotabilidad coincide con el de gravedad.

Impulsores hacia adelante

En la primera simulación ambos impulsores están alineados con el eje x del marco de referencia del robot. De esta forma se tiene que $\alpha_1 = \alpha_2 = -\pi/2$ y $\beta_1 = \beta_2 = 0$. La fuerza de impulsión es constante e igual a 20 N en ambos impulsores.

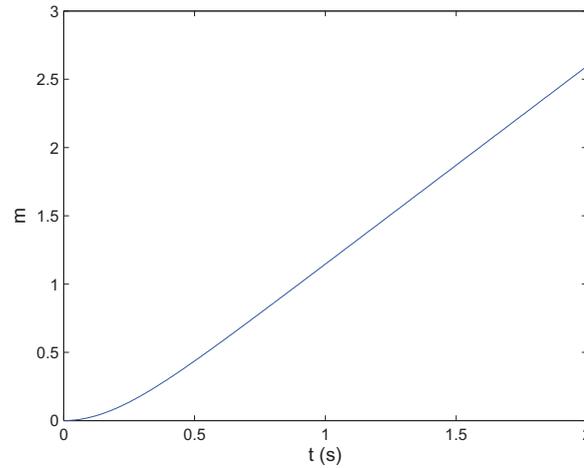


Figura 7.9: Posición en el eje x , $t_1 = t_2 = 20N$.

En la figura 7.9 se presenta el movimiento del robot en el eje de x . De la simulación fue posible comprobar que el robot no realiza movimientos en y y z , ni tampoco rotaciones en ningún eje.

En la figura 7.10 se presenta la velocidad lineal de robot. Se puede apreciar que la velocidad del robot se incrementa hasta los 1.25 s donde alcanza una velocidad cercana a los 1.5 m/s y después permanece constante. Por otro lado, en la figura 7.11 se presenta la aceleración del robot. Alrededor de los 1.2 s la aceleración es igual a cero. Esto se debe a que el sistema alcanzado una velocidad en la que la fuerza en los impulsores se equilibran con las fuerzas de arrastre.

Impulsores con sentidos opuestos

En este caso ambos impulsores son paralelos al eje x pero su sentido es contrario. Se tiene que $\alpha_1 = -\alpha_2 = -\pi/2$ y $\beta_1 = \beta_2 = 0$. La fuerza de impulsión es constante e igual a 1 N en ambos impulsores.

En la figura 7.12 se presenta la rotación robot en el eje de z . De la simulación fue posible comprobar que el robot no realiza movimientos en x , y y z , ni tampoco rotaciones en los ejes x e y .

En la figura 7.13 se presenta la velocidad angular de robot en el eje z . Se puede apreciar que la velocidad del robot se incrementa proporcionalmente al tiempo, y su gráfica forma una pendiente. La aceleración angular en este caso es constante.

7.4. Resumen

En este capítulo se presento una serie de análisis sobre robots submarinos con múltiples impulsores vectorizado y por otro lado se presento el diseño de un simulador

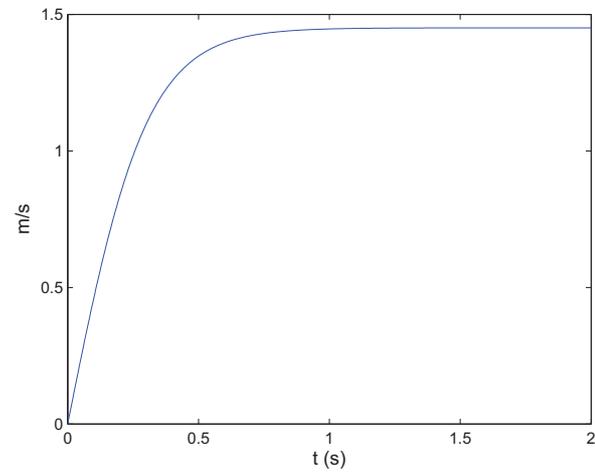


Figura 7.10: Velocidad en el eje x , $t_1 = t_2 = 20N$.

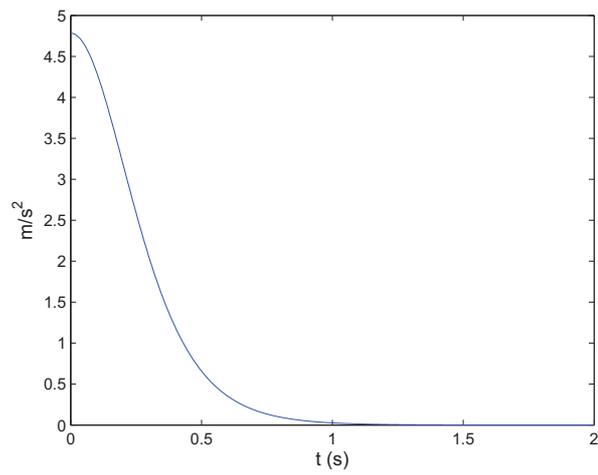


Figura 7.11: Aceleración en el eje x , $t_1 = t_2 = 20N$.

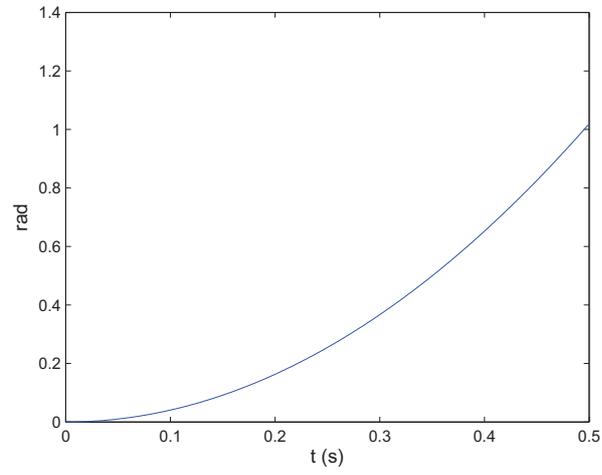


Figura 7.12: Rotación en el eje z , $t_1 = t_2 = 1N$.

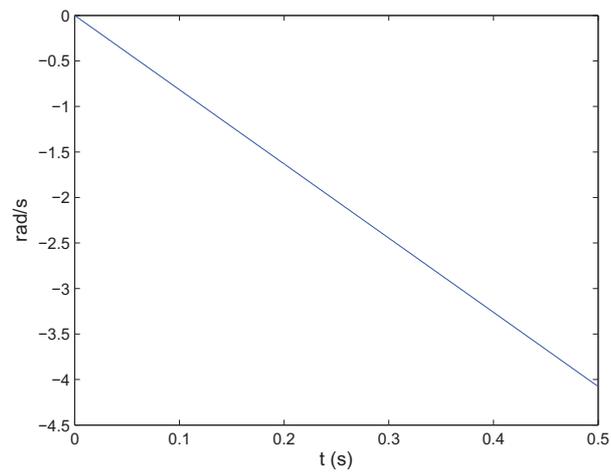


Figura 7.13: Velocidad angular en el eje z , $t_1 = t_2 = 1N$.

para robots submarinos. Ambos trabajos se relacionaron al final de este capítulo al presentarse algunas simulaciones de un robot submarino con impulsores vectorizados.

El sistema con dos impulsores no puede generar un momento sobre el eje y , es decir que el robot es incapaz de controlar el cabeceo. En tal caso es necesario un elemento adicional que genere un momento en este eje, como puede ser que el centro de flotabilidad no coincida con el centro de masa (Walker, 2005), o un actuador activo como lo es un giróscopo.

Para el robot con dos impulsores se observo (en la ecuación 7.4) que si $t_1 = t_2$, y además $\alpha_1 = \alpha_2$ y $\beta_1 = \beta_2$, el sistema de propulsión genera únicamente una fuerza lineal y no se generar ningún momento. Si el robot tiene una geometría simétrica, como lo puede ser una esfera, esto permitirá que el robot pueda realizar traslaciones puras.

Por otro lado si $\alpha_2 = \alpha_1 + \pi$ con $\beta_1 = \beta_2 = 0$ el sistema genera un momento puro alrededor del eje z . De manera similar si $\beta_2 = \beta_1 + \pi$ con $\alpha_1 = \alpha_2 = 0$ genera un momento puro alrededor del eje x .

Para el robot con tres impulsores se observo que si los centros de los actuadores vectorizados se encuentren en los vértices de un triangulo equilátero el robot puede genera fuerzas lineales y torques puros. Para esta distribución de los actuadores, se tiene que cuando todos los impulsores tienen la misma dirección y generan la misma fuerza de impulsión los momentos son nulos. Esta es una característica deseable en algunas ocasiones, ya que en un movimiento de traslación la fuerza se distribuye uniformemente en los tres actuadores o de visto desde otro punto de vista se puede utilizar la máxima potencia de los impulsores para generar un movimiento en determinado sentido. Como se puede observar la distribución de los impulsores tiene efectos en el desempeño del sistema de propulsión. Esto es igual para el robot con cuatro impulsores.

Como se indico en la introducción uno de los objetivos de esta tesis es el desarrollo de herramientas de simulación para robots submarinos. Como parte del seguimiento de este objetivo se realizo el diseño e implementación de un simulador dinámico para robots submarinos. Este simulador consiste de una librería de software desarrollada en el lenguaje de programación orientado a objetos C++. El simulador esta compuesto por 4 tipos de clases principales: `uRobot`, `uRigidBody`, `Actuator` y `Sensor`. Estas se encargan de modelar la dinámica del cuerpo rígido en el ambiente submarino, los actuadores y los sensores.

Finalmente se presentaron un par de simulaciones que permiten verificar de manera sencilla que una distribución simétrica de los impulsores y a la vez una geometría simétrica de la cabina del robot, hacen posible que se puedan realizar rotaciones y traslaciones puras.

Parte III

Conclusiones

Capítulo 8

Conclusiones Generales

La impulsión vectorial representa una alternativa válida como sistema de propulsión. Permite el uso de una menor cantidad de impulsores, esto a su vez tiene consecuencias en el volumen y costo del robot a medida que una mayor potencia de propulsión es requerida. El empleo de mecanismos seriales o paralelos para controlar la postura del impulsor dependerá en cierta medida del problema de diseño y del criterio del diseñador del sistema.

En esta tesis se tomo como objeto de estudio el robot Remo 2, cuya estructura paralela representa una gran parte del volumen del robot y esto hace que su análisis sea el más complejo que se pueda tener en los robots de esta categoría. El propósito de este trabajo de investigación fue explorar los aspectos mayor relevancia relativos al modelado, control y diseño de robots submarinos con impulsores vectorizados de estructura paralela.

Una contribución importante de este trabajo fue el planteamiento (y solución) de los problemas cinemático y dinámico inverso para un robot submarino de estructura paralela. Por otro lado, se demostró por primera vez el funcionamiento del concepto del robot submarino originalmente presentado en (Yime, 2008). Esto se realizó empleando un potente paquete de simulación cuyas prestaciones han sido validadas por diversas instituciones alrededor del mundo. Haciendo uso de esta herramienta se exploró el desempeño del robot bajo diversos esquemas de control, se encontró que el robot es capaz de ejecutar con éxito diversas maniobras empleando controladores clásicos. Posteriormente se presento un apartado cuyo propósito es exponer aspectos importantes sobre el diseño de este tipo de robots, sobre todo aquellos que influyen en su desempeño cinetostático. Se plantea el problema del diseño óptimo de un robot tipo Remo II, y se proponen un par de índices de desempeño globales. Otra contribución fue, en condición de coautoría, el diseño y análisis de una nueva estructura paralela la cual no habia sido considerada anteriormente en la literatura.

Teniendo en cuenta lo anterior, se consideran las siguientes conclusiones generales, que resumen a las presentadas al final de cada capítulo de este documento de tesis.

En el Capítulo 1 se plantea el tema de interés de este trabajo, que fue realizar un estudio sobre el modelado, control y diseño de robots submarinos con impulsores vectorizados de estructura paralela. Se realizó una discusión sobre el concepto de este

tipo de robot y un análisis comparativo con otras estructuras. Se concluyó que el aspecto en el que este robot puede tener una ventaja notable con respecto a otros, es en el uso de su estructura para realizar tareas adicionales como colaborar en las misiones de intervención o realizar operaciones de trepado. De esta manera tener una máquina robótica submarina multipropósito.

Los dos capítulos posteriores se encargan de poner en contexto al lector. En el Capítulo 2, se hizo una revisión del estado de la técnica de la robótica submarina, con particular interés en los sistemas de propulsión. Además, se presenta una descripción de las estrategias que se han propuesto para controlar estos dispositivos. El propósito de este capítulo fue aclarar el lugar en el que encajan los robots submarinos con impulsores vectorizados de estructura paralela. Por otro lado, en el Capítulo 3 se presentaron los fundamentos matemáticos sobre el modelado de robots submarinos.

En el Capítulo 4 se derivaron las ecuaciones de la cinemática y dinámica inversa de una estructura paralela donde ninguna de las plataformas está unida al marco inercial. Esta es la primera vez que se presenta un trabajo como este. La formulación presentada es fácilmente paralelizable lo cual puede ser ventajoso si hay múltiples procesadores disponibles. El modelo desarrollado aquí es útil en esquemas de control basados en la planta y la planificación de trayectorias.

En el Capítulo 5 se presentó una exploración sobre el desempeño de algunas estrategias de control para el robot submarino Remo 2. Por un lado se analizó el comportamiento del robot empleando diversas leyes de control, y por otro lado mediante la simulación del robot fue posible observar aspectos de la dinámica del robot en ambientes submarinos.

Se observó que la simetría de la arquitectura del robot facilitaba la realización de maniobras. Por ejemplo, cuando el robot se mueve en línea recta hacia adelante, las fuerzas de arrastre que se generan en los eslabones de las piernas, se anulaban perfectamente unas a otras. Se realizaron pruebas en las que la distribución de las fuerzas de arrastre en piernas no era simétrica y observó que se generaba un desbalanceo que hacía que se perdiera el equilibrio de fuerzas y a su vez se provocaran movimientos indeseables.

Los esquemas de control que se estudiaron fueron P con retroalimentación de velocidad, PID, PD con Precompensación, tanto para controlar a los impulsores, como para controlar los actuadores lineales de las piernas. Se revisó el desempeño de estos controladores con tres maniobras de prueba: línea recta hacia adelante, movimiento rotacional puro, y movimiento lateral.

Se observó que para controlar la posición de un robot con las características de Remo 2, un controlador P con retroalimentación de velocidad es más eficiente que un controlador PID. La componente integral hacía más lenta, en general, la llegada del sistema a la posición de referencia. En las pruebas que se realizaron para evaluar el control de la estructura paralela del robot, se observó que el controlador PID fue más eficaz que un controlador P con retroalimentación de velocidad. En un escenario en el que la estructura del robot está sometida a fuerzas persistentes, como cuando se mueve a una velocidad constante, se mostró que el PID sí era capaz de reducir el error hasta

alcanzar la referencia deseada.

Posteriormente se presento un estudio sobre un controlador basado en el modelo del robot. Este es el controlador PD con Precompensación. El controlador emplea el modelo dinámico inverso del robot, el cual se explico en el Capítulo 4. Esta clase controlador tiene aplicación en el seguimiento de una trayectoria deseada. Para esto, la trayectoria debe estar definida en posición velocidad y aceleración en cada instante de tiempo. Se observo que en el caso del controlador PD con Precompensación, los errores de posición y velocidad eran prácticamente nulos. Esto demuestro que bajo las condiciones del experimento (sin perturbaciones y con parámetros conocidos del sistema) el empleo de la dinámica inversa es notablemente superior.

En el Capítulo 7 se presento una serie de análisis sobre robots submarinos con múltiples impulsores vectorizados y por otro lado se presento el diseño de un simulador para robots submarinos. Ambos trabajos se relacionaron al final de este capítulo al presentarse algunas simulaciones de un robot submarino con impulsores vectorizados. Se realizó el diseño e implementación de un simulador dinámico para robots submarinos. Este simulador consiste de una librería de software desarrollada en el lenguaje de programación orientado a objetos C++. El simulador esta compuesto por 4 tipos de clases principales: `uRobot`, `uRigidBody`, `Actuator` y `Sensor`. Estas se encargan de modelar la dinámica del cuerpo rígido en el ambiente submarino, los actuadores y los sensores. Se presentaron un par de simulaciones que permiten verificar de manera sencilla que una distribución simétrica de los impulsores y a la vez una geometría simétrica de la cabina del robot, hacen posible que se puedan realizar rotaciones y traslaciones puras.

Parte IV

Apéndice

Apéndice A

Análisis del Robot de 6 g.d.l. desacoplado

En este apéndice se presenta el análisis cinemático de un robot paralelo desacoplado de 6 g.d.l., véase la figura A.1. El robot está compuesto de seis piernas con articulaciones prismáticas actuadas. Tres piernas controlan la posición de un punto de la plataforma móvil y las piernas restantes controlan la orientación. Este robot es cinemáticamente equivalente a la arquitectura propuesta por Innocenti Innocenti and Parenti-Castelli (1992).

La gran dificultad para construir un robot real basado en la arquitectura de Innocenti es la construcción de una articulación esférica tripe. Se llama articulación esférica tripe a la unión cinemática que permite que cuatro eslabones giren alrededor de un punto común. Se han presentado algunos trabajos sobre articulaciones esféricas triples, Bosscher and Elbert-Upholf (2003), Zanganeh and Angeles (1994), S. Song and Kim (2003), D. Bernier and Li (1995) and Y. Takeda and Sugimoto (2005). En Zanganeh and Angeles (1994) Zanganeh y Angeles presentan un robot paralelo redundante, donde seis eslabones se conectan en el centro del mecanismo mediante una articulación esférica y seis articulaciones de horquilla. Song et al. S. Song and Kim (2003), presento una articulación esférica para conectar tres o más eslabones. Esta articulación esférica está hecha de una bola una cubierta esférica con hoyos que permite que múltiples eslabones se conecten con la bola mediante discos de soporte que están puestos dentro del espacio que existe entre la superficie de la bola y la superficie interna de la superficie esférica. En D. Bernier and Li (1995), Bernier et al. se presentó el robot Nabla 6, un robot desacoplado de 6 grados con seis articulaciones prismáticas sobre un el plano horizontal. En estos robots pares de articulaciones prismáticas comparten el mismo eje. Tres piernas están articuladas en la misma articulación universal. El diseño de las articulaciones esféricas se basa en el uso de un apilamiento de articulaciones rotacionales con un eje de rotación en común. Los eslabones se conectan a estas articulaciones mediante un eslabón con dos articulaciones rotacionales que se intersecan en el punto central de la articulación. En Bosscher and Elbert-Upholf (2003), Bosscher y Ebert-Uphoff presentó un mecanismo para implementar múltiples articulaciones esféricas. Este mecanismo consiste de dos eslabones intermedios con dos articulaciones rotacionales

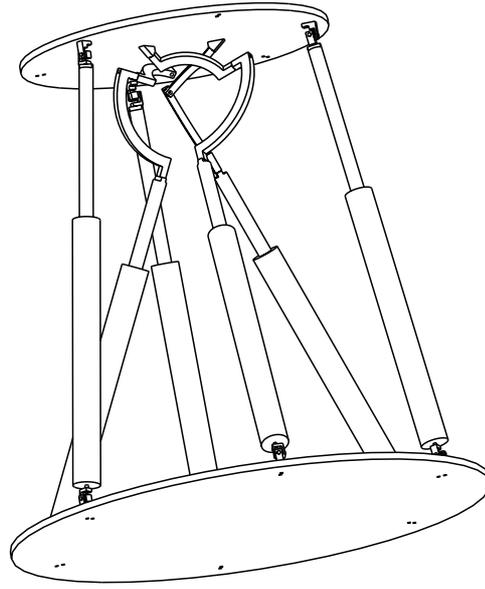


Figura A.1: Robot de 6 grados de libertad con cinemática desacoplada

entre los eslabones principales. En Y. Takeda and Sugimoto (2005), Takeda et al. presenta un estudio con el objetivo de construir robots paralelos con la arquitectura de de Innocenti. La articulación esférica presentada en ese artículo estaba compuesta de una pequeña plataforma que estaba conectada a cada eslabón principal mediante dos eslabones intermedios con articulaciones de revoluta cuyas articulaciones eran ortogonales. La plataforma además contenía una articulación esférica cuya bola estaba conectada a la plataforma móvil del robot.

Nosotros proponemos una nueva articulación esférica. Descripción del sistema El robot tiene seis pierna con articulaciones prismáticas actuadas. Tres piernas internas controlar la posición de la plataforma móvil, mientras que las piernas externas restantes controlan la orientación del efector final. Las tres piernas interna son cadenas cinemáticas $U\text{PRRR}$ (donde U denota una articulación universal, \underline{P} denota una articulación prismática actuada, y R denota una articulación rotacional). Las piernas externas son cadenas cinemáticas $U\text{PS}$ (donde S denota una articulación esférica) . Con el objetivo de obtener una cinemática desacoplada, deben de haber una articulación esférica tripe entre las tres piernas internas y el efector final. Esto se logra intersecando en un punto común el eje de las últimas tres articulaciones rotacionales de cada pierna, vea figura A.2. En este caso la segunda articulación rotacional es ortogonal a la primera y tercera articulación, por lo tanto todas las posibles orientaciones de las piernas con respecto a la plataforma móvil pueden ser alcanzadas (exceptuando aquellas donde hay singularidades). Por el otro lado, el efector final del robot tiene sus ejes rotacionales dispuestos de tal forma que se intersecan en un punto como se ilustra en la figura A.3. Cuanto las tres piernas internas se ensamblan con la plataforma móvil, el punto de intersección de cada pierna será coincidente con el punto de intersección del efector final,

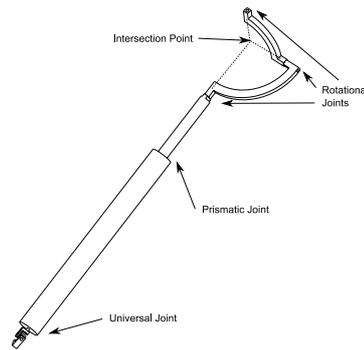


Figura A.2: Pierna interna

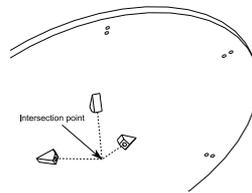


Figura A.3: Articulaciones rotacionales en la plataforma superior

formando una articulación esférica triple. La figura A.4 ilustra un esquema cinemático equivalente de este robot. Con el objetivo de verificar los grado de libertad del robot paralelo presentado aquí, utilizamos el criterio de Grüber - Kutzbach Tsai (1999).

$$F = \lambda(n - j - 1) + \sum_i^j f_i \quad (\text{A.1})$$

Donde F son los grados de libertad del mecanismo, λ son los grados de libertad del espacio en el cual el mecanismo se pretende mover, n es el numero de eslabones del mecanismo, j es el numero de articulaciones del mecanismo, f_i son los grados de movimiento relativo permitido por la articulación i . Para el robot paralelo tenemos los siguientes valores: $\lambda = 6$, $n = 20$, $j = 24$, $\sum f_i = 36$. Por lo tanto, los grados de libertad de el robot son $F = 6$. Consecuentemente, usando la longitud de los seis actuadores prismáticos como entradas, la posición y orientación de la plataforma móvil puede ser definida. A continuación se presenta el análisis cinemático

A.1. Análisis Cinemático

En esta sección la cinemática directa e inversa del robot se presentara. La cinemática directa calcula la posición y orientación del efector final conocido el valor de la longitud de las seis piernas. Por otro lado, la cinemática inversa calcula la longitud de los seis

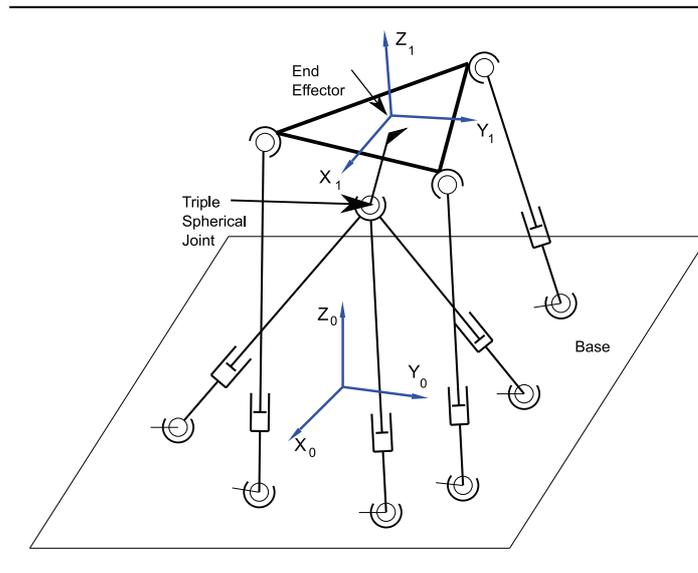


Figura A.4: Diagrama esquemático del robot

actuadores conocida la posición y orientación del efector final. Finalmente, la Jacobiana del robot se calcula usando teoría de tornillos.

A.1.1. Cinemática Directa

La cinemática directa de este robot se resuelve en dos partes. La primera parte calculara la posición del centro del efector final de la longitud de las tres piernas. La segunda parte calculara la orientación del efector final dada la longitud de las tres piernas externas.

A.1.1.1. Problema traslacional

Como se puede observar en la figura A.5, las tres piernas internas forman una pirámide donde el vértice es el centro rotacional del efector final. Los siguiente vectores están diefinidos en la figura previamente señalada, $\mathbf{d} = [d_x \ d_y \ d_z]^T$ Relacionada con la posición de la articulación esférica triple, $\mathbf{a}_i = [a_{i,x} \ a_{i,y} \ a_{i,z}]^T$, en conexión a la posición de la articulación universal en la base del cuerpo, y L_i relacionada a las longitudes de las aristas de la pirámide. Con estos vectores, la longitud total de los tres primeros actuadores, ρ_i es

$$\rho_i = L_i - h_i \quad (\text{A.2})$$

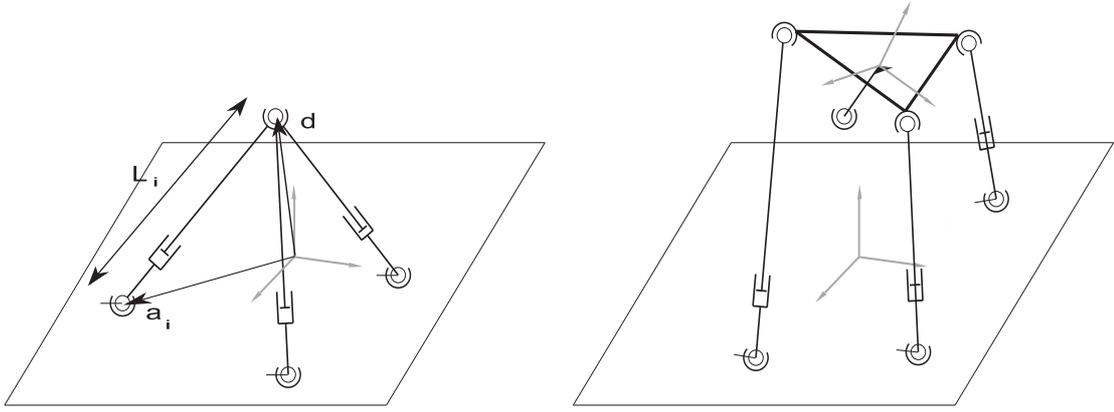


Figura A.5: Cadena cinemática interna de 3 grados de libertad.

Donde h_i es la distancia entre la articulación universal ínfero y el vértice superior de la pirámide La longitud interna se calcula con la siguiente expresión:

$$\begin{aligned} L_1^2 &= (d_x - a_{1,x})^2 + (d_y - a_{1,y})^2 + (d_z - a_{1,z})^2 \\ L_2^2 &= (d_x - a_{2,x})^2 + (d_y - a_{2,y})^2 + (d_z - a_{2,z})^2 \\ L_3^2 &= (d_x - a_{3,x})^2 + (d_y - a_{3,y})^2 + (d_z - a_{3,z})^2 \end{aligned} \quad (\text{A.3})$$

(A.4)

Resolviendo d_x y d_y ,

$$d_x = \frac{k_1(a_{1,y}a_{2,y}) - k_2(a_{1,y}a_{3,y})}{2k_3} \quad (\text{A.5})$$

$$d_y = \frac{-k_1(a_{1,y}a_{2,y}) + k_2(a_{1,y}a_{3,y})}{2k_3} \quad (\text{A.6})$$

Donde k_i están definidas con los siguientes valores,

$$\begin{aligned} k_1 &= L_1^2 - L_3^2 + a_{3,x}^2 - a_{1,x}^2 + a_{3,y}^2 - a_{1,y}^2 \\ k_2 &= L_1^2 - L_2^2 + a_{2,x}^2 - a_{1,x}^2 + a_{2,y}^2 - a_{1,y}^2 \\ k_3 &= a_{1,x}a_{2,y}a_{1,y}a_{2,x} + a_{3,x}(a_{1,y} - a_{2,y}) \\ &\quad - a_{3,y}(a_{1,x} - a_{2,x}) \end{aligned} \quad (\text{A.7})$$

Las coordenadas z se obtienen después de reemplazar los valores de d_x y d_y en cualquiera de las ecuaciones (A.4).

A.1.1.2. Problema rotacional

Analizando la figura A.6 se puede notar que la longitud de cada actuador traslacional se determina mediante la siguiente ecuación

$$L_i = \|\mathbf{d} + \mathbf{R}(\mathbf{c}_i - \mathbf{e}) - \mathbf{b}_i\| \quad (\text{A.8})$$

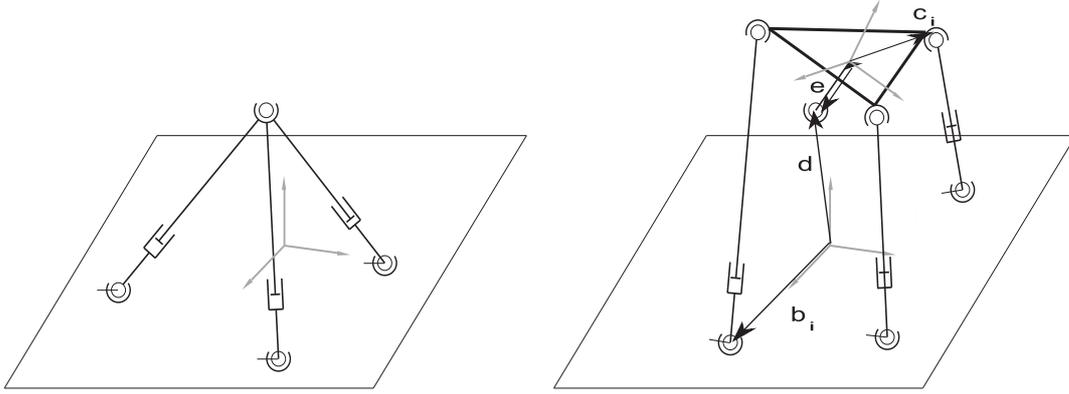


Figura A.6: Cadena cinemática externa de 3 grados de libertad.

Donde \mathbf{R} es la matriz del efector final, \mathbf{e} es el vector de excentricidad del anillo superior, \mathbf{c}_i es el punto de intersección del actuador con el efector final y \mathbf{b}_i es el punto de intersección de cada actuador con la base. Por otro lado, el cuadrado de la ecuación (A.8) es,

$$L_i^2 = \|\mathbf{d} + \mathbf{R}(\mathbf{c}_i - \mathbf{e}) - \mathbf{b}_i\|^2 \quad (\text{A.9})$$

Lo cual puede ser también expresado como:

$$L_i^2 = (\mathbf{d} + \mathbf{R}(\mathbf{c}_i - \mathbf{e}) - \mathbf{b}_i)^T (\mathbf{d} + \mathbf{R}(\mathbf{c}_i - \mathbf{e}) - \mathbf{b}_i) \quad (\text{A.10})$$

O en una forma simplificada como

$$L_i^2 = \mathbf{d}^T \mathbf{d} + 2(\mathbf{d} - \mathbf{b}_i)^T \mathbf{R}(\mathbf{c}_i - \mathbf{e}) - 2\mathbf{d}^T \mathbf{b}_i + (\mathbf{c}_i - \mathbf{e})^T (\mathbf{c}_i - \mathbf{e}) + \mathbf{b}_i^T \mathbf{b}_i \quad (\text{A.11})$$

Si subsecuentemente se ponen las variables para calcular en la izquierda:

$$2(\mathbf{d} - \mathbf{b}_i)^T \mathbf{R}(\mathbf{c}_i - \mathbf{e}) = L_i^2 - \mathbf{d}^T \mathbf{d} + 2\mathbf{d}^T \mathbf{b}_i - (\mathbf{c}_i - \mathbf{e})^T (\mathbf{c}_i - \mathbf{e}) - \mathbf{b}_i^T \mathbf{b}_i \quad (\text{A.12})$$

Y los siguientes vectores son definidos

$$\begin{aligned} \mathbf{u}_i &= \begin{bmatrix} u_{x,i} & u_{y,i} & u_{z,i} \end{bmatrix}^T \\ &= 2(\mathbf{d} - \mathbf{b}_i)^T \end{aligned} \quad (\text{A.13})$$

$$\begin{aligned} \mathbf{v}_i &= \begin{bmatrix} v_{x,i} & v_{y,i} & v_{z,i} \end{bmatrix}^T \\ &= 2(\mathbf{c}_i - \mathbf{e}) \end{aligned} \quad (\text{A.14})$$

$$\begin{aligned} \mathbf{w}_i &= \begin{bmatrix} w_{x,i} & w_{y,i} & w_{z,i} \end{bmatrix}^T \\ &= L_i^2 - \mathbf{d}^T \mathbf{d} + 2\mathbf{d}^T \mathbf{b}_i \\ &\quad - (\mathbf{c}_i - \mathbf{e})^T (\mathbf{c}_i - \mathbf{e}) - \mathbf{b}_i^T \mathbf{b}_i \end{aligned} \quad (\text{A.15})$$

Se obtendrá un conjunto de tres ecuaciones no lineal que determinan la solución de la cinemática directa,

$$\mathbf{u}_i^T \mathbf{R} \mathbf{v}_i = \mathbf{w}_i \quad (\text{A.16})$$

A.1.2. Cinemática Inversa

Si se define un vector de posición de el efector con respecto a la base \mathbf{p} , las longitudes de seis actuadores se definen mediante las siguientes ecuaciones,

$$L_i = \|\mathbf{p} + \mathbf{R}\mathbf{c}_i - \mathbf{b}_i\|, \forall i = \{1, \dots, 3\} \quad (\text{A.26})$$

$$L_j = \|\mathbf{p} + \mathbf{R}\mathbf{e} - \mathbf{a}_{j-3}\|, \forall j = \{4, \dots, 6\} \quad (\text{A.27})$$

Donde $\mathbf{e} = \begin{bmatrix} 0 & 0 & e_z \end{bmatrix}^T$

A.1.3. Análisis de Velocidad

Para cada pierna el Screw resultante sera:

$$\mathcal{S}_r = \omega_{1,i} \hat{\mathcal{S}}_{1,i} + \omega_{2,i} \hat{\mathcal{S}}_{2,i} + v_i \hat{\mathcal{S}}_{3,i} + \omega_{4,i} \hat{\mathcal{S}}_{4,i} + \omega_{5,i} \hat{\mathcal{S}}_{5,i} + \omega_{6,i} \hat{\mathcal{S}}_{6,i} \quad (\text{A.28})$$

Donde $\mathcal{S}_r = \begin{bmatrix} \mathbf{v}^T & \omega^T \end{bmatrix}^T$, $\hat{\mathcal{S}}_{j,i}$ es el tornillo a lo largo de la j-esima articulación en la i-esima pierna, $\omega_{j,i}$ es la velocidad angular de la j-esima articulación en la i-esima pierna, v_i es la velocidad lineal de la articulación traslacional de la i-esima pierna. Sin embargo el recíproco del tornillo para cada pierna es $\hat{\mathcal{S}}_{3,i}$. Si multiplicamos (A.28) por este tornillo obtenemos

$$\langle \mathcal{S}_r, \hat{\mathcal{S}}_{3,i} \rangle = v_i = \hat{\mathcal{S}}_{3,i}^T \mathcal{S}_r \quad (\text{A.29})$$

Entonces, la expresion de velocidad de cada pierna puede ser agrupada en un vector,

$$\begin{bmatrix} v_1 \\ \vdots \\ v_6 \end{bmatrix} = \begin{bmatrix} \hat{\mathcal{S}}_{3,1}^T \\ \vdots \\ \hat{\mathcal{S}}_{3,6}^T \end{bmatrix} \mathcal{S}_r \quad (\text{A.30})$$

O en forma matricial,

$$\mathbf{v} = \mathbf{J} \mathcal{S}_r \quad (\text{A.31})$$

donde la Jacobiana, \mathbf{J} , es,

$$\mathbf{J} = \begin{bmatrix} \mathbf{u}_1^T & -\mathbf{u}_1^T \tilde{\mathbf{b}}_1 \\ \mathbf{u}_2^T & -\mathbf{u}_2^T \tilde{\mathbf{b}}_2 \\ \mathbf{u}_3^T & -\mathbf{u}_3^T \tilde{\mathbf{b}}_3 \\ \mathbf{u}_4^T & -\mathbf{u}_4^T \tilde{\mathbf{e}} \\ \mathbf{u}_5^T & -\mathbf{u}_5^T \tilde{\mathbf{e}} \\ \mathbf{u}_6^T & -\mathbf{u}_6^T \tilde{\mathbf{e}} \end{bmatrix} \quad (\text{A.32})$$

\mathbf{u}_i son vectores unitarios con la dirección de los actuadores traslacionales, y $\tilde{\mathbf{e}}$ es la matriz de producto cruz,

$$\tilde{\mathbf{e}} = \begin{bmatrix} \tilde{e}_x \\ e_y \\ e_z \end{bmatrix} = \begin{bmatrix} \mathbf{0} & -e_z & e_y \\ e_z & \mathbf{0} & -e_x \\ -e_y & e_x & \mathbf{0} \end{bmatrix} \quad (\text{A.33})$$

Apéndice B

Librería de Simulación Dinámica

En este apéndice se describirá la librería de simulación de cuerpos rígidos sin restricciones que se ha desarrollado en parte para construir el simulador de robots submarinos. Esta librería está compuesta de tres elementos principales, el módulo de simulación, el módulo de cálculo de fuerzas de contacto, y el módulo de detección de colisiones. Cada uno de estos módulos se puede utilizar por separado o en conjunto. El diseño modular de la librería la hace flexible para utilizar sus funciones con otras herramientas de simulación o dispositivos, como lo son el MSIM y el Dispositivo Haptico como lo es el Phantom de Sensable (Moreno). En este simulador utilizamos la formulación basada en penalizaciones y por eso el simulador es llamando MinPBS que es un acrónimo de Minimal Library for Penalty Based Simulation (en castellano es Librería Mínima para Simulaciones Basadas en Penalizaciones).

Antes de comenzar es importante saber que una simulación dinámica se realiza en tres fases:

1. La detección de colisiones
2. La determinación de fuerzas de contacto
3. La solución de las ecuaciones de movimiento

En la fase de detección de colisiones se determinan los puntos de contacto entre los diversos cuerpos a partir de la geometría y posición de los mismos. Una vez conocidos los puntos de contacto se calculan las fuerzas de contacto que finalmente se utilizan para calcular el movimiento de los cuerpos después de un paso de integración de las ecuaciones de movimiento.

A continuación se presenta la manera de calcular las fuerzas de contacto entre cuerpos y las geometrías consideradas. Posteriormente se describen los paquetes que contienen las funciones del simulador.

B.1. Simulación Basada en Penalizaciones

Cuando dos objetos se tocan o colisionan, se debe aplicar una Respuesta de Colisión para prevenir que exista interpenetración. Un método para calcular las

Respuestas de Colisión es la inserción de resortes rígidos en los puntos de contacto. Este método se inspira en el hecho de que, cuando los objetos colisionan, existen pequeñas deformaciones en las regiones de contacto, y esas deformaciones pueden ser modeladas con resortes, aunque los objetos son geoméricamente rígidos.

Para calcular la fuerza penalizante considérese que el contacto \mathbf{k} , entre los dos cuerpos i y j con centro de masa en \mathbf{r}_i y \mathbf{r}_j , respectivamente, está localizado en \mathbf{c}_k . El contacto \mathbf{k} tiene la normal de contacto \mathbf{n}_k apuntando desde el cuerpo i hacia el cuerpo j y una medida de la profundidad de la penetración \mathbf{d}_k . La fuerza de penalización actuando sobre el cuerpo j se calcula de la siguiente manera:

$$\mathbf{f}_j = -(\mathbf{k}_s \mathbf{d}_k - b \mathbf{u}_k \cdot \mathbf{n}_k) \mathbf{n}_k \quad (\text{B.1})$$

donde \mathbf{u}_k denota la velocidad relativa del punto \mathbf{k} , esto es:

$$\mathbf{u}_k = (\mathbf{v}_i + \boldsymbol{\omega}_i \times \mathbf{r}_{ki}) - (\mathbf{v}_j + \boldsymbol{\omega}_j \times \mathbf{r}_{kj})$$

\mathbf{r}_{ki} y \mathbf{r}_{kj} representan la posición del contacto con respecto al centro de masa de los cuerpos i y j . El amortiguamiento se debe aplicar unicamente en la dirección de la restricción.

Existe un momento debido a la fuerza de contacto:

$$\boldsymbol{\tau}_j = \mathbf{r}_{kj} \times \mathbf{f}_j \quad (\text{B.2})$$

La fuerza y el momento en el cuerpo i se obtienen mediante la segunda ley de Newton.

Los métodos basados en penalizaciones tienen como ventaja que el modelo de fuerza es local para cada contacto y computacionalmente simple, la interpenetración es inherentemente permitida. Las desventajas son que no existe control directo sobre los parámetros físicos, como lo es el coeficiente de restitución. Las restricciones de no penetración son forzadas por medio de coeficientes de rigidez muy altos. Para los propósitos del simulador presentado aquí estas desventajas no representan una limitación importante.

B.2. Geometrias consideradas

En esta tesis se utilizan diversas primitivas para modelar la geometría de los cuerpos. Las primitivas utilizadas son las siguientes: esfera, barra, colchón, marco, cubo, plano, disco y toroide. La ventaja de utilizar estas primitivas es que determinar un contacto es un problema que se puede resolver directamente. Para determinar el contacto solo es necesario encontrar los puntos más cercanos entre los elementos geoméricos que definen la primitiva ya partir de ahí determinar los puntos mas cercanos sobre las superficies. En la figura B.1 se muestran las primitivas consideradas.

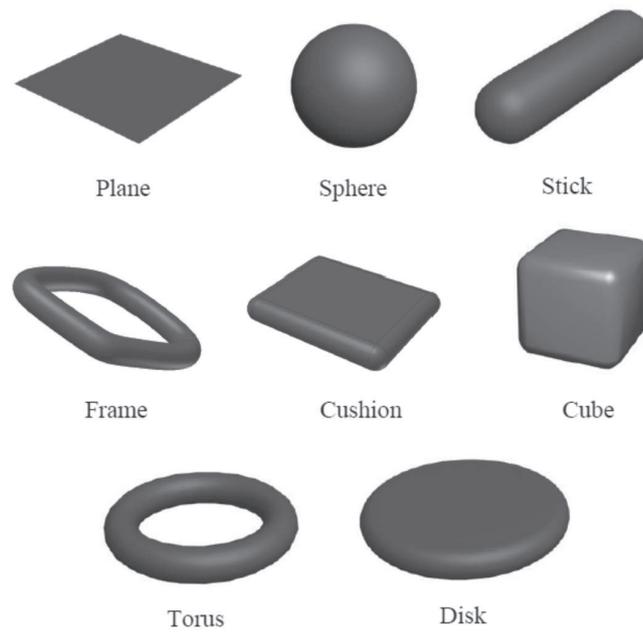


Figura B.1: Geometrias consideradas para la detección de colisiones

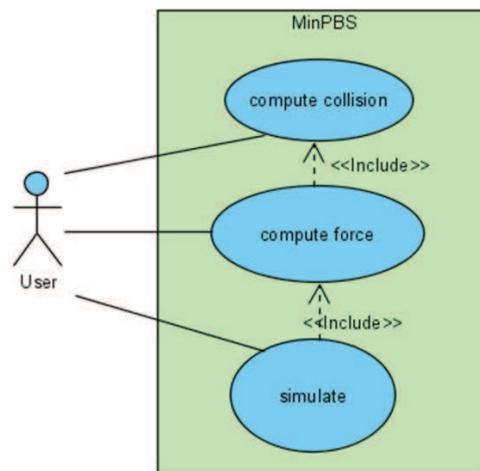


Figura B.2: Diagrama de casos de uso de la librería MinPBS.

B.3. Descripción de la Librería

Para describir la librería se utiliza el lenguaje UML, (Rumbaugh et al., 2000). El diagrama de los casos de uso del simulador se muestra en la figura B.2.

En la figura B.3 se presentan los paquetes y clases que conforman el simulador MinPBS. Los paquetes principales son: Dynamic Simulator, Force Computation y Collision Detection. Además se ha agregado otro paquete que es Graphics que permite la graficación de los cuerpos rígidos simulados utilizando funciones de OpenGL.

B.4. Paquete de Detección de Colisiones

Este paquete contiene todas las clases para determinar las colisiones entre diversas primitivas presentadas anteriormente. Para cada primitiva existe una clase que permite modelarla e instanciarla. La principal tarea de este paquete es la determinación de contactos entre primitivas. Esta información puede ser utilizada para saber si los cuerpos están en contacto y en donde, para poder determinar las fuerzas de contacto. En la figura B.4 se muestra el diagrama de clases con métodos y atributos de este paquete.

En la tabla B.1 se describen las clases de la figura B.4.

B.4.0.1. Clase Geometry

Esta clase es una clase abstracta y sirve para modelar una geometría. Esta clase se generaliza en varias clases hijas: Sphere, Stick, Cube, Cushion, Disk, Torus, Plane, Frame. En la siguiente figura se presenta la clase Geometry.

En la tabla B.2 se presenta una descripción de los métodos de esta clase.

Clase GeometryList

Esta clase permite gestionar la lista de geometrías utilizadas para realizar la simulación. Los objetos de esta clase son utilizados por el módulo de cómputo de fuerzas de contacto y el módulo de simulación.

En la tabla B.3 se presenta una descripción de los métodos de esta clase.

Clase ContactsQueue

Para guardar los contactos entre cuerpos se utiliza una cola, en la que se puede almacenar un número variable de contactos. La cola es rellena en la fase de detección de contactos y posteriormente se limpia cuando se calculan las fuerzas de contacto. El elemento base de la cola es Contact que es una clase que contiene la información relevante del contacto como lo es: los puntos de contacto, la dirección y la distancia de penetración.

Clase CollisionDetection

Esta es la clase principal donde se realiza la detección de colisiones. Se revisan cada uno de los elementos de la lista y se le asignan los contactos que tiene con los demás elementos de la lista.

En la tabla B.5 se presentan una descripción de las funciones de la clase.

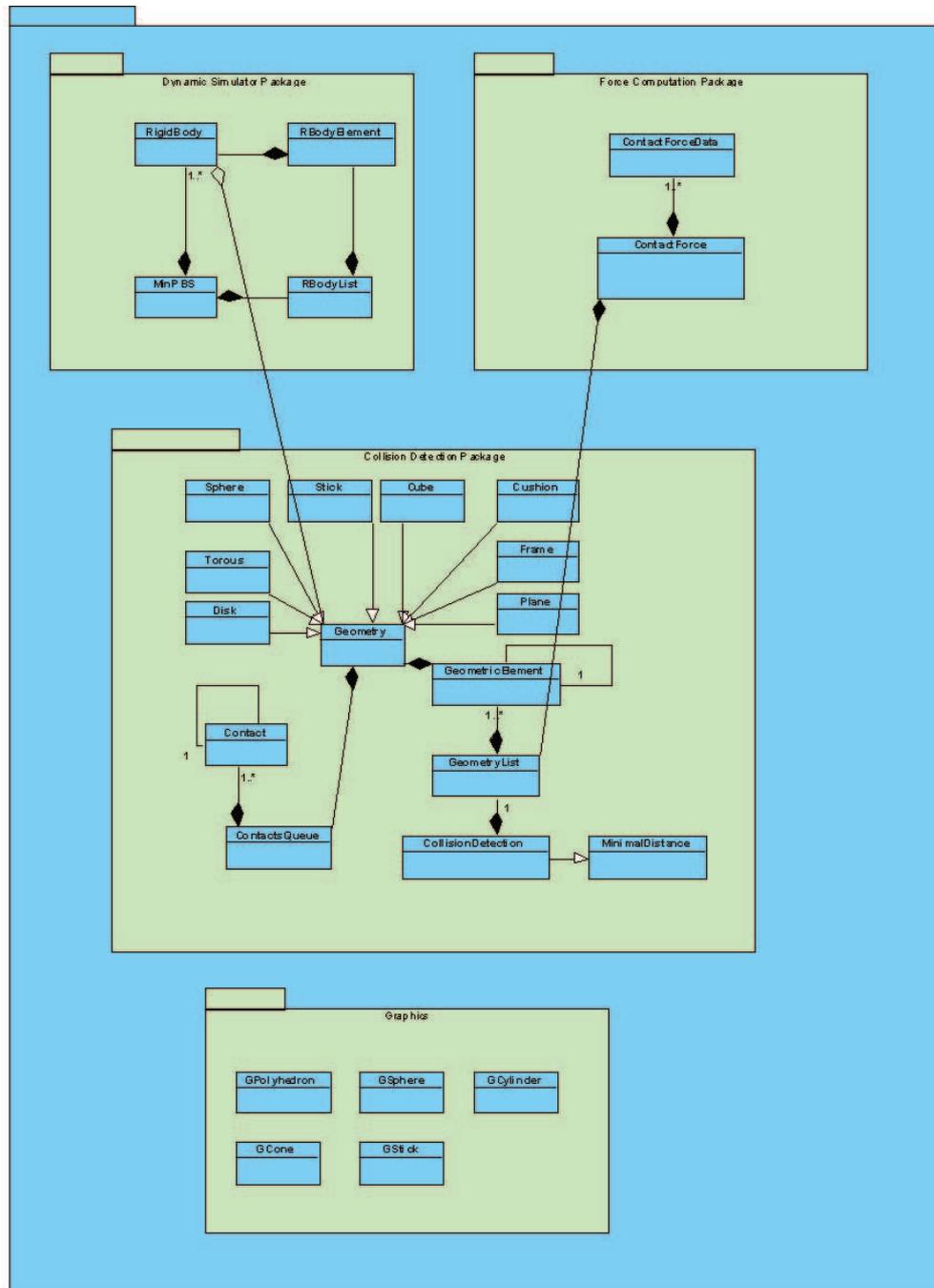


Figura B.3: Diagrama de Clases del simulador MinPBS (repetida).

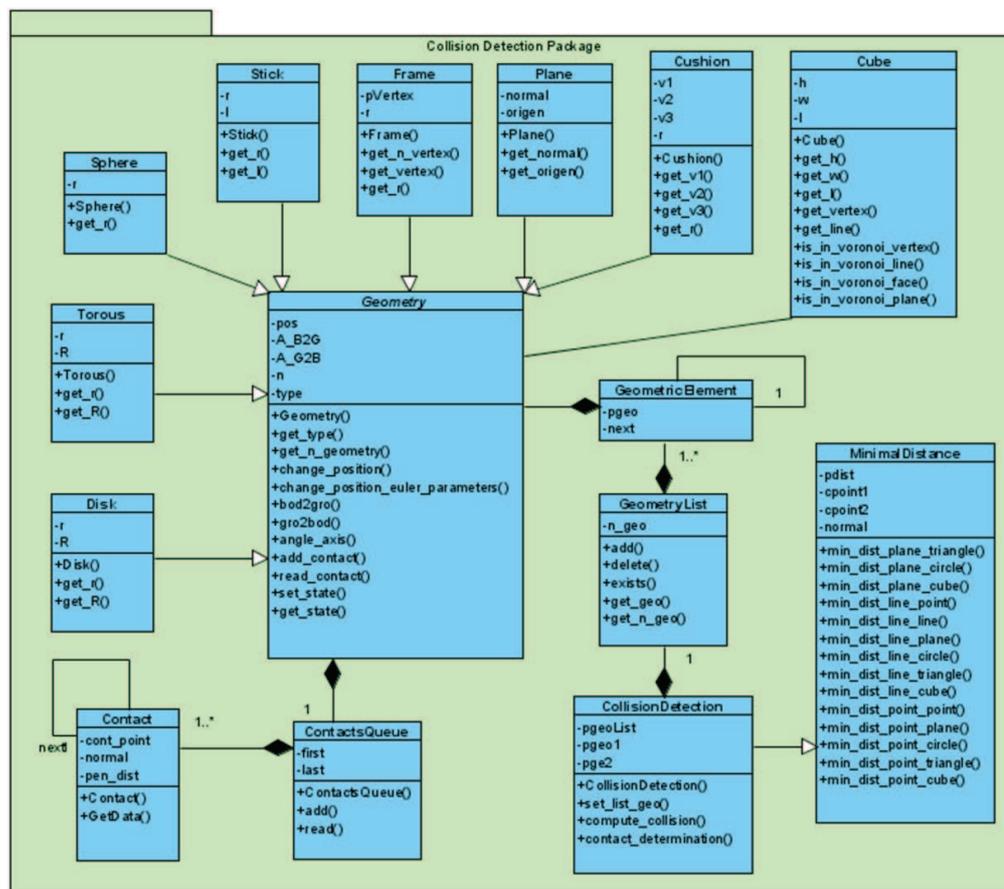


Figura B.4: Paquete de Detección de Colisiones.

Tabla B.1: Descripción de las clases

Nombre de la Clase	Descripción
ColisionDetection	Este es la clase principal, se encarga de realizar la detección de colisiones entre cada uno de los elementos geométricos contenidos en la lista
GeometryList	Esta clase se encarga de gestionar los elementos de una lista de elementos geométricos
GeometricElement	Esta clase es el elemento con el que se construye la lista
Geometry	Esta clase es una clase abstracta, sirve para modelar una geometría y después se generaliza en cada una de las entidades previamente descritas
Sphere	Esta clase sirve para modelar una esfera
Stick	Esta clase sirve para modelar un bastón
Cube	En esta clase se modela un cubo, contiene las funciones para determinar si un punto se encuentra dentro de la región de Voronoi de una de sus características
Cushion	Este elemento consiste de diversos triángulos, contiene funciones para determinar si un punto se encuentra dentro de alguna de las regiones de Voronoi de triangulo
Disk	Esta clase sirve para modelar un disco
Torus	Esta clase sirve para modelar un toroide
Plane	Esta clase modela un plano
Frame	Esta clase modela un marco
ContactQueue	Esta clase es el elemento con el que se construye la cola de contactos para cada una de las geometrías
Contact	Esta clase contiene la información principal que modela un contacto
MinimalDistance	Esta clase contiene todas las funciones que determinan la distancia mínima y los puntos más cercanos entre las diversas entidades geométricas

Tabla B.2: Descripción de las funciones de la clase Geomtry

Nombre de la Función	Descripción
get_type	Esta función sirve para identificar que tipo de geometría es (Sphere, Stick, etc.)
get_n_geometry	Permite obtener el numero de la geometría en la lista
change_position	Esta función permite definir la posición de la geometría con las coordenadas cartesianas y los Ángulos de Euler 3-1-3.
change_position_euler_parameters	Esta función al igual que la anterior permite definir la posición pero la orientación se expresa en Parámetros de Euler
bod2gro	Transforma un vector expresado en coordenadas del cuerpo a un vector expresado en coordenadas del mundo
gro2bod	Transforma un vector expresado en coordenadas del mundo a coordenadas del cuerpo
angle_axis	Entrega la orientación del cuerpo en términos de eje-angulo
add_contact	Agrega un contacto a la cola de contactos de la geometría
read_contact	Entrega un contacto de la cola de contactos y lo borra.
set_state	Permite actualizar el estado en términos de posiciones y velocidades del cuerpo, esta función es utilizada por la clase RigidBody después e un paso de integración
get_state	Entrega el estado del cuerpo (posición y velocidad).

Tabla B.3: Descripción de las funciones de la clase GeometryList

Nombre de la Función	Descripción
add	Agrega una geometría a la lista
delete	Elimina una geometría de la lista
exists	Pregunta si existe un elemento en la lista
get_geo	Entrega un puntero a cierta geometría
get_n_geo	Entrega el numero de geometrías

Tabla B.4: Descripción de las funciones de la clase ContactsQueue

Nombre de la Función	Descripción
add	Agrega un contacto a la cola.
read	Entrega la información contenida en un contacto y lo borra
GetData	Entrega la información

Tabla B.5: Descripción de las funciones de la clase CollisionDetection

Nombre de la Función	Descripción
set_list_geo	Se establece la lista de geometrías
compute_collision	Esta función revisa cada una de las geometrías para determinar los contactos entre los cuerpos
contact_determination	Esta función calcula el contacto entre dos primitivas

La función `compute_collision` revisa cada uno de los elementos de la lista y determina sus contactos con los demás objetos. Para hacer esto se realizan dos bucles, de tal manera que se analiza la distancia entre todos los posibles pares de cuerpos y si existe penetración se determina el contacto y se guarda en la cola de contactos de cada una de las geometrías. En la figura B.5 se presenta un diagrama de secuencia del método `compute_collision`.

Función ContactDetermination

La función `ContactDetermination` es una función sobrecargada que calcula el contacto entre dos primitivas. Tiene como argumentos de entrada dos primitivas geométricas, y como argumentos de salida un par de variables tipo `Contact` correspondientes a cada geometría y que contiene la información del contacto. En la figura B.6 se puede apreciar el diagrama de secuencia de esta función.

B.5. Paquete de Fuerzas de Contacto

En este paquete se encuentran las funciones que calculan las fuerzas de contacto. La clase principal es `ContactForce` que contiene todas las funciones necesarias para calcular las fuerzas.

En la tabla B.6 se presenta una descripción de las clases.

Clase CollisionDetection

En la tabla B.7 se describen las funciones miembro de la clase `ContactForce`.

La función `compute_force` (ver fig. B.8) está compuesta de dos bucles, el primero en el que se revisan cada uno de las geometrías de la lista y otro que esta anidado y a través

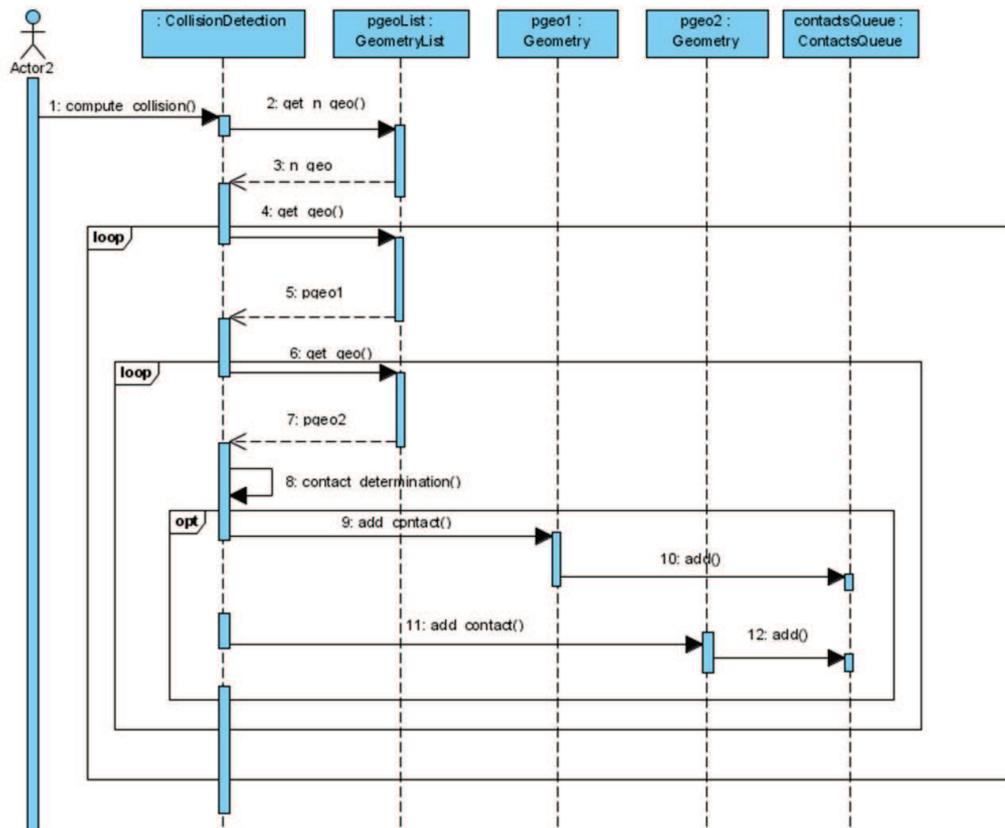


Figura B.5: Diagrama de secuencia que se ejecuta cuando el usuario llama a la función `compute_collision`.

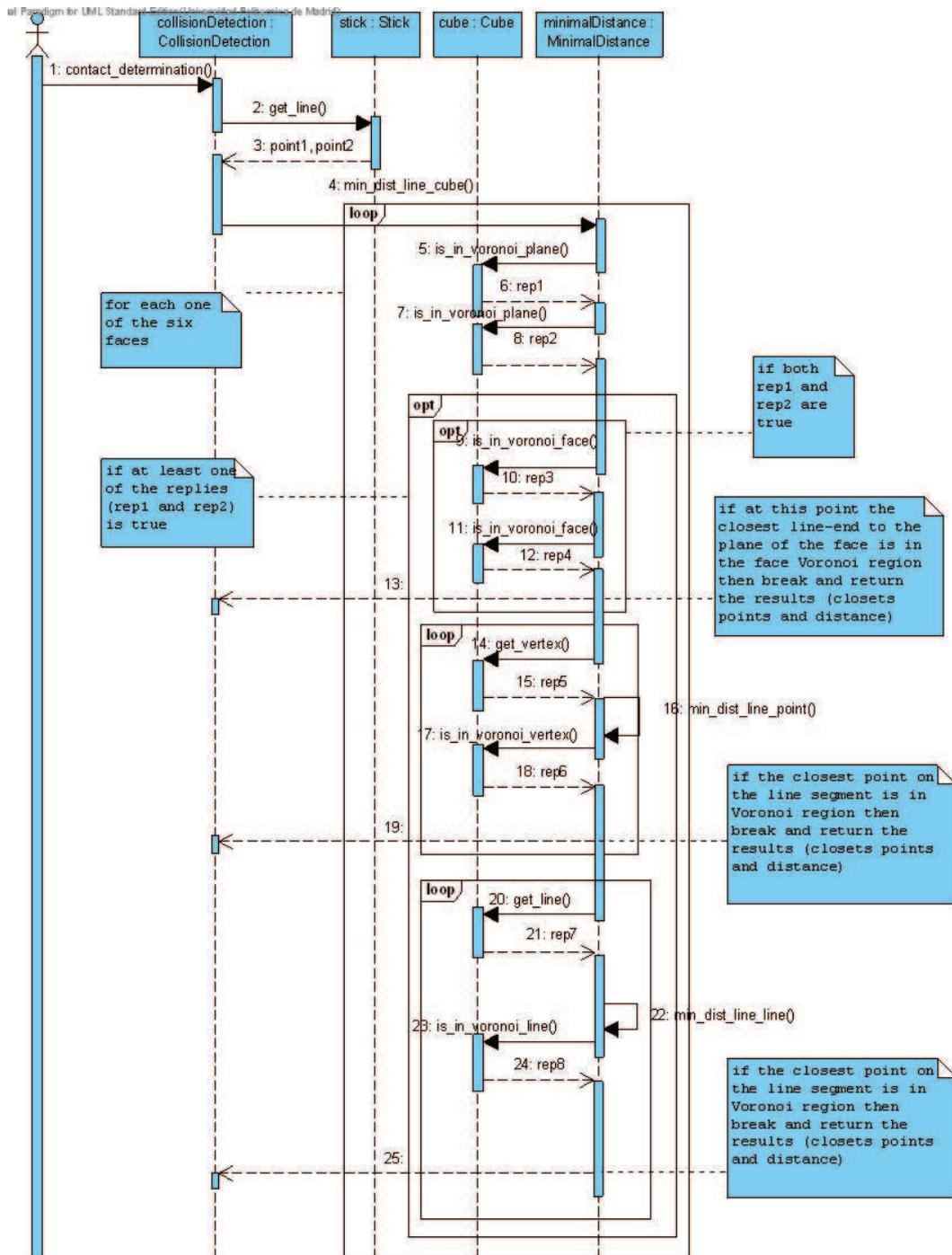


Figura B.6: Diagrama de secuencia de contact_determination.

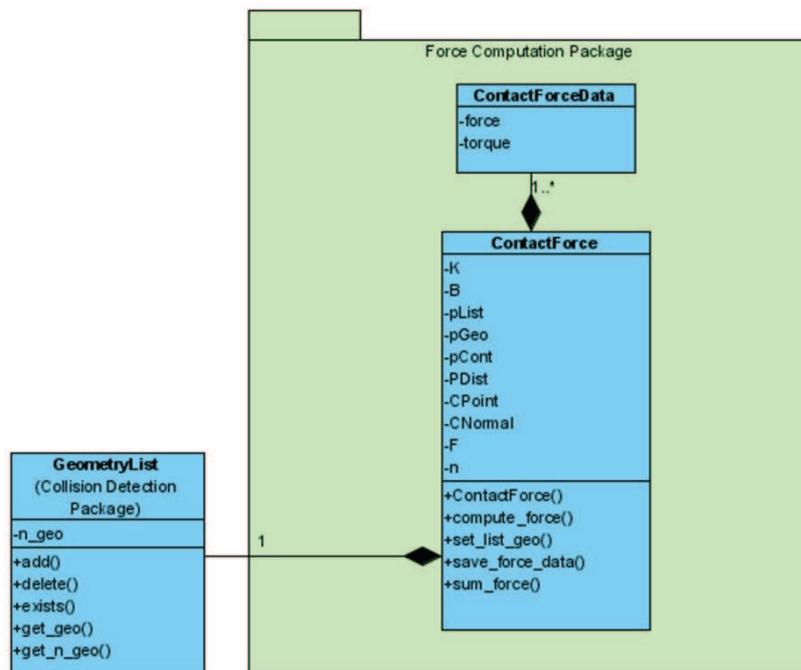


Figura B.7: Paquete de cálculo de fuerzas de contacto.

Tabla B.6: Descripción de las clases para el cálculo de fuerzas de contacto

Nombre de la Función	Descripción
ContactForceData	Esta es una estructura que sirve para contener la fuerza y el par sobre cada uno de los cuerpos
ContactForce	Esta clase contiene los métodos para calcular la fuerza y el torque en cada uno de los elementos geométricos contenidos en la lista de geometrías.

del cual se leen cada uno de los contactos de la cola de contactos para posteriormente calcular las fuerzas correspondientes.

B.6. Paquete de Simulación

Este paquete contiene las clases principales del motor de simulación. Para realizar la simulación, los cuerpos se modelan mediante instancias de la clase *RigidBody*. Para actualizar cada uno de los objetos, estos son almacenados en una lista de tal manera que se puede acceder sistemáticamente a los métodos de cada una de las clases.

Clase RigidBody

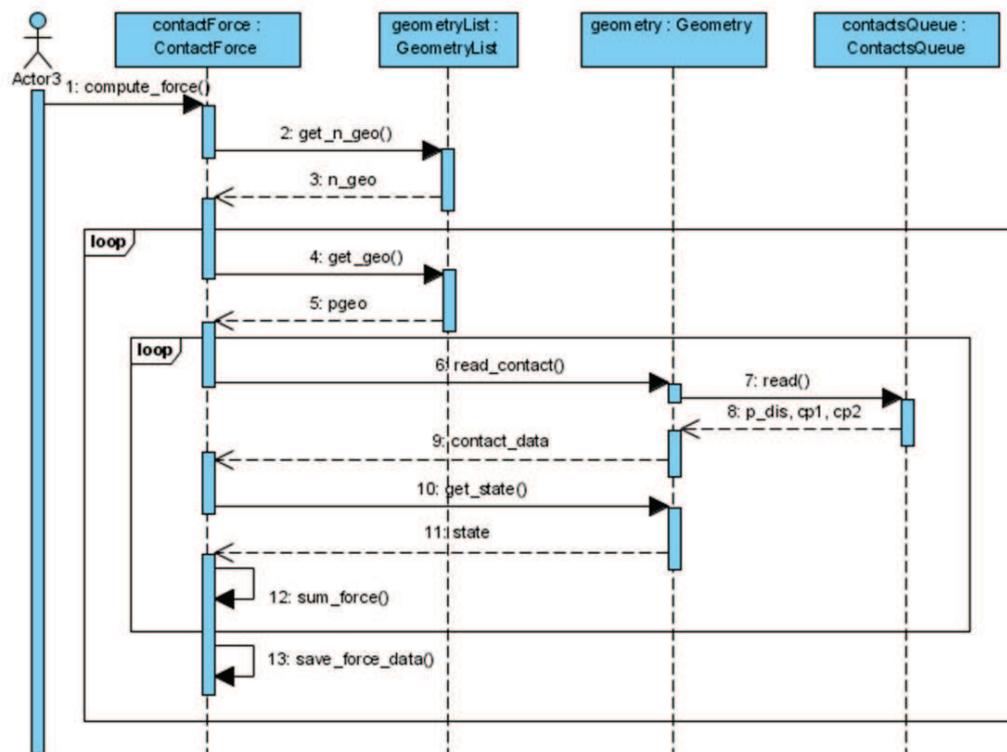
Figura B.8: Diagrama de secuencia de `compute_force`.

Tabla B.7: Descripción de las clases

Nombre de la Función	Descripción
compute_force	Esta es la función principal de la clase ContactForce y calcula la fuerza a partir de los contactos guardados en la cola de cada una de las geometrías.
set_list_geo	Se establece la lista de geometrías
save_force_data	Este comando guarda la fuerza total asignada al cuerpo en términos de fuerza lineal y torque sobre el centro de masa
sum_force	Este comando realiza la operación de sumar las fuerzas correspondientes a cada contacto

Tabla B.8: Descripción de las clases del Paquete de Simulación

Nombre de la Función	Descripción
MinPBS	Esta clase contiene los métodos para actualizar las fuerzas de colisión sobre cada cuerpo y mandar ejecutar la integración de las ecuaciones de movimiento
RBodyElement	Elemento para construir la lista
RbodyList	Se encarga de gestionar la lista que contiene a los cuerpos rígidos
RigidBody	Esta clase contiene toda la información relativa a los cuerpos rígidos. A través de esta clase se puede definir los parámetros inerciales de los cuerpos y calcular las ecuaciones de movimiento e integrarlas

Esta clase sirve para modelar un cuerpo rígido. Contiene los parámetros inerciales del cuerpo (masa y matriz de inercia) y el estado en un instante de tiempo. En la tabla B.9 se describen las funciones de esta clase.

Clase MinPBS

Esta clase se encarga de actualizar la fuerza de cada uno de los cuerpos para que se incluya en las ecuaciones de movimiento para después integrarlas y determinar las posiciones y velocidades actuales. En la tabla B.10 se presentan las funciones de la clase MinPBS.

Por otra parte en la figura B.10 se presenta el diagrama de secuencia para la función simulate. Básicamente se compone de un bucle en el que se revisan cada uno de los

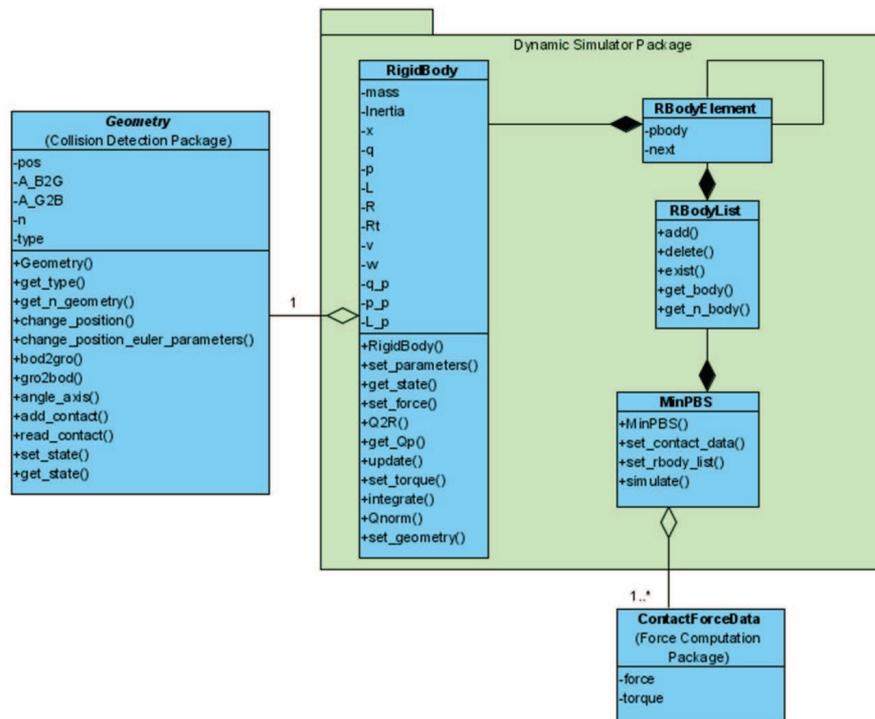


Figura B.9: Diagrama de Clases del paquete de simulación.

cuerpos, en la primera fase se actualiza el valor de las fuerzas externas y posteriormente se llama la función `update` de los objetos de la clase `RigidBody`, al final se actualiza también el estado de la geometría asociada al cuerpo.

B.7. Uso de la Librería

En esta sección se presenta un ejemplo construido en el simulador. En el ejemplo se crea un ambiente con múltiples cuerpos que caen desde cierta altura y chocan entre ellos y contra el suelo. Para construir la aplicación se escribe una rutina en un archivo de C++. El primer paso es incluir los archivos de cabecera que contienen las clases para realizar la simulación. Posteriormente se crean los objetos geométricos. En este ejemplo se utilizan tres tipos de geometrías, el plano, la esfera y el bastón. A continuación se crean los cuerpos rígidos. Finalmente se crean los módulos de detección de colisiones, cálculo de fuerzas de contacto y simulación.

```
// First include the header files
#include "cont_det.h" #include "cont_force.h" #include
"rigid_body.h" #include "plane.h" #include "stick.h" #include
"sphere.h"

//Create the geometric objects
```

Tabla B.9: Funciones de la clase RigidBody

Nombre de la Función	Descripción
set_parameters	A través de esta función se pueden definir los parámetros inerciales del cuerpo
get_state	Esta función entrega el estado del cuerpo (posición y velocidad)
set_torque	Mediante esta función se aplica el torque al cuerpo
set_force	Mediante esta función se aplica la fuerza al cuerpo
set_geometry	Define la geometría asociada a este cuerpo
Q2R	Esta función transforma un vector de parámetros de Euler en una Matriz de Rotación
get_Qp	Entrega la velocidad del quaternion
update	Esta función ejecuta diversas operaciones para encontrar la posición y velocidad del cuerpo después de un paso de integración
Integrate	Integra las ecuaciones de movimiento
Qnorm	Normaliza el cuaternion de parámetros de Euler

Tabla B.10: Descripción de las funciones de MinPBS

Nombre de la Función	Descripción
set_contact_data	Se especifica el arreglo que contiene las fuerzas de colisión en cada uno de los cuerpos
set_rbody_list	Se especifica la lista de cuerpos
simulate	Se realiza calcula la posición y velocidad de todos los cuerpos después de un paso de integración

```
Plane Plane1(1); Sphere Sph1(1, RSph1); Sphere Sph2(2, RSph2);
Sphere Sph3(3, RSph3); Sphere Sph4(4, RSph4); Sphere Sph5(5, RSph5);
Stick Sti1(6, LSti1, RSti1); Stick Sti2(7, LSti2, RSti2); Stick
Sti3(8, LSti3, RSti3); Stick Sti4(9, LSti4, RSti4);
```

```
//Create the rigid bodies
RigidBody RBody1(1); RigidBody RBody2(2); RigidBody RBody3(3);
RigidBody RBody4(4); RigidBody Rbody5(5); ...
```

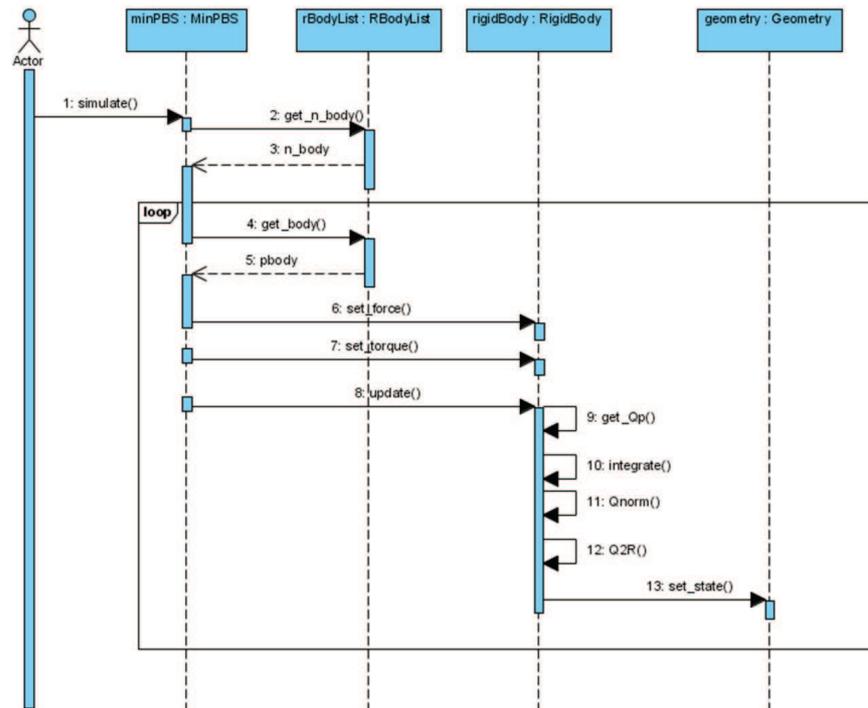


Figura B.10: Diagrama de secuencia de la función simulate .

```

//Create the list of geometries
GeometryList GeoList;

CollisionDetection MDC; ContactForce MFC(2.0, 1,0);
contact_force_data pCFD[10];

RbodyList RBLList; MinPBS PBS;
  
```

Posteriormente dentro del *main* se definen las propiedades y condiciones iniciales de los objetos geométricos y los cuerpos rígidos. A continuación se agregan los objetos a sus respectivas listas y se le asignan estas a los diferentes módulos. Finalmente se inicializa la ventana de gráficos utilizando las funciones del OpenGL y el glut.

```

int main(int argc, char** argv) {
    double xr1[7]={0.0, 0.0, -4.0, 1.0, 0.0, 0.0, 0.0};
    double vr1[6]={0.0, 0.0, 0.0, 0.0, 0.0, 0.0};
    Plane1.change_position_euler_parameters(xr1);
    Plane1.SetVel(vr1);

    //Define the parameters and initial conditions of the bodies
  
```

```

double xr2[7]={0.0, -1.0, 5.0, 1.0, 0.0, 0.0, 0.0};
double vr2[6]={0.0, 0.0, 0.0, 0.0, 2.0, 0.0};
RBody1.set_parameters(m_1, I_1, xr2, vr2);
RBody1.set_geometry(&Sph1);
Sph1.change_position_euler_parameters(xr2);
Sph1.SetVel(vr1);
...

//Add geometries to the list
GeoList.add(&Plane);
GeoList.add(&Sph1);
GeoList.add(&Sph2);
GeoList.add(&Sph3);
...

RBList.add(&RBody1);
RBList.add(&RBody2); RBList.add(&RBody3); RBList.add(&RBody4); ...
//Define the list of geometries
MDC.set_list_geo(&GeoList);
MFC.set_list_geo(&GeoList);
PBS.set_contact_data(pCFD);
PBS.set_rbody_list(&RBList);

//OpenGL functions
glutInit(&argc, argv);
glutInitDisplayMode (GLUT_DOUBLE | GLUT_RGB);
glutInitWindowSize (500, 500);
glutInitWindowPosition (100, 100);
glutCreateWindow ("MinPBS");
init();
glutDisplayFunc(display);
glutReshapeFunc(reshape);
glutKeyboardFunc(keyboard);
glutSpecialFunc(keybspecial);
glutIdleFunc(IdleDisplay);
glutMainLoop();
}

```

En cada ciclo del bucle de gráficos se extraen las nuevas posiciones de los cuerpos y se actualiza la imagen en la ventana de gráficos.

```

void display(void) {
    //Compute collision force and simulation
    MDC.compute_collision();
    MFC.compute_force(pCFD);
    t=t+dt;
}

```

```
PBS.simulate(t,dt);

//Get position and orientation in angle-axis representation
Plane1.get_current_position(X1);
Plane1.angle_axis(ori1);

Sph1.get_current_position(X2);
Sph1.angle_axis(ori2);
... ..

//Draw figures
glPushMatrix();
    glTranslatef(X1[0], X1[1], X1[2]);
    glRotatef(ori1[0]*180/PI, ori1[1], ori1[2], ori1[3]);
    glColor3f(0,.5,0);
    gPlane.draw();
glPopMatrix();

glPushMatrix();
    glTranslatef(X2[0], X2[1], X2[2]);
    glRotatef(ori2[0]*180/PI, ori2[1], ori2[2], ori2[3]);
    glColor3f(0,.5,0);
    gSph1.draw(RSph1,50,false);
glPopMatrix();
... ..

glutSwapBuffers();

}
```

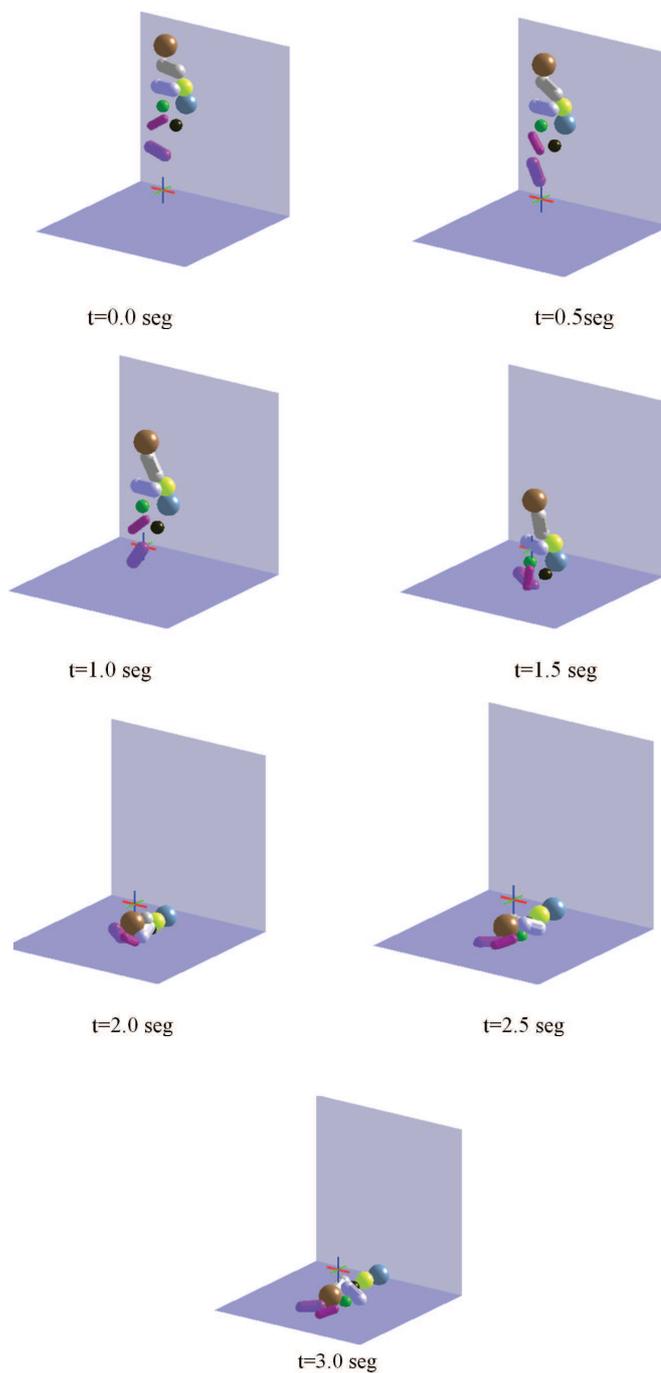


Figura B.11: Simulación de varios objetos cayendo desde una altura.

Apéndice C

Detección de Colisiones

C.1. Detección de colisiones

La detección de colisiones es un problema estrictamente geométrico. Los algoritmos utilizan la información sobre la geometría de los objetos y la posición de los mismos para determinar si están en contacto y como lo están. Los módulos del componente de detección de colisiones son (Erleben et al., 2005):

1. Modulo de Detección de Colisiones Basto
2. Modulo de Detección de Colisiones
3. Modulo de Determinación de Contactos

Modulo de Detección de Colisiones Basto. Este modulo es básicamente un algoritmo de detección de colisiones que determina las posibles intersecciones entre todos los objetos de la escena, esto implica encontrar las $n(n-1)/2$ posibles intersecciones de los n objetos de la escena. El propósito de este Modulo es disminuir el número de detección de colisiones finas entre pares de objetos. Esto permite reducir considerablemente la carga computacional de la detección de colisiones fina. Para determinar la proximidad, los objetos generalmente se aproximan por Volúmenes Evolventes. Existen diferentes volúmenes evolventes, tales como: Cajas Evolventes (BB), Cajas Evolvente con Ejes Alineados (AABB) y Esferas Evolventes (BS).

Modulo de Detección de Colisiones Fino. Este modulo examina pares de objetos con el objetivo de determina si están en contacto o no. Los algoritmos utilizados por este modulo son generalmente capaces de entregar los puntos de contacto, las normales del contacto, las características penetrantes, etc.

Modulo de Determinación de Contactos. El modulo de determinación de contactos calcula las regiones de contacto entre los objetos que se tocan o penetran. En términos matemáticos, la región de contacto es la intersección de dos objetos. El punto de contacto se representa por dos características, una de cada poliedro, y se calcula usualmente como el punto mas cercano entre las dos características. Tal representación es llamada Contacto Principal. Para objetos poliédricos se tiene tres diferentes Características, vértice (V), arista (A), y cara (C). Debido a la simetría, solo

se tiene 6 diferentes tipos de contactos principales: (V,V), (V,A), (V, C), (A, A), (A, C), y (C, C). A veces no solamente se define un solo punto de contacto, sino que existen varios o una región donde existe una infinidad de contactos. Una región de contacto es definida como la intersección de dos objetos. Se pueden tener múltiples regiones entre dos objetos las regiones no tiene que ser necesariamente planas. Para objetos poliédricos, la región de contacto es una área poligonal, la cual consiste de:

- Puntos
- Segmentos de línea
- Polígonos cerrados (tal vez con hoyos)

Para poliedros convexos, las regiones de contacto son particularmente simples. Puede ser un punto, un segmento de línea, o un polígono cerrado sin hoyos. Esto indica que podemos representar las regiones de contactos por los vértices de las regiones poligonales. En la figura, la región de contacto consiste de contactos principales $PC_1(V_A, C_B)$, $PC_2(A_A, A_B)$, $PC_3(C_A, V_B)$ y $PC_4(A_A, A_B)$.

Determinación de Contactos para el Cálculo de Fuerzas Penalizantes En esta tesis se utiliza la formulación basada en penalizaciones para calcular las fuerzas de contacto entre los cuerpos. Una fuerza penalizante tiene dos componentes una que es proporcional a la penetración y otra que es proporcional a la velocidad de penetración. Considerando lo anterior se define un contacto C entre dos objetos mediante los siguientes parámetros:

- Un punto p_1 en el objeto 1
- Un punto p_2 en el objeto 2
- Una normal de contacto n
- La profundidad de penetración d del contacto

C.2. Cálculo de la Distancia Mínima entre Elementos Geométricos

En esta sección se describe los fundamentos matemáticos para determinar la distancia mínima entre dos elementos geométricos y los puntos más cercanos entre ellos. Algunas de las técnicas presentadas en esta sección se han recopilado de (Ericson, 2005) y (Schneider and Eberly, 2003). Los elementos geométricos considerados aquí son: el punto, la línea, el plano, el triángulo, el círculo, y el cubo.

Determinar la distancia mínima entre los elementos geométricos permite la determinación de contactos entre cada una de las primitivas presentadas en la anterior subsección.

Distancia Mínima entre un Punto y un Plano

Dado un plano π , definido por un punto \mathbf{p} y una normal \mathbf{n} , cualquier punto \mathbf{x} sobre el plano satisface la ecuación $\mathbf{n} \cdot (\mathbf{x} - \mathbf{p}) = 0$. Dado un punto \mathbf{q} en el espacio, el punto más cercano sobre plano es \mathbf{r} , que es la proyección ortogonal de \mathbf{q} sobre el plano, esto es $\mathbf{r} = \mathbf{q} - d\mathbf{n}$ para algún valor de t .

La distancia mínima entre el punto y el plano viene dada por:

$$d = \frac{\mathbf{n} \cdot (\mathbf{q} - \mathbf{p})}{\mathbf{n} \cdot \mathbf{n}} \quad (\text{C.1})$$

El vector \mathbf{n} es un vector unitario alineado con el eje z_B del plano, por lo tanto:

$$d = \mathbf{n} \cdot (\mathbf{q} - \mathbf{p}) \quad (\text{C.2})$$

Distancia Mínima entre un Punto y un Segmento de Línea

Sea un punto \mathbf{p} y una línea parametrizada $\mathbf{l}(t) = \mathbf{b} + t\mathbf{m}$, donde \mathbf{b} es un punto en la línea, \mathbf{m} es la dirección de la línea, y $t \in [0, 1]$. Los puntos extremos de la línea son \mathbf{b} y $\mathbf{b} + \mathbf{m}$.

El punto más cercano a \mathbf{p} sobre la línea, es la proyección de \mathbf{p} sobre la línea, $\mathbf{q} = \mathbf{b} + t_0\mathbf{m}$, donde

$$t_0 = \frac{\mathbf{m} \cdot (\mathbf{p} - \mathbf{b})}{\mathbf{m} \cdot \mathbf{m}} \quad (\text{C.3})$$

la distancia \mathbf{p} a la línea es:

$$d = \|\mathbf{p} - (\mathbf{b} + t_0\mathbf{m})\| \quad (\text{C.4})$$

si $t_0 \leq 0$ entonces el punto más cercano entre la línea y el punto \mathbf{p} es \mathbf{b} . Para $t_0 > 0$, la proyección $\mathbf{b} + t_0\mathbf{m}$ es el punto más cercano. Finalmente, si $t_0 > 1$, entonces el punto más cercano sobre el segmento de línea al punto \mathbf{p} es $\mathbf{b} + \mathbf{m}$. La distancia desde \mathbf{p} al segmento de línea es:

$$d = \begin{cases} \|\mathbf{p} - \mathbf{b}\| & \text{si } t_0 \leq 0, \\ \|\mathbf{p} - (\mathbf{b} + t_0\mathbf{m})\| & \text{si } 0 < t_0 < 1, \\ \|\mathbf{p} - (\mathbf{b} + \mathbf{m})\| & \text{si } t_0 \geq 1. \end{cases} \quad (\text{C.5})$$

Distancia Mínima entre un punto y un marco

Para determinar la distancia mínima entre un punto y un marco de n vértices, se revisa la distancia entre cada una de las líneas del marco y el punto y se elige aquella que sea menor a las demás.

Distancia Mínima entre un punto y un Triángulo

Para explicar el algoritmo para determinar la distancia mínima y los puntos más cercanos entre un punto y un triángulo primero se introducirán dos conceptos: las *coordenadas baricéntricas* y la *región de Voronoi*.

Coordenadas baricéntricas. Las coordenadas baricéntricas parametrizan el espacio a través de una combinación ponderada de un conjunto de puntos de referencia. Una aplicación típica de las coordenadas baricéntricas es parametrizar triángulos (o los planos de los triángulos). Considérese un triángulo ABC especificado por tres puntos no colineales A , B y C . Cualquier punto P en el plano de estos puntos puede ser expresado únicamente como:

$$P = uA + vB + wC \quad (\text{C.6})$$

para algunas constantes que satisfagan $u + v + w = 1$. La tripleta (u, v, w) corresponde a las coordenadas baricéntricas del punto. En general, un punto está dentro de un triángulo si sus coordenadas baricéntricas cumplen que $0 \leq u, v, w \leq 1$. Si consideramos que $u = 1 - v - w$, tenemos que

$$P = (1 - v - w)A + vB + wC = A + v(B - A) + w(C - A) \quad (\text{C.7})$$

En la última formulación, los dos vectores independientes con dirección AB y AC forman un sistema de coordenadas con origen en A , permitiendo que cualquier punto P en el plano esté parametrizado en únicamente términos de v y w .

Las coordenadas baricéntricas del punto P pueden ser calculadas como los radios de las áreas de los triángulos PBC , PCA , PAB y con respecto a el área del triángulo ABC . Utilizando las áreas con signo de los triángulos, esas expresiones son válidas para puntos fuera del triángulo. Las coordenadas baricéntricas pueden estar dadas como:

$$u = \frac{A_{PBC}}{A_{ABC}} \quad (\text{C.8})$$

$$u = \frac{A_{PCA}}{A_{ABC}} \quad (\text{C.9})$$

$$u = \frac{A_{PAB}}{A_{ABC}} \quad (\text{C.10})$$

Debido a que los factores constantes se cancelan, cualquier función proporcional a el área del triángulo se puede utilizar para calcular esos radios. En particular, la magnitud del producto cruz de las dos aristas del triángulo se puede utilizar. El signo correcto del área se obtiene utilizando el producto punto con la normal de ABC . Por ejemplo, el área con signo del triángulo PBC se calcula como:

$$A_{PBC} = [(\mathbf{b} - \mathbf{p}) \times (\mathbf{c} - \mathbf{p})] \cdot \frac{(\mathbf{b} - \mathbf{a}) \times (\mathbf{c} - \mathbf{a})}{\|(\mathbf{b} - \mathbf{a}) \times (\mathbf{c} - \mathbf{a})\|} \quad (\text{C.11})$$

De igual manera se obtienen las otras dos áreas de los triángulos.

Las coordenadas baricéntricas dividen el plano de el triángulo ABC en siete regiones dependiendo del signo de las componentes de las coordenadas.

Región de Voronoi. Un concepto importante en la detección de colisiones es el concepto de las Regiones de Voronoi. Dado un conjunto S de puntos en el plano, la región de Voronoi de un punto P en S se define como el conjunto de puntos en el

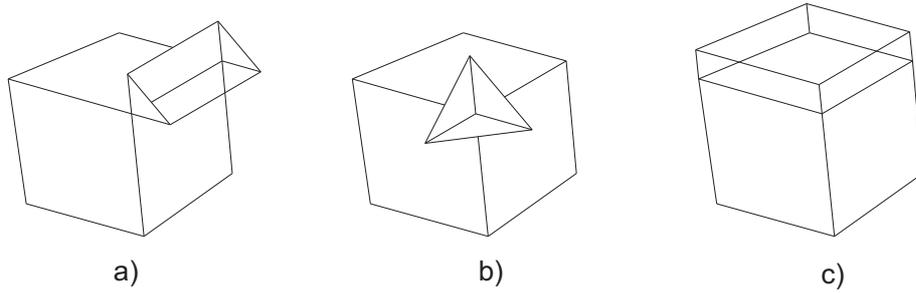


Figura C.1: Tres tipos de regiones de Voronoi de un cubo tridimensional. c) región de una arista, b) región de un vértice c) región de una cara.

plano más cercanos a P que cualquiera otro conjunto de puntos en S . En la detección de colisiones entre polígonos y poliedros un concepto asociado a las regiones de Voronoi son las características geométricas. Un polígono está compuesto por características geométricas, tales características son los vértices y aristas que lo conforman. Por otra parte un Poliedro, está compuesto también de características geométricas como lo son los vértices y aristas y además las caras que lo conforman.

Dado un poliedro P , la región de Voronoi de una característica geométrica F (un vértice, una arista o una cara) es el conjunto de puntos en el espacio más cercanos a F que ha cualquier otra característica de P . Por ejemplo para el caso de un polígono en el plano en la siguiente figura se muestran las regiones de Voronoi de las características de un triángulo.

Por otro lado, en la siguiente figura se presentan tres tipos de regiones de Voronoi correspondientes a las características de un cubo.

Dado un poliedro convexo P , todos los punto en el espacio exterior a pueden ser clasificados como pertenecientes a una región de Voronoi correspondiente a un vertice, arista o cara de P , con una frontera que pertenece a una de las regiones. Debido a que la región de Voronoi crea un particionamiento del espacio exterior al poliedro, se puede utilizar, por ejemplo, para determinar el punto más cercano sobre el cuerpo convexo a un punto Q en el espacio. Esto se puede hacer revisando cada una de las regiones hasta encontrar una donde Q este contenido. La proyección de Q en tal característica es el punto más cercano a Q .

Teorema de Voronoi. Dados un par de poliedros que no se interceptan A y B , considérese que \mathbf{a} y \mathbf{b} son los puntos más cercanos entre la característica F_a de A , y la característica F_b de B , respectivamente. Si \mathbf{a} y \mathbf{b} son los puntos más cercanos entre A y B entonces $\mathbf{a} \in V(F_b)$ (es decir, pertenece a la región de Voronoi de F_b) y $\mathbf{b} \in V(F_a)$.

Regiones de Voronoi de un Triángulo. Para que un punto P este dentro la región de Voronoi de la arista AB , el punto P debe estar fuera o sobre la arista AB , esta condición se cumple cuando el valor de la área A_{PAB} es menor que 0, además para estar en la región de Voronoi debe estar dentro de los semi-espacios positivos definidos

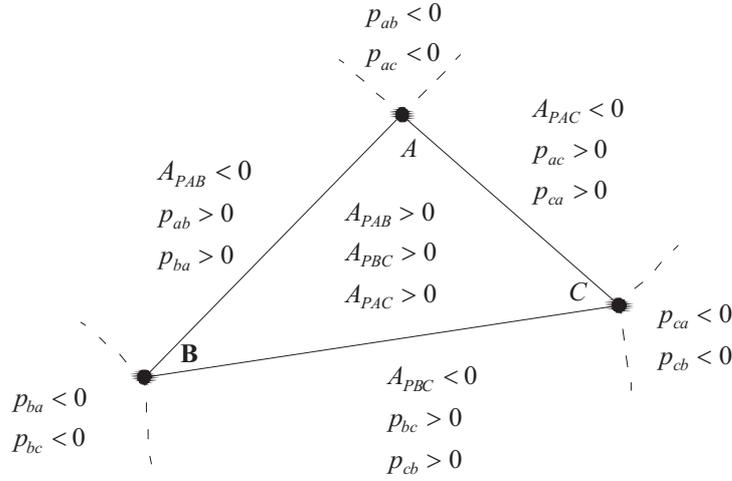


Figura C.2: Condiciones en las diferentes regiones de Voronoi.

por los planos $(\mathbf{p} - \mathbf{a}) \cdot (\mathbf{b} - \mathbf{a}) = 0$ y $(\mathbf{p} - \mathbf{b}) \cdot (\mathbf{a} - \mathbf{b}) = 0$. A continuación se definen algunas variables auxiliares para determinar la región de Voronoi en que se encuentra el punto de estudio:

$$p_{ab} = (\mathbf{p} - \mathbf{a}) \cdot (\mathbf{b} - \mathbf{a}) \quad (\text{C.12})$$

$$p_{ba} = (\mathbf{p} - \mathbf{b}) \cdot (\mathbf{a} - \mathbf{b}) \quad (\text{C.13})$$

$$p_{ac} = (\mathbf{p} - \mathbf{a}) \cdot (\mathbf{c} - \mathbf{a}) \quad (\text{C.14})$$

$$p_{ca} = (\mathbf{p} - \mathbf{c}) \cdot (\mathbf{a} - \mathbf{c}) \quad (\text{C.15})$$

$$p_{bc} = (\mathbf{p} - \mathbf{c}) \cdot (\mathbf{c} - \mathbf{b}) \quad (\text{C.16})$$

$$p_{cb} = (\mathbf{p} - \mathbf{c}) \cdot (\mathbf{b} - \mathbf{c}) \quad (\text{C.17})$$

En la figura C.2 se expresan las condiciones que definen cada una de las regiones de Voronoi.

El algoritmo para determinar el punto sobre el triángulo más cercano al punto \mathbf{P} en el espacio consiste en ir revisando cada una de las regiones de Voronoi y encontrar aquella en la que este contenido el punto \mathbf{P} . La manera más eficiente es comenzar revisando cada uno de los vértices, posteriormente las aristas y finalmente la cara central.

Distancia mínima entre un Punto y un Círculo

Considérese un círculo tridimensional representado por un centro \mathbf{c} , un radio R , y un plano que contiene al círculo, $\mathbf{n} \cdot (\mathbf{x} - \mathbf{c})$ donde \mathbf{n} es un vector unitario normal al plano. Si \mathbf{u} y \mathbf{v} son vectores unitarios y forman con \mathbf{n} un sistema de coordenadas ortogonal, entonces el círculo:

$$\mathbf{x} = \mathbf{c} + R(\cos \theta \mathbf{u} + \sin \theta \mathbf{v}) \tag{C.18}$$

$$\mathbf{x} = \mathbf{c} + R\mathbf{w}(\theta) \tag{C.19}$$

para ángulos $\theta \in [0, 2\pi)$. Nótese que $\|\mathbf{x} - \mathbf{c}\| = R$, por lo tanto los valores \mathbf{x} son equidistantes a \mathbf{c} . Además, $\mathbf{n} \cdot (\mathbf{x} - \mathbf{c}) = 0$ debido a que \mathbf{u} y \mathbf{v} son perpendiculares a \mathbf{n} , por lo tanto \mathbf{x} se encuentra en el plano.

Para ángulos $\theta \in [0, 2\pi)$, el cuadrado de la distancia desde un punto específico \mathbf{p} al correspondiente punto en el círculo es

$$F(\theta) = \|\mathbf{c} + R\mathbf{w}(\theta) - \mathbf{p}\|^2 \tag{C.20}$$

$$F(\theta) = R^2 + \|\mathbf{c} - \mathbf{p}\|^2 + 2R(\mathbf{c} - \mathbf{p}) \cdot \mathbf{w} \tag{C.21}$$

El problema es minimizar $F(\theta)$ buscando θ_0 de tal manera que $F(\theta_0) \leq F(\theta)$ para todo $\theta \in [0, 2\pi)$. Realizando la derivación de F para encontrar las condiciones para obtener un mínimo de la función, se observa que si se define un vector \mathbf{q} como la proyección de \mathbf{p} sobre el plano, entonces $\mathbf{q} - \mathbf{c} = \mathbf{p} - \mathbf{c} - (\mathbf{n}(\mathbf{p} - \mathbf{c}))\mathbf{n}$ el vector $\mathbf{w}(\theta)$ el vector $(\mathbf{q} - \mathbf{c})$. El punto más cercano en el círculo al punto \mathbf{p} es:

$$\mathbf{x} = \mathbf{c} + R \frac{\mathbf{q} - \mathbf{c}}{\|\mathbf{q} - \mathbf{c}\|} \tag{C.22}$$

Asumiendo que $\mathbf{q} \neq \mathbf{c}$. La distancia mínima es entonces:

$$d = \|\mathbf{p} - \mathbf{x}\| \tag{C.23}$$

Distancia Mínima entre un punto y un Cubo

Para este cálculo lo que se hace es utilizar el Teorema de Voronoi. El algoritmo consiste en revisar cada una de las regiones de Voronoi correspondientes al cubo y determinar en cual se encuentra el punto. En el algoritmo utilizado aquí primero se revisa cada plano correspondiente a una cara. Si el punto se encuentra en la parte positiva del espacio dividido por el plano, entonces se hace una revisión de las regiones de Voronoi correspondientes a esa cara y cada una de sus características. Primero se revisa la cara y posteriormente las aristas y los vértices. Determinar si el punto se encuentra dentro de la región de Voronoi de una cara rectangular es sencillo, ya que solo hay que revisar si las coordenadas no rebasen dos umbrales (uno positivo y uno negativo) correspondientes al ancho y largo de la cara. De igual manera sucede con las aristas y vértices.

Distancia Mínima entre un Segmento de Línea y un Plano

Considérese un plano π , definido por un punto \mathbf{p} y una normal \mathbf{n} . Considérese además un segmento de línea que esta definido por los puntos de sus extremos, \mathbf{q}_1 y \mathbf{q}_2 .

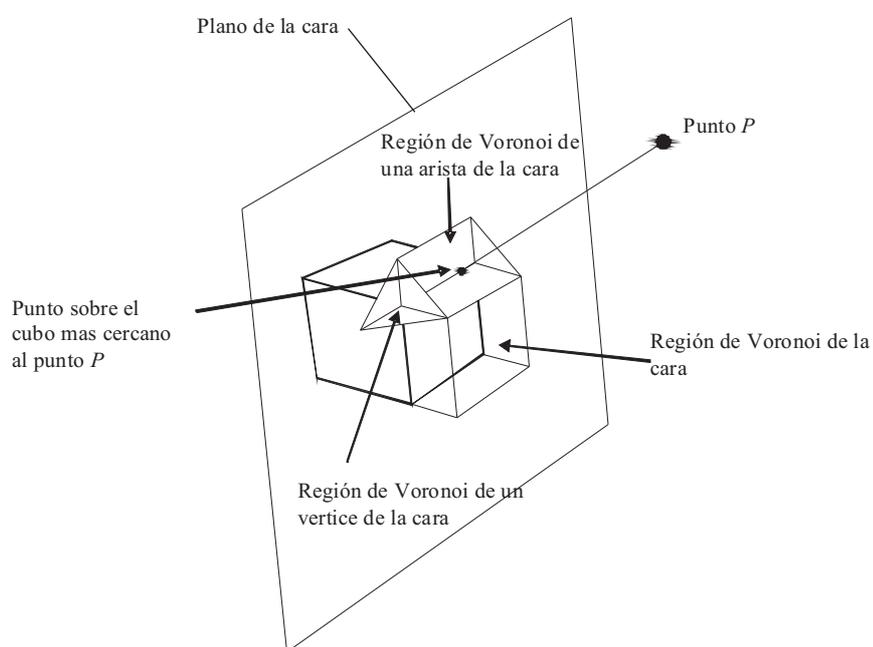


Figura C.3: Distancia mínima entre un punto y un cubo.

La distancia mínima de los puntos extremos de la línea esta dada por:

$$d_1 = \frac{\mathbf{n} \cdot (\mathbf{q}_1 - \mathbf{p})}{\mathbf{n} \cdot \mathbf{n}} \quad (\text{C.24})$$

$$d_2 = \frac{\mathbf{n} \cdot (\mathbf{q}_2 - \mathbf{p})}{\mathbf{n} \cdot \mathbf{n}} \quad (\text{C.25})$$

El vector \mathbf{n} es un vector unitario alineado con el eje del plano. La distancia mínima entre la línea y el plano es:

$$d = \begin{cases} \|\mathbf{n} \cdot (\mathbf{q}_1 - \mathbf{p})\| & \text{si } d_1 < d_2, \\ \|\mathbf{n} \cdot (\mathbf{q}_2 - \mathbf{p})\| & \text{si } d_1 > d_2, \end{cases} \quad (\text{C.26})$$

en caso de que $d_1 = d_2$, entonces la distancia mínima es igual alguna de las dos variables.

Distancia Mínima entre Dos Segmentos de Línea

Considérese dos segmentos de línea dadas por las siguientes expresiones:

$$\mathbf{l}_0 = \mathbf{b}_0 + s\mathbf{m}_0 \quad (\text{C.27})$$

$$\mathbf{l}_1 = \mathbf{b}_1 + t\mathbf{m}_1 \quad (\text{C.28})$$

con $s \in [0, 1]$ y $t \in [0, 1]$.

La distancia mínima se calcula ubicando los valores \bar{s} y \bar{t} que corresponden a los puntos más cercanos entre los segmentos de línea.

Para encontrar la distancia mínima entre los segmentos de línea se define primero una función del cuadrado de la distancia:

$$Q(s, t) = |\mathbf{l}_0(s) - \mathbf{l}_1(t)|^2 \quad (\text{C.29})$$

La función anterior es cuadrática si se expresa en términos de s y t queda de la siguiente forma:

$$Q(s, t) = as^2 + 2bst + ct^2 + 2ds + 2et + f \quad (\text{C.30})$$

donde las variables auxiliares están dadas de la siguiente manera:

$$\begin{aligned} a &= \mathbf{m}_0 \cdot \mathbf{m}_0 \\ b &= -\mathbf{m}_0 \cdot \mathbf{m}_1 \\ c &= \mathbf{m}_1 \cdot \mathbf{m}_1 \\ d &= \mathbf{m}_0 \cdot (\mathbf{b}_0 - \mathbf{b}_1) \\ e &= -\mathbf{m}_1 \cdot (\mathbf{b}_0 - \mathbf{b}_1) \\ f &= (\mathbf{b}_0 - \mathbf{b}_1) \cdot (\mathbf{b}_0 - \mathbf{b}_1) \end{aligned}$$

si $ac - b^2 > 0$ los dos segmentos de línea no son paralelos y la grafica de Q es un paraboloide. Si $ac - b^2 = 0$ los dos segmentos de línea son paralelos y la grafica de Q es un cilindro parabólico.

Para encontrar los puntos más cercanos se debe de minimizar la función Q sobre un cuadrado unitario en el plano $s - t$. Debido a que Q es una función continua y diferenciable, el mínimo puede estar dentro del cuadrado unitario donde el gradiente es

$$\nabla Q = 2 \begin{bmatrix} as + bt + d \\ bs + ct + e \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (\text{C.31})$$

o puede estar en un punto sobre la frontera del cuadrado.

Alrededor del plano $s - t$ existen 8 regiones que representan los diferentes casos cuando los puntos más cercanos entre las dos líneas quedan fuera de los segmentos de línea. Mediante la exploración de ∇Q se puede determinar a qué caso corresponde la situación y así determinar la distancia mínima entre los dos segmentos (Schneider and Eberly, 2003).

Distancia Mínima entre una Línea y un Marco

Para determinar la distancia mínima entre un segmento de línea y un marco de n vértices, se revisa la distancia entre cada una de las líneas del marco y el segmento de línea y se elige aquella que sea menor a las demás.

Distancia mínima entre un segmento de línea y un triángulo

Los puntos más cercanos entre un segmento de línea y un triángulo no son necesariamente únicos. Cuando el segmento es paralelo al plano del triángulo puede existir un número infinito de puntos igualmente cercanos. Sin embargo, sin importar si el segmento es paralelo al plano o no siempre es posible encontrar una par de puntos tales que la distancia mínima puede ocurrir en a) entre un punto extremo del segmento y el interior del triángulo b) entre el segmento y una arista del triángulo.

El caso a) puede ocurrir solo si la proyección de un extremo del segmento de línea sobre el plano (del triángulo) está contenido dentro del triángulo. Sin embargo, aun cuando un extremo del segmento se proyecta dentro del triángulo puede ser que un punto de una arista sea el más cercano al segmento de línea. Por lo tanto, los puntos más cercanos pueden ser encontrados calculando el par de puntos más cercanos de las siguientes entidades:

- El segmento PQ y la arista AB
- El segmento PQ y la arista BC
- El segmento PQ y la arista CA
- El punto P y el plano del triángulo (cuando P se proyecta dentro de ABC)
- El punto Q y el plano del triángulo (cuando Q se proyecta dentro de ABC)

Finalmente se seleccionan los puntos más cercanos.

Distancia Mínima entre una línea y un círculo

Considérese un círculo tridimensional representado por un centro \mathbf{c} , un radio R , y un plano que contiene al círculo, $\mathbf{n} \cdot (\mathbf{x} - \mathbf{c}) = 0$ donde \mathbf{n} es un vector unitario normal al plano. Si \mathbf{u} y \mathbf{v} son vectores unitarios y forman con \mathbf{n} un sistema de coordenadas ortogonal, entonces el círculo:

$$\mathbf{x} = \mathbf{c} + R(\cos \theta \mathbf{u} + \sin \theta \mathbf{v}) \quad (\text{C.32})$$

$$\mathbf{x} = \mathbf{c} + R\mathbf{w}(\theta) \quad (\text{C.33})$$

para ángulos $\theta \in [0, 2\pi)$. Nótese que $\|\mathbf{x} - \mathbf{c}\| = R$, por lo tanto los valores \mathbf{x} son equidistantes a \mathbf{c} . Además, $\mathbf{n} \cdot (\mathbf{x} - \mathbf{c}) = 0$ debido a que \mathbf{u} y \mathbf{v} son perpendiculares a \mathbf{n} , por lo tanto \mathbf{x} se encuentra en el plano.

Para ángulos $\theta \in [0, 2\pi)$, el cuadrado de la distancia desde un punto específico \mathbf{p} al correspondiente punto en el círculo es

$$F(\theta) = \|\mathbf{c} + R\mathbf{w}(\theta) - \mathbf{p}\|^2 \quad (\text{C.34})$$

$$F(\theta) = R^2 + \|\mathbf{c} - \mathbf{p}\|^2 + 2R(\mathbf{c} - \mathbf{p}) \cdot \mathbf{w} \quad (\text{C.35})$$

Donde

$$\mathbf{w} = \frac{\mathbf{q} - \mathbf{c}}{|\mathbf{q} - \mathbf{c}|} \quad (\text{C.36})$$

Si $\mathbf{p} = \mathbf{b} + t\mathbf{m}$, entonces F es una función de t . El problema es minimizar F para toda t .

Al obtener la derivada de $F(\theta)$, igualarla a cero, y realizando operaciones algebraicas se obtiene la siguiente expresión:

$$(a_0 + a_1 t)^2 (c_0 + 2c_1 t + c_2 t^2) = R^2 (b_0 + b_1 t)^2 \quad (\text{C.37})$$

donde:

$$\begin{aligned} a_0 &= \mathbf{m} \cdot \mathbf{d} \\ a_1 &= \mathbf{m} \cdot \mathbf{m} \\ b_0 &= \mathbf{m} \cdot \mathbf{d} - (\mathbf{n} \cdot \mathbf{m})(\mathbf{n} \cdot \mathbf{d}) \\ b_1 &= \mathbf{m} \cdot \mathbf{m} - (\mathbf{n} \cdot \mathbf{m})^2 \\ c_0 &= \mathbf{d} \cdot \mathbf{d} - (\mathbf{n} \cdot \mathbf{d})^2 \\ c_1 &= b_0 \\ c_2 &= b_1 \end{aligned}$$

Se obtiene pues una ecuación cuadrática. Las raíces de esta ecuación se pueden obtener mediante un solucionador de polinomios estándar. Para cada raíz, se calcula el valor correspondiente de F . El valor mínimo de entre las raíces corresponde a la distancia mínima.

Distancia mínima entre una línea y un cubo

Primero se revisa si los dos extremos están en el semi-espacio positivo de los planos paralelos a cada una de las caras del cubo. Si se encuentra una cara que satisfaga la anterior condición, entonces se revisa la distancia mínima entre la línea y cada una de las características de la cara. Primero se revisa si la proyección del punto más cercano al plano está dentro de la cara, si esto es así se ha encontrado el punto más cercano entre la línea y el cubo. Si no, se comienza por revisar la cada una de las aristas. En cada arista se determinan los puntos más cercanos entre la arista y la línea, si el punto correspondiente a la línea se encuentra en la región de Voronoi de la arista entonces se termina la búsqueda y se han encontrado los puntos más cercanos. Si no, entonces se continúa realizando la misma prueba en las otras aristas. Si no se encuentran los puntos más cercanos en alguna de las líneas, entonces se continúa de manera análoga con los demás vértices. Si no se encuentran los puntos más cercanos se realiza el estudio en otra cara, hasta haberse revisado las 6 caras.

Distancia mínima entre un Plano y un Marco

Para determinar la distancia mínima entre un punto y un marco de vértices, se revisa la distancia entre cada uno de los vértices y el plano y se elige aquel punto que corresponda a la menor distancia como el punto más cercano al plano.

Distancia mínima entre un Plano y un Triángulo

De igual manera que en el caso anterior se revisa cada uno de los tres vértices del triángulo y se elige el que este más cercano al plano, solamente revisando su coordenada en el eje del marco de referencia asociado al plano.

Distancia mínima entre un Plano y un Círculo

Para determinar el punto más cercano al plano primero se calcula el vector que une los puntos más cercanos entre el plano y el centro del círculo. Posteriormente se calcula un vector unitario que tiene origen en el marco de referencia del círculo y que tiene la misma dirección (diferente sentido) al vector previamente calculado. Finalmente se calcula la proyección del vector sobre el marco de referencia del círculo y se multiplica ese vector proyectado por el radio del círculo.

Distancia mínima entre un Plano y un Cubo

De igual manera que en el caso del triángulo se revisa cada uno de los vértices del cubo y se elige el que este más cercano al plano, solamente revisando su coordenada en el eje del marco de referencia asociado al plano.

Distancia mínima entre dos Triángulos

Al igual que en el caso de determinar la distancia mínima entre un segmento y un triángulo, puede existir un número infinito de puntos igualmente cercanos entre ambos triángulos. Sin embargo los puntos más cercanos entre dos triángulos T_1 y T_2 pueden ser siempre encontrados de tal manera que un punto está en un límite del triángulo.

Los puntos más cercanos entre T_1 y T_2 pueden ocurrir en las aristas de cada triángulo o como un vértice de un triángulo y un punto interior del otro. El problema es calcular los puntos más cercanos entre todos los pares de aristas, una de cada triángulo, y los puntos más cercanos en el triángulo opuesto para cada vértice de cada triángulo (esto solamente cuando el vértice cae dentro del triángulo). Por todo se harán seis pruebas entre triángulo-vértice y 9 pruebas para arista-arista. De todos los pares de puntos encontrados se selecciona aquel que tiene la mínima distancia

Bibliografía

- Alba, O. Wenger, P. y P. J. (2005). Consistent kinetostatic indices for planar 3-dof parallel manipulators, application to the optimal kinematic inversion. In *Proceedings of the ASME DETC*, pages 765–774.
- Alvarez, C. (2008a). *Concepción y desarrollo de un vehículo submarino robótico de estructura paralela de geometría variable*. PhD thesis, Universidad Politécnica de Madrid, Madrid, España.
- Alvarez, C. (2008b). *Concepción y desarrollo de un vehículo submarino robótico de estructura paralela de geometría variable*. PhD thesis, Univesidad Politecnica de Madrid, Madrid, España.
- Amat, J., Escote, O., Frigola, M. and Giralt, X., and Hernansanz, A. (2006). Milana: a lowcost glider used for building a map of barcelona sea bed. In *Robotics and Automation in the Maritime Industries AUTOMAR*.
- Anderson, J. M. and Chhabra, N. K. (2002). Maneuvering and stability performance of a robotic tuna. *Integrative and Comparative Biology*, 42(1):118–126.
- Angeles, J. (2007). *Fundamentals of Robotic Mechanical Systems, Theory, Methods, and Algorithms*. Springer.
- Antonelli, G., Fossen, T. I., and Yoerger, D. R. (2008). Underwater Robotics. In Siciliano, B. and Khatib, O., editors, *Springer Handbook of Robotics*, chapter 44, pages 987–1008. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Bachmayer, R., Whitcomb, L., and Grosenbaugh, M. (2000). An accurate four quadrant nonlinear dynamical model for marine thrusters. *IEEE Journal of Oceanic Engineering*, 25(1):146 – 159.
- Baron L. y Bernier, G. (2001). The design of parallel manipulators of star topology under isotropic constraint. In *Proc. DETC ASME*.
- Bhattacharya, S., Hatwal, H., and Ghosh, A. (1997). An on-line parameter estimation scheme for generalized stewart platform type parallel manipulators. *Mechanism and Machine Theory*, 32(1):79 – 89.
- Bonev, I. y Ryu, J. (2001). A new approach to orientation workspace analysis of 6-dof parallel manipulators. *Trans. ASME, Journal of Mechanical Design*, 36(1):15–28.

- Bosscher, P. and Elbert-Upholf, I. (2003). A novel mechanism for implementing multiple collocated spherical joints. In *Proceedings IEEE International Conference on Robotics and Automation*, pages 336–341.
- Boyer, F., Chablat, D., Lemoine, P., and Wenger, P. (2009). The eel-like robot. In *Proceedings of the ASME IDETC/CIE 2009*.
- Bradley, A., Feezor, M., Singh, H., and Sorrell, F. (2001). Power systems for autonomous underwater vehicles. *IEEE Journal of Oceanic Engineering*, 26(4):526538.
- Briota, S., Arakelian, V., and Guàrdia, S. (2009). Paminsa: A new family of partially decoupled parallel manipulator. *Mechanism and Machine Theory*, 44(2):425–444.
- Caffaz, A., Caiti, A., Casalino, G., and Turetta, A. (2010). The hybrid glider/auv folaga. *Robotics Automation Magazine, IEEE*, 17(1):31–44.
- Callegari, M., Palpacelli, M.-C., and Principi, M. (2006). Dynamics modelling and control of the 3-rcc translational platform. *Mechatronics*, 16(10):589–605.
- Cavallo, E. (2003). *Parallel robotic system design for the steering and guidance mechanism of an autonomous underwater vehicle*. PhD thesis, Università degli studi di Genova, Genoa, Italy.
- Cavallo, E. and Michelini, R. (2004). A robotic equipment for the guidance of a vectored thruster auv. volume 1, pages 1–6.
- Chablat, D. y Wenger, P. (2003). Architecture optimization of a 3-dof parallel mechanism for machining applications, the orthoglide. *IEEE Transactions on Robotics and Automation*, 19(3):403–410.
- Chablat, D. Wenger, P. C. S. y. A. J. (2002). The isoconditioning loci of planar three-dof parallel manipulators. In *Proc. DETC ASME*.
- Chablat, D. Wenger, P. y. A. J. (1998a). The isoconditioning loci of a class of closed-chain manipulators. In *Proc. IEEE International Conference on Robotics and Automation*, pages 1970–1976.
- Chablat, D. Wenger, P. y. A. J. (1998b). Working modes and aspects in fully-parallel manipulators. In *Proc. IEEE International Conference on Robotics and Automation*, pages 1964–1969.
- Chen, I., Li, H., and Cathala, A. (1999). Design and simulation of amoebot a metamorphic underwater vehicle. In *Proceedings of the International Conference of Robotics and Automation*.
- D. Bernier, J. M. C. and Li, X. (1995). A new parallel structure with six degree of freedom. In *9th World Congress on the Theory of Machines and Mechanism*.
-

- Damus, R., Manley, J., Dasset, S., Morash, J., and Chryssostomidis, C. (2002). Design of an inspection class autonomous underwater vehicle. In *OCEANS '02 MTS/IEEE*, volume 1, pages 180 – 185 vol.1.
- Davis, Russ E.; Eriksen, C. C. and Jones, C. (2002). Autonomous buoyancy-driven underwater gliders. *The Technology and Applications of Autonomous Underwater Vehicles*. G.Griffiths, ed., London, England.
- DeBitetto, P. (1995). Fuzzy logic for depth control of unmanned undersea vehicles. *IEEE Journal of Oceanic Engineering*, 20(3):242 – 248.
- DeNovi, G., Melchiorri, C., García, J., Sanz, P., Ridao, P., and Oliver, G. (2010). A new approach for a reconfigurable autonomous underwater vehicle for intervention. *IEEE Aerospace and Electronic Systems Magazine*, 25(11):32–36.
- Dasset, S., Damus, R., Hover, F., Morash, J., and Polidoro, V. (2005). Closer to deep underwater science with odyssey iv class hovering autonomous underwater vehicle (hauv). In *IEEE Oceans 2005 - Europe*, volume 2, pages 758 – 762.
- Di Gregorio, R. (2006). Dynamic model and performances of 2-dof manipulators. *Robotica*, 24(1):51–60.
- Di Gregorio, R. y Parenti-Castelli, V. (2002). Dynamic performance indices for 3-dof parallel manipulators. In *Proc. Advances in Robot Kinematics*, pages 11–20.
- Doty, K. Melchiorri, C. y. B. C. (1993). A theory of generalized inverse applied to robotics. *Int. J. Robot. Res.*, 12(1):1–19.
- Dudek, G., Giguere, P., Prahacs, C., Saunderson, S., Sattar, J., Torres-Mendez, L.-A., Jenkin, M., German, A., Hogue, A., Ripsman, A., Zacher, J., Milios, E., Liu, H., Zhang, P., Buehler, M., and Georgiades, C. (2007). Aqua: An amphibious autonomous robot. *Computer*, 40(1):46 –53.
- Ericson, C. (2005). *Real Time Collision Detection*. Elsevier.
- Erleben, K., Sporning, J., Henriksen, K., and Dohlmann, H. (2005). *Physics-based animation*. Charles River Media.
- Evan, J., Redmond, P., Plakas, C., Hamilton, K., and Lane, D. (2003). Autonomous docking for intervention-aufs using sonar and video-based real-time 3d pose estimation. volume 4, pages 2201 – 2210.
- Fossen, T. (1991). *Nonlinear Modeling and Control of Underwater Vehicles*. PhD thesis, Norwegian University of Science and Technology, Trondheim.
- Fossen, T. (2002). *Marine Control Systems. Guidance, Navigation, and Control of Ships, Rigs and Underwater Vehicles*. Marine Cybernetics.
-

- Fossen, T. and Sagatun, S. (1991). Adaptive control of nonlinear underwater robotic systems. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1687–1695.
- Goheen, K. and Jeffery, R. (1990). Multivariable self-tuning autopilots for autonomous and remotely operated underwater vehicles. *IEEE Journal of Oceanic Engineering*, 15(3):144–151.
- Gosselin, C. (1990a). Dexterity indices for planar and spatial robotic manipulators. In *IEEE International Conference on Robotics and Automation*, pages 650–655 vol.1.
- Gosselin, C. (1996). Parallel computational algorithms for the kinematics and dynamics of planar and spatial parallel manipulators. *ASME Journal of Dynamic Systems, Measurement, and Control*, 118(1):22–28.
- Gosselin, C. and Angeles, J. (1990). Singularity analysis of closed-loop kinematic chains. *Robotics and Automation, IEEE Transactions on*, 6(3):281–290.
- Gosselin, C. y A. J. (1990b). Kinematic inversion of parallel manipulators in the presence of incompletely specified tasks. *ASME J. Mechanical Design*, 112(4):494–500.
- Gosselin, C. y Angeles, J. (1991). A global performance index for the kinematic optimization of robotic manipulators. *J. Mech. Des.*, 113(3):220–226.
- Graver, J. (2005). *Underwater Gliders: Dynamics, Control and Design*. PhD thesis, Princeton University, USA.
- Guo, J., Chiu, F.-C., and Huang, C.-C. (2003). Design of a sliding mode fuzzy controller for the guidance and control of an autonomous underwater vehicle. *Ocean Engineering*, 30(16):2137–2155.
- Guo, J. and Huang, S. (1996). Adaptive control of nonlinear underwater robotic systems. In *Proceedings of the Symp. on Autonomous Underwater Vehicle Technology*, pages 285–289.
- Healey, A. and Lienard, D. (1993). Multivariable sliding mode control for autonomous diving and steering of unmanned underwater vehicles. *IEEE Journal of Oceanic Engineering*, 18(3):327–339.
- Hillson, R. and Jones, C. (1989). Evolution of an auv mission simulation testbed. In *Unmanned Untethered Submersible Technology, 1989. Proceedings of the 6th International Symposium on*, pages 462–470.
- Huang, T. y Whitehouse, D. (1998). Local dexterity, optimal architecture and optimal design of parallel machine tools. *Ann. CIRP*, 47(1):347–351.
- Huang, T. Li, M. L. Z. C. D. y W. D. (2004). Optimal kinematic design of 2-dof parallel manipulators with well-shaped workspace bounded by a specified conditioning index. *IEEE Transactions on Robotics and Automation*, 20(3):538–543.
-

- Innocenti, C. and Parenti-Castelli, V. (1992). Direct kinematics of the 6-4 fully parallel manipulator with position and orientation uncoupled. In Tzafestas, S. G., editor, *Robotic Systems: Advanced Techniques and Applications*, pages 3–10. Kluwer Academic Publishers, London.
- Innocenti, M. and Campa, G. (1999). Robust control of underwater vehicles: Sliding mode vs. lmi synthesis. In *LMI Synthesis, American Controls Conference*.
- Iwasaki, M., Akizono, J., Takahashi, H., Umetani, T., Nemoto, T., Azakura, O., and Asayama, K. (1987). Development on aquatic walking robot for underwater inspection. *Report of the Port and Harbour Research Institute*, 26(5):393–422.
- Khalil, W. and Guegan, S. (2004). Inverse and direct dynamic modeling of gough-stewart robots. *IEEE Transactions on Robotics*, 20(4):754 – 761.
- Khatib, O. y Burdick, J. (1987). Optimization of dynamics in manipulator design: The operational space formulation. *The International Journal of Robotics and Automation*, 2(2):90–98.
- Kim, E. and Yourn, Y. (2004). Design and dynamic analysis of fish robotuna. In *Proceedings of the IEEE International Conference on Robotics and Automation, New Orleans, USA*.
- Kim, S. y Ryu, J. (2003). New dimensionally homogeneous jacobian matrix formulation by three end-effector points for optimal design of parallel manipulators. *IEEE Transactions on Robotics and Automation*, 19(4):731–737.
- Kinsey, J., Yoerger, D., Jakuba, M., Camilli, R., Fisher, C., and Christopher, R. (2011). Assessing the deepwater horizon oil spill with the sentry autonomous underwater vehicle. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 261 – 267.
- Klein, C. y Blaho, B. (1987). Dexterity measures for the design and control of kinematically redundant manipulators. *International Journal of Robotics Research*, 6(2):72–83.
- Kumar, A. y Waldron, K. (1985). The workspaces of a mechanical manipulator. *Trans. ASME, Journal of Mechanical Design*, 4(2):3–9.
- Lane, D., Falconer, G., Randall, G., Duffy, N., Herd, J., Chernett, P., Hunter, J., Colley, M., Standeven, J., Callaghan, V., Smith, J., Evans, J., Woods, A., Penrose, J., Whittaker, G., Smith, D., and Edwards, I. (1998). Mixing simulations and real subsystems for subsea robot development. specification and development of the core simulation engine. In *OCEANS '98 Conference Proceedings*, pages 1382 – 1386.
- Le Page, Y. and Holappa, K. (2000). Simulation and control of an autonomous underwater vehicle equipped with a vectored thruster. In *OCEANS 2000 MTS/IEEE Conference and Exhibition*, pages 2129 –2134 vol.3.
-

- Li, D. and Salcudean, S. (1997). Modeling simulation and control of a hydraulic stewart platform. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3360 – 3366.
- Li, J. Wang, S. W. X. y. H. C. (2010). Optimization of a novel mechanism for a minimally invasive surgery robot. *Int. J. Med. Robotics Comput. Assist. Surg.*, 6(1):83–90.
- Li, M. Huang, T. Z. D. (2005). Conceptual design and dimensional synthesis of a reconfigurable hybrid robot. *ASME J Manufact Sci Eng*, 127(3):647–653.
- Lin, T. and Gilbert, J. (1991). Analyses of magnetohydrodynamic propulsion with sea water for underwater vehicles. American Institute of Aeronautics and Astronautics.
- Lin, X. and Guo, S. (2012). Development of a spherical underwater robot equipped with multiple vectored water-jet-based thrusters. *J Intell Robot Syst*, 63:307–321.
- Lipkin, H. y Duffy, J. (1989). Hybrid twist and wrench control of a robotic manipulator. *ASME J. Mechanisms Transmissions Automation Des.*, 110:110–144.
- Liu, M.-J., Li, C.-X., and Li, C.-N. (2000). Dynamics analysis of the gough-stewart platform manipulator. *Robotics and Automation, IEEE Transactions on*, 16(1):94–98.
- Low, K. H. and Willy, A. (2005). Development and initial investigation of ntu robotic fish with modular flexible fins. In *Proceedings of the IEEE International Conference on Mechatronics and Automation*.
- Ma, O. and Angeles, J. (1991). Optimum architecture design of platform manipulators. In *Fifth International Conference on Advanced Robotics, 1991. 'Robots in Unstructured Environments'*, pages 1130 –1135 vol.2.
- Marani, G., Choi, S. K., and Yuh, J. (2009). Underwater autonomous manipulation for intervention missions auvs. *Ocean Engineering*, 36(1):15 – 23.
- Merlet, J. (2007). Jacobian, manipulability, condition number and accuracy of parallel robots. In *Robotics Research*, volume 28, pages 175–184.
- Merlet, J. P. (2006). *Parallel Robots*. Springer.
- Morel, Y. and Leonessa, A. (2003). Adaptive Nonlinear Tracking Control of an Underactuated Non-minimum Phase Model of a Marine Vehicle Using Ultimate Boundedness. In *42nd IEEE Conference on Decision and Control*.
- Moreno, H. Herramientas de simulación dinámica para interacciones hápticas. Trabajo fin de master. Universidad Politécnica de Madrid.
- Moreno, H. A., Puglisi, L. J., Saltaren, R. J., and Carrera, I. (2011). Kinematic analysis of an underwater parallel robot. In *OCEANS 2011 IEEE Spain*, pages 1–6.
-

- Moreno, H. A. y Pamanes, J. (2011). Isotropic design of a 2 dof parallel kinematic machine with a translational workpiece table. In *13th IFToMM World Congress in Mechanism and Machine Science*.
- Moreno, H. Pamanes, J. W. P. y. C. D. (2006). Global optimization of performance of a 2prp parallel manipulator for cooperative tasks. In *ICINCO-RA '06*, pages 516–522.
- Moreno, H. Saltaren, R. P. J. y. A. R. (2010). Motion strategy for the climbing robot on a metallic orthogonal structure. In *Proceedings of the 13th International Conference on Climbing and Walking Robots*, pages 1161–1168.
- Nie, J., Yuh, J., Kardash, E., and Fossen, T. I. (1998). On-board sensor-based adaptive control of small uuv's in very shallow water. In *Proc. of IFAC-Control applications in Marine Systems*, pages 201–206.
- Pamanes, A. y Zegloul, S. (1991). Optimal placement of robotic manipulators using multiple kinematic criteria. In *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, pages 933–938 vol.1.
- Paster, D. (1986). Importance of hydrodynamic considerations for underwater vehicle design. *OCEANS*, 18:1413–1422.
- Phoha, S., Peluso, E., and Culver, R. (2001). A high-fidelity ocean sampling mobile network (samon) simulator testbed for evaluating intelligent control of unmanned underwater vehicles. *IEEE Journal of Oceanic Engineering*, 26(4):646 – 653.
- Polsenberg, A., Milano, M., Gsell, M., and Fischer, K. (2005). Synthetic jet propulsion for small underwater vehicles. In *Proceedings of the International Conference of Robotics and Automation*, pages 181–187.
- Potter, M., Wiggert, D., and Hondzo, M. (1998). *Mecanica de Fluidos*. Pretince Hall.
- Prats, M., Ribas, D., Palomeras, N., Garcia, J. C., Nannen, V., Wirth, S., Fernandez, J. J., Beltran, J. P., Campos, R., Ridao, P., Sanz, P. J., Oliver, G., Carreras, M., Gracias, N., Marin, R., and Ortiz, A. (2012). Reconfigurable AUV for intervention missions: A case study on underwater object recovery. *Journal of Intelligent Service Robotics*, 5(1):19–31.
- P.Wenger and Chablat, D. (2000). Kinematic analysis a of new parallel machine-tool: the orthoglide. In *ARK*, pages 305–314.
- Ridao, P., J., Y., Battle, J., and Sugihara, K. (2000). On auv control architecture. In *Proceedings of the International Conference on Intelligent Robots and Systems*, pages 855–860.
- Ross, C. (2006). A conceptual design of an underwater vehicle. *Ocean Engineering*, 33(16):2087–2104.
-

- Rumbaugh, J., Jacobson, I., and Booch, G. (2000). *El Lenguaje Unificado de Modelado, Manual de Referencia*. Pearson Addison- Wesley.
- S. Song, D. K. and Kim, W. (2003). Spherical joint for coupling three or more links together at one point.
- Salisbury, K. y Craig, J. (1982). Articulated hands: force and kinematic issues. *The Int. J. of Robotic Research*, 1(1):4–17.
- Saltaren, R., Aracil, R., Alvarez, C., Yime, E., and Sabater, J. (2007a). Field and service applications - exploring deep sea by teleoperated robot - an underwater parallel robot with high navigation capabilities. *Robotics Automation Magazine, IEEE*, 14(3):65–75.
- Saltaren, R., Sabater, J. M., Yime, E., Azorin, J. M., Aracil, R., and Garcia, N. (2007b). Performance evaluation of spherical parallel platforms for humanoid robots. *Robotica*, 25(3):257–267.
- Schneider, P. and Eberly, D. (2003). *Geometric Tools for Computer Graphics*. Elsevier.
- Seaeye (2012). *Web page Panther-XT*. Online:<http://www.seaeye.com/pantherxt.html>.
- Staicu, S. (2011). Dynamics of the 6-6 stewart parallel manipulator. *Robot. Comput.-Integr. Manuf.*, 27:212–220.
- Staicu, S., Zhang, D., and Rugescu, R. (2006). Dynamic modelling of a 3-dof parallel manipulator using recursive matrix relations. *Robotica*, 24(01):125–130.
- Stommel, H. (1989). The slocum mission. *Oceanography*, pages 22–25.
- Tsai, L. W. (1999). *Robot Analysis*. John Wiley & Sons.
- Tsai, L.-W. (2000). Solving the inverse dynamics of a stewart-gough manipulator by the principle of virtual work. *Journal of Mechanical Design*, 122(1):3–9.
- Valavanis, K., Gracanin, D., Matijasevic, M., and Kolluru, R. (1997). Control architectures for autonomous underwater vehicles. *IEEE Control Systems*, 17(6):48–64.
- van de Ven, P. W. J., Flanagan, C., and Toal, D. (2005). Neural network control of underwater vehicles. *Eng. Appl. Artif. Intell.*, 18(5):533–547.
- Walker, D. (2005). Design and control of an high maneuverability remotely operated vehicle with multi-degree of freedom thrusters. Thesis. Massachusetts Institute of Technology.
- Wang, W., Engelaar, R., Chen, X., and Chase, J. (2009). The state-of-art of underwater vehicles - theories and applications. *Mobile Robots - State of the Art in Land, Sea, Air, and Collaborative Missions*, X.Q. Chen, Y.Q. Chen and J.G. Chase (Ed.), InTech.
-

- Webb, D., Simonetti, P., and Jones, C. (2001). Slocum: an underwater glider propelled by environmental energy. *Oceanic Engineering, IEEE Journal of*, 26(4):447–452.
- Williams, R. (1999). Planar parallel 3rpr manipulator. In *Proceedings of the Sixth Conference on Applied Mechanisms and Robotics*.
- Y. Takeda, K. Kamiyama, Y. M. M. H. and Sugimoto, K. (2005). Development of position-orientation decoupled spatial in-parallel actuated mechanisms with six degrees of freedom. *Journal of Robotics and Mechatronics*, 17(1):59–68.
- Yang, C., Huang, Q., Jiang, H., Peter, O. O., and Han, J. (2010). Pd control with gravity compensation for hydraulic 6-dof parallel manipulator. *Mechanism and Machine Theory*, 45(4):666 – 677.
- Yen, P.-L. and Lai, C.-C. (2009). Dynamic modeling and control of a 3-dof cartesian parallel manipulator. *Mechatronics*, 19(3):390 – 398.
- Yime, E. (2008). *Modelo Matemático y Control Vectorial de Robots Submarinos de Geometría Variable*. PhD thesis, Univesidad Politecnica de Madrid, Madrid, España.
- Yime, E., Moreno, H., and Saltaren, R. (2011). A novel 6 dof parallel robot with decoupled translation and rotation. In *13th World Congress in Mechanism and Machine Science*.
- Yoerger, D., Cooke, J., and J., S. (1990). The influence of thruster dynamics on underwater vehicle behavior and their incorporation into control system design. *IEEE Journal of Oceanic Engineering*, 15(3):167–178.
- Yoerger, D. and Slotine, J. (1985). Robust trajectory control of underwater vehicles. *IEEE Journal of Oceanic Engineering*, 10(4):462–470.
- Yoshikawa, T. (1985). Manipulability of robotic mechanisms. *The International Journal of Robotics Research*, 4(2):3–9.
- Yuh, J. (1990). A neural net controller for underwater robotic vehicles. *IEEE Journal of Oceanic Engineering*, 15(3):161–166.
- Yuh, J. (1994). Learning control of underwater robotic vehicles. *IEEE Control System*, 14(2):39–46.
- Yuh, J. (2000). Design and control of autonomous underwater robots: A survey. *Auton. Robots*, 8(1):7–24.
- Yuh, J., Nie, J., and Lee, C. (1999). Experimental study on adaptive control of underwater robots. In *Proceedings of the IEEE International Conference on Mechatronics and Automation*.
- Zabalza, I., Ros, J., Gil, J., Pintor, J., Jimenez, J., and Tri-Scott (2002). A new kinematic structure for a 6-dof decoupled parallel manipulator. In *Proceedings*
-

- of the Workshop on fundamental issues and future research directions for parallel mechanisms and manipulators*, pages 12–15.
- Zanganeh, K. y Angeles, J. (1997). Kinematic isotropy and the optimum design of parallel manipulators. *Int. J. Robot. Res.*, 16(2):185–197.
- Zanganeh, K. E. and Angeles, J. (1994). Instantaneous kinematics and desing of a novel redundant parallel manipulator. In *Proceedings IEEE International Conference on Robotics and Automation*, pages 3043–3048.
- Zhang, Y., Tian, J., Su, D., and Wang, S. (2006). Research on the hierarchical supervisory control of underwater glider. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 5509 –5513.
- Zhu, Z., Li, J., Gan, Z., and Zhang, H. (2005). Kinematic and dynamic modelling for real-time control of tau parallel robot. *Mechanism and Machine Theory*, 40(9):1051 – 1067.
- Zyda, M.J. McGhee, R., Kwak, S., Nordman, D., Rogers, R., and Marco, D. (1990). Three-dimensional visualization of mission planning and control for the nps autonomous underwater vehicle. *IEEE Journal of Oceanic Engineering*, 15(3):217–221.
-