

Centro de Investigación y Estudios Avanzados
del Instituto Politécnico Nacional

Unidad Zacatenco
Departamento de Control Automático

**Aprendizaje por demostración en el espacio articular
para el seguimiento de trayectorias aplicado en un
exoesqueleto de 4 grados de libertad.**

Tesis que presenta:

M. en C. Javier Garrido Meléndez

Para presentar el examen:

Doctoral

En la Especialidad de

Control Automático

Director de Tesis:

Dr. Wen Yu Liu

México, D.F.,

Enero 2015

Índice general

| | |
|---|----------|
| 1. Introducción | 1 |
| 1.1. Motivación | 3 |
| 1.2. Objetivos | 5 |
| 1.3. Estructura | 6 |
| 1.4. Publicaciones | 6 |
| | |
| 2. Diseño de un exoesqueleto para la extremidad superior. | 9 |
| 2.1. Cinemática y dinámica de la extremidad superior | 11 |
| 2.1.1. Modelado de la extremidad superior | 13 |
| 2.2. Diseño mecánico de un exoesqueleto para la extremidad superior | 14 |
| 2.2.1. Actuadores | 16 |
| 2.2.2. Eslabones | 18 |
| 2.2.3. Sistema de transmisión | 20 |
| 2.2.4. Protocolo de comunicación | 24 |
| 2.3. Control maestro | 26 |
| 2.4. Seguridad | 28 |
| 2.5. Modelo del Exoesqueleto | 29 |
| 2.5.1. Cinemática directa | 30 |
| 2.5.2. Modelo dinámico | 34 |
| 2.5.3. Singularidades | 36 |

| | |
|--|-----------|
| 3. Aprendizaje por Demostración | 39 |
| 3.1. ¿Qué es el ApD? | 40 |
| 3.2. Métodos para enseñar una tarea | 42 |
| 3.2.1. Mapeo de funciones. | 43 |
| 3.2.2. Modelo del sistema dinámico | 44 |
| 3.3. Reproducción de una tarea automáticamente | 44 |
| 3.3.1. La técnica basada en demostración de trayectorias | 44 |
| 3.3.2. La técnica basada en modelos probabilísticos | 45 |
| 3.3.3. Técnica basada en sistemas dinámicos | 49 |
| 3.4. Evaluación de la tarea aprendida | 50 |
| 4. Modelos Ocultos de Markov para el seguimiento de Trayectorias | 53 |
| 4.1. Modelos Ocultos de Markov | 53 |
| 4.1.1. Elementos de los Modelos Ocultos de Markov | 54 |
| 4.1.2. Tipos de Modelos Ocultos de Markov | 56 |
| 4.1.3. Los 3 problemas de los Modelos Ocultos de Markov | 58 |
| 4.2. Cuantización de trayectorias en el espacio articular usando el algoritmo de Lloyd's | 67 |
| 4.2.1. Algoritmo de Lloyd's | 68 |
| 4.3. Modelos Ocultos de Markov Modificados | 72 |
| 4.3.1. Generación de la trayectoria usando MOM modificado | 74 |
| 4.4. Simulaciones para un Robot de 2 gdl. | 77 |
| 4.4.1. Demostraciones con diferencias en el tiempo. | 79 |
| 4.4.2. Demostraciones con diferencias de amplitudes en el espacio de trabajo | 80 |
| 4.5. Simulación para un Robot Quirúrgico Raven. | 81 |
| 4.6. Comparación del algoritmo MOM modificado con otra investigación | 83 |
| 5. Resultados experimentales | 87 |
| 5.1. Controlador PD | 87 |
| 5.2. Controlador Maestro para el seguimiento de trayectorias. | 91 |

| | |
|--|------------|
| ÍNDICE GENERAL | III |
| 5.2.1. Ejemplo de seguimiento de trayectorias con diferentes amplitudes . . | 91 |
| 5.2.2. Ejemplo de seguimiento de trayectorias con diferentes velocidades . . | 93 |
| 6. Conclusiones y Trabajo Futuro | 101 |

Índice de figuras

| | |
|--|----|
| 1.1. Robot humanoide para aplicaciones de ApD | 2 |
| 2.1. Distribución estadística de las articulaciones del brazo humano durante 19 movimientos [56]. | 10 |
| 2.2. CinvesRobot-1: exoesqueleto de 4 gdl para la extremidad superior | 11 |
| 2.3. Anatomía de la extremidad superior | 12 |
| 2.4. Movimientos del hombro, codo y muñeca | 14 |
| 2.5. Grados de libertad simplificado de la extremidad superior. | 15 |
| 2.6. Grados de libertad para el exoesqueleto CinvesRobot-1 | 16 |
| 2.7. Exoesqueleto CinvesRobot-1 de 4 gdl. | 17 |
| 2.8. Modulo Power Cube (MPC). | 18 |
| 2.9. Diagrama a bloques del MPC. | 19 |
| 2.10. Eslabones del exoesquelto para las articulaciones a) 1, 2 y 3, b) 4, y c) soporte para el brazo y antebrazo | 20 |
| 2.11. a) Partes del Harmonic Drive, b)Harmonic Drive ensamblado | 21 |
| 2.12. Partes internas del MPC. | 23 |
| 2.13. Tarjeta de interface de protocolo USB a CAN | 25 |
| 2.14. Comunicación de los modulos con el control maestro | 27 |
| 2.15. Boton de paro | 28 |
| 2.16. Modulo de Seguridad en simulink | 29 |
| 2.17. Exoesqueleto de 4 gdl. | 30 |

| | |
|--|----|
| 2.18. Marco de referencia para los parámetros D-H. | 32 |
| 2.19. Exoesqueleto de 4 grados de Libertad | 37 |
| 3.1. Cómo enseñar una tarea | 42 |
| 3.2. Cómo reproducir una tarea | 45 |
| 4.1. MOM ergódico con $N = 4$ | 57 |
| 4.2. MOM no ergódico con matriz A triangular superior y $N = 4$ | 58 |
| 4.3. Secuencia de operaciones para el calculo de la variable hacia adelante $\alpha_T(i)$. | 60 |
| 4.4. secuencia de operaciones para el calculo de la variable hacia atras $\beta_t(i)$. . . | 61 |
| 4.5. Secuencia de operaciones para el cálculo del evento conjunto de que el sistema se encuentre en el estado S_i en el tiempo t y en el estado S_j en el t+1 | 65 |
| 4.6. El codebook $N = 3$ (lineas discontinuas) y El codebook $N = 4$ (lineas continuas) | 71 |
| 4.7. Diferentes demostraciones con el mismo números de símbolos de observación | 72 |
| 4.8. Trayectorias de un robot planar de 2 gdl. A) En el espacio articular y B) En el espacio de trabajo | 73 |
| 4.9. MOM para la trayectoria de la Figura 4.6 | 75 |
| 4.10. Generación de trayectorias por medio de MOM modificado usando el algoritmo de Lloyd's | 77 |
| 4.11. Robot de dos grados de Libertad | 78 |
| 4.12. Demostraciones en el espacio de trabajo | 79 |
| 4.13. Demostraciones en el espacio articular con diferencias en tiempo | 80 |
| 4.14. Trayectoria generada en el espacio articulado 2 veces más rápida que las demostraciones. | 81 |
| 4.15. Demostraciones con diferentes amplitudes en el espacio de trabajo | 82 |
| 4.16. Trayectorias en el espacio articular | 83 |
| 4.17. EL sistema quirúrgico Raven | 84 |
| 4.18. Trayectorias de Raven para la figura 8 en el espacio de trabajo | 85 |
| 4.19. Trayectorias de Raven para la figura 8 en el espacio articulado | 85 |
| 4.20. Raven dibujo de la figura "8" dos veces más rapido en el espacio articulado . | 86 |

| | |
|---|----|
| 4.21. Usando la técnica propuesta por [12] ,se calculó los puntos claves para q_1 para cada demostración N=11, 18 y 22. | 86 |
| 5.1. Diagrama a bloques para realizar seguimiento de trayectorias utilizando los MOM | 88 |
| 5.2. Exoesqueleto CinvesRobot-1 | 89 |
| 5.3. Controlar PD en Simulink para los MPC | 90 |
| 5.4. Respuesta a una señal cuadrada para cada eslabón | 91 |
| 5.5. Controlador Maestro | 92 |
| 5.6. Trayectoria de la articulación q_1 en el espacio articulado con diferentes amplitudes | 93 |
| 5.7. Trayectoria de la articulación q_2 en el espacio articulado con diferentes amplitudes | 94 |
| 5.8. Trayectoria de la articulación q_3 en el espacio articulado con diferentes amplitudes | 95 |
| 5.9. Trayectoria de la articulación q_4 en el espacio articulado con diferentes amplitudes | 95 |
| 5.10. Trayectoria en el espacio de Trabajo X,Y,Z | 96 |
| 5.11. Fotografía de las trayectorias en el espacio de trabajo | 96 |
| 5.12. Demostraciones en el espacio articulado con distintas velocidades para la articulación q_1 | 97 |
| 5.13. Demostraciones en el espacio articulado con distintas velocidades para la articulación q_2 | 97 |
| 5.14. Demostraciones en el espacio articulado con distintas velocidades para la articulación q_3 | 98 |
| 5.15. Demostraciones en el espacio articulado con distintas velocidades para la articulación q_4 | 98 |
| 5.16. Trayectorias en el espacio de trabajo con diferentes velocidades | 99 |
| 5.17. Fotografía de las trayectorias en el espacio de trabajo | 99 |

Capítulo 1

Introducción

En este trabajo se presenta el diseño y construcción de un exoesqueleto de 4 grados de libertad, con el objetivo de realizar el seguimiento de trayectorias usando la metodología de Aprendizaje por Demostración (ApD) aplicando la técnica de Modelos Ocultos de Markov (MOM).

El exoesqueleto construido es para la extremidad superior del ser humano (hombro y codo), está formado por 4 grados de libertad (gdl), donde 3 gdl son para la articulación del hombro, y 1 gdl para el codo utilizando articulaciones de tipo esférica y rotacional, respectivamente; para su diseño se investigó el modelo mecánico y cinemático del hombro y codo del ser humano.

Para la etapa de control se utilizó la metodología de ApD para el seguimiento de trayectorias por las ventajas que ésta ofrece, ya que se pueden generar las trayectorias con la ayuda de demostraciones, las cuales pueden ser en el espacio de trabajo o en el espacio articular donde la trayectoria generada puede aplicarse directamente en el espacio articular, sin la ayuda del modelo de la cinemática inversa.

El ApD se basa principalmente en el uso de demostraciones, las cuales son realizadas por seres humanos y sirven para entrenar un modelo del movimiento, el cual es usado para generar la trayectoria a realizar. Estos modelos de movimientos usualmente implican algún proceso para obtener un promedio de las trayectorias o un modelo probabilístico.



Figura 1.1: Robot humanoide para aplicaciones de ApD

El ApD tiene muchas aplicaciones en distintas áreas como por ejemplo: el control en robots que juegan futbol soccer [9], el control de vuelo en un helicóptero [14], robots que realizan tareas de traslación y sujeción de objetos [53] y robots móviles que evitan obstáculos [28].

Dentro del ApD existen diversas técnicas para el aprendizaje y generación de trayectorias las cuales se pueden clasificar como:

- Técnicas basadas en demostraciones de trayectorias:
 - Métodos de aprendizaje iterativo.
 - Aprendizaje por demostración con múltiples datos de entradas.

- Técnicas basadas en modelados probabilísticos:
 - Modelos de mezclas gaussianas.
 - Regresión de mezclas gaussianas.
 - Regresión gaussiana local.
 - Aprendizaje flojo.
 - Modelos Ocultos de Markov.
 - Máquina de vectores soportados.

- Técnicas basadas en modelos dinámicos:
 - Preplaneación del movimiento de un sistema dinámico.

Las técnicas antes mencionadas realizan el seguimiento de trayectorias en el espacio de trabajo, por lo que es necesario utilizar el modelo de cinemática inversa para aplicarlo a un robot; uno de los objetivos planteados en este trabajo es evitar el uso del modelo de cinemática inversa, por lo tanto al aplicar la metodología del ApD debe ser capaz de generar las trayectorias en el espacio articular. La técnica más viable que se puede adaptar para lograr este objetivo son los MOM.

En el ApD se utilizan los modelos probabilísticos como los MOM, debido a que en las demostraciones existen variaciones en las trayectorias en amplitud y velocidad, la ventaja de aplicar las técnicas estadísticas es la capacidad de manejar la alta variabilidad inherente en los datos de las demostraciones, con el fin de codificar las variaciones temporales y espaciales de los datos, así como de codificar un modelo para recuperar y reproducir la trayectoria generada.

1.1. Motivación

La construcción de un exoesqueleto de la extremidad superior para realizar el seguimiento de trayectorias, surgió por la necesidad de tener una plataforma para poder comprobar diversas técnicas de control, en este trabajo la técnica que se desea comprobar es la metodología de ApD, utilizando la herramienta estadística de los MOM.

El exoesqueleto de la extremidad superior de 4 gdl “CinvesRobot-1” es el primer prototipo construido en el laboratorio del departamento de Control Automático del Cinvestav, cuyo diseño tiene como objetivo realizar los movimientos del hombro y codo, para que en un futuro se pueda utilizar en distintas aplicaciones como amplificadores de fuerza o en ejercicios para rehabilitación. Para llevar a cabo las pruebas de simulación antes de aplicarlas directamente al exoesqueleto fue necesario calcular la cinemática directa y el modelo dinámico.

El exoesqueleto a diferencia de un robot está unido por algún mecanismo a los eslabones del ser humano, por lo tanto el espacio de trabajo del robot está limitado al movimiento de

las articulaciones del mismo, en el caso del exoesqueleto su espacio de trabajo está limitado al rango de movimiento de sus articulaciones y de las articulaciones del ser humano, debido a esto para poder llevar a cabo el seguimiento de trayectorias se deben de tomar en cuenta los movimientos que puede realizar el ser humano.

La principal característica del APD es que se basa en demostraciones, las cuales son realizadas por el ser humano por lo que las trayectorias están limitadas por el espacio de trabajo de la persona, dichas trayectorias deberán ser reproducidas por el exoesqueleto.

Actualmente la metodología de ApD para seguimiento de trayectorias en robots se realiza en el espacio de trabajo, existen pocas investigaciones que lo aplican en el espacio articular, la ventaja de aplicarlo directamente en el espacio articular es evitar el cálculo de la cinemática inversa, la cual en el caso de los exoesqueletos entre mas gdl tenga, más complejo sera obtener el modelo de cinemática inversa.

Las investigaciones que aplican el ApD en el espacio articular utilizan la técnica de Dynamic Time Warping (DTW) que en español se le conoce como Distancia de Alineamiento Temporal no Lineal, la cual tiene como función alinear las trayectorias en el tiempo y promediar las mismas para así poder realizar el seguimiento de trayectorias; la idea de aplicar los MOM es evitar el uso de DTW.

Existen diferentes trabajos que aplican los MOM para reconocimiento de trayectorias, en el trabajo de [12], el cual tiene por objetivo identificar las trayectorias realizadas por un robot, el cual aprende el movimiento de 4 letras del abecedario, para realizar esto, los MOM generan una secuencia llamada cadena de Markov [57], [79], la cual decodifica las trayectorias del movimiento de un robot y encuentra la probabilidad más alta de la secuencia de estados con ayuda del algoritmo de Viterbi [78].

Para el entrenamiento de los MOM [3], utiliza los puntos claves, los cuales se obteniendo calculando los máximos y mínimos en las trayectorias o cuando existen cambios en las posiciones o en las velocidades para lo cual se utilizan ciertos valores de umbral, tiene el inconveniente que si se modifica la trayectoria se tiene que volver a asignar los valores de umbral, en este trabajo se propone el algoritmo de Lloyd's para calcular los puntos claves de forma automática solo asignando un número de particiones para la trayectoria.

El trabajo presentando por [73], utilizan los MOM en el aprendizaje de trayectorias, utilizan los puntos claves y usando la técnica de Dynamic Time Warping, para evitar el uso de esta técnica se propone el uso del algoritmo de Lloyd's.

Trabajos sobre Aplicaciones de planeamiento de trayectorias usando MOM se basan en el uso del algoritmo forward el cual es usado para buscar el conjunto con la máxima probabilidad en los modelos entrenados para generar la trayectoria deseada, el algoritmo de Viterbi [78] es aplicado para encontrar la secuencia de estados más probable, otras investigaciones se presentan en [79], [35].

El objetivo de usar los MOM y el algoritmo de Lloyd's es para evitar tres problemas:

1. El problema del cálculo de la Dynamic Time Warping en el espacio articular.
2. Generar las trayectorias en el espacio de trabajo y tener que tilzar el modelo de cinemática inversa del robot.
3. El problema de seleccionar los puntos claves en el espacio articular de forma manual.

El clásico HMM es además modificado para que sea mas factible generar una trayectorias en el espacio articular y en el espacio de trabajo.

1.2. Objetivos

1. Diseñar y construir un exoesqueleto de la extremidad superior de 4 gdl para realizar el control de seguimiento de trayectorias.

2. Aplicar la metodología de Aprendizaje por Demostración para el seguimiento de trayectorias en un exoesqueleto.

3. Diseñar un algoritmo que permita a los MOM modelar el movimiento de trayectorias en las demostraciones para generar una trayectoria generalizada.

4. Diseñar un algoritmo utilizando la técnica de Lloyd's para calcular los puntos clave para entrenar los MOM.

5. Aplicar la metodología de ApD utilizando los MOM para realizar el seguimiento de trayectorias en el exoesqueleto de 4 gdl.

1.3. Estructura

En el Capítulo 1 se describe el diseño del exoesqueleto de 4 grados de libertad, se explica la cinemática de la extremidad superior del ser humano (hombro y codo), se describe el diseño mecánico para la selección de sus componentes además de describir el cálculo de la cinemática directa y el modelo dinámico, usando los parámetros de Denavit-Hartenberg (D-H) para su cálculo.

En el Capítulo 2 se describe la metodología del ApD, donde se explica que significa aprender una trayectoria y como se genera la trayectoria aprendida, así como sus principales características y las distintas técnicas para aplicar esta metodología.

En el Capítulo 3 se explica que son los Modelos Ocultos de Markov, los problemas que pueden resolver, su arquitectura, además de describir el algoritmo diseñado para entrenar los modelos, usando demostraciones y generar una trayectoria para el seguimiento.

En el Capítulo 4 se muestran los resultados de la aplicación de la metodología de ApD usando los MOM para la generación de trayectorias en el exoesqueleto de 4 gdl.

Finalmente en el Capítulo 5 se presentan las conclusiones y el trabajo futuro.

1.4. Publicaciones

Revista

1. Javier Garrido, Wen Yu, Human Behavior Learning for Robot in Joint Space, *Neuro-computing*, Aceptado.
2. Javier Garrido, Alberto Soria, Modular Design and Control of an Upper Limb Exoskeleton, *Journal of Mechanical Science and Technology*, en revisión.
3. Javier Garrido, Wen Yu, Robot Trajectory Generation Using Modified Hidden Markov Model and Lloyd's Algorithm in Joint Space, *Engineering Applications of Artificial Intelligence*, en revisión.

Congreso.

1. Javier Garrido, Wen Yu, Alberto Soria, Modular Design and Modeling of an Upper Limb Exoskeleton, *5th IEEE. International Conference on Biomedical Robotics and Biomechatronics*, Sao Paulo, Brazil, 508-513, 2014.
2. Javier Garrido, Wen Yu, Trajectory Generation in Joint Space Using Modified Hidden Markov Model, *23rd IEEE. International Symposium on Robot and Human Interactive Communication*, Edinburgh, Scotland, UK, 429-434, 2014.

Capítulo 2

Diseño de un exoesqueleto para la extremidad superior.

En este capítulo se explica el diseño y construcción de un exoesqueleto de 4 gdl para la extremidad superior (hombro y codo), se presentan las consideraciones que se tomaron en cuenta para su diseño, se explican las partes que componen al exoesqueleto y finalmente el modelo matemático para calcular su cinemática directa y modelo dinámico.

Los exoesqueletos se usan fundamentalmente como: dispositivos teleoperados [56], amplificadores de fuerza en seres humanos [18], apoyo de terapia física en ejercicios de rehabilitación [74], para rehabilitación de las extremidades superiores [51] y extremidades inferiores [32].

La primera generación de exoesqueletos se diseñó como amplificadores de fuerza para humanos conocidos como Hardiman [39], donde sus articulaciones eran accionadas hidráulicamente por un operador. La segunda generación de exoesqueletos usaba directamente la fuerza de contacto entre el humano y el exoesqueleto como señal de control, por lo cual el operador estaba en completo contacto físico con el exoesqueleto durante su manipulación [32]. La tercera generación de exoesqueletos está definida por un alto nivel fisiológico por lo que su uso requiere sensores electromiográficos [22] y [54].

Para realizar el diseño es importante estudiar la cinemática y dinámica del brazo humano durante las actividades diarias para determinar el rango de movimiento de las articu-

laciones en la investigación [56], se realizó un estudio donde se muestran el histograma de los movimientos del brazo cuando se realizan 19 actividades diarias, frecuentemente las velocidades de las articulaciones son bajas al inicio o al final del movimiento de las trayectorias, como se muestra en la figura 2.1, algunas distribuciones tienen la forma de distribución normal, binomial o trinomial donde el centro corresponde al punto clave de las configuraciones antropomórficas.

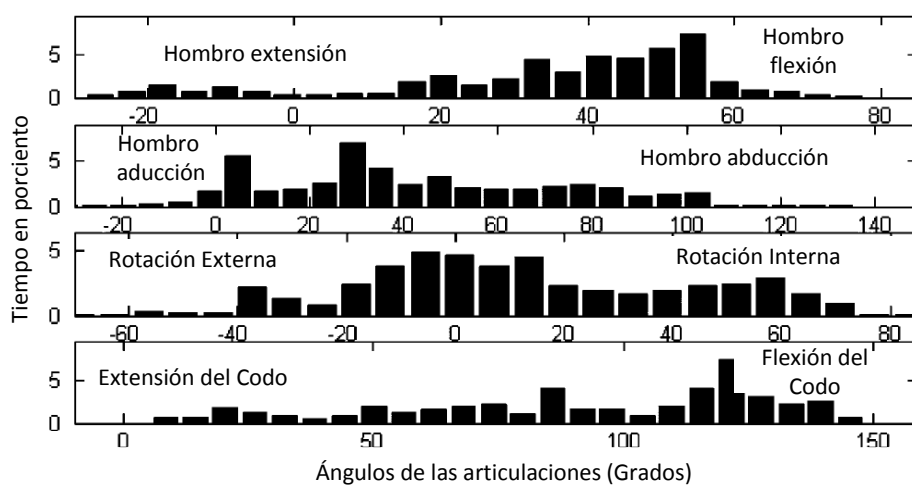


Figura 2.1: Distribución estadística de las articulaciones del brazo humano durante 19 movimientos [56].

Los exoesqueletos para la extremidad superior comenzaron con [33] el cual fue de los primeros en estudiar los 7 gdl de la extremidad superior del ser humano, en el trabajo de [15] se presenta un diseño 9 gdl y [40] presenta un diseño de 11 gdl, el tener mayor gdl en un exoesqueleto tiene la ventaja de generar movimientos que otros de menor gdl no puede, el diseño de [51] no toma en cuenta el movimiento de traslación y rotación de la escápula y de la clavícula. Analizando los trabajos antes mencionados, los modelos de 7 gdl dan una buena combinación de exactitud y su diseño no es tan complejo, si solo se toman en cuenta los movimientos del codo y hombro, estos movimientos solo tienen 4 gdl como en el diseño de [44].

Una parte esencial de los exoesqueletos son los actuadores, en la actualidad existen una gran variedad como son: motores eléctricos [47], músculos neumáticos [59] y actuadores hidráulicos [42].

La forma de llevar a cabo la transmisión de potencia en los exoesqueletos es a través de un tren de engranes [38], accionamiento por cables [51], y diferentes mecanismos de acoplamiento.

El diseño del exoesqueleto está montado en el piso, lo que permite ajustarse a diferentes niveles de altura de las personas, para la sujeción del brazo tiene un mecanismo deslizante lo que permite trabajar con diferentes tamaños de brazos, como se ve en la figura 2.2-1. El exoesqueleto CinvesRobot tiene 4 gdl, tres para el hombro y una para el codo.

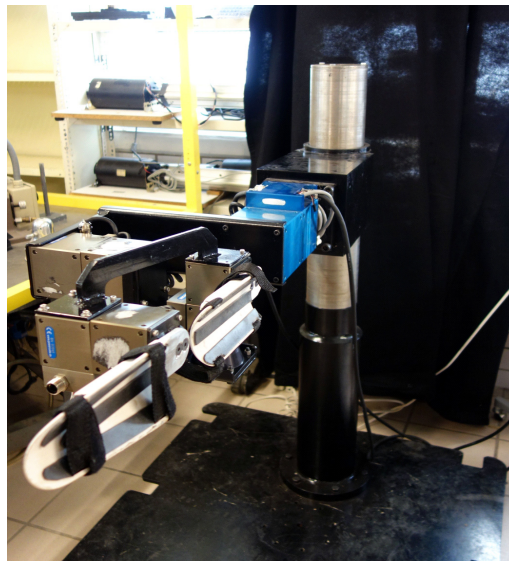


Figura 2.2: CinvesRobot-1: exoesqueleto de 4 gdl para la extremidad superior

2.1. Cinemática y dinámica de la extremidad superior

La extremidad superior está formada principalmente por el hombro, codo y muñeca y huesos como se muestra en la figura 2.3. El hombro es la parte del cuerpo donde se une el

brazo con el tronco, la cual está formado por la conjunción de los extremos de tres huesos: la clavícula, la escápula y el húmero; así como por músculos, ligamentos y tendones. El hombro es la articulación con mayor amplitud de movimientos del cuerpo humano.

El brazo se encuentra entre la cintura escapular que lo fija al tronco y el antebrazo, el hueso del brazo es el húmero, que se articula con la escápula por su epífisis proximal, y con el cúbito y radio en su epífisis distal.

Las manos están localizadas en los extremos de los antebrazos, son prensiles y tienen cinco dedos cada una. Abarcan desde la muñeca hasta la yema de los dedos. La mano está unida al antebrazo por una unión llamada muñeca (cuyos huesos forman el carpo) y consiste en una palma central (cuyos huesos forman el metacarpo) de la que surgen cinco dedos (también denominados falanges). Además la mano está compuesta de varios músculos y ligamentos diferentes que permiten una gran cantidad de movimientos y destreza.

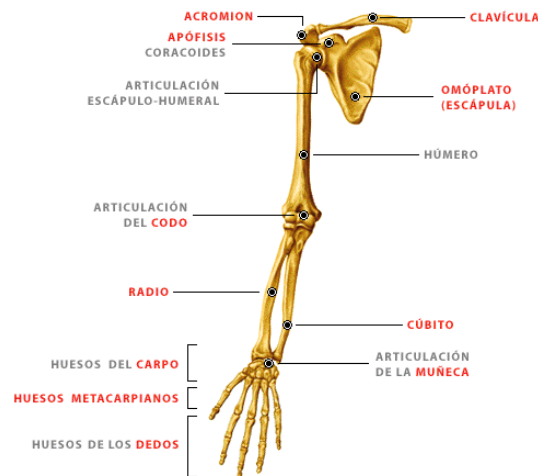


Figura 2.3: Anatomía de la extremidad superior

2.1.1. Modelado de la extremidad superior

La extremidad superior puede ser representada como un conjunto de cuerpos rígidos conectados por uniones. La extremidad superior tiene un total de 9 gdl sin tomar en cuenta los dedos.

La articulación del glenohumeral del hombro es compleja, es una articulación esférica la cual habilita al húmero a rotar en 3 gdl, con los siguientes movimientos: flexión-extensión, abducción- aducción y rotación medial-lateral.

La articulación esternoclavicular tiene 2 gdl, sus movimientos son elevación-depresión del hombro y retracción-prolongación, por lo tanto el hombro tiene un total de 5 gdl.

El codo tiene 2 gdl, el movimiento es flexión-extensión del codo y pronación-supinación del antebrazo. La articulación de la muñeca tiene 2 gdl, la muñeca flexión-extensión y radial-cubital. Los movimientos se muestran en la figura 2.5

Los exoesqueletos de la extremidad superior están diseñados para actuar en las articulaciones glenohumeral, codo y muñeca y son pocos los exoesqueletos que incluyen los grados de libertad de la articulación esternoclavicular, si se desea que opere en todo el rango de los movimientos de la extremidad superior del ser humano es necesario el uso de actuadores pasivos y activos para las articulaciones esternoclavicular y glenohumera.

Los movimientos de la extremidad superior puede ser simplificados por siete grados de libertad, con los cuales se puede obtener un espacio de trabajo bastante funcional, donde 3 gdl son para el hombro 2 gdl para el codo y 2 gdl para la muñeca

El brazo humano se puede dividir en tres articulaciones complejas las cuales son: el hombro, el codo y la muñeca; para el diseño del exoesqueleto del hombro y codo se realizan los movimientos que se muestran en la figura 2.4, en la tabla 2.1 se muestran los grados de movimientos de cada articulación al hacer actividades diarias.

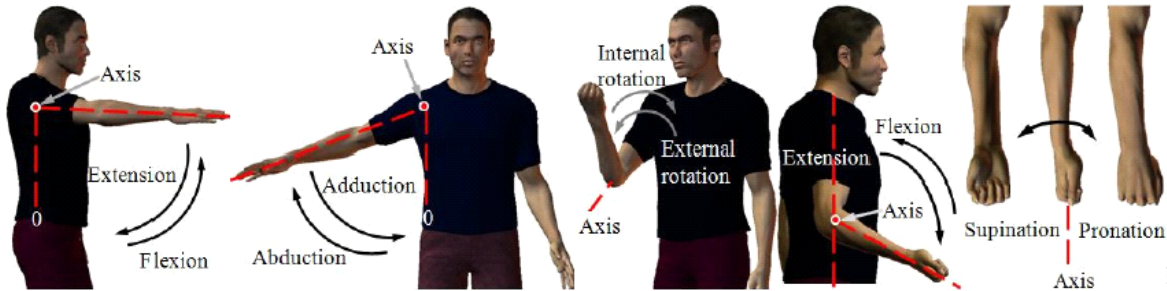


Figura 2.4: Movimientos del hombro, codo y muñeca

| Extremidad superior | Movimientos | Grados de movimiento |
|---------------------|----------------------------|---------------------------|
| Hombro. (3 gdl). | Abducción-aducción. | 0° a 180° |
| | Flexión-extensión. | -60° a 180° |
| | Interno-externo(rotación). | -80° a 180° |
| Codo. (1 gdl). | Flexión-extensión. | 0° a 150° |

Tabla 2.1 Movimientos del hombro y codo

2.2. Diseño mecánico de un exoesqueleto para la extremidad superior

El análisis mecánico se realiza para validar la estructura cinemática del robot. Como ya se mencionó anteriormente el hombro es anatómicamente complejo, debido a que su centro de rotación cambia con los movimientos que realiza, si el exoesqueleto está directamente unido a la parte superior del brazo del usuario, se debe tomar en cuenta el poder cambiar el centro de rotación lo cual es una gran dificultad en el diseño, además de que el centro de rotación del hombro del usuario está adentro de él, por lo tanto es necesario modelar la articulación del hombro como una articulación esférica.

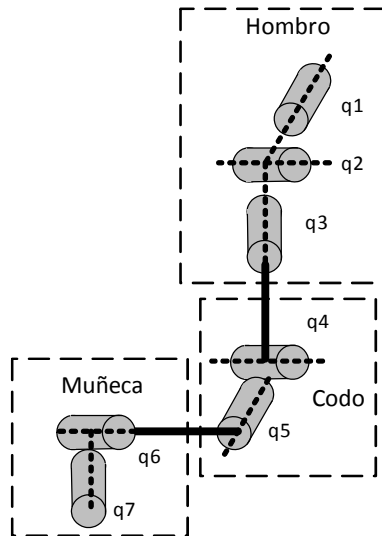


Figura 2.5: Grados de libertad simplificado de la extremidad superior.

La articulación del codo es modelada como una unión tipo bisagra, aun cuando consiste en tres huesos, alinear el centro del eje de rotación de la articulación con el del usuario es bastante sencillo.

El diseño del exoesqueleto se hizo con 4 gdl divididos en tres gdl para el hombro y 1 gdl para el codo, como se muestra en la figura 2.6. Para el movimiento del hombro es necesario una articulación esférica, por lo tanto se busca que los 3 ejes de cada gdl coincidan en un mismo punto, para el caso del codo se requiere un solo gdl tipo bisagra como se muestra en la figura 2.7.

Las articulaciones del exoesqueleto están representados por q_1 a q_4 donde q_1 a q_3 son los encargados de los movimientos del hombro: flexión-extensión, abducción-aducción y rotación interna-externa, q_4 es el responsable del movimiento del codo flexión-extensión.

A continuación se explican las partes que componen el exoesqueleto.

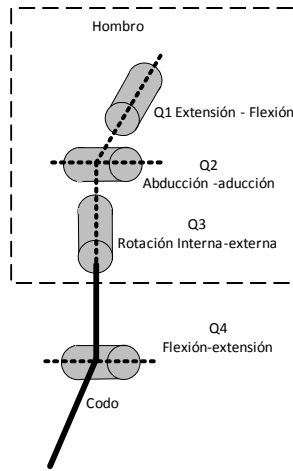


Figura 2.6: Grados de libertad para el exoesqueleto CinvesRobot-1

2.2.1. Actuadores

Los Motores usados en el exoesqueleto son Módulos Power Cube (MPC) de la marca Shunk como se muestran en la figura 2.8, cada módulo puede trabajar como controlador distribuido. El controlador maestro (computadora) es el encargado de generar la secuencia del programa y enviar la señal de control o de referencias a cada uno de los MPC, debido a que cada motor se puede controlar de tres formas diferentes: Control de Corriente, Velocidad y Posición. El controlador maestro es el encargado de monitorear cada motor, leer las variables y comprobar que estén dentro de los parámetros nominales para su correcto funcionamiento.

Las variables que en los MPC se pueden monitorear son:

- Posición.
- Temperatura.
- Corriente.
- Voltaje.
- Velocidad.
- Aceleración.

En la parte de seguridad cada motor tiene instalados frenos electromagnéticos los cuales

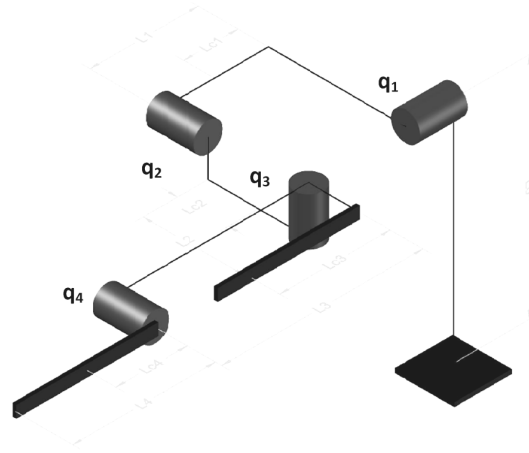


Figura 2.7: Exoesqueleto CinvesRobot-1 de 4 gdl.

se pueden activar por diferentes causas: si el motor se sale de su rango de operación de trabajo, si uno de sus parámetros está fuera de sus valores nominales o por programación para evitar una colisión.

Internamente cada uno de los módulos está integrado por un controlador, un servoamplificador, un motor sin escobillas (brushless), slipping por el que se introducen los cables, un encoder absoluto para medir la posición y sensores, cuenta con una interface de comunicación que soporta dos protocolos: protocolo UART RS-232 y protocolo CAN (Controller Area Network)), además de contar con sensores de límite (limit switch) que se pueden programar para que se detenga el motor cuando alcance ciertos límites de trabajo; en la figura 2.9 se muestra el diagrama de flujo de datos del modulo Power Cube.

Para el diseño del exoesqueleto se calculó la masa a mover, la cual está formada por el peso de los eslabones, los motores y las articulaciones del usuario; una vez calculado lo anterior se propusieron los motores con el par requerido, tomando en cuenta que entre más cerca éste de la base más par se requiere, los modelos Power Cube que se usaron se muestran en la siguiente tabla 2.2:



Figura 2.8: Modulo Power Cube (MPC).

| Articulación | Modelo Powe Cube | Voltaje | Par | Peso |
|--------------|------------------|---------|--------|--------|
| 1 | PR 110-161 | 48 VDC | 142 Nm | 5.6 Kg |
| 2 | PR 90-161 | 24 VDC | 72 Nm | 3.4 Kg |
| 3 | PR 70-161 | 24 VDC | 23 Nm | 1.7 Kg |
| 4 | PR 70-161 | 24 VDC | 23 Nm | 1.7 Kg |

Tabla 2.2 Modulos Power Cube usados por articulación

Los MPC se unen unos con otros a través de eslabones los cuales tienes como objetivo alienar las articulaciones del exoesqueleto a continuación se menciona como se seleccionaron los eslabones.

2.2.2. Eslabones

Los eslabones se construyeron de acero al carbón y dependiendo de su función es la forma y longitud de cada uno, como se explicó anteriormente para realizar los movimientos del hombro se necesitan 3 gdl, por lo tanto fue necesario diseñar los eslabones en forma de "L", la longitud y ancho de cada eslabón se seleccionó en función de las medidas de los MPC

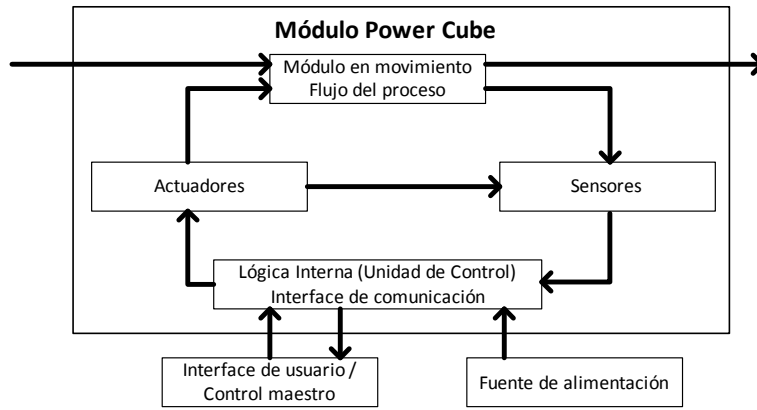


Figura 2.9: Diagrama a bloques del MPC.

para que sus ejes de rotación coincidan, además que con esto se logró que los ejes de los motores estén ortogonales uno con otro, como se muestra en la figura 2.10(a).

El eslabón utilizado para unir la articulación 3 con la articulación 4 tiene la forma que se muestra en la figura 2.10(b), se seleccionó esta forma debido a las diferencias que tienen los motores en tamaño y es necesario por diseño que los ejes que sirven para la sujeción del brazo y el antebrazo estén alineados.

Para la sujeción del brazo y antebrazo se construyeron unos soportes de plástico duro (PVC), el cual tiene dos correas con velcro que sujetan el brazo y antebrazo respectivamente, además cuentan con un mecanismo deslizante que sirve para sujetar brazos de diferentes tamaños como se muestra en figura 2.10(c).

Las dimensiones de los eslabones así como las masas se muestran en la tabla 2.3

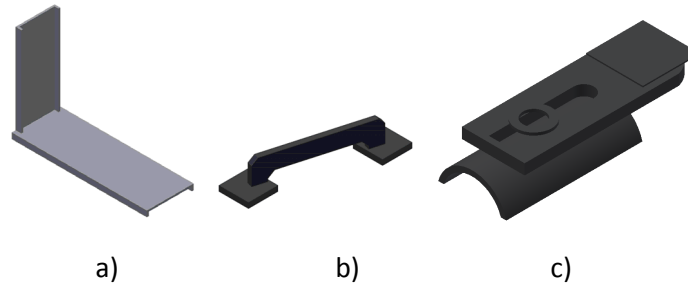


Figura 2.10: Eslabones del exoesqueleto para las articulaciones a) 1, 2 y 3, b) 4, y c) soporte para el brazo y antebrazo

| Eslabón | Tipo de eslabón | Masa(g) | Dimensiones(mm) |
|-------------------------------|-----------------|----------------|---|
| 1 | a) | 2875 | Largo=345, ancho=228 Altura=100, espesor=4.76 |
| 2 | a) | 1578 | Largo=190, ancho=175 Altura=100, espesor=4.76 |
| 3 | a) | 1006.13 | Largo=345, ancho=228 |
| 4 | b) | 577.48 | Largo=220, ancho =95 |
| Soporte del brazo y antebrazo | c) | 577.48 1578 | Largo=220, ancho superior=95 Ancho inferior=72, espesor=4.76 |

Tabla 2.3 Dimensiones y masa de los eslabones

Los MPC cuentan con un sistema de transmisión para aumentar el par de los motores, los cuales se mencionan a continuación.

2.2.3. Sistema de transmisión

Los MPC usan un sistema de transmisión de potencia tipo Harmonic Drive (HD) el cual es capaz de desarrollar elevadas relaciones de transmisión, con exactitud en su posicionamiento y

una relación peso-volumen baja. El Harmonic Drive está compuesto por 3 elementos distintos y con funciones bien definidas, El Wave Generator (WG), El FlexSpline (FS) y El Circular Spline (CS), los elementos se muestran en la figura. 2.11.a).

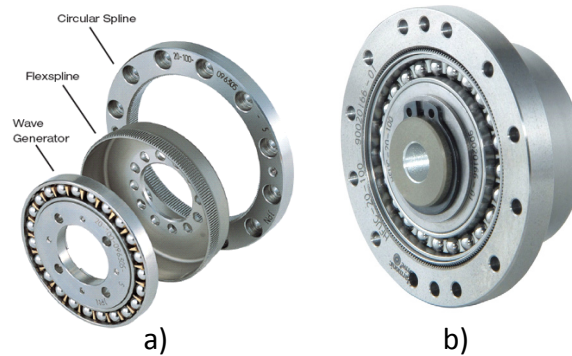


Figura 2.11: a) Partes del Harmonic Drive, b) Harmonic Drive ensamblado

Se dice que el HD tiene tres posibles configuraciones de funcionamiento, debido a que cualquiera de los tres componentes antes mencionados puede servir de entrada o salida de movimiento de rotación, lo cual determinará la configuración del dispositivo.

La configuración básica, consiste en la fijación del CS, que permite una rotación de alta velocidad en el WG y una baja velocidad en el FS, de manera similar, con el FS fijo, el par en el WG puede incrementarse y transmitirse a través del Circular Spline. Si tanto el Circular Spline como el FlexSpline son utilizados como entradas, el Wave Generator puede ser manejado a altas velocidades y bajo par. En general, usando diferentes combinaciones de rotación de los tres componentes del HD, se pueden lograr diferentes relaciones de transmisión y par.

Fundamentos de operación del HD

Cuando se insertan en el FlexSpline y el Wave Generator en el Circular Spline como se muestra en la figura 2.11.b), los dientes externos del FlexSpline engranan con los dientes

internos del Circular Spline a lo largo del eje mayor de la elipse del Wave Generator. Los tres componentes de la transmisión pueden rotar a velocidades diferentes, aun estando en el mismo eje de rotación. El mecanismo de rotación del Harmonic Drive comienza su recorrido en el momento que el Wave Generator tiene la zona de engrane en su eje mayor, y cuando esta zona se desplaza 180° alrededor de la circunferencia del Circular Spline, entonces el FlexSpline, que posee dos dientes menos que el Circular Spline, tendrá un retraso de un diente relativo al Circular Spline. Por este engrane gradual y continuo de los dientes del engranaje, cada rotación del Wave Generator mueve al FlexSpline dos dientes hacia atrás sobre el Circular Spline. Las relaciones de transmisión típicas del Harmonic Drive están en el rango de 50:1 a 200:1, en una transmisión simple [60]. Las eficiencias del mecanismo normalmente rondan el 85 %, aunque pueden alcanzar un 90 %; las transmisiones más pequeñas pueden proporcionar un par mínimo de salida de alrededor de 1.3 N.m, a altas revoluciones (3000 rpm) y con relaciones de transmisión mínimas (50:1), por el contrario, los Harmonic Drives de alta potencia pueden llegar hasta 10.000 N.m de capacidad de par.

El Harmonic Drive Ideal

Al Harmonic Drive se le asigna una relación de transmisión N , que describe su posición, velocidad y comportamiento de par.

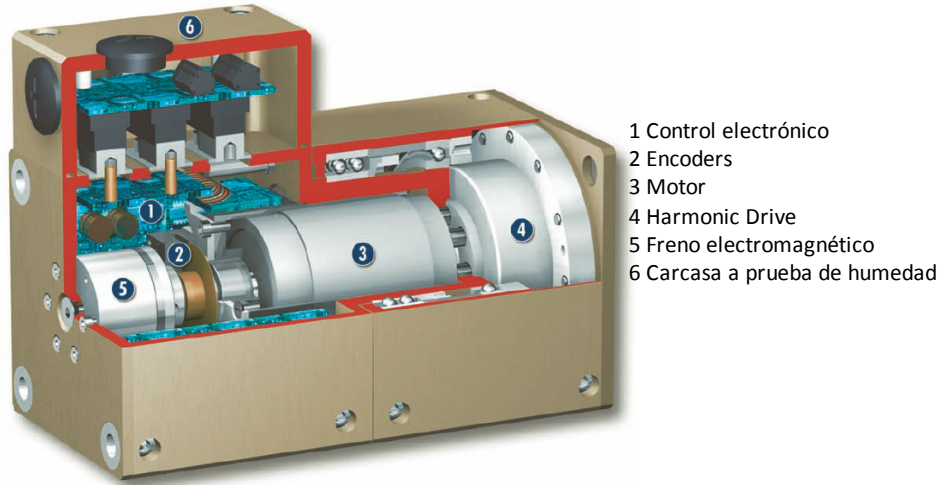
En particular, conociendo el valor de N y la posición de dos de los tres componentes del HD, puede saberse la rotación ideal del tercer componente, mediante la ecuación 2.1:

$$\theta_{WG} = (N + 1)\theta_{CS} - N\theta_{FS} \quad (2.1)$$

donde ω_{WG} es la Rotación angular del Wave Generator, θ_{CS} es la Rotación angular del Circular Spline y θ_{FS} es la Rotación angular del FlexSpline.

De manera similar, dado que N es una constante bajo suposiciones ideales, la derivada de esta relación produce una restricción de velocidad angular dada por la ecuación 2.2:

$$\omega_{WG} = (N + 1)\omega_{CS} - N\omega_{FS} \quad (2.2)$$



- 1 Control electrónico
- 2 Encoders
- 3 Motor
- 4 Harmonic Drive
- 5 Freno electromagnético
- 6 Carcasa a prueba de humedad

Figura 2.12: Partes internas del MPC.

De las ecuaciones 2.1 y 2.2 se aprecia que, si $\omega_{CS}=0$, el Wave Generator rotará N veces más rápido que el FlexSpline en la dirección opuesta. En una configuración diferente, un FlexSpline fijo produce que el Wave Generator rote $(N + 1)$ veces más rápido que el Circular Spline en la misma dirección. Aplicando la ley de conservación de la energía a los 3 componentes del HD, el par ideal se define como:

$$\tau_{WG} = \frac{1}{N + 1} \cdot \tau_{CS} = -\frac{1}{N} \cdot \tau_{FS} \quad (2.3)$$

De esta identidad, se puede comprobar que, el par en cada puerto del HD conduce al par ideal en los otros dos. También se observa que el par en el FlexSpline es semejante al par en el Circular Spline, que es aproximadamente N veces más grande que el par en el Wave Generator. Con las ecuaciones 2.1, 2.2 y 2.3, el comportamiento dinámico de un HD bajo suposiciones ideales, está completamente definido.

Los MPC usados tienen una relación de transmisión de 161:1 a una velocidad de 25 rpm, la ubicación del Harmonic Drive dentro de los MPC se muestra en la figura 2.12

Los módulos de comunicación es una parte esencial en la interface hombre-máquina del control maestro y los MPC, a continuación se explica el protocolo de comunicación usado.

2.2.4. Protocolo de comunicación

El protocolo de comunicación usado es CAN (acrónimo del inglés Controller Area Network) desarrollado por la firma alemana Robert Bosch GmbH [11], basado en una topología bus para la transmisión de mensajes en entornos distribuidos [11]. El procesador anfitrión (host) delega la carga de comunicaciones a un periférico inteligente, por lo tanto el procesador anfitrión dispone de mayor tiempo para ejecutar sus propias tareas.

Principales características de CAN

CAN se basa en el modelo productor/consumidor, el cual es un concepto, o paradigma de comunicaciones de datos, que describe una relación entre un productor y uno o más consumidores, CAN es un protocolo orientado a mensajes, es decir la información que se va a intercambiar se descompone en mensajes, a los cuales se les asigna un identificador y se encapsulan en tramas para su transmisión. Cada mensaje tiene un identificador único dentro de la red, con el cual los módulos deciden aceptar o no dicho mensaje.

De acuerdo al modelo de referencia OSI (Open Systems Interconnection, Modelo de interconexión de sistemas abiertos), la arquitectura de protocolos CAN incluye tres capas: física, de enlace de datos y aplicación, además de una capa especial para gestión y control del módulo llamada capa de supervisor.

Cuando un módulo necesita enviar información a través de una red CAN, puede ocurrir que varios módulos intenten transmitir simultáneamente. CAN resuelve lo anterior al asignar prioridades mediante el identificador de cada mensaje, donde dicha asignación se realiza durante el diseño del sistema en forma de números binarios y no puede modificarse dinámicamente. El identificador con el menor número binario es el que tiene mayor prioridad.

En cuanto a la detección y manejo de errores, un controlador CAN cuenta con la capacidad de detectar y manejar los errores que surjan en una red. Todo error detectado por un nodo, se notifica inmediatamente al resto de los nodos.

Para hacer la interface entre el control maestro y los MPC, se usó una tarjeta con un DSPIC 30f4012, para la transferencia de datos de la tarjeta con el control maestro, se usa

una Interface USB, para la comunicación entre la tarjeta y los MPC se realiza a través de protocolo Can, las prioridades de los mensajes y el manejo de errores se realizan con ayuda de las librerías propias del DSPIC. como se muestra en la figura 2.13.



Figura 2.13: Tarjeta de interface de protocolo USB a CAN

Toda la programación del DSPIC se realizó en lenguaje C, a continuación se explica el algoritmo 2.1 para la interface de comunicación.

1. Cuando se energiza la tarjeta de interface, el primer paso es llamar a la función stop la cual es la responsable de detener todos los MPC y posicionarlos en Home, se comprueba que hayan llegado si no se espera hasta que lleguen a esta posición.
2. El siguiente paso es iniciar un Ciclo donde se leen las posiciones generadas por el Controlador Maestro y se envían a cada MPC, una vez hecho esto se leen las posiciones actuales de los MPC y se mandan al Control Maestro, este Ciclo se repite hasta que se presione el Botón stop.
3. Cuando se presiona el botón Stop, se llama a la Función Stop la cual ya se explicó y se sale del Ciclo.

A continuación se muestra el algoritmo que se programó en la interface DSPIC.

Algoritmo 2.1 *Interface de comunicación*

Stop: *Detener todos los MPC "Halt"*
 Posicionar los MPC a "Home"

Hogar: *Si (Pos de MPC <> "Home")*
 ir a Hogar
 end Si

end Función

LLamar Función Stop:
 Leer Vel y Acel del Control Maestro
 Mandar Vel y Acel a cada MPC

Ciclo: *Leer Pos del Control Maestro*
 Escribir la Pos a MPC
 Pedir la Pos actual a MPC
 Enviar la Pos actual al Control Maestro
 Si (Boton_stop = verdadero)
 Llamar Función Stop
 ir a Fin
 end Si
 Ir a Ciclo

Fin End

2.3. Control maestro

El control maestro es el encargado de enviar las señales de control o señales de referencia y leer los sensores de cada uno de los MPC, todo esto con la finalidad de controlar en lazo abierto o lazo cerrado los MPC y comprobar el correcto funcionamiento de los mismos como se muestra en la figura 2.14.

El controlador maestro puede trabajar en lazo abierto o lazo cerrado:

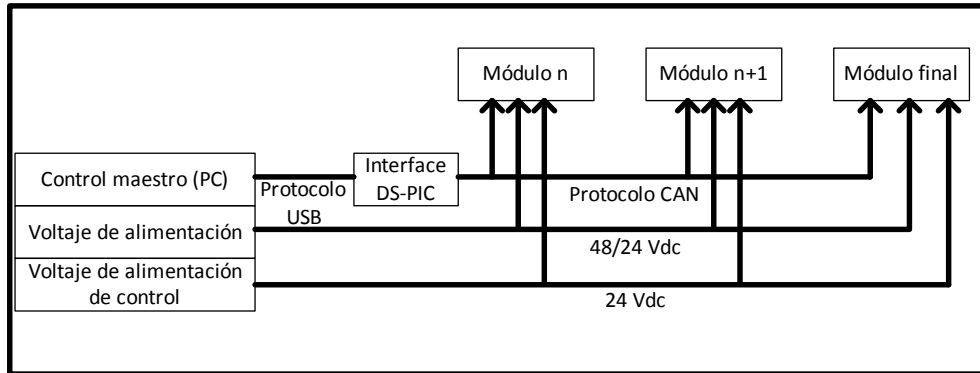


Figura 2.14: Comunicación de los módulos con el control maestro

Si se desea trabajar en lazo abierto, se debe seleccionar el modo de trabajo en posición. El controlador maestro es el responsable de generar las trayectorias así como de proporcionar las velocidades y aceleraciones de cada motor, el controlador de cada módulo es el encargado de llevarlo a la posición deseada.

En el caso de trabajar en lazo cerrado el controlador maestro selecciona el modo velocidad o corriente, para esto es necesario medir la posición del motor y compararla con la señal de referencia para obtener el error y calcular una ley de control.

EL control maestro (PC) calcula la ley de control con el programa de Simulink de Matlab y usando el software Quarc Real-Time Control para trabajar en tiempo real, los datos generados se transmiten a través del puerto USB hacia la interface cuya función es convertir los datos enviados en protocolo USB y transmitirlo a los MPC por protocolo Can.

La tarjeta de Interface también tiene la función de solicitar a cada MPC los valores de la posición medidos por los encoders absolutos, una vez que recibe los datos son enviados a través del protocolo USB al control Maestro para calcular la ley de control, el tiempo de muestreo es de 1 kHz.

2.4. Seguridad

La seguridad en el exoesqueleto es muy importante por los daños que pudiera ocasionar al momento de ser usado por alguna persona, por lo tanto existen 5 mecanismos de seguridad para evitar algún accidente, 3 son externos y 2 están integrados dentro de los MPC a continuación se describe cada uno de ellos.

1. Interruptores de límites integrados en los MPC, los cuales se activan cuando los motores alcanzan un valor angular, lo cual se utiliza para detener el motor.
2. Configuración de cada MPC se programan para que los motores alcancen ciertos ángulos los actuadores se detengan y no lastimen al usuario.
3. Boton de paro de emergencia controlado por el operador que supervisa el funcionamiento del exoesqueleto y detiene los actuadores en cualquier momento que sea oprimido, como se muestra en la figura 2.15.



Figura 2.15: Boton de paro

- 4 Programación desde el controlador maestro si la posición angular leída esta fuera de los límites permisibles de los movimiento del ser humano manda a detener los actuadores, como se muestra en la figura2.16.

- 5 Boton de paro de emergencia controlado por el usuario que detienen a los actuadores en cualquier momento que se active.

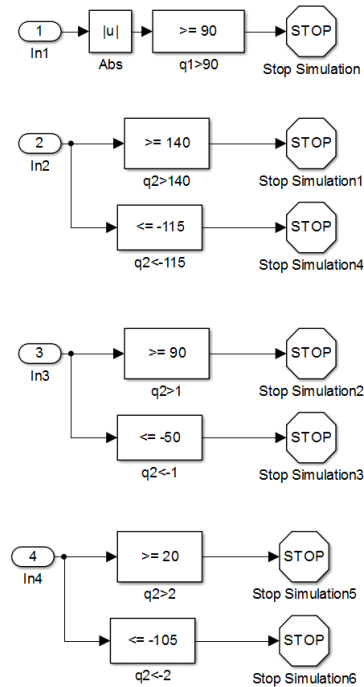


Figura 2.16: Módulo de Seguridad en simulink

2.5. Modelo del Exoesqueleto

La cinemática del robot estudia el movimiento del mismo con respecto a un sistema de referencia, describe el movimiento espacial del Robot como una función del tiempo y en particular por las relaciones entre la posición y la orientación del extremo final del Robot con los valores que toman sus coordenadas articulares.

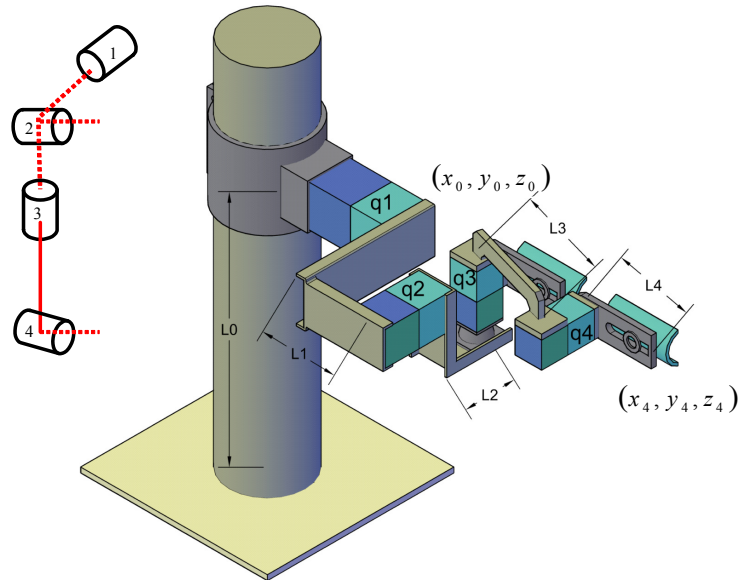


Figura 2.17: Exoesqueleto de 4 gdl.

2.5.1. Cinemática directa

Para calcular la cinemática directa del exoesqueleto de 4 grados de libertad que se muestra en la figura 2.19, se utilizó los parámetros de Denavit-Hartenberg (D-H) [71], por lo que es necesario cumplir la siguiente convención:

1. El eje z se selecciona en la dirección de eje de la articulación (articulaciones en revolución).
2. El eje x es paralelo a la normal: $x_n = z_n \times z_{n-1}$
3. El eje y se selecciona una vez seleccionado los ejes x y z , tomando en cuenta la regla de la mano derecha para sistemas coordenados (Producto Cruz : donde los dedos son: Pulgar **a**, índice **b**, medio **c**, $\mathbf{a} \times \mathbf{b} = \mathbf{c}$)

Los parámetros DH se seleccionan tomando en cuenta las siguientes reglas:

- a_i es la distancia entre el eje z_{i-1} y z_i , y es medido a lo largo del eje x_i
- d_i es la distancia ente el origen o_{i-1} y la intersección del eje x_i con z_{i-1} medido a lo largo del eje z_{i-1}
- α_i es el ángulo entre los ejes x_{i-1} y x_i medido en el plano normal a z_{i-1}
- θ_i es el ángulo entre los ejes x_{i-1} y x_i , medido en el plano normal a z_{i-1}

Tomando en cuenta la convención D-H se definió los marcos de referencia para cada articulación como se muestra en la figura 2.18, para la selección de los parámetros D-H de las tres primeras articulaciones, las cuales representan una articulación esférica, por lo tanto solo existe movimiento de los ángulos q_1 , q_2 y q_3 y no existe desplazamiento ya que los tres ejes coinciden en un mismo punto sin importar el giro de las articulaciones, por lo tanto se analizó como una articulación esférica con un desplazamiento en la articulación q_3 , para calcular los parámetros D-H de la articulación q_4 se consideró una articulación en revolución, quedando los parámetros como se muestra en la tabla 2.1

| Eslabón | a_i | d_i | α_i | θ_i |
|---------|-------|-------|------------------|------------|
| 1 | 0 | 0 | $\frac{\pi}{2}$ | q_1 |
| 2 | 0 | 0 | $-\frac{\pi}{2}$ | q_2 |
| 3 | l_3 | 0 | 0 | q_3 |
| 4 | l_4 | 0 | 0 | q_4 |

Tabla 2.1 Parámetros D-H del exoesqueleto.

El uso de coordenadas homogéneas para la localización de sólidos en un espacio dimensional se realiza a través de coordenadas de un espacio $(n + 1)$ -dimensional. Es decir, un espacio n -dimensional se encuentra representado en coordenadas homogéneas por $(n + 1)$ dimensiones, de tal forma que un vector $p(x, y, z)$ vendrá representado por $p(wx, wy, wz, w)$, donde w tiene un valor arbitrario y representan un factor de escala [7] .

A partir de la definición de las coordenadas homogéneas surge inmediatamente el concepto de matriz de transformación homogénea. Se define como matriz de transformación

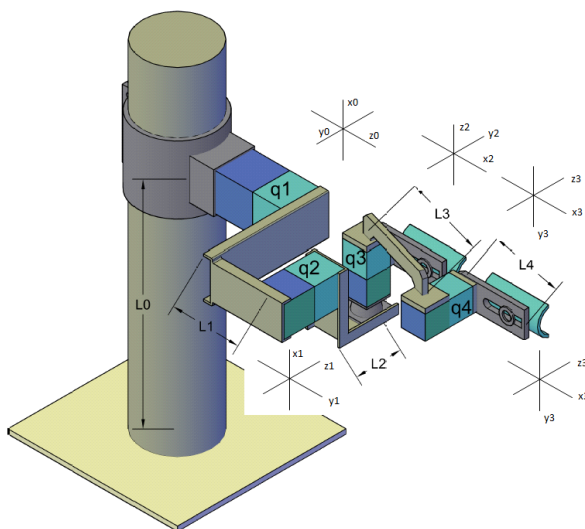


Figura 2.18: Marco de referencia para los parámetros D-H.

homogénea T a una matriz de dimensión 4×4 que representa la transformación de un vector de coordenadas homogéneas de un sistema de coordenadas a otro. Si se considera la transformación de perspectiva nula y el escalado global unitario, la matriz de transformación homogénea T queda de la siguiente forma:

$$T_n^0 = A_1 A_2 \cdots A_n = \begin{bmatrix} R_0^n & o_0^n \\ 0 & 1 \end{bmatrix} \quad (2.4)$$

$$A_i = \begin{bmatrix} \cos \theta_i & -\cos \alpha_i \sin \theta_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

donde R_0^n corresponde a una matriz de rotación, o_0^n corresponde al vector de traslación, a_i, d_i, α_i y θ_i son los parámetros D-H. Usando los valores de la tabla 2.1 y la ecuación 2.5,

las matrices homogénea A_i para cada eslabón son:

$$A_1 = \begin{bmatrix} \cos q_1 & 0 & \sin q_1 & 0 \\ \sin q_1 & 0 & -\cos q_1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (2.6)$$

$$A_2 = \begin{bmatrix} \cos q_2 & 0 & -\sin q_2 & 0 \\ \sin q_2 & 0 & \cos q_2 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (2.7)$$

$$A_3 = \begin{bmatrix} \cos q_3 & -\sin q_3 & 0 & 0 \\ \sin q_3 & \cos q_3 & 0 & 0 \\ 0 & 0 & 1 & l_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ y} \quad (2.8)$$

$$A_4 = \begin{bmatrix} \cos q_4 & -\sin q_4 & 0 & l_4 \cos q_4 \\ \sin q_4 & \cos q_4 & 0 & l_4 \sin q_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.9)$$

Donde la matriz de transformación homogénea es:

$$\begin{aligned} T_0^4 &= A_1 A_2 A_3 A_4 \\ &= \begin{bmatrix} -(c4)(s1s3 - c1c2c3) - c1s2s4 & (s4)(s1s3 - c1c2c3) - c1c4s2 & c3s1 + c1c2s3 & o_1^4 \\ (c4)(c1s3 + c2c3s1) - s1s2s4 & -(s4)(c1s3 + c2c3s1) - c4s1s2 & c2s1s3 - c1c3 & o_2^4 \\ c2s4 + c3c4s2 & c2c4 - c3s2s4 & s2s3 & o_2^4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

$$\begin{aligned}
o_1^4 &= l_2 s_1 - l_4 (c_4) (s_1 s_3 - c_1 c_2 c_3) - l_3 s_1 s_3 + l_3 c_1 c_2 c_3 - l_4 c_1 s_2 s_4 \\
o_2^4 &= l_4 (c_4) (c_1 s_3 + c_2 c_3 s_1) - l_2 c_1 + l_3 c_1 s_3 + l_3 c_2 c_3 s_1 - l_4 s_1 s_2 s_4 \\
o_3^4 &= l_3 + l_3 c_3 s_2 + l_4 c_2 s_4 + l_4 c_3 c_4 s_2
\end{aligned}$$

donde, $c_i = \cos q_i$ y $s_i = \sin q_i$ para $i = 1, 4$.

Para calcular la cinemática directa se toma en cuenta el vector de traslación, $o_0^4 = [x_4, y_4, z_4]^T$ que representa los componentes x_0, y_0, z_0 del origen con respecto al marco de referencia o_4 donde:

$$\begin{aligned}
x &= l_2 \sin q_1 - l_4 (\cos q_4) (\sin q_1 \sin q_3 - \cos q_1 \cos q_2 \cos q_3) - l_3 \sin q_1 \sin q_3 + \quad (2.10) \\
&\quad l_3 \cos q_1 \cos q_2 \cos q_3 - l_4 \cos q_1 \sin q_2 \sin q_4 \\
y &= l_4 (\cos q_4) (\cos q_1 \sin q_3 + \cos q_2 \cos q_3 \sin q_1) - l_2 \cos q_1 + l_3 \cos q_1 \sin q_3 + \\
&\quad l_3 \cos q_2 \cos q_3 \sin q_1 - l_4 \sin q_1 \sin q_2 \sin q_4 \\
z &= l_3 + l_3 \cos q_3 \sin q_2 + l_4 \cos q_2 \sin q_4 + l_4 \cos q_3 \cos q_4 \sin q_2
\end{aligned}$$

2.5.2. Modelo dinámico

El exoesqueleto para la extremidad superior es fijado en el brazo humano y su comportamiento es el mismo que el del brazo humano, anteriormente se mencionó q_1, q_2 y q_3 son las articulaciones del hombro y q_4 para el codo como se muestra en la figura 2.6, la cinemática y dinámica de las articulaciones son obtenidas del brazo humano. El marco de referencia para q_1, q_2, q_3 y q_4 son calculados de la tabla 2.1

Para calcular el modelo dinámico, se calcula primero la velocidad lineal la cual se obtiene por la diferenciación aplicando la regla de la cadena del efector final \dot{o}_0^n quedando como:

$$\dot{o}_0^n = \sum_{i=1}^n \frac{\partial o_0^n}{\partial q_i} \dot{q}_i = [J_{v_1} \cdots J_{v_n}] \begin{bmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_n \end{bmatrix}$$

donde

$$J_{v_i} = \frac{\partial o_n^0}{\partial q_i},$$

$$\begin{aligned} T_0^4 &= \begin{bmatrix} R_0^4(q) & o_0^4 \\ 0 & 1 \end{bmatrix} \\ &= T_0^{i-1} T_{i-1}^i T_i^4 \\ &= \begin{bmatrix} R_0^4 & R_0^i o_i^4 + R_0^{i-1} o_{i-1}^i + o_0^{i-1} \\ 0 & 1 \end{bmatrix}, \end{aligned}$$

o_i esta dado por los tres primeros elementos de la cuarta columna de T_0^i .

La dinámica del exoesqueleto incluye la cinemática de traslación, la cinemática de rotación y la energía potencial, donde la energía de traslación se calcula como:

$$K_T = \frac{1}{2} \dot{q}^T \left[\sum_{i=1}^n m_i J_{v_i}^T(q) J_{v_i}(q) \right] \dot{q}$$

La energía de rotación es

$$\begin{aligned} K_R &= \frac{1}{2} \sum_{i=1}^4 (\omega_0^i)^T I_i \omega_0^i \\ &= \frac{1}{2} \dot{q}^T \left[\sum_{i=1}^n J_{\omega_i}^T I_i J_{\omega_i} \right] \dot{q}. \end{aligned}$$

La energía potencial del manipulador es la suma de las energías en cada uno de los eslabones que se calcula como la masa multiplicada por la aceleración gravitacional y el peso en cada centro de masa.

$$\begin{aligned}
V_1 &= m_1 g l_{c1} \sin q_1 \\
V_2 &= m_2 g l_{c1} l_1 [\sin q_1 + l_2 \sin (q_1 + q_2)] \\
V_3 &= g m_3 (l_3 \cos q_1 \sin q_3 - l_2 \cos q_1 + l_3 \cos q_2 \cos q_3 \sin q_1) \\
V_4 &= l_2 \sin q_1 - l_4 (\cos q_4) (\sin q_1 \sin q_3 - \cos q_1 \cos q_2 \cos q_3) - l_3 \sin q_1 \sin q_3 + \\
&\quad l_3 \cos q_1 \cos q_2 \cos q_3 - l_4 \cos q_1 \sin q_2 \sin q_4 \\
V &= V_1 + V_2 + V_2 + V_3
\end{aligned}$$

La dinámica del exoesqueleto es derivada de la ecuación Euler-Lagrange

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = \tau, \quad L = K - V, \quad K = K_T + K_R$$

La ecuación dinámica es

$$M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + G(q) = \tau$$

Donde

$$\begin{aligned}
M(q) &= D(q) \\
G(q) &= [\phi_1, \dots, \phi_4]^T, \quad \phi_i = \frac{\partial V}{\partial q_i} \\
C(q, \dot{q}) &= \{c_{kj}\}, \quad c_{kj} = \sum_{i=1}^n c_{ijk} \dot{q}_i, \quad k, j = 1 \dots n
\end{aligned}$$

c_{ijk} son los símbolos de Christoffel [71]

$$c_{ijk} = \frac{1}{2} \left(\frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k} \right)$$

2.5.3. Singularidades

Como se explicó anteriormente el modelo del hombro se puede representar como una articulación esférica (ball and socket) como se muestra en la figura 2.7, cuando se mueve el exoesqueleto a la posición que se muestra en la figura 2.19, se puede observar que en esta posición se alinean las articulaciones q_1 y q_3 , lo que genera una singularidad, una singularidad se presenta cuando la configuración de un dispositivo pierde 1 gdl como resultado de la

alineación de dos ejes de rotación, lo que significa que dos articulaciones son colineales por lo tanto no son linealmente dependientes, esto ocurre cuando el ángulo de $q_1 = 0$ y $q_2 = -90$ grados, estos valores se evitan para evitar la singularidad. Para minimizar la frecuencia de estas ocurrencias, el eje de la articulación q_1 es ubicado de tal forma que la singularidad con la articulación q_3 , si se llega a presentar se de un movimiento que es antropomórficamente difícil de alcanzar..

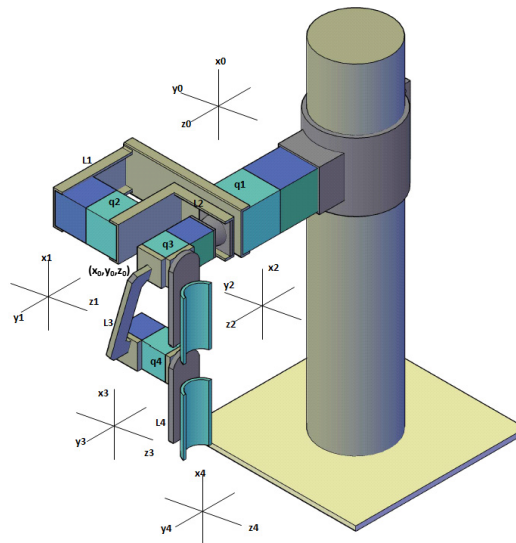


Figura 2.19: Exoesqueleto de 4 grados de Libertad

La existencia o no existencia de singularidades dependerá enteramente del espacio de trabajo, los espacios de trabajo esféricos siempre contendrán posiciones singulares, el desafío consiste en colocar la singularidad en una posición inalcanzable como es el límite del espacio de trabajo.

En este capítulo se explicó el diseño del exoesqueleto de 4 gdl de la extremidad superior, el criterio que se utilizó para la selección de sus articulaciones, para generar los movimientos del hombro y codo, se explicó los rangos de movimientos que realiza cada articulación tomando

en cuenta la cinemática del ser humano, se describió las partes que forman el exoesqueleto: actuadores, eslabones, sistemas de comunicación MPC y finalmente se presenta el modelo de cinemática directa y el modelo dinámico. En el siguiente capítulo se explica la metodología del aprendizaje por demostración para realizar el seguimiento de trayectorias en el exoesqueleto

Capítulo 3

Aprendizaje por Demostración

En este capítulo se presenta el Aprendizaje por Demostración ApD como un técnica que permite el seguimiento de trayectorias en robots, se describen los factores a tomar en cuenta al aplicar esta metodología, se explican los métodos para aprender una tarea y generar la tarea aprendida, además de cómo se evalúa la tarea aprendida con respecto a la tarea enseñada.

Las primeras investigaciones se iniciaron, a principios de los años 80's, el primer enfoque fue a través de razonamiento simbólico y fue aplicado en robots [19], [36], [58], la tarea de demostración era segmentada en una secuencia de transiciones estados-acciones-estados, las demostraciones normalmente consistían en mover manualmente el robot a través de una secuencia de movimientos y grabar la posición y orientación proporcionada por los sensores durante los movimientos guiados, usando estas posiciones se diseñaba un controlador de máquinas de estados finitos, la cual usaba un lenguaje simbólico con reglas “si-entonces”, describiendo los estados y las acciones de acuerdo a la relación simbólica, como por ejemplo “en contacto con”. “cerrar gripper”, “mover hacia”, etc.

Actualmente la forma de aplicar el ApD en los robots se puede dividir en dos partes: cómo enseñar una trayectoria y cómo generar la trayectoria aprendida, la primera parte se refiere a como se mapean las demostraciones hacia el robot es decir la forma en que se direccionan los estados y como se relaciona con las acciones, la segunda parte se refiere a

como el controlador del robot en base a los valores de los estados genera las salidas para reproducir la trayectoria deseada.

La ventaja de utilizar el ApD es que el robot aprende a realizar tareas en base a las necesidades y condiciones que existen en su entorno. Uno de los factores que se debe tomar en cuenta es que quede claramente especificado que es lo que se desea que aprenda el Robot, entre mas habilidad tenga el maestro mejor será la trayectoria aprendida.

3.1. ¿Qué es el ApD?

El Aprendizaje por Demostración en los robots se enfoca en la generación y desarrollo de algoritmos de forma genérica. El ApD, se basa en ejemplos o en demostraciones de ciertas habilidades o tareas que realiza un sujeto llamado maestro para enseñarle al robot que se le denomina aprendiz, con la finalidad que reproduzca la tarea enseñada [4].

Demostraciones

El ApD se basa en demostraciones, para llevar a cabo esto se necesitan tener todos los estados disponibles, donde los estados lo denotaremos como E y las acciones como A existe la posibilidad que los estados no se encuentren disponibles por lo tanto es necesario utilizar observadores por lo cual se utilizara los estados observados O , donde las acciones corresponden a la salida del robot.

Es importante resaltar que las demostraciones en algunos casos quedan muy por debajo del desempeño de un robot, por ejemplo la velocidad y precisión que tiene un robot comparada con las habilidades del maestro humano, por lo tanto el ApD aparte de realizar una tarea debe tener la capacidad de mejorar el desempeño de la tarea aprendida, es decir, si la velocidad de demostración es lenta con el algoritmo para reproducir la tarea aprendida debe mejorar la velocidad y precisión. En la investigación de [66], se desea coser a un paciente después de una operación, donde en las demostraciones se realizan a una velocidad y la tarea reproducida por el robot se puede aumentar 5 veces la velocidad enseñada.

Una parte esencial en el ApD es como se obtienen los datos de los estado-acciones, para los primeros estudios se utilizaban robots teleoperados. Actualmente se utilizan técnicas como la visión [31] el uso de guantes de datos los cuales tienen sensores [61], medidores de rango laser [29], a través de visión usando camaras [55], ó enseñanza kinésica [28], en este tipo de demostración el maestro con ayuda de su mano mueve las articulaciones para que el robot grabe sus estados-acciones con ayuda de los sensores montados en el robot.

La forma en que el maestro ejecuta las demostraciones y se las transmite al Robot se clasifica como:

- Demostración: la cual se refiere cuando el sensor está montado en el Robot y por lo tanto los estados de la demostración coinciden con los estados del aprendiz.
- Imitación: los sensores están en el maestro por lo tanto se requiere realizar un direccionamiento entre los estados del maestro y del aprendiz, en este caso existe la posibilidad que el número de estados del maestro no coincide con el del aprendiz.

Factores a tomar en cuenta al aplicar el ApD.

Son tres factores que hay que tomar en cuenta al aplicar el ApD, el problema de correspondencia, la generalización de trayectorias y el rechazo a perturbaciones a continuación se describen.

El problema de correspondencia se refiere a que las articulaciones del robot y del humano a veces no coinciden en el número de gdl y es necesario procesar los datos de las demostraciones para poder entrenar el robot. La correspondencia es más difícil cuando el maestro y el robot tienen diferentes arquitecturas, en ese caso, el robot puede sólo tomar ciertos estados del maestro y reproducir solo ciertos movimientos de las demostraciones. La correspondencia consiste en definir cuáles son los estados que deben reproducirse [16], propuso un marco matemático para expresar como una función de asignación, en términos de funciones de transferencia a través de distintos espacios, una aplicación de lo anterior es el movimiento de piezas de ajedrez en un tablero [?].

La generalización de trayectorias es necesaria porque el número de demostraciones es limitado, debido a esto no se pueden tomar todas las trayectorias posibles para realizar una misma tarea, por lo que es necesario que las demostraciones puedan ser generalizadas para que abarquen diferentes condiciones finales o iniciales.

El rechazo a perturbaciones es necesario, debido a que las tareas se pueden realizar en ambientes dinámicos y las condiciones pueden cambiar y el robot deberá poder generar la trayectoria aprendida, algunos ejemplos recientes muestran a robots poder adaptarse en línea para cumplir con alguna tarea, o esquivando obstáculos que no estaban en las demostraciones [49].

3.2. Métodos para enseñar una tarea

Los métodos para enseñar una tarea a un robot a través del ApD se clasifican como se muestra en la Figura 3.1:

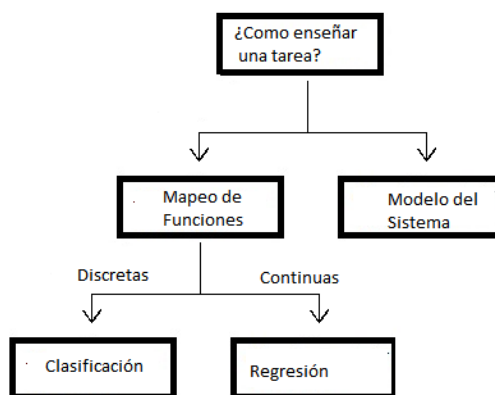


Figura 3.1: Cómo enseñar una tarea

3.2.1. Mapeo de funciones.

En base a las demostraciones se recolectan los datos de los estados y acciones, con estos datos se trata de aproximar una función probabilística que relacione los estados E a las acciones A .

$$f() : E \rightarrow A \quad (3.1)$$

Donde E son los estados, A son las acciones, y $f()$ es la función que mapea los estados con las acciones. El mapeo de funciones comprende los enfoques basados en clasificación y regresión.

Enfoque de clasificación

El enfoque basado en clasificación sus salidas y entradas son discretas, donde las entradas del clasificador son los estados del robot, y las acciones son las salidas. [70].

Las técnicas más comunes para llevar a cabo la tarea de clasificación son:

- Modelos de Mezclas Gaussianas (MMG).
- Árboles de decisión.
- Redes Bayesian.
- Modelos Ocultos de Markov.

Enfoque de Regresor

El enfoque basado en Regresor usa la información de las demostraciones para mapear los estados a las acciones donde los estados y las acciones son continuas.

3.2.2. Modelo del sistema dinámico

En base a las demostraciones se recolectan los datos de los estados y acciones con los cuales se determina un modelo con la ayuda de funciones de recompensa $R(s)$ se determina los estados E en función de las acciones A para asegurar que el modelo tiene las dinámicas que asegura que el robot cumple la tarea específica.

Los sistemas dinámicos ofrecen una solución de interés particular en el proceso de ApD, el objetivo es proporcionar robustez a las perturbaciones de cambios dinámicos en el ambiente. Las perturbaciones se pueden clasificar en tres partes: perturbaciones espaciales vs temporales, perturbaciones externas vs internas y perturbaciones instantáneas vs continuas [?].

3.3. Reproducción de una tarea automáticamente

Una vez que se llevan a cabo las demostraciones y los datos de entrada-acción son aprendidos por algún modelo, el siguiente paso es reproducir la tarea aprendida, en la figura 3.2 se muestra la clasificación de las técnicas de ApD a utilizar:

- Técnica basado en demostración de trayectorias.
- Técnica basado en modelado probabilístico.
- Técnica basada en modelos dinámicos.

3.3.1. La técnica basada en demostración de trayectorias

El número de demostraciones por parte del maestro generalmente es reducido, el enfoque es que las demostraciones se utilizan como una semilla para generar las trayectorias deseadas, para después ser optimizado mediante un proceso de auto mejora. Este auto-aprendizaje ajusta las diferencias del movimiento de las trayectorias reproducidas entre el robot y el maestro, y además, garantiza la estabilidad del comportamiento una aplicación de esta técnica es el aprendizaje de un golpe de derecha de tenis por un Robot [43].

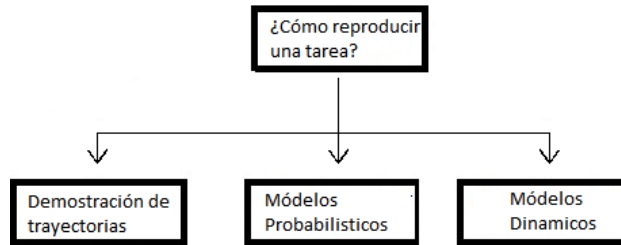


Figura 3.2: Cómo reproducir una tarea

Métodos de aprendizaje iterativo

El objetivo es que el robot empiece realizando tareas de manera correcta y gradualmente mejore su desempeño mientras es monitoreado por el maestro. Para aplicar esta técnica el primer paso es grabar las trayectorias que se desean realizar a través de humanos expertos los cuales operan al robot ya sea a través de un joystick o alguna otra forma para llevar a cabo las demostraciones, como son operados por humanos estas trayectorias se pueden desviar de la trayectoria deseada y entre más trayectorias más desviaciones existen aun cuando se cumpla el objetivo o se llegue al punto final.

Aprendizaje por demostración con múltiples datos de entradas

Esta técnica utiliza datos de entrada de diferentes maestros, si los maestros no tienen la misma habilidad para llevar a cabo una tarea es necesario utilizar un aprendizaje experto el cual decide que demostraciones tendrá mas peso para llevar a cabo la tarea de enseñarle al Robot, para esto se requiere utilizar aprendizaje por reforzamiento el cual se basa en la idea de tomar una recompensa o castigo si la trayectoria se aproxima a la deseada o si se aleja.

3.3.2. La técnica basada en modelos probabilísticos

Esta técnica se basa en utilizar la alta variabilidad inherente de los datos en las demostraciones, usando una técnica estadística, como Modelos de Mezcla Gaussianas (MMG) o ex-

plotar la robustez de los Modelos Ocultos de Markov (MOM) con el fin de codificar las variaciones temporales y espaciales de las señales, así como codificar un modelo, para recuperar y reproducir varios tipos de movimientos.

La desventaja de este enfoque es que se generan discontinuidades al tratar de recuperar la trayectoria del sistema, por lo tanto es necesario usar técnicas de interpolación para resolver este problema [13].

Modelos de Mezclas Gaussianas MMG

Para esta técnica el primer paso es realizar demostraciones, el MMG extrae la información más importante, las demostraciones sirven para generar los patrones del movimiento del robot al seguir las trayectorias deseadas. Con estos valores se estiman los parámetros utilizando la Probabilidad de Máxima Esperanza, una vez que se obtienen estos parámetros, el siguiente paso es utilizar alguna técnica de regresión para reproducir las trayectorias deseadas.

El modelo extraerá una representación continua de las restricciones con la información local de las variaciones y correlaciones de las variables. Por lo tanto provee una descripción genérica de los aspectos importantes de la tarea.

Dentro de los inconvenientes de esta técnica es que aún no se puede alcanzar la precisión que se requiere en el mundo real, además para este tipo de algoritmo entre más grados de libertad tenga el robot se requiere más tiempo de computo, por lo tanto es necesario hacer diferentes modificaciones a esta técnica.

Regresión Gausiana Local RGL

Esta técnica divide en cluster los entrenamientos generando una mejor respuesta que el proceso gaussiano simple, la ventaja que tiene es que a través de pocas demostraciones se puede lograr que el robot reproduzca la tarea especificada, como en [67], donde el robot se le enseñan con tres demostraciones y el aprende a realizar la tarea de servir una tasa de café.

Se puede descomponer la trayectoria dentro de un conjunto de puntos claves y recuperar una versión generalizada de las trayectorias a través de técnicas de ajuste spline [3], cuando

se utiliza la regresión, las entradas del regresor son los estados y las salidas continuas son las acciones del robot, la regresión típicamente es para un comportamiento de movimientos de bajo nivel.

Aprendizaje flojo

Este método de aprendizaje detiene el procesamiento de los datos de entrenamiento hasta que una pregunta específica se responde, por lo general, implica el almacenamiento de los datos de entrenamiento y realiza una búsqueda de datos relevantes en la base de datos para responder una consulta determinada. La relevancia es frecuentemente medida usando una función de distancia, con los puntos cercanos que tienen alta relevancia [75].

En la mayoría de los métodos de aprendizaje un modelo global es utilizado para ajustar todo los datos de entrenamiento si se responde a la consulta específica y es conocida durante el procesamiento de los datos de entrenamiento.

Los modelos locales tratan de ajustar los datos de entrenamiento sólo en una región alrededor de la ubicación de la consulta específica. En cada uno de estos modelos locales se combinan los puntos cercanos a un punto de consulta para estimar la salida apropiada.

Modelos Ocultos de Markov (MOM)

Las demostraciones se codifican en MOM con el fin de descartar la variabilidad intrínseca de los movimientos de las trayectorias y para permitir la generalización de las trayectorias demostradas. Sin embargo, la codificación de datos utilizando MOM pueden llegar a un modelo que no sea el correcto si una parte de los datos es sometido a un ruido con gran varianza, por lo tanto es necesario descartar los datos que puedan contribuir a una mala generalización.

La Máxima Esperanza (EM) es el proceso utilizado para entrenar a las MOM que tiene como objetivo encontrar un local óptimo para representar los datos considerando la transición y las distribuciones de las probabilidades de salida. Cuando la variabilidad de las señales de salida aumentan, el proceso de entrenamiento no pueden encontrar una solución local

satisfactoria por ejemplo, cuando se codifica dos señales de salida, que consta de una señal aleatoria y una señal invariable, el proceso de entrenamiento EM tratara de encontrar una representación probabilística para ambas señales, a veces a expensas de la señal invariante, que podría ser la señal relevante.

Este problema es particularmente evidente cuando tratamos de modelar señales múltiples con un modelo completamente conectado y/o una matriz de covarianza completa que representa la distribución de salida, esto es debido a que una amplio conjunto de parámetros deben ser estimados con pocos datos provenientes de las demostraciones.

Sin embargo, estas capacidades MOM son todavía ventajosas en la codificación de señales para la representación de patrones recurrentes y/o correlaciones. Por lo tanto, un modelo MOM genérico puede ser usado, al menos en algún punto del proceso, para codificar señales generalizadas.

Máquina de vectores soportados (MVS)

Esta técnica necesita los datos de entrada-salida como si fuera una red neuronal artificial (RNA), la principal diferencia entre una red neuronal y un MVS es que la primera requiere un amplia cantidad de datos para el entrenamiento y así lograr un óptimo desempeño y la MVS no lo requiere debido a que se basa en un procesos estadístico, con lo cual a diferencia de una RNA que al momento de entrenarse puede quedarse en un mínimo local, el SVM asegura el mínimo global si existe.

Diferentes investigaciones se han enfocados en el aprendizaje de la arquitectura de control utilizando RNA [2], [82] ó MOM [65]. Por ejemplo, las RNA y MOM tiene problemas relacionados con los mínimos locales a diferencia de la MVS.

La MVS no es sólo una herramienta teórica para análisis, sino también una herramienta para la creación de algoritmos prácticos para la estimación de funciones multidimensionales. Ejemplos de MVS son en la operación de seguimiento y captura de robots espaciales basado en aprendizaje y transferencia de estrategias de control por humanos [27].

Una de las ventajas de esta técnica es que con ayuda de las demostraciones se obtienen los pesos para una trayectoria deseada si se realizan ciertas modificaciones como cambios en

la trayectoria, los pesos se actualizan para seguir la nueva trayectoria.

3.3.3. Técnica basada en sistemas dinámicos

Este enfoque combina dos ingredientes: sistemas dinámicos no lineales para robustez a las trayectorias y técnicas de regresión no paramétrica para dar forma a los atractores de acuerdo a las trayectorias demostradas.

El enfoque es empezar con un sistema dinámico simple y transformarlo dentro de un sistema no lineal con dinámica de atractores prescritas para definir un término de aprendizaje forzado. El punto atractor es usado para movimientos primitivos discretos y el atractor ciclo límite para movimientos primitivos rítmicos

El controlador integrado consiste en un sistema dinámico con un único atractor estable, es necesario obtener un modelo de trabajo el cual sirve para modular las trayectorias generadas por el sistema dinámico cuando reproduce una tarea. Esto resulta en un marco general dirigido para aprender y reproducir una tarea, a pesar de las diferentes condiciones iniciales y los cambios que ocurra durante la ejecución de tareas. [41]

Preplaneación del movimiento de un sistema dinámico

El enfoque de los movimientos de aprendizaje con algoritmos de planificación para sistemas dinámicos autónomos utilizando el ApD, codifica las trayectorias a través una función independiente del tiempo que define la evolución temporal de la trayectoria. Las principales ventajas [?].

- Robustez a las perturbaciones temporales.
- Movimiento de planificación con sistemas dinámicos, lo que permite en línea la adaptación de perturbaciones espaciales, eliminando así la necesidad de algoritmos adicionales o volver a planificar una trayectoria completa.
- Ofrece un medio para generalizar los movimientos en las áreas de espacio de trabajo no cubiertos durante las demostraciones.

Una de las principales limitaciones sobre el uso de los sistemas dinámicos para la codificación de movimiento es cuando una demostración aprendida puede ser inestable. Una preocupación primordial para la generación de movimientos con sistemas dinámicos es por lo tanto garantizar la estabilidad de una estimación.

3.4. Evaluación de la tarea aprendida

Actualmente existen pocos estudios para comparar un algoritmo de ApD con otro, una razón es que la mayoría de los enfoques son probados en un solo dominio y una plataforma de Robot, con esto resulta que los algoritmos son optimizados para un dominio particular y no representa una solución general.

Un enfoque para explicar una métrica de desempeño de la tarea aprendida, sirve para expresar cuantitativamente la intención del usuario durante la demostración y para evaluar la fidelidad del rRobot al reproducir la tarea.

Una forma de hacer una función métrica, consiste en crear diferentes modelos basados en las demostraciones de la tarea a ejecutar con ligeras diferencias en las condiciones. La generalización de este proceso consiste en explotar la variabilidad inherente en varias demostraciones y extraer los componentes esenciales de las tareas como son: amplitud, velocidad, etc. Estos componentes deberán permanecer sin cambios a través de las demostraciones [21]. Existen métodos de evaluación que incluyen diferentes propósitos para áreas específicas de ApD [1] y potencialmente algunas para interacción con robots humanoides [62].

Existen investigaciones [63] donde se proponen funciones de costo como métricas para medir la reproducción de una tarea aprendida por el Robot.

$$J = g(E(t), A(t), t)$$

Donde $E(t)$ son los estados del robot, $A(t)$ son las acciones del Robot y J puede ser cualquier función de costo, sin pérdida de generalidad ésta podrá ser minimizada, como ejemplo si se desea que el robot llegue a un punto final J puede estar definido en variables de un sistema coordinado.

Definir una función de costo para medir el desempeño de la reproducción de una tarea aprendida es un problema complejo, en un escenario ideal J podría capturar el objetivo de la tarea y la calidad de reproducción de la tarea, por ejemplo si deseamos alcanzar una tasa con la mano, la función de costo podría penalizar la distancia al cuadrado de la mano a la tasa, pero si por alguna razón la trayectoria enseñada por el maestro es una curva, la función de costo requiere un término adicional que penalice la desviación de la trayectoria mostrada por el maestro y la realizada por la mano, o si se desea que la velocidad al ejecutar la tarea aumente, se tendría que tener un término extra que sea capaz de medir la velocidad de desempeño de la tarea especificada, con lo anterior se puede decir que seleccionar una función de costo dependerá muchas veces de lo que se desea medir.

En este capítulo se explicó en que consiste el ApD, que es una demostración, que factores deben tomarse en cuenta para llevar acabo el ApD, los distintos métodos que existen para enseñar una tarea y los métodos para reproducir la tarea aprendida, de los métodos aquí mencionadas se selecciono los Modelos Ocultos de Markov debido a que si existen diferencias en amplitud o velocidad en las demostraciones este descarta las diferencias en los movimientos de las trayectorias y permite la generalización de las trayectorias demostradas

Capítulo 4

Modelos Ocultos de Markov para el seguimiento de Trayectorias

En este capítulo, se describen los Modelos Ocultos de Markov (MOM) para el seguimiento de trayectorias, el cual es un método estadístico utilizado en el ApD, además se explica el uso del algoritmo de Lloyd para calcular los puntos claves de las trayectorias de entrenamiento, los cuales sirven para entrenar los MOM y estos puedan generar una trayectoria que se puede aplicar directamente en el espacio articular o en el espacio de trabajo.

Los MOM utilizados para el seguimiento de trayectorias se basan en la obtención de un modelo probabilístico, la forma de llevar a cabo esto es obteniendo estadísticamente las características intrínsecas de cada demostración y eliminando las variaciones que no tengan información relevante.

4.1. Modelos Ocultos de Markov

Para definir una MOM utilizaremos la definición de Rabiner 1989 [57] donde define una MOM como:

Definición 4.1 *doble proceso estocástico con un proceso subyacente que no es observable*

(oculto) pero que puede ser observado a través de otro conjunto de procesos estocásticos que generan la secuencia de observaciones.

Cuando las observaciones realizadas son caracterizadas como símbolos discretos seleccionados de un alfabeto finito se usa una función de densidad discreta dentro de cada estado del modelo.

En la aplicación de los MOM discretos se puede entender como una máquina de estados finitos donde en cualquier instante de tiempo se encuentra en cualquiera de N distintos estados $S = \{S_1, S_2, S_3 \dots S_N\}$. En intervalos de tiempo regulares, el sistema cambia de estado (el cual puede ser el mismo estado) o cualquier otro dependiendo de la configuración que tenga. Los instantes de tiempo asociados con los cambios de estado, se denotan por $t = 1, 2, \dots$, y el estado en el momento t se denota por q_t . Para el caso de los procesos modelados por una cadena de Markov discreta de grado 1, es decir que la probabilidad de transición solo depende del estado actual t_i y del estado anterior t_{i-1} se puede considerar que:

$$\begin{aligned} P(q_t = S_j | q_{t-1} = S_i, q_{t-2} = S_k, \dots) \\ = P(q_t = S_j | q_{t-1} = S_i) \end{aligned} \quad (4.1)$$

Además, solo se consideran los procesos en los que el lado derecho de la igualdad anterior son independientes del tiempo, por lo tanto creando un conjunto de probabilidades de transición de estados a_{ij} de la forma:

$$a_{ij} = P(q_t = S_j | q_{t-1} = S_i), 1 \leq i, j \leq N \quad (4.2)$$

4.1.1. Elementos de los Modelos Ocultos de Markov

A continuación, se definen formalmente todos los elementos que componen a un MOM.

1. El número de estados en el modelo N . Los estados serán denotados como $S = \{S_1, S_2, S_3 \dots S_N\}$, y el estado en el instante t como q_t .

2. El número de símbolos de observación distintos por estado M . Los símbolos corresponden usualmente a la salida física u observable del sistema modelado. Se denotan los símbolos como $V = \{v_1, v_2, \dots, v_M\}$.
3. La distribución de probabilidad de transición $A = \{a_{ij}\}$. Usualmente representada por una matriz en donde:

$$a_{ij} = P(q_t = S_j | q_{t-1} = S_i), 1 \leq i, j \leq N \quad (4.3)$$

Para el caso especial en el que cada estado puede llevar a cualquier otro estado en una sola transición, $a_{ij} > 0$ para todo i, j . En otros tipos de MOM, se tiene que $a_{ij} = 0$ para uno o más pares (i, j) .

4. La distribución de la probabilidad de observación del símbolo en el estado j , $B = \{b_j(k)\}$. Esta distribución de probabilidad representa la probabilidad de observar el símbolo k estando en el estado j donde:

$$b_j(k) = P[v_k \text{ en } t | q_t = S_j], \quad \begin{array}{l} 1 < j < N \\ 1 < k < M \end{array} \quad (4.4)$$

5. La distribución del estado inicial $\pi = \{\pi_i\}$, donde:

$$\pi_i = P[q_1 = S_i], 1 < i < N \quad (4.5)$$

Dados valores apropiados para N, M, A, B y π , el MOM puede ser utilizado como el generador de una secuencia de observaciones donde:

$$O = O_1 O_2 \dots O_T \quad (4.6)$$

(donde cada observación O_t es uno de los símbolos del conjunto V , y T es el número de observaciones en la secuencia) de acuerdo al siguiente algoritmo:

- 1 Seleccionar un estado inicial $q_1 = S_i$ de acuerdo con la distribución inicial π .
- 2 Establecer el instante de tiempo $t = 1$.
- 3 Seleccionar $O_t = v_k$ de acuerdo a la distribución de probabilidad de observación de los símbolos en el estado S_i , es decir, $b_i(k)$.
- 4 Hacer una transición al nuevo estado $q_{t+1} = S_j$ de acuerdo a la distribución de probabilidad para el estado S_i , es decir, a_{ij} .
- 5 Establecer el intervalo de tiempo $t = t + 1$; regresar al paso 3 si $t < T$; de otro modo terminar el proceso.

Con el procedimiento anterior, se podrían generar secuencias de observaciones correspondientes al modelo especificado por los parámetros N, M, A, B y n . Por conveniencia, se representan los parámetros de un MOM a través de la siguiente notación:

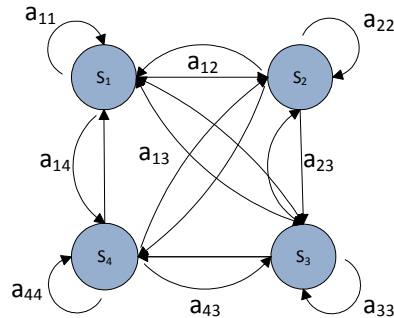
$$\lambda = (A, B, \pi) \tag{4.7}$$

4.1.2. Tipos de Modelos Ocultos de Markov

La siguiente clasificación de los MOM, no se debe exclusivamente a alguna de sus características. Esto quiere decir, que los dos primeros tipos cuya clasificación depende de los valores de las matrices de probabilidad de transición, son excluyentes entre ellos pero no con el tercer tipo. Así, podemos tener MOM que sean no ergódicos y autoregresivos o bien ergódicos y autoregresivos al mismo tiempo.

4.2.1. Ergódicos

Cuando un MOM tiene una matriz de probabilidad de estados completa (es decir, que no es cero para ningún a_{ij}) se dice que el MOM es ergódico. En este tipo de MOM cualquier estado puede ser visitado nuevamente y estas visitas no deben tomar necesariamente lugar en intervalos de tiempo periódicos. La figura 4.1 muestra un ejemplo de este tipo de MOM.

Figura 4.1: MOM ergódico con $N = 4$

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \quad (4.8)$$

4.2.2. No Ergódicos

En los casos en los que las matrices de transición del MOM pueden tener algunos valores "0", se dice que no son ergódicos. Por ejemplo, si se tiene una matriz triangular superior, se tendría un MOM como el de la figura 4.2. A estos modelos se les conoce también como modelos "izquierda-derecha", debido a que la secuencia de estados producida por la secuencia de observaciones siempre deberá proceder desde el estado de la izquierda, hasta el que esté a la derecha.

La propiedad fundamental de los modelos izquierda a derecha es que los coeficientes de las matrices de transición

$$a_{ij} = 0, \quad j < i \quad (4.9)$$

Además la probabilidad del estado inicial tiene la propiedad

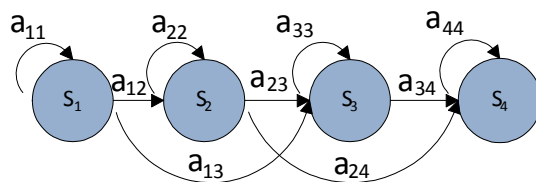


Figura 4.2: MOM no ergódico con matriz A triangular superior y $N = 4$

$$\pi_i = \begin{cases} 0, & i \neq 1 \\ 1, & i = 1 \end{cases} \quad (4.10)$$

Donde la secuencia de estados debe empezar en el estado 1, un ejemplo de una matriz de transición es:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 \\ 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & a_{44} \end{bmatrix} \quad (4.11)$$

4.1.3. Los 3 problemas de los Modelos Ocultos de Markov

Es común encontrar en la literatura que los MOM, son descritas en términos de 3 problemas fundamentales:

- Problema 1. Dada una secuencia de observaciones $O = O_1 O_2 \dots O_T$ y un modelo $\lambda = (A, B, \pi)$, ¿cómo se puede calcular eficientemente la probabilidad de la secuencia de observaciones $P(O|\lambda)$, de haber sido generada por el modelo λ ?
- Problema 2. Dada la secuencia de observaciones $O = O_1 O_2 \dots O_T$ y un modelo λ , ¿cómo se selecciona una secuencia de estados particular $Q = q_1 q_2 \dots q_T$ que sea óptima en algún sentido útil?

- Problema 3. ¿Cómo se pueden ajustar los parámetros del modelo $\lambda = (A, B, \pi)$, para maximizar $P(O|\lambda)$?

Solución al problema 1

La manera directa de calcular la probabilidad de que la secuencia de observación $O = O_1O_2\dots O_T$ dado un modelo un modelo $\lambda = (A, B, \pi)$ es enumerando cada posible secuencia de estados de longitud T (el número de observaciones) considerando una secuencia de estados fijos

$$Q = q_1q_2\dots q_T \quad (4.12)$$

Donde q_1 es el estado inicial, la probabilidad de la secuencia de observación O para la secuencia de estados es

$$P(O|Q, \lambda) = \prod_{t=1}^T P(O_t|q_t, \lambda) \quad (4.13)$$

Donde se asume independencia estadística de las observaciones por lo que tenemos que

$$P(O|Q, \lambda) = b_{q_1}(O_1) \cdot b_{q_2}(O_2) \cdot \dots \cdot b_{q_T}(O_T) \quad (4.14)$$

La probabilidad de una secuencia de estados Q puede ser escrita como:

$$P(Q|\lambda) = \pi_{q_1} a_{q_1q_2} a_{q_2q_3} \cdot \dots \cdot a_{q_{T-1}q_T} \quad (4.15)$$

La probabilidad conjunta de O y Q , es decir la probabilidad de que ocurran de manera simultánea, es el producto de los dos términos anteriores

$$P(O|Q, \lambda) = P(O|Q, \lambda)P(Q|\lambda) \quad (4.16)$$

La probabilidad de O se calcula sumando esta probabilidad conjunta sobre todas las posibles secuencias de estado q .

$$\begin{aligned}
 P(O|\lambda) &= \sum_{\text{all } Q} P(O|Q, \lambda)P(Q|\lambda) \\
 &= \sum_{q_1 q_2 \dots q_T} \pi_{q_1} b_{q_1}(O_1) a_{q_1 q_2} \cdots a_{q_{T-1} q_T} b_{q_T}(O_T)
 \end{aligned}
 \tag{4.17}$$

Un algoritmo que brinda una solución se llama algoritmo de Avance-Retroceso, apareció por primera vez en [5] y sirve para resolver los problemas 1 y 3 de los MOM. A continuación se describe el algoritmo.

Algoritmo de Avance Considérese la variable de avance

$$\alpha_t(i) = P(O_1, O_2, \dots, O_t, q_t = S_i | \lambda)
 \tag{4.18}$$

es decir, la probabilidad de la secuencia de observación parcial (hasta el instante de tiempo $t < T$) y el estado S_i en el instante t , dado el modelo λ como ejemplo de este algoritmo se puede ver la figura 4.3, para resolver $\alpha_t(i)$ inductivamente de la siguiente manera:

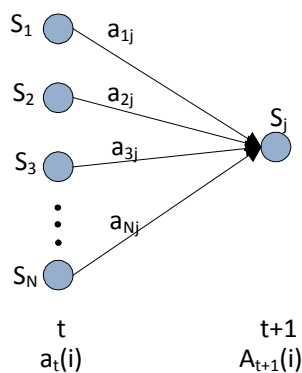


Figura 4.3: Secuencia de operaciones para el cálculo de la variable hacia adelante $\alpha_T(i)$

1. Inicialización

$$\alpha_1(i) = \pi_i b_i(O_1), \quad 1 \preceq i \preceq N
 \tag{4.19}$$

2. Inducción

$$a_{t+1} = \left[\sum_{i=1}^N \alpha_t(i) a_{i,j} \right] b_j(O_1), \quad 1 \leq t \leq T-1 \quad (4.20)$$

$$1 \leq j \leq N \quad (4.21)$$

3. Terminación

$$P(O|\lambda) = \sum_{i=1}^N \alpha_t \quad (4.22)$$

Algoritmo de Retroceso De manera similar, se puede calcular una variable de retroceso:

$$\beta_t(i) = P(O_{t+1}, O_{t+2}, \dots, O_T, q_t = S_i | \lambda) \quad (4.23)$$

Es decir, una variable que represente la probabilidad de la observación parcial de la secuencia desde $t + 1$, hasta el final dado el estado S_i en el momento t y el modelo λ , como ejemplo de este algoritmo se puede ver la figura 4.4, De nuevo, se puede resolver inductivamente de la siguiente manera:

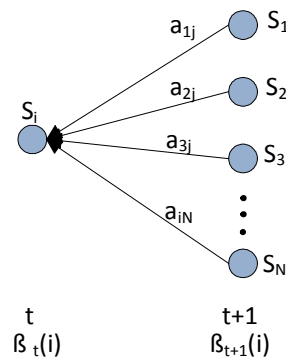


Figura 4.4: secuencia de operaciones para el cálculo de la variable hacia atrás $\beta_t(i)$

1. Inicialización

$$\beta_T(i) = 1 \quad 1 \leq i \leq N \quad (4.24)$$

2. Inducción

$$\begin{aligned} \beta_t(i) &= \sum_{j=1}^N a_{ij} b_j(O_{t+1}) \beta_{t+1}(j), \\ t &= T-1, T-2, \dots, 1 \quad 1 < i < N \end{aligned} \quad (4.25)$$

3. Terminación

$$P(O|\mu) = \sum_{j=1}^N \beta_{t+1}(j) \pi_j b_j(O_1) \quad (4.26)$$

Solución al problema 2

A diferencia del problema 1 de los MOM, para el problema 2 no existe una solución exacta. El problema consiste en encontrar una secuencia $S = S_1, S_2, \dots, S_T$ tal que la probabilidad de ocurrencia de la secuencia $O = O_1, O_2, \dots, O_T$ de esta secuencia de estados sea optima que cualquier otra secuencia de estados la dificultad consiste en decidir cuál es la secuencia optima es decir que criterio es el que se tiene que utilizar para optimizar.

La técnica de programación dinámica para encontrarla es el algoritmo de Viterbi [23].

Algoritmo de Viterbi Para encontrar una secuencia de estados óptima $Q = \{q_1, q_2, \dots, q_T\}$ para la secuencia de observaciones $O = \{O_1, O_2, \dots, O_T\}$ se necesita definir la cantidad:

$$\delta_t(i) = \max_{q_1, q_2, \dots, q_{t-1}} P(q_1 q_2 \dots q_t = i, O_1 O_2 \dots O_t | \lambda) \quad (4.27)$$

Esta cantidad representa la más alta probabilidad a través de un solo camino, en el instante t, que considera las primeras t observaciones y termina en el estado S_i . Por inducción se tiene que:

$$\delta_{t+1}(j) = [\text{máx} \delta_t(i) a_{ij}] b_j(O_{t+1}) \quad (4.28)$$

Para poder tener la secuencia de estados se debe llevar la cuenta de los argumentos que maximizan la ecuación para cada t y j . Esto se hace a través del vector $\Psi_t(j)$, siguiendo el procedimiento recursivo descrito a continuación.

1. Inicialización:

$$\delta_t(i) = \pi_i b_i(O_1), \quad 1 \leq i \leq N \quad (4.29)$$

$$\psi_t(i) = O \quad (4.30)$$

2 Recursión:

$$\delta_t(j) = \text{máx}[\delta_{t-1}(i) a_{ij}] b_j(O_t) \quad 2 < t < T \quad (4.31)$$

$$1 < j < N$$

$$\psi_t(j) = \underset{1 \leq i \leq N}{\text{argmax}}[\delta_{t-1}(i) a_{ij}] \quad 2 < t < T \quad (4.32)$$

$$1 \leq j \leq N$$

3 Terminación:

$$p^* = \underset{1 \leq i \leq N}{\text{max}}[\delta_T(i)] \quad (4.33)$$

$$q_T^* = \underset{1 \leq i \leq N}{\text{arg máx}}[\delta_T(i)] \quad (4.34)$$

4 Secuencia de estados:

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \quad t = T - 1, T - 2, \dots, 1 \quad (4.35)$$

Solución al problema 3

El objetivo de la solución a este problema es entrenar al MOM usando una secuencia de observaciones de modo que si se presentara otra secuencia con características similares después, la pueda reconocer (es decir, se tenga un valor de certeza de haber sido generada por el mismo MOM).

Formulas de reestimación de Baum-Welch El procedimiento iterativo de actualizar y mejorar los parámetros de un MOM para maximizar $P(O|\lambda)$ conocido como re-estimación de Baum-Welch [6], no es más que un equivalente del método Esperanza-Maximización (EM) como se menciona en [17].

Este procedimiento asume que se cuenta con un MOM conocido y que se seleccionarán parámetros $\lambda = (A, B, \pi)$ para maximizar $P(O|\lambda)$ localmente. Para describir el procedimiento, se define una nueva variable:

$$\xi_t(i, j) = P(q_t = S_i, q_{t+1} = S_j | O, \lambda) \quad (4.36)$$

donde $\xi(i, j)$ representa la probabilidad de estar en el estado S_i en el instante t y en el estado S_j en $t + 1$ dados el modelo λ y la secuencia de observaciones O . La secuencia de pasos requeridos para cumplir las condiciones que implica $\xi(i, j)$ se muestran en la figura 4.5

$\xi_t(i, j)$ puede ser definida en términos de las variables de avance-retroceso de la siguiente manera:

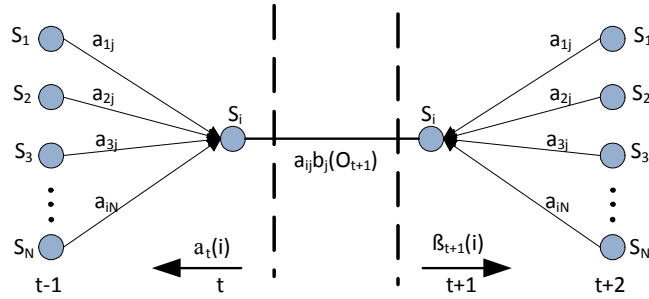


Figura 4.5: Secuencia de operaciones para el cálculo del evento conjunto de que el sistema se encuentre en el estado S_i en el tiempo t y en el estado S_j en el $t+1$

$$\xi_t(i, j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O|\lambda)} \quad (4.37)$$

$$\frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^N \sum_{j=1}^N \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}$$

Esta nueva definición de ξ tiene en el numerador solamente $P(q_t = S_i, q_{t+1} = S_j, O|\lambda)$. El numerador $P(O|\lambda)$ da a ξ la característica de ser una medida de probabilidad.

Se define $\gamma_t(i)$ como la probabilidad de estar en el estado S_i en el instante t dada la secuencia de observaciones O y el modelo λ ; a partir de esta definición se puede relacionar $\gamma_t(i)$ con $\xi_t(i, j)$ de la manera siguiente:

$$\gamma_t(i) = \sum_{j=1}^N \xi_t(i, j) \quad (4.38)$$

Si se suma $\gamma_t(i)$ a través del índice de tiempo t , se obtiene una cantidad que puede ser interpretada como el número esperado de veces que el estado S_i es visitado. De manera similar, la suma de $\xi_t(i, j)$ sobre t (desde $t = 1$ hasta $t = T - 1$) puede ser interpretada como la cantidad esperada de transiciones del estado S_i al estado S_j , esto es:

$$\sum_{t=1}^{T-1} \gamma_t(i) = \text{número esperado de transiciones desde } S_i \quad (4.39)$$

$$\sum_{t=1}^{T-1} \xi_t(i, j) = \text{número esperado de transiciones de } S_i \text{ a } S_j \quad (4.40)$$

Usando los conceptos anteriores, se propone las siguientes fórmulas de reestimación:

$$\bar{\pi}_i = \text{frecuencia (número de veces visitado) esperada del estado } S_i \text{ en } t = 1 \quad (4.41)$$

$$\begin{aligned} \bar{a}_{ij} &= \frac{\text{número esperado de transiciones de } S_i \text{ a } S_j}{\text{número esperado de transiciones de } S_i} \\ &= \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \end{aligned} \quad (4.42)$$

$$\begin{aligned} \bar{b}_j(k) &= \frac{\text{número de veces en el estado } j \text{ y observar el simbolo } v_k}{\text{número esperado de veces en el estado } j} \\ &= \frac{\sum_{t=1}^{T-1} \gamma_t(i)}{\sum_{t=1}^{T-1} \gamma_t(i)} \quad \text{s.t. } O_t = v_k \end{aligned} \quad (4.43)$$

Si usamos los parámetros del modelo inicial $\lambda = (A, B, \pi)$ para calcular los valores de las formulas anteriores para obtener $\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi})$, se puede probar que:

1. La definición del modelo inicial λ es un punto crítico de la función de probabilidad en cuyo caso $\lambda = \bar{\lambda}$; o bien

2. El modelo $\bar{\lambda}$ es más probable que el modelo λ en el sentido de $P(O|\bar{\lambda}) > P(O|\lambda)$. Basándose en el procedimiento anterior, si se reemplaza iterativamente $\bar{\lambda}$ por λ y repetimos la reestimación, se puede mejorar la probabilidad de que O sea observado por el modelo hasta alcanzar un límite seleccionado. Al resultado final de la reestimación se le llama estimado de mayor posibilidad del MOM.

Si se desea entrenar los MOM es necesario tener la secuencia de observación, para obtenerla se utiliza el algoritmo de Lloyd's para cuantizar las trayectorias, a continuación se explica la forma de obtener la secuencia de observación

4.2. Cuantización de trayectorias en el espacio articular usando el algoritmo de Lloyd's

Para llevar el proceso de entrenamiento de los MOM el primer paso consiste en mapear las trayectorias continuas de las demostraciones en valores discretos este proceso se le conoce como cuantización, los valores discretos obtenidos se les conoce como Libro de Código (Codebook) $C_i(t)$, donde t es el tiempo de entrenamiento.

La razón de usar los codebook se debe a que es imposible usar todos los datos muestreados en una demostración para entrenar el MOM, para poder entrenarla se usan los puntos claves, los cuales incluyen la información necesaria para reconstruir la trayectoria cuantizada.

Los métodos usados para la selección de puntos claves es tomando solo en cuenta la forma de la trayectoria de la demostración, donde se puede calcular solo basándose en la posición [80] o en la posición y velocidad [73]. El uso del algoritmo de Lloyd's es para seleccionar los puntos claves de forma automática solo seleccionando el número de puntos del codebook.

El codebook inicial es definido como $C_i(1)$. Una mala selección de condiciones iniciales podrían alcanzar un mínimo local. Un método heurístico [81] es usado para encontrar un mejor $C_i(1)$, éste método necesita bastante tiempo, otra forma es seleccionar $C_i(1)$ aleatoriamente de X_i^r , donde la r -ésima trayectoria de la i -ésima articulación.

Un codebook puede ser considerado como un vector de dimensión N_i por ejemplo, $C_i = [c_{i1} \cdots c_{iN_i}]^T$. La dimensión del codebook N_i era seleccionado anteriormente por el conocimiento de las geometrías de la trayectorias. Las demostraciones de cada articulación tienen su propio codebook definido como $C_i, i = 1 \cdots n$.

4.2.1. Algoritmo de Lloyd's

Se utilizo el algoritmo de Lloyd's para la selección de los puntos claves en una trayectoria es a través de la particion de los datos en células de tamaño uniformemente convexas [37]. Es un proceso repetitivo donde se calcula el centroide de cada conjunto particionado usando los diagramas de Voronoi, la ventaja de usar el algoritmo de Lloyd's es que el cálculo de los puntos claves se realiza de forma automática.

El objetivo de usar el algoritmo de Lloyd's es crear el codebook que minimiza el error de cuantización de la distribución de datos, para esto se usa la condición de vecino-cercano. Para un punto $q_i(l_1)$ en la trayectoria, se desea encontrar el elemento del codebook mas cercano, para esto se calcula

$$\min_{1 \leq j \leq N} |q_i(l_1) - c_{ij}|, \quad l_1 = 1 \cdots T, \quad i = 1 \cdots n \quad (4.44)$$

Si el elemento del codebook mas cercano es c_{ik} . La región R_{ij} ($i = 1 \cdots n, j = 1 \cdots N_i$) es definida como

$$R_{ik} = \{q_i(l_1)\} = [r_{ik1} \cdots r_{ikp_{ik}}], \quad k = 1 \cdots s_i^r \quad (4.45)$$

Donde p_{ik} es la longitud de la región R_{ik} , s_i^r es el número de la región del eslabón i y la demostración es r .

Si el centro de la región es R_{ij} ahora deberá ser c_{ij} , el nuevo centro de R_{ij} esta es la condición de centroide y puede ser calculada como

$$c_{ij} = \frac{1}{p_{ij}} \sum_{l=1}^{p_{ij}} r_{ijl} \quad (4.46)$$

El algoritmo normal de Lloyd's usa los puntos seleccionados para construir los centros [24], se calcula el promedio de los valores de todos los puntos de la región R_{ij} . La ventaja es que puede ser actualizado en línea. Con (4.45) y (4.46) se puede calcular $c_{ij}(t)$ recursivamente.

4.2 Cuantización de trayectorias en el espacio articular usando el algoritmo de Lloyd's 69

Se usa el error de cuantización para designar el criterio de paro, que se calcula como el promedio del error de cuantización para $X_{ij} = \{q_i(l)\}$, $l = 1 \cdots T$, y se define como:

$$\varepsilon_{ij}(t) = \frac{1}{T} \sum_{l=1}^T |q_i(l) - c_{ij}(t)| \quad (4.47)$$

Donde t es el tiempo calculado recursivamente. Se define el error relativo de cuantización como

$$\Delta\varepsilon_{ij}(t) = \left| \frac{\varepsilon_{ij}(t+1) - \varepsilon_{ij}(t)}{\varepsilon_{ij}(t+1)} \right| \quad (4.48)$$

El cálculo del codebook se detiene cuando el error relativo (4.48) es menor que $\bar{\varepsilon}$

$$\Delta\varepsilon_{ij}(t) \leq \bar{\varepsilon} \quad (4.49)$$

Donde $\bar{\varepsilon}$ es previamente definido como un límite superior.

En este trabajo se utilizó el algoritmo de Lloyd's para cuantizar las trayectorias de las articulaciones y obtener el codebook, tomando en cuenta la mínima distancia como se calculó en la ecuación (4.44), el codebook obtenido es un método recursivo y óptimo [24]. El promedio de cada ángulo de las articulaciones q_i tiene una distancia mínima con respecto a los elementos del codebook. La cuantización es con respecto al eje Y (eje del ángulo de la articulación), no se considera el eje del tiempo. Entonces la semejanza de las trayectorias con diferencias en el tiempo o velocidad puede ser medidas directamente, por lo tanto no es necesario el método Dynamic Time Warping como en [73].

Para entrenar el MOM, se necesitan los puntos claves y los símbolos de observación, los cuales son generados por el codebook mencionado anteriormente, Los puntos claves toman en cuenta el inicio, el final y los puntos centrales en el eje del tiempo del codebook, si el índice de tiempo de c_{ij} en (4.46) es definido como t_{ij} ($i = 1 \cdots n$, $j = 1 \cdots N_i$), entonces los puntos claves k_{ij} son calculados como

$$k_{ij} = t_{ij} + \frac{t_{i(j+1)} - t_{ij}}{2} \quad (4.50)$$

Por cada demostración el conjunto de puntos claves es:

$$K_i^r = [k_{i1} \cdots k_{i,v_i^r}]^T \quad (i = 1 \cdots n, r = 1 \cdots m)$$

Donde , v_i^r es el número de puntos claves K_i^r donde los puntos claves están en el centro de R_{ij} , en el inicio y el final de los puntos de las trayectorias son puntos claves, por lo tanto

$$v_i^r = s_i^r + 2. \quad (4.51)$$

Los símbolos de observación son los ángulos de las articulaciones en el tiempo en el que se calculan los puntos claves

$$O_i^r = [o_{i1} \cdots o_{iN_i}]^T \quad (i = 1 \cdots n, r = 1 \cdots m). \quad (4.52)$$

$$\begin{cases} o_{ij}(, q) = c_{i,j+1} & \text{Si } c_{ij} \leq q_i^r(k) \leq c_{i,j+1} \\ o_{ij}(s,) = j + 1 & \text{en otro caso} \end{cases} \quad (4.53)$$

donde $j = 1 \cdots N_i, k = 1 \cdots v_i^r, s$ es el símbolo de observación, q es el valor del codebook.

El algoritmo de Lloyd's permite fijar la dimensión de los puntos claves y de los símbolos de observación que son los mismos que los del codebook por lo que se evita el proceso de selección manual como [24],[80] y [73].

El siguiente algoritmo explica el proceso del cálculo del codebook C_i , los puntos claves K_i^r , y los símbolos de observación O_i^r .

Algoritmo 4.1 *cálculo del codebook, puntos claves y los símbolos de observación*

Obtener las demostraciones $Q = [Q_1 \cdots Q_n]^T$, $Q_i = [X_i^1 \cdots X_i^m]^T$

Fijar el tamaño del codebook N

For $i = 1$ a n

For $r = 1$ a m

$t = 1$

While $\Delta\varepsilon_{ij}(t) \leq \bar{\varepsilon}$

For $j = 1$ a N_i

Calcular la Región $R_{ij}(t)$ con ((4.45)

Calcular el centro $c_{ij}(t)$ con (4.46)

Calcular el error de cuantización relativo (4.48)

End for

$t = t + 1$

End while

Calcular los puntos claves K_i^r con (4.50)

Calcular los símbolos de observación O_i^r con (4.53)

End for

End For

Se usa el siguiente ejemplo para explicar cómo se usó el algoritmo de Lloyd's para generar el codebook, los puntos claves y los símbolos de observación.

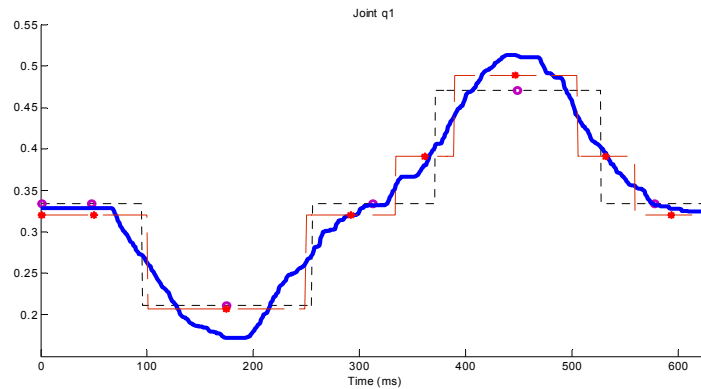


Figura 4.6: El codebook $N = 3$ (líneas discontinuas) y El codebook $N = 4$ (líneas continuas)

Ejemplo 4.1 *Un Robot dibuja un círculo en el espacio de trabajo, el ángulo de la articulación q_1 es mostrado en la figura 4.6, se muestra como el número de codebook N afectan el resultado. Las dos ondas cuadradas corresponden a la cuantización de un codebook con $N = 3$ y $N = 4$ como se muestra en la figura 4.6*

Cuando $N = 3$, se generan 3 cuantizaciones en el eje Y, lo cual genera 5 regiones en la trayectoria, $s_1^1 = 5$, $R_{11} \cdots R_{15}$. Los puntos claves son en el centro de cada segmento de

las señales cuadradas como se muestra en la figura 4.6, los puntos de inicio y final de las trayectorias se anexan como puntos claves, el número de puntos claves es K_1^1 es 7, $v_1^1 = s_1^1 + 2 = 7$.

Cuando $N = 4$ el número de regiones de la ecuación 4.45 es $s_1^1 = 7$ el número de puntos claves es $v_1^1 = 9$.

Si se tienen 3 demostraciones, entonces $r = 1, 2, 3$, como se muestra en la figura 4.7, en cada demostración las trayectorias tienen diferentes velocidades, para este tipo de problemas generalmente se usa Dynamic Time Warping para alinear las demostraciones en el tiempo, se uso el algoritmo de Lloyd's generando el mismo número de puntos claves donde, $N = 3$, $s_1^1 = s_1^2 = s_1^3 = 5$, para cada demostración y los símbolos de observación serian iguales.

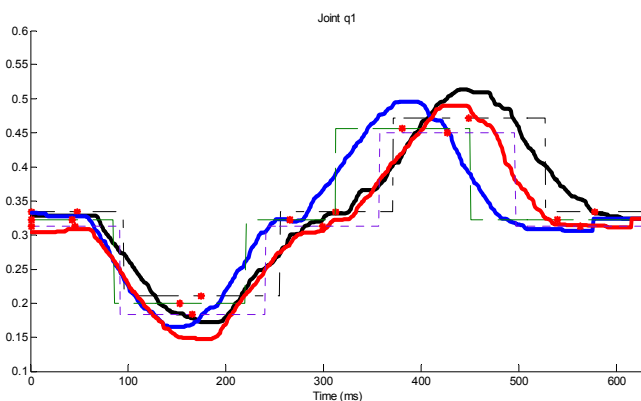


Figura 4.7: Diferentes demostraciones con el mismo números de símbolos de observación

Una vez que sé que se determina la secuencia de observación, el siguiente paso es entrenar los modelos MOM para obtener un modelo del movimiento.

4.3. Modelos Ocultos de Markov Modificados

EL uso de MOM como modelos probabilísticos para modelar los movimientos de demostraciones en Robots, tienen un buen desempeño cuando las demostraciones tienen variaciones

en amplitud y velocidad, que generalmente son las diferencias que se presentan en un entrenamiento de ApD, la ventaja de usar las MOM es que se puede aprender la trayectoria en el espacio articular o en el espacio de trabajo dependiendo de la trayectoria.

El planeamiento de trayectorias en el espacio articular evita el uso de la cinemática inversa, pero las demostraciones en el espacio articular son dependientes del tiempo, por ejemplo en la figura 4.8.A) se muestran las trayectorias de los ángulos q_1 y q_2 para dos demostraciones con diferentes velocidades, la gráficas en el espacio de trabajo se muestra en la 4.8.B) debido a que en el espacio de trabajo las trayectorias solo dan una relación del espacio y son independientes del tiempo, es indiferente la velocidad a la cual se realizaron, por lo que la planeación de las trayectorias es más fácil en el espacio de trabajo que en el espacio articular [73], [34], [68], y[25], existen pocos artículos en el espacio articular [10] y [52]

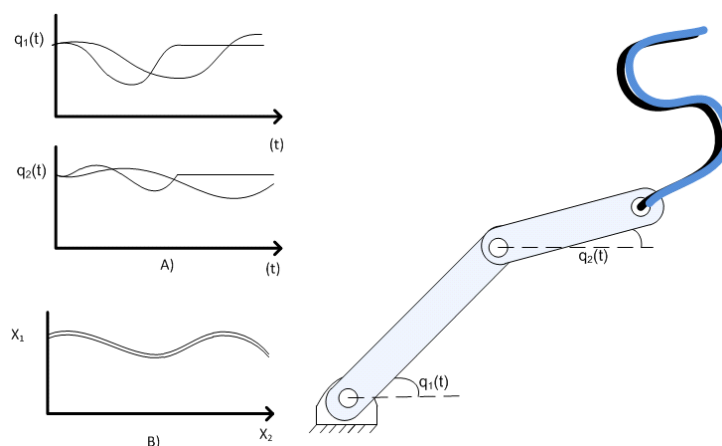


Figura 4.8: Trayectorias de un robot planar de 2 gdl. A) En el espacio articular y B) En el espacio de trabajo

El uso de MOM para planeamiento de trayectorias generalmente usan el algoritmo Dynamic Time Warping (DTW) [72], el cual sirve para la medición de similitud entre dos secuencias temporales que pueden variar en el tiempo o velocidad, en el caso de las demostraciones que tienen diferentes velocidades sirve para alinearlas en el dominio del tiempo.

La dinámica de un Robot de n -eslabones en el espacio articular puede ser escrito como

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) = \tau \quad (4.54)$$

donde $q \in \mathfrak{R}^n$ denota los ángulos de los eslabones, $\dot{q} \in \mathfrak{R}^n$ denota la velocidad de los eslabones, $M(q) \in \mathfrak{R}^{n \times n}$ es la matriz de inercia, $C(q, \dot{q}) \in \mathfrak{R}^{n \times n}$ es la matriz centrípeta y de coriolis, $G(q) \in \mathfrak{R}^n$ es el vector gravedad, $F \in \mathfrak{R}^n$ es el término de fricción, y $\tau \in \mathfrak{R}^n$ es el vector de control de entrada.

El objetivo de usar los MOM es generar un modelo de las demostraciones y generar una trayectoria deseada q^* para esto es necesario entrenar los MOM con ayuda de las demostraciones en el espacio articular $Q = [Q_1 \cdots Q_n]^T$, $Q_i = [X_i^1 \cdots X_i^m]^T$ donde la trayectoria X_i^r define la i -ésima articulación y r -ésima trayectoria, la cual es definida como $X_i^r = \{q_i(l)\}$, $l = 1 \cdots T_i^r$, T_i^r es el total de números de muestras de los ángulos de las articulaciones. Donde m son las trayectorias de cada articulación. La longitud de los datos T_i^r es diferente de una demostración a otra, debido a que tienen diferentes velocidades por lo que terminan en diferentes tiempos.

4.3.1. Generación de la trayectoria usando MOM modificado

Los MOM son usados para el planeamiento de trayectorias para la i -ésima articulación, el MOM λ_i puede ser generalizado de la ecuación 4.7 como:

$$\lambda_i = (A_i, B_i, \pi_i) \quad (4.55)$$

Donde A_i es la matriz de transición, B_i es la matriz de probabilidad de observación, π_i es el vector de probabilidad de los estados iniciales.

El número de estado A_i es igual al número de puntos claves v_i^r y depende de cada codebook C_i . El estado i -ésimo de la articulación es definido como:

$$S_i = [s_{i1}, \dots, s_{iv_i}]^T. \quad (4.56)$$

El número de estados ocultos v_i es el máximo número de puntos claves.

$$v_i = \max_{1 \leq r \leq m} \{v_i^r\} \quad (4.57)$$

Donde m es el número de la demostración, todas las demostraciones deberán tener el mismo número de puntos claves para poder entrenar el MOM.

El número de símbolos de observación es el mismo en todas las demostraciones, debido a que es igual al número de Codebook.

$$O_i = [o_{i1}, \dots, o_{iN_i}]^T \quad (i = 1 \dots n). \quad (4.58)$$

Configuración del MOM Una representación gráfica del MOM se muestra en la figura 4.9. El número de codebook $N_1 = 4$, el número de estados ocultos $v_1 = 7$. La configuración del MOM es conocida como topología izquierda-derecha.

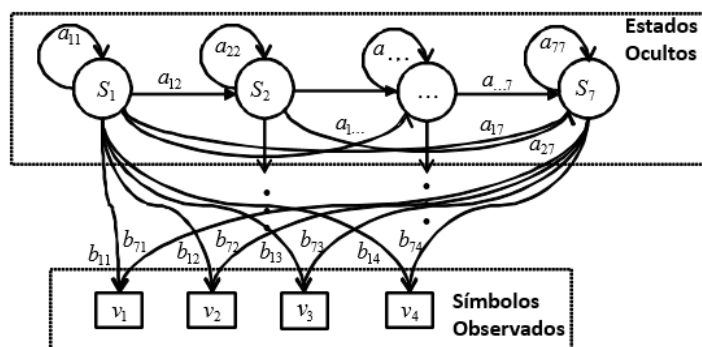


Figura 4.9: MOM para la trayectoria de la Figura 4.6

Los elementos iniciales a_{lj}^i de la matriz de transición $A_i(1)$ se seleccionan como una distribución uniforme que cumpla la siguiente condición:

$$\begin{cases} \sum_j a_{lj}^i = 1 & \text{Si } l \leq j \\ a_{lj}^i = 0 & \text{En otro caso} \end{cases} \quad i = 1 \dots n \quad (4.59)$$

Por lo tanto los estados no puede regresar a un estado anterior, pero pueden permanecer en el mismo estado o saltar a los siguientes estados.-

La condición de estado inicial para b_{lj}^i es: $b_{lj}^i = 0$ si o_{il} es el símbolo de la demostración l ; de otra forma

$$b_{lj}^i = P[o_{il} \text{ en } t | r(1) = s_{ij}] = 1. \quad (4.60)$$

Se propone modificar el MOM y asignar una cuarta tarea como:

$$\lambda_i = (A_i, B_i, \pi_i, F_i) \quad (4.61)$$

Donde $F_i = [f_{i1} \cdots f_{iN_i}]^T$ es la salida del MOM, lo cual no es mayor que una demostración en el MOM tradicional.

Para generar un modelo de la trayectoria, se debe entrenar el MOM con el algoritmo de Baum-Welch, para calcular la matriz de transición A_i y la matriz de probabilidad de observación B_i , donde $B_i = \{b_{lj}^i\} \in R^{v_i \times N_i}$, aquí v_i es el número de estados ocultos, N_i es el número de codebook. El primer paso es encontrar el estado con la máxima probabilidad posible del codebook por ejemplo, se quiere encontrar

$$\max_{1 \leq l \leq v_i} \{b_{lj}^i\}, \quad j = 1 \cdots N_i \quad (4.62)$$

El índice del máximo b_{lj}^i es sc_j . entonces se obtiene el conjunto de símbolos $\{sc_1 \cdots sc_{N_i}\}$ por lo que es necesario decodificar estos símbolos dentro de valores. La salida F puede ser representada como el valor codificado de los símbolos observados del codebook.

$$F_i = [f_{i1} \cdots f_{iN_i}] = \left[o_{i,sc_1}(\cdot, q) \cdots o_{i,sc_{N_i}}(\cdot, q) \right] \quad (4.63)$$

La salida del MOM F_i es un estado discreto, para poder generar una trayectoria suave, se usa la interpolación spline de 3-er orden (cubica)

$$s_t(x) = a_t(x - x_t)^3 + b_t(x - x_t)^2 + c_t(x - x_t) + d_t \quad (4.64)$$

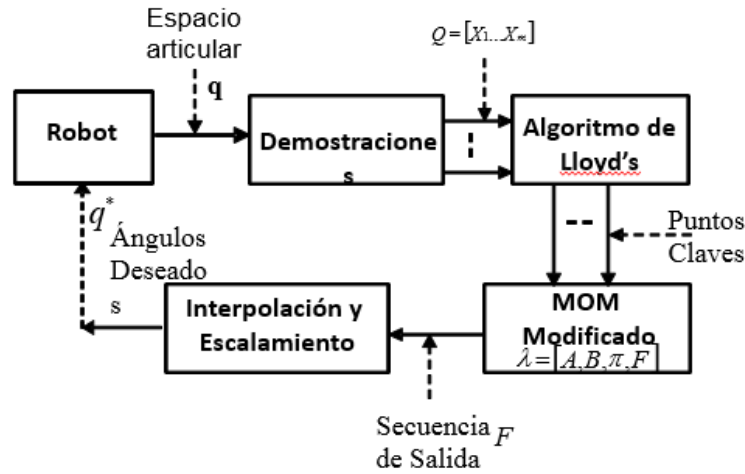


Figura 4.10: Generación de trayectorias por medio de MOM modificado usando el algoritmo de Lloyd's

Donde $t = 1 \dots q$, es el número de funciones por segmentos, el índice del tiempo de la salida son los puntos claves K_i^r en (4.50), el tiempo total de la trayectoria q^* es

$$T_i = \frac{1}{v_i} \sum_{j=1}^{v_i} k_{ij}, \quad i = 1 \dots n$$

Donde el punto discreto x_t es la salida del MOM f_{ij} , la función spline da una trayectoria suave $q^*(t)$ la cual se puede escalar en el tiempo, La trayectoria deseada es

$$q^*(\alpha t) \quad (4.65)$$

Donde α es un factor de escala del tiempo en el espacio articular. Finalmente se presenta un esquema completo del algoritmo presentado en la figura 4.10.

4.4. Simulaciones para un Robot de 2 gdl.

En esta sección se evalúa la efectividad de método propuesto aplicado a un brazo de robot de 2 gdl (manipulador tipo codo planar). El experimento es simulado en Matlab sin

optimización de código. La computadora trabaja con un procesador Intel Core i3 3.30Ghz. Se usó el Toolbox Murphy's HMM [45] para entrenar el MOM.

Para el robot de 2 gdl, la cinemática directa y la cinemática inversa son:

$$\begin{aligned} x &= l_1 \cos q_1 + l_2 \cos(q_1 + q_2) \\ y &= l_1 \sin q_1 + l_2 \sin(q_1 + q_2) \end{aligned} \quad (4.66)$$

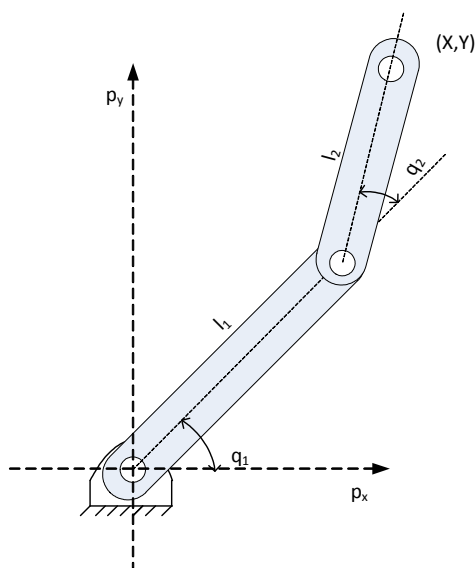


Figura 4.11: Robot de dos grados de Libertad

$$q_2 = \arctg \left(\frac{\pm \sqrt{1 - \cos^2 q_2}}{\cos q_2} \right) \text{ donde} \quad (4.67)$$

$$\cos q_2 = \frac{p_x^2 + p_y^2 - l_1^2 - l_2^2}{2l_1 l_2} \quad (4.68)$$

$$q_1 = \arctg \left(\frac{p_y}{p_x} \right) - \arctg \left(\frac{l_2 \sin q_2}{l_1 + l_2 \cos q_2} \right)$$

Se proponen dos ejemplos:

- Demostraciones con diferencias en tiempo
- Demostraciones con diferencias de amplitud en el espacio de trabajo

4.4.1. Demostraciones con diferencias en el tiempo.

Este ejemplo muestra que el método propuesto puede generar una trayectoria aun cuando existen diferencias en las demostraciones en el tiempo como se muestra en la figura 4.8, estas diferencias no se presentan en el espacio de trabajo solo en el espacio articular por lo que otras técnicas para generar la trayectoria en el espacio articular utilizan el método Dynamic Time Warping.

Para llevar a cabo el ejemplo se dibujaron 3 trayectorias en el espacio articular (q_1, q_2) como se muestra en la figura 4.13, utilizando la ecuación (4.66) se obtienen las trayectorias en el espacio de trabajo (x, y) como se muestra en la figura 4.12.

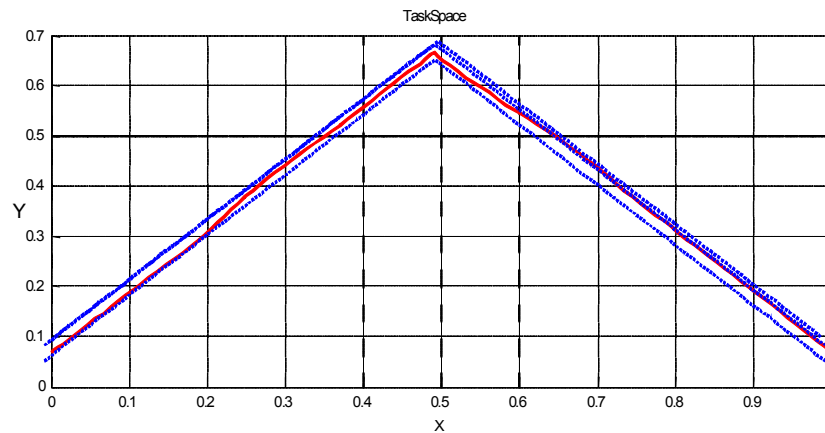


Figura 4.12: Demostraciones en el espacio de trabajo

Se utilizaron las trayectorias en el espacio articular (q_1, q_2) para entrenar los MOM modificados. Para el primer ejemplo se utilizó $N_1 = N_2 = 100$ como el número de codebook para cada articulación respectivamente. Estos valores son seleccionados de forma heurística para balancear la complejidad computacional y la exactitud del modelo. El número de puntos

claves obtenidos por el el algoritmo de Lloyd's para la articulación q_1 , $v_1^1 = 62$, $v_1^2 = 63$, $v_1^3 = 61$; para la articulación q_2 , $v_2^1 = 76$, $v_2^2 = 77$, $v_2^3 = 75$ por (4.57), El número de estados del MOM para cada articulación es $v_1 = 63$, $v_2 = 77$.

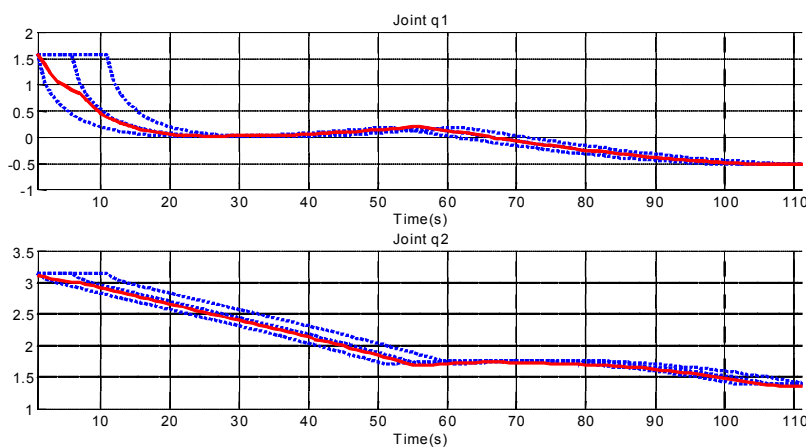


Figura 4.13: Demostraciones en el espacio articular con diferencias en tiempo

Una vez que el HMM es entrenada, se genera la trayectoria deseada (q_1^*, q_2^*) se selecciona $\alpha = 1$ de la ecuación (4.65), se genera la trayectoria q_1^* y q_2^* a la velocidad promedio de las demostraciones, la línea sólida en la figura 4.13 es el ángulo deseado para las articulaciones. Si se desea que el robot se mueva dos veces más rápido que en las demostraciones, se selecciona $\alpha = 0,5$, esto se muestra por una línea sólida en la figura 4.14.

Se usa la ecuación 4.66 para transformar el espacio articular (q_1^*, q_2^*) al espacio trabajo (x^*, y^*) , la trayectoria generada en el espacio de trabajo tiene la misma forma que las demostraciones como se muestra en la figura 4.12.

4.4.2. Demostraciones con diferencias de amplitudes en el espacio de trabajo

Para el segundo ejemplo, se muestran diferencias de amplitudes en el espacio de trabajo, se dibujan 3 figuras con diferentes amplitudes como se muestra en la figura 4.15, se utiliza

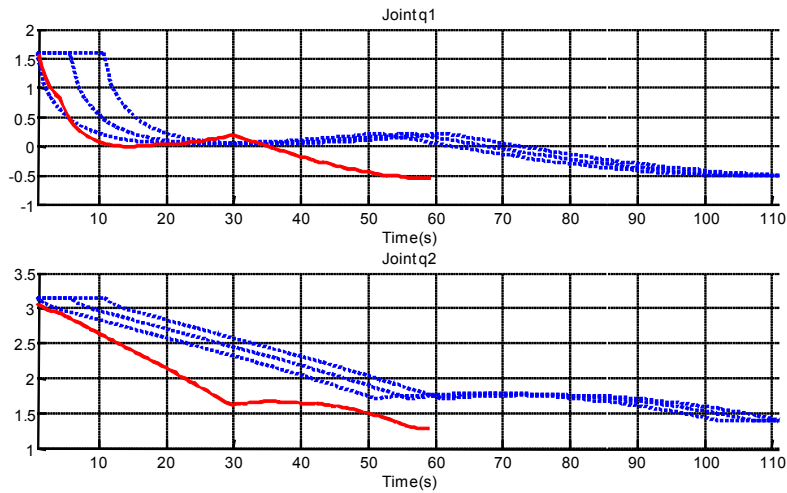


Figura 4.14: Trayectoria generada en el espacio articulado 2 veces más rápida que las demostraciones.

la ecuación de 4.67 para calcular la trayectoria en el espacio articulado como se muestra en la figura 4.16 donde la línea sólida es (q_1^*, q_2^*) , el cual es generada por el MOM modificado, el número de codebook seleccionado es $N_1 = N_2 = 100$. Los puntos claves son $v_1^1 = 60$, $v_1^2 = 51$, $v_1^3 = 55$, $v_2^1 = 64$, $v_2^2 = 68$, $v_2^3 = 71$. La trayectoria deseada (x^*, y^*) en el espacio de trabajo se muestra como una línea sólida en la figura 4.15 con lo que se puede apreciar que el objetivo es generar una trayectoria que tenga la misma forma que las demostraciones.

4.5. Simulación para un Robot Quirúrgico Raven.

EL sistema quirúrgico Raven es desarrollado en la Universidad de California - Santa Cruz, el cual incluye 4 articulaciones y 2 cámaras ver la Figura 4.17, el brazo quirúrgico es cinemáticamente construido como un pivote alrededor de un punto fijo lo cual puede ser imaginado como el punto donde el Robot entra en el cuerpo del paciente. El efector final puede ser movido en el espacio de 3 dimensiones y se utiliza un controlador maestro Omni

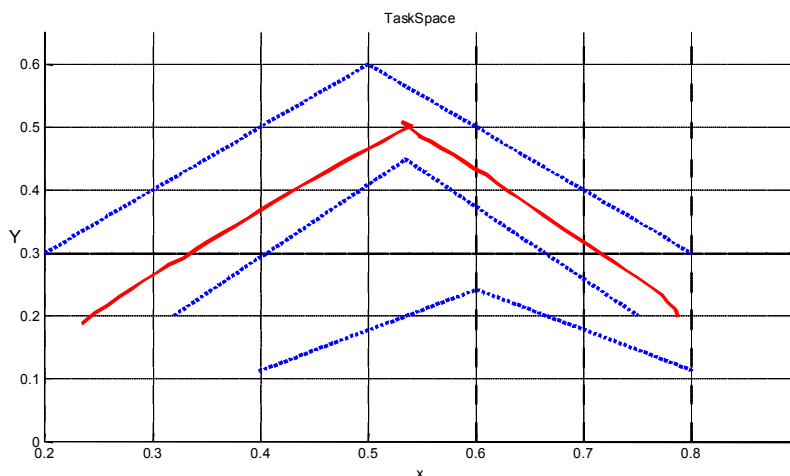


Figura 4.15: Demostraciones con diferentes amplitudes en el espacio de trabajo

Háptico Phantom. El Robot es controlado por 7 servomotores, cuatro motores son montados dentro del brazo para el control del pitch, roll, rotación y para activar las mordazas de la pinza, existen tres motores para controlar la posición en 3D.

Se dibuja la figura "8" en el espacio de trabajo tres veces, como se muestra en la figura 4.18, los ángulos son almacenados como trayectorias para el entrenamiento del MOM modificado como se muestra en la figura .4.19, cada demostración está formado por diferente número de datos los cuales son 629, 576 y 615.

El siguiente paso es seleccionar el número de codebook como $N_1 = N_2 = N_3 = 14$. los puntos claves son $v_1^1 = 29$, $v_1^2 = 32$, $v_1^3 = 30$, con estos valores se entrena el MOM para generar la trayectoria deseada en el espacio articular (q_1^*, q_2^*, q_3^*) , como se muestra en la figura 4.19. donde las líneas solidas es la trayectoria generada y las líneas punteadas son las demostraciones de las articulaciones q_i .

Las trayectorias deseadas sirven de referencia al controlador de bajo nivel, los tres motores usan un controlador PID para seguir las trayectorias como resultado el robot dibuja la figura "8" en el espacio de trabajo, que es la línea sólida y las líneas punteadas son las

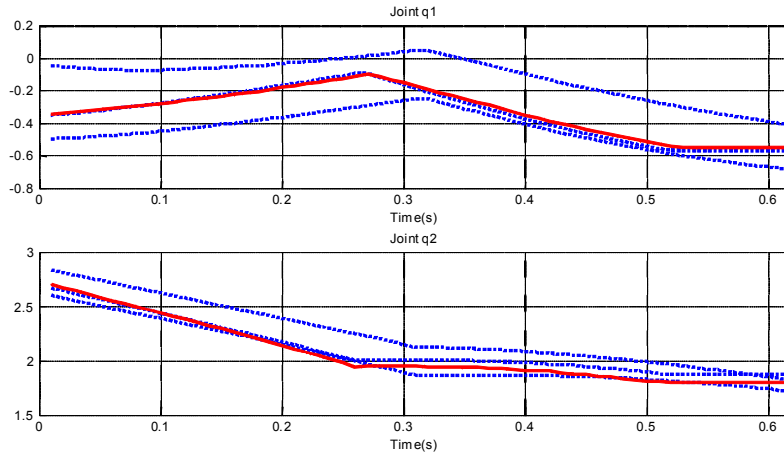


Figura 4.16: Trayectorias en el espacio articular

demostraciones como se muestra en la figura 4.18.

Una ventaja de este método es que la escala del tiempo de la trayectoria deseada puede ser modificada usando la ecuación (4.65), el funcionamiento superhumano puede ser realizado fácilmente por el MOM modificado como se muestra en la figura 4.20 donde se muestra que la trayectoria generada es dos veces más rápido que en las demostraciones

4.6. Comparación del algoritmo MOM modificado con otra investigación

Se comparó el resultado con el trabajo presentado con [12], el cual usan los MOM para la reproducción y reconocimiento de una tarea en el espacio de trabajo y el espacio articular aplicando la técnica propuesta se toma en cuenta los máximos y mínimos en las trayectorias para calcular los puntos claves como se muestra en la figura 4.21.

Uno de los requisitos al usar el algoritmo de Baulm-Welch para el entrenamiento de los MOM, es que el número de estados sea el mismo, si los estados son iguales al número de



Figura 4.17: EL sistema quirúrgico Raven

puntos claves, se necesita tener el mismo número de puntos claves para cada demostración, cuando eso no ocurre [12] propone el siguiente criterio "Una demostración es excluida del conjunto de entrenamiento si el número total de puntos claves está por debajo de un umbral de fijado" por lo tanto para la articulación q_1 es $N = 11, 18$ y 22 , no es posible tomar las tres trayectorias para entrenar el MOM.

Con la técnica de los MOM modificados, el número de puntos claves fue de $v_1^1 = 29$ por lo que fue posible realizar el proceso de entrenamiento como se muestra en la figura 4.19.

En este capítulo se explicó los MOM, sus principales características y los 3 problemas que puede resolver, se describió el algoritmo de Lloyd's para calcular los puntos claves, lo cuales se utilizan para obtener la secuencia de observación de salida, con la cual entrenamos los MOM modificados para generar una trayectoria en el siguiente capítulo se utilizan los MOM modificados para realiza el seguimiento de trayectorias en un exoesqueleto de 4 gdl.

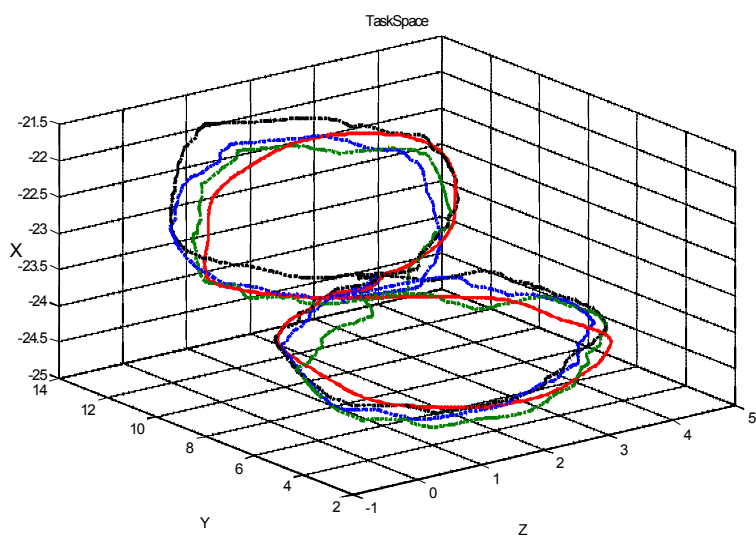


Figura 4.18: Trayectorias de Raven para la figura 8 en el espacio de trabajo

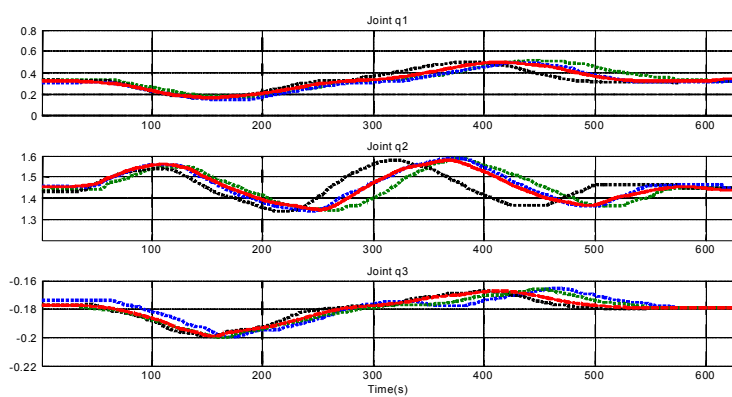


Figura 4.19: Trayectorias de Raven para la figura 8 en el espacio articulado

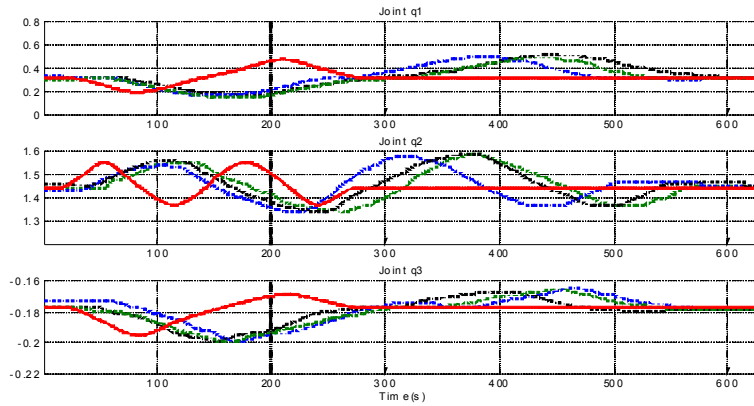


Figura 4.20: Raven dibujo de la figura "8" dos veces más rapido en el espacio articulado

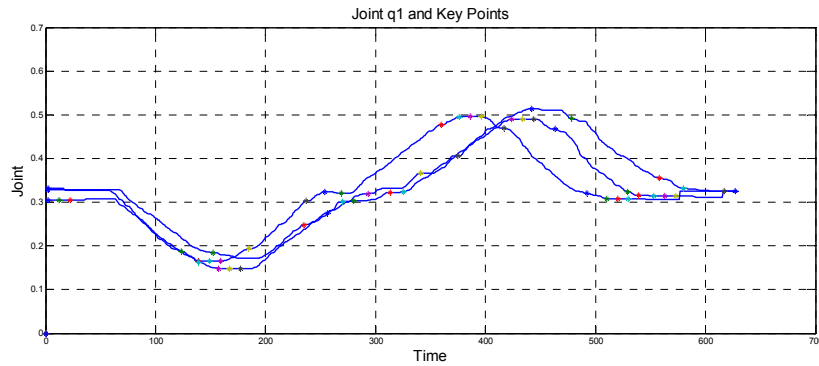


Figura 4.21: Usando la técnica propuesta por [12], se calculó los puntos claves para q_1 para cada demostración $N=11, 18$ y 22 .

Capítulo 5

Resultados experimentales

En este capítulo se presentan los resultados en tiempo real al aplicar los MOM modificados para realizar el seguimiento de trayectorias en el exoesqueleto de 4 gdl CinvesRobot-1 para la extremidad superior, se explica el diseño del controlador PD y el controlador maestro para

Para llevar a cabo la aplicación en tiempo real se realizó la secuencia de pasos que se muestran en el figura 5.1, el primer paso es hacer tres demostraciones, las cuales se le aplica el algoritmo de Lloyd's para obtener el codebook y puntos claves, con los cuales se entrena los MOM, una vez que se obtiene el modelo del movimiento, el siguiente paso es generar una trayectoria e interpolarla para generar una trayectoria continua mediante la técnica spline de tercer orden y finalmente aplicarla en el exoesqueleto.

Los pasos para aplicar el algoritmo de Lloyd's y MOM modificados ya se explicó en el capítulo anterior, a continuación se explica cómo se aplicó la trayectoria generada por los MOM para realizar seguimiento en el exoesqueleto de 4 gdl.

5.1. Controlador PD

La trayectoria generada por los MOM modificados se toma como señal de referencia por lo que es necesario aplicar un controlador para hacer el seguimiento de las trayectorias, por lo ue se utilizó un controlador PD con la siguiente ley de control

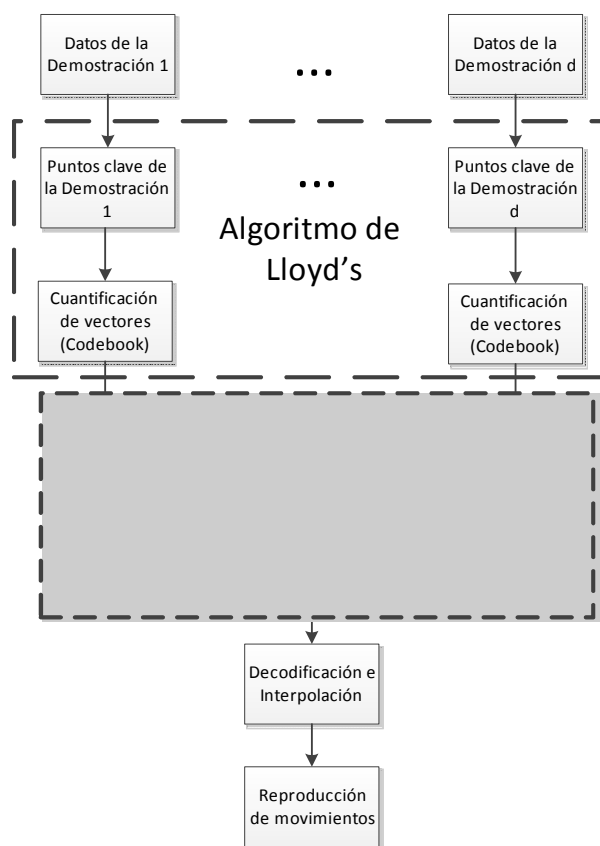


Figura 5.1: Diagrama a bloques para realizar seguimiento de trayectorias utilizando los MOM

$$\begin{aligned}\tau_i &= k_{pi}e_i + k_{di}\dot{e}_i \\ e_i &= \theta_{ri} - \theta_i\end{aligned}$$

Donde τ_i es el par del actuador, e_i , \dot{e}_i es el error en la articulación y derivada del error respectivamente, θ_{ri} es la trayectoria de referencia, θ_i es la trayectoria medida de la articulación, k_{pi} es la ganancia proporcional, k_{di} es la ganancia derivativa e i es la articulación q_i , $i = 1 \dots 4$.

τ_i es el par producido por el eje del motor, el subsistema de los componentes eléctricos y

mecánicos están acoplados a través de la siguiente ecuación:

$$\tau_i = K_i I_i$$

Donde K_i es la constante de torque del motor e I_i es la corriente del motor.

Para obtener la derivada del error se calculó a través de un filtro

$$\dot{e}_i = \frac{350s}{s + 350}$$

Se utiliza un filtro pasabajos para suavizar la derivada del error.

$$Filtro = \frac{500}{s + 500}$$



Figura 5.2: Exoesqueleto CinvesRobot-1

Una vez diseñado el controlador PD se realizaron pruebas en tiempo real, en el exoesqueleto CinvesRobot-1 que se muestra en la figura 2.2, se utilizó una computadora con

las siguientes características: procesador Intel Pentium 4@2.4 GHz y 2G RAM. El software de trabajo es Windows 7 con Matlab R2012, se utilizó el programa Quarc de Quanser para trabajar en tiempo real, se usa una tarjeta de interface de comunicación USB a CAN diseñada para la comunicación del control maestro los MPC y la frecuencia de muestreo es de 500 Hz.

El controlador PD se programó en el software de simulink como se muestra en la figura 5.3, los valores para la ganancia proporcional y derivativa son diferentes para cada eslabón, por ejemplo para q_1 la ganancia proporcional $K_{p1} = 74,2$ y la ganancia derivativa $K_{d1} = 1,6$ como se muestra en la figura 5.3.

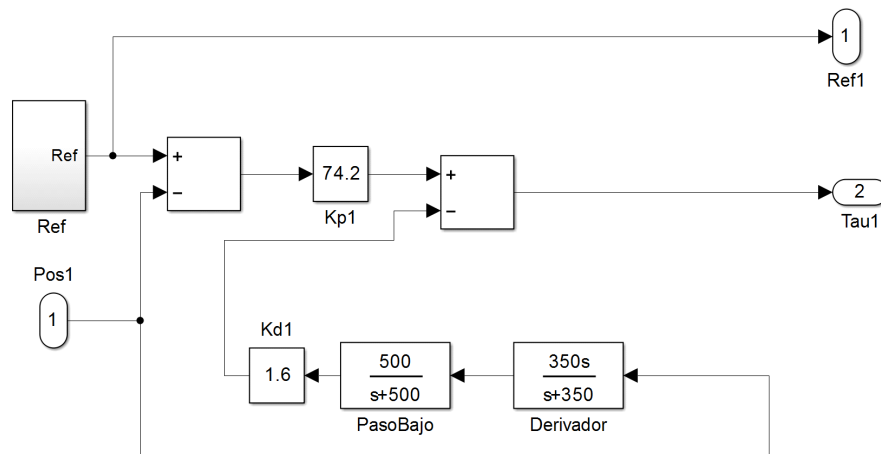


Figura 5.3: Controlar PD en Simulink para los MPC

Se aplicó una señal de referencia cuadrada a cada articulación, en la figura 5.4, se muestra que la señal de referencia y la posición angular de cada articulación, donde se puede ver que si se realiza la función de seguimiento.

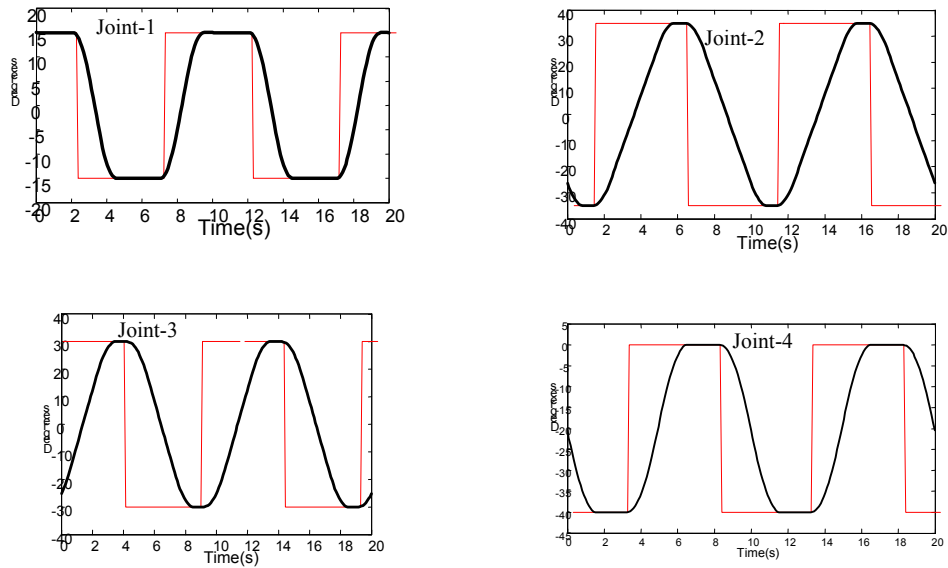


Figura 5.4: Respuesta a una señal cuadrada para cada eslabón

5.2. Controlador Maestro para el seguimiento de trayectorias.

El controlador Maestro se programó en Simulink, es el encargado de leer las señales de referencia, medir las posiciones de cada articulación y calcular las salidas de los controladores PD's, esta información se envía a cada MPC, como se muestra en la figura 5.5.

5.2.1. Ejemplo de seguimiento de trayectorias con diferentes amplitudes

Se eligió una trayectoria circular para realizar el seguimiento de trayectorias, se utilizó el algoritmo de Lloyd's, para lo cual se seleccionó el número de codebook como $N_1 = N_4 = 0$ y $N_2 = N_3 = 100$, donde las articulaciones q_1 y q_4 no son necesarias entrenar MOM, debido que su salida es igual a cero por lo tanto las articulaciones que se seleccionaran para entrenar

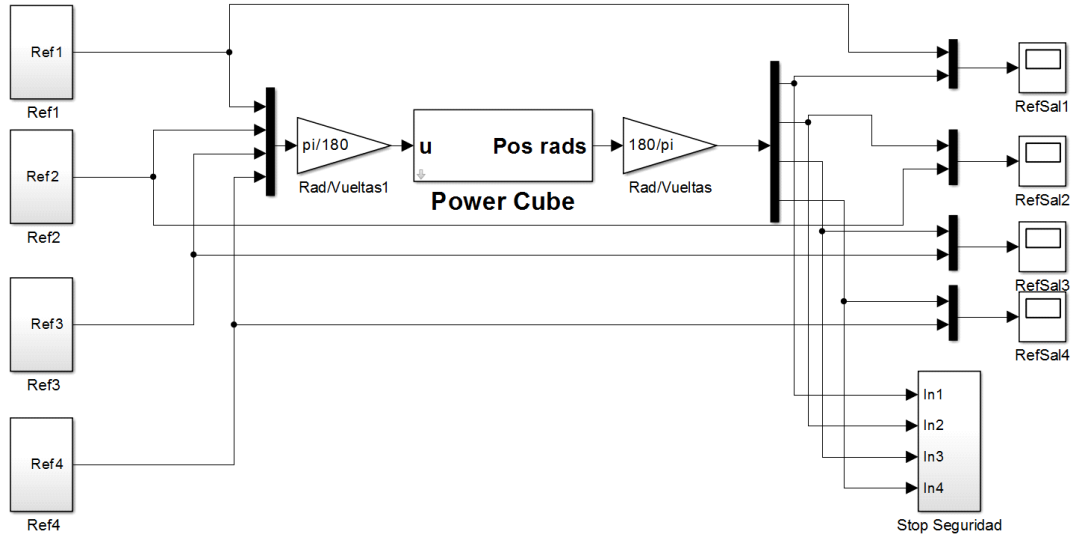


Figura 5.5: Controlador Maestro

a los MOM son q_2 y q_3 obteniendo los puntos claves $v_2 = 281$, $v_3 = 323$, las demostraciones se muestran en las figuras 5.6 a 5.9.

Una vez obtenidos los puntos claves se usa el algoritmo de Baum-Welch para entrenar los MOM para generar la trayectoria deseada en el espacio articular (q_2^*, q_3^*) , como se muestran en las figuras 5.7 y 5.8, donde las líneas punteadas es la trayectoria generada y las líneas solida son las demostraciones de las articulaciones q_i .

La trayectoria generada en el espacio de trabajo se muestra en la figura 5.10, para ver en simulación la trayectoria se utilizó la ecuación (2.10) del modelo de cinemática directa, donde las líneas delgadas son las trayectorias de demostración y la línea gruesa es la generada por el MOM, en el espacio de trabajo se muestran una fotografía de las trayectorias en la figura 5.17.

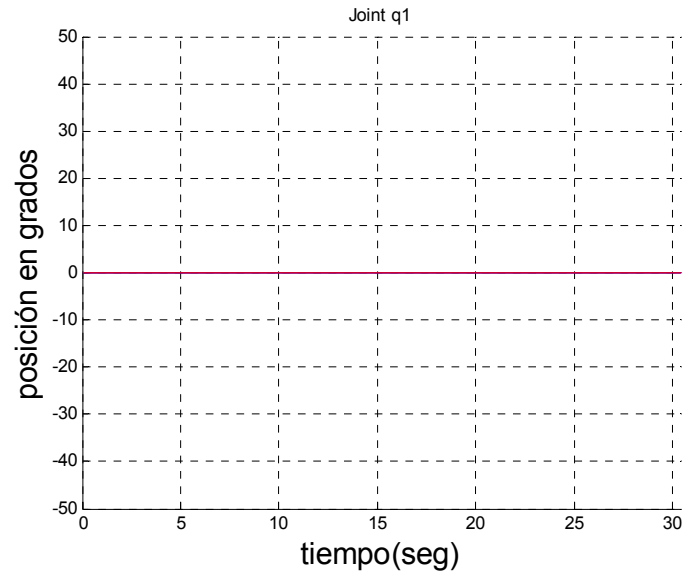


Figura 5.6: Trayectoria de la articulación q_1 en el espacio articulado con diferentes amplitudes

5.2.2. Ejemplo de seguimiento de trayectorias con diferentes velocidades

Se realizaron demostraciones con distintas velocidades para cada una de las 4 articulaciones como se muestran en las figuras 5.12- 5.15, el objetivo es que los MOM modificados sean capaz de generar la trayectoria deseada, para su entrenamiento se selecciono el Codebook $N_1 = N_2 = N_3 = N_4 = 100$ con estos valores se obtuvieron los siguientes puntos claves $v_1 = 746$, $v_2 = 288$, $v_3 = 732$ y $v_4 = 730$.

Una vez obtenidos los puntos claves se usa el algoritmo de Baum-Welch para entrenar los MOM para generar la trayectoria deseada en el espacio articular (q_2^*, q_3^*) , como se muestran en las figuras 5.12 a 5.15, donde las líneas punteadas es la trayectoria generada y las líneas solida son las demostraciones de las articulaciones q_i .

La trayectoria en el espacio de trabajo se muestra en la figura 5.10 se obtuvieron con la ecuación 2.10 de la cinemática directa del exoesqueleto, donde las líneas delgadas son las

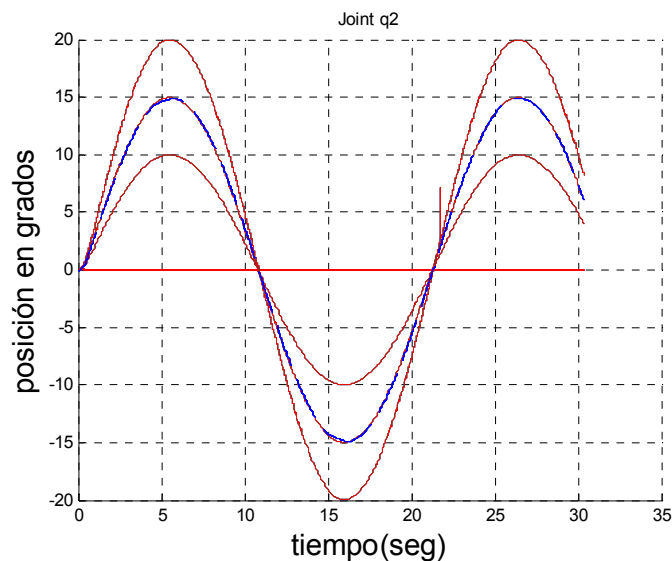
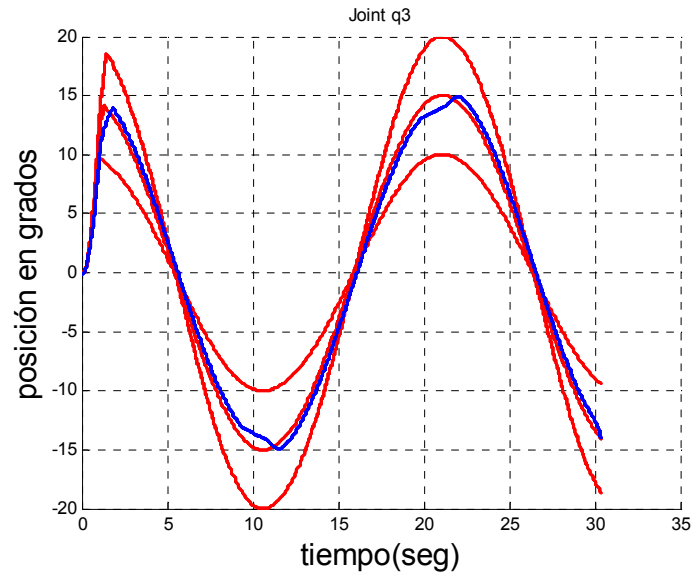
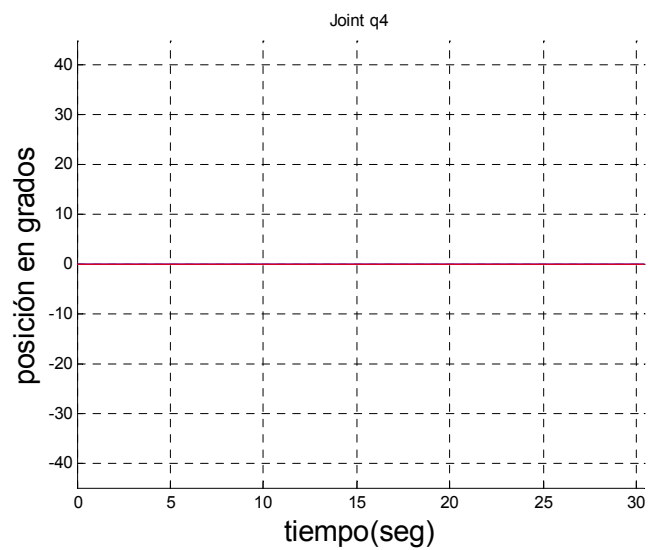


Figura 5.7: Trayectoria de la articulación q_2 en el espacio articulado con diferentes amplitudes trayectorias de demostración y la línea gruesa es la generada por el MOM, una fotografía de las trayectorias se muestra en la figura 5.17.

En este capítulo se mostro que el exoesqueleto es capaz de reproducir la trayectorias con dos diferentes tipos de problemas, demostraciones con diferencias en amplitud y demostraciones con diferencias en velocidad, en ambos casos el robot reproduce la tarea aprendida en base a tres demostraciones de manera correcta.

Figura 5.8: Trayectoria de la articulación q_3 en el espacio articulado con diferentes amplitudesFigura 5.9: Trayectoria de la articulación q_4 en el espacio articulado con diferentes amplitudes

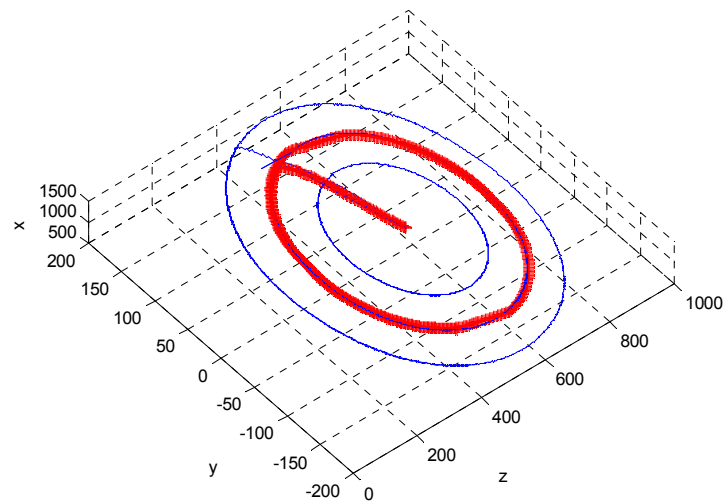


Figura 5.10: Trayectoria en el espacio de Trabajo X,Y,Z

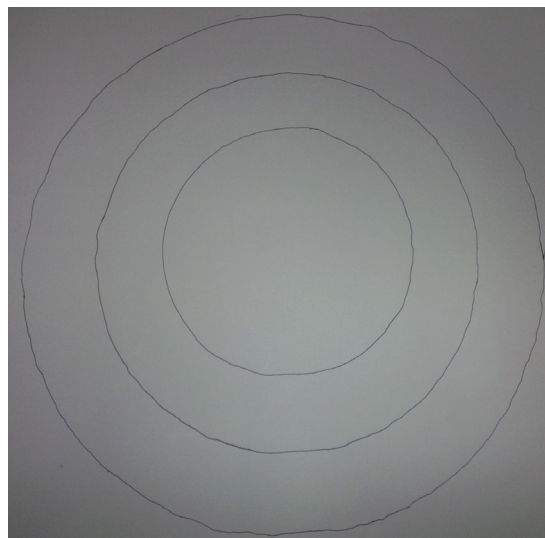


Figura 5.11: Fotografía de las trayectorias en el espacio de trabajo

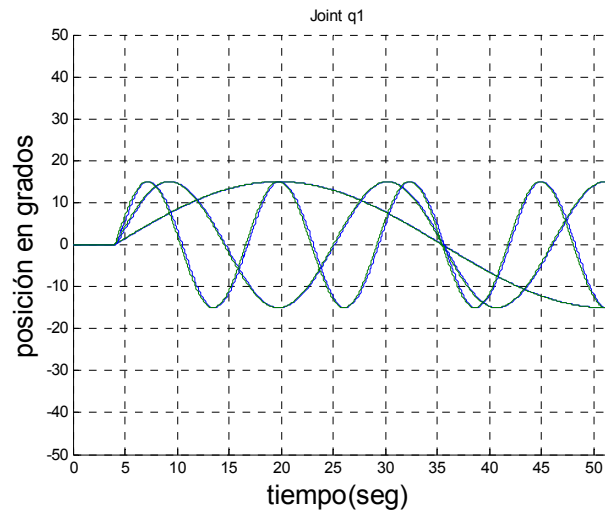


Figura 5.12: Demostraciones en el espacio articulado con distintas velocidades para la articulación q_1

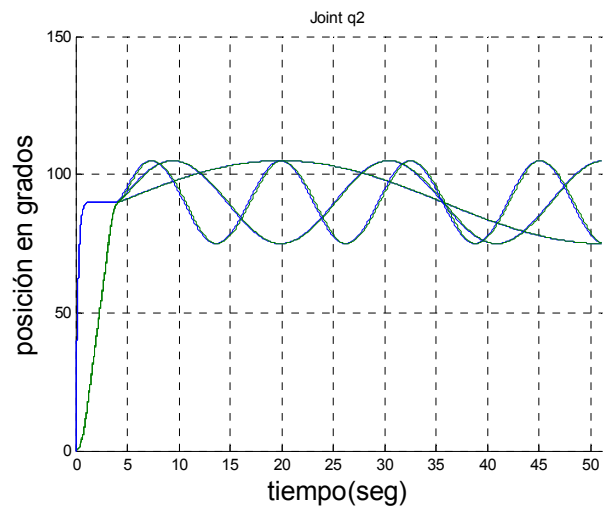


Figura 5.13: Demostraciones en el espacio articulado con distintas velocidades para la articulación q_2

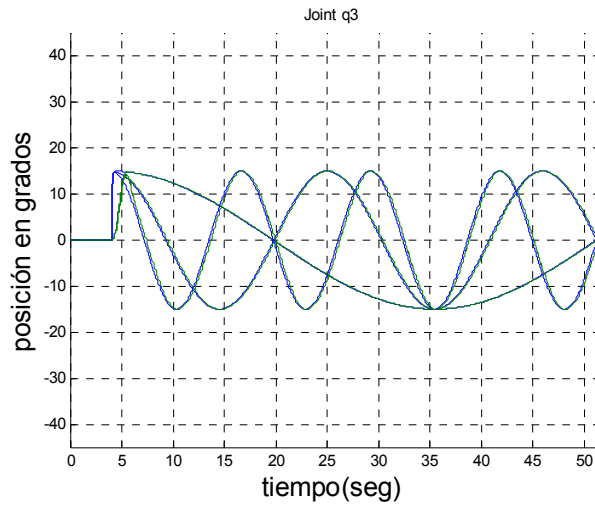


Figura 5.14: Demostraciones en el espacio articulado con distintas velocidades para la articulación q_3

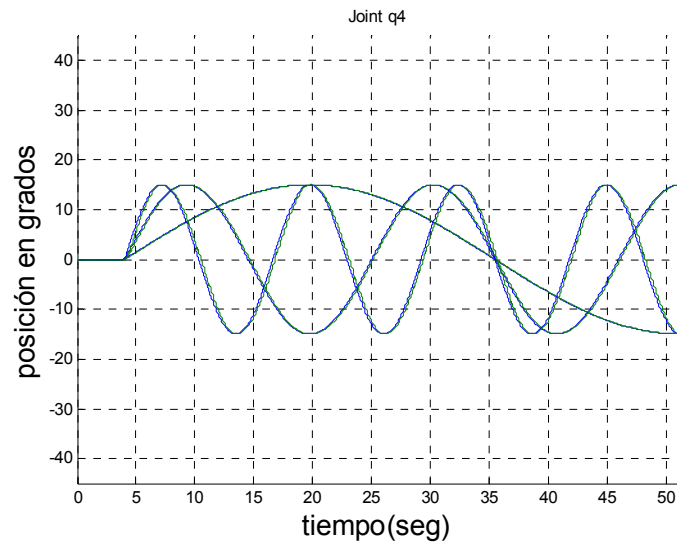


Figura 5.15: Demostraciones en el espacio articulado con distintas velocidades para la articulación q_4

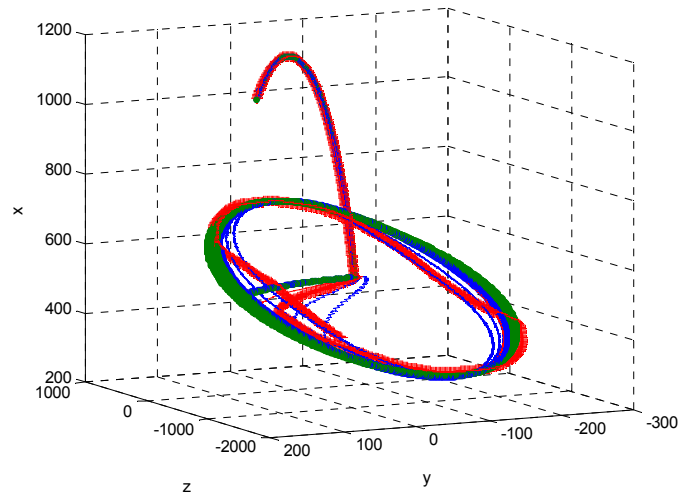


Figura 5.16: Trayectorias en el espacio de trabajo con diferentes velocidades

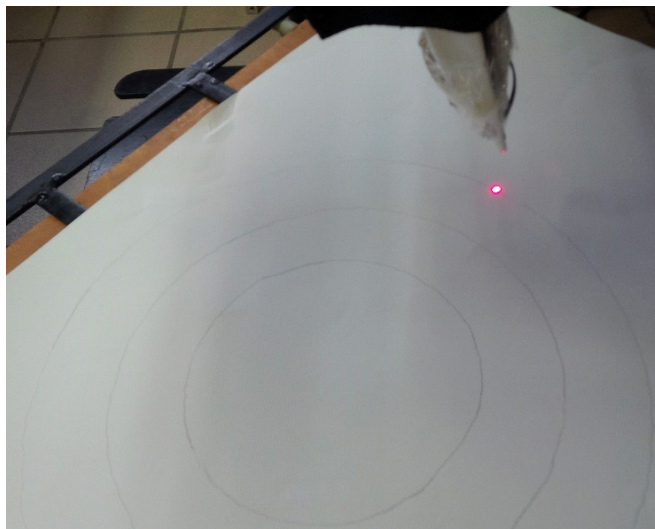


Figura 5.17: Fotografía de las trayectorias en el espacio de trabajo

Capítulo 6

Conclusiones y Trabajo Futuro

En esta tesis se presentó el Aprendizaje por Demostración aplicando los Modelos Ocultos de Markov para seguimiento de trayectorias en un exoesqueleto de 4 gdl para la extremidad superior.

El primer objetivo planteado fue el diseñar y construir un exoesqueleto de 4 gdl, para la articulación del hombro y codo, para lo cual se tomaron en cuenta la cinemática del hombro y codo del ser humano para poder realizar los movimientos sin ningún daño para el usuario, se diseñó y se mandó a construir la parte mecánica como eslabones y soportes, para la comunicación entre el control maestro y los MPC se diseñó una interface para comunicar entre protocolos de comunicación USB y CAN,

La mayoría de las técnicas que se usan para aplicar la metodología de ApD se aplican en el espacio de trabajo, y las que se aplican en el espacio articular hacen uso de la técnica Dynamic time Warping, dentro de los objetivos de esta tesis era demostrar que se puede aprenderá en el espacio articular sin el uso de esta técnica como se demostró.

Se explicó el uso del algoritmo de Lloyd's para el cálculo de puntos claves los cuales se realizaban usando criterios como máximo, mínimo y variaciones en amplitud y velocidad de una trayectoria para calcular lo anterior se proponen ciertos valores de umbral, la desventaja de usar esta técnica es que si la trayectoria se modifica hay que volver a proponer los valores de umbral, entre mas gdl tenga el robot más son valores que se tienen que modificar; el uso

de algoritmo de Lloyd's evita lo anterior ya que solo se propone el tamaño del codebook y con esto se calculan los puntos claves.

El objetivo de generar una trayectoria en base a modelar el movimiento de las trayectorias es que los MOM, ofrece la ventaja que genere una trayectoria que tome en cuenta estadísticamente las características que permanecen en todas las demostraciones y las que solo aparecen en una las minimiza, una vez que genera la trayectoria es necesario interpolar para generar una trayectoria suave.

Se presentaron dos tipos de problemas diferentes:

- Demostraciones con diferencia en amplitud
- Demostraciones con diferencia en velocidad

En cada uno de ellos se comprueba que se puede llevar a cabo el seguimiento de trayectorias.

El problema de las demostraciones con distintas amplitudes, se demostró que los MOM modificados tienen un buen desempeño, la ventaja es que se pueden aprender en el espacio articular o en el espacio de trabajo como se demostró en los distintos ejemplos presentados.

El problema de las demostraciones con distintas velocidades se demostró que el uso de los MOM modificados no requiere el uso del modelo de cinemática inversa ya que directamente se puede generar la trayectoria en el espacio articular.

Trabajo Futuro

Como trabajo futuro queda el aumentar de gdl el exoesqueleto, como se mencionó anteriormente el diseño del hombro está formado por 2 gdl pasivos los cuales no están implementados en este trabajo.

El uso de sensores para medir el torque en el exoesqueleto y poder realizar otro tipo de aplicaciones

Bibliografía

- [1] A. Alissandrakis, C.L. Nehaniv, K. Dautenhahn, J. Saunders, Evaluation of Robot imitation attempts: Comparison of the system's and the human's perspectives, in: Proceedings of the 1st ACM/IEEE International Conference on Human_Robot Interactions, HRI'06, 2006.
- [2] Antsaklis, P. J., Guest Editor: Special Issue on Neural Network in Control Systems. IEEE Control System Magazine 12(2) (1992) 8-57
- [3] T. Asfour, F. Gyarfas, P. Azad, and R. Dillmann. Imitation learning of dual-arm manipulation tasks in humanoid Robots. In Proceedings of the IEEE-RAS International Conference on Humanoid Robots (Humanoids), pages 40–47, December 2006.
- [4] B. D. Argall, S. Chernova, M. Veloso and B. Browning (2009). A Survey of Robot Learning from Demonstration. In Robotics and Autonomous Systems.
- [5] Baum, L. E. y Egon, J. A. An inequality with applications to statistical estimation for probabilistic functions of a markov process and to a model for ecology. Bulletin of the American Meteorology Society, 73:páginas 360–363, 1967
- [6] Baum, L. E. An inequality and associated maximization technique in statistical estimation for probabilistic functions of markov processes. Inequalities, 3:páginas 1–8, 1972
- [7] Antonio Barrientos,Luis Felipe Peñin,Carlos Balaguer,Rafael Aracil Santoja, Fundamentos de Robotica, Mcgraw-Hill, 2007.

- [8] Aude G. Billard , Sylvain Calinon, Florent Guenter, Discriminative and adaptive imitation in uni-manual and bi-manual tasks in: *Robotics and Autonomous Systems* 54, pages 370–384, (2006)
- [9] B. Browning, L. Xu, M. Veloso, Skill acquisition and use for a dynamicallybalancing soccer Robot, in: *Proceeding of the 19th National Conference on Artificial Intelligence, AAAI'04*, 2004.
- [10] J. V. D. Berg , S.Miller , D.Duckworth , H.Hu , A.Wan , X.Fu , K.Goldberg , P.Abbeel. Superhuman Performance of Surgical Tasks by Robots using Iterative Learning from Human-Guided. 2010
- [11] http://www.bosch-semiconductors.de/media/pdf_1/canliteratur/can2spec.pdf
- [12] Calinon, S.; Billard, A., "Stochastic gesture production and recognition model for a humanoid Robot," *Intelligent Robots and Systems, 2004. (IROS 2004)*. *Proceedings. 2004 IEEE/RSJ International Conference on* , vol.3, no., pp.2769,2774 vol.3, 28 Sept.-2 Oct. 2004
- [13] S. Calinon and A. Billard. Learning of gestures by imitation in a humanoid Robot. In K. Dautenhahn and C.L. Nehaniv, editors, *Imitation and Social Learning in Robots, Humans and Animals: Behavioural, Social and Communicative Dimensions*, pages 153–177. Cambridge University Press, 2007.
- [14] A. Ng, A. Coates, M. Diel, V. Ganapathi, J. Schulte, B. Tse, E. Berger, E. Liang, Inverted autonomous helicopter flight via reinforcement learning, in: *International Symposium on Experimental Robotics*, 2004.
- [15] P. Culmer, A. Jackson, M.C. Levesley, J. Savage, R.Richardson, J.A.Cozens, and B.B Bhakta, An admittance control scheme for a Robotic upper-limb stroke rehabilitation system, *27th Annual Conference on Engineering in Medicine and Biology*, 5081-5084, 2005

- [16] Dautenhahn, K. & Nehaniv, C. L. , Imitation in animals and artifacts. Cambridge, MA: MIT Press. (2002)
- [17] Dempster, A.P.; Laird, N.M.; y Rubin., D.B. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, B39(1):páginas 1 – 38, 1977.
- [18] A.M. Dollar and H.Herr, Lower Extremity Exoskeletons and Active Orthoses: Challenges and State-of-the-Art, *IEEE Transactions on Robotics*, Volume 24, No. 1, pp. 1-15, August 2008
- [19] B. Dufay and J.-C. Latombe. An approach to automatic Robot programming based on inductive learning. *The International Journal of Robotics Research*, 3(4):3–20, 1984.
- [20] <http://www.efisioterapia.net/articulos/protocolo-hombro>
- [21] S. Ekvall and D. Kragic. Learning task models from multiple human demonstrations. In *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 358–363, September 2006. Kaufmann.
- [22] C.Ettore, J.Rosen, J.C. Perry, S.Burns, Myoprocessor for Neural Controlled Powered Exoskeleton Arm, *IEEE Transactions on Biomedical Engineering*, pp. 2387-2396, Vol. 53, No. 11, November 2006
- [23] G. D. Forney, Jr. The viterbi algorithm. *Proc. IEEE (Invited Paper)*, 61:páginas 268–278, Marzo 1973
- [24] A. F. Gernot. (2008). *Markov Models for Pattern Recognition: From Theory to Applications*, Springer, Berlin, 2008.
- [25] E.Gribovskaya, S.M.Khansari-Zadeh, A.Billard. Learning Non-linear Multivariate Dynamics of Motion in Robotic Manipulators, *International Journal of Robotics Research*, Vol.30 No.1, 80-117. 2011

- [26] Micha Hersch, Florent Guenter, Sylvain Calinon, and Aude Billard, Dynamical System Modulation for Robot Learning via Kinesthetic Demonstrations in:IEEE Transactions On Robotics, Vol. 24, NO. 6, pages 1463-1467 (2008)
- [27] P. Huang, W. Xu, Y Xu, and B Liang³ Learning Control for Space Robotic Operation Using Support Vector Machines in: Advances in Neural Networks - ISNN 2006 Lecture Notes in Computer Science Volume 3972, 2006, pp 1208-1217
- [28] T. Inamura, M. Inaba, H. Inoue, Acquisition of probabilistic behavior decision model based on the interactive teaching method, in: Proceedings of the Ninth International Conference on Advanced Robotics, ICAR'99, 1999
- [29] K. Ikeuchi and T. Suchiro. Towards an assembly plan from observation, part I: Assembly task recognition using face-contact relations (polyhedral objects). In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), volume 3, pages 2171–2177, May 1992.
- [30] A.J. Ijspeert, J. Nakanishi, and S. Schaal. Learning control policies for movement imitation and movement recognition. In Neural Information Processing System (NIPS), volume 15, pages 1547–1554, 2003.
- [31] S.B. Kang and K. Ikeuchi. A Robot system that observes and replicates grasping tasks. In Proceedings of the International Conference on Computer Vision (ICCV), pages 1093–1099, June 1995.
- [32] H.Kazerooni, R.Steger, The Berkeley lower extremity exoskeleton, *Journal of Dynamic Systems, Measurements, and Control-Transactions of the ASME*, Vol.128, 14-25, 2006
- [33] J.U. Korein, *A Geometric Investigation of Reach*, The MIT Press,1985
- [34] Y. Kuniyoshia, Y.Ohmuraa, K.Teradaa, A.Nagakuboc, S.Eitokua, T.Yamamotob. Embodied basis of invariant features in execution and perception of whole-body dynamic

- actions-knacks and focuses of Roll-and-Rise motion, *Robotics and Autonomous Systems*, Vol.48, No.4, 189-201. 2004
- [35] J.Kwon, F. C. Park, Natural Movement Generation Using Hidden Markov Models and Principal Components, *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, Vol.38, No.5, 184-1194 , 2008
- [36] A. Levas and M. Selfridge. A user-friendly highlevel Robot teaching system. In Proceeding of the IEEE International Conference on Robotics, pages 413–416, March 1984.
- [37] S. P. Lloyd, Least squares quantization in PCM, *IEEE Transactions on Information Theory*, Vol.28, No.2, 129-137. 1982
- [38] F. Martinez, etc., Design of a Five Actuated DoF Upper Limb Exoskeleton Oriented to Workplace Help, *2nd Biennial IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics*, 169-175, Scottsdale, USA, 2008
- [39] B. J. Makinson, Research and development prototype for machine augmentation of human strength and endurance, Hardiman I project, Report S-71-106, GE Company, Schenectady, New York, 1971.
- [40] W.Maurel, *3D Modeling of the Human Upper Limb Including the Biomechanics of Joints, Muscles and Soft Tissues*, PhD thesis, Ecole Polytechnique Federale de Lausanne, 1998.
- [41] Micha Hersch, Florent Guenter, Sylvain Calinon, and Aude Billard, Dynamical System Modulation for Robot Learning via Kinesthetic Demonstrations in *IEEE Transactions on Robotics*, vol. 24, no. 6, pages 1463-1467 2008
- [42] M. Mistry, P. Mohajerian, and S. Schaal, An Exoskeleton Robot for Human Arm Movement Study, *IEEE/RSJ Int. Conf. on Intell. Robots and Syst.*, 4071- 4076, Alberto, Canada, 2005.

- [43] Miyamoto, H. & Kawato, M., A tennis serve and upswing learning Robot based on bi-directional theory. in *Neural Networks* 11, 1331-1344. (1998)
- [44] S. Moubarak, M.T. Pham, T. Pajdla and T. Redarce, Design and Modeling of an Upper Extremity Exoskeleton, *11th International Congress on Medical Physics and Biomedical Engineering*, 476-482, Munich, Germany, 2009
- [45] K. Murphy, *Hidden Markov Model (HMM) Toolbox for MATLAB*, 1998.
- [46] C. Nehaniv and K. Dautenhahn, “Of hummingbirds and helicopters: An algebraic framework for interdisciplinary studies of imitation and its applications,” in *Interdisciplinary Approaches to Robot Learning*, J. Demiris and A. Birk, Eds. World Scientific Press, 2000, vol. 24, pp. 136–161.
- [47] T. Nef, M. Mihelj, G. Colombo, and R. Riener, ARMin-Robot for Rehabilitation of the Upper Extremities, *IEEE Int. Conf. on Robotics and Automat.*, 3152-3157, Orlando, USA, 2006.
- [48] M. Ito, K. Noda, Y. Hoshino, and J. Tani. Dynamic and interactive generation of object handling behaviors by a small humanoid Robot using a dynamic neural network model. *Neural Networks*, 19(3):323–337, 2006.
- [49] Peter Pastor, Heiko Hoffmann, Tamim Asfour, and Stefan Schaal, Learning and Generalization of Motor Skills by Learning from Demonstration in: 2009 IEEE International Conference on Robotics and Automation, pages 763-768, Kobe, may 2009
- [50] J. C. Perry, J. Rosen, and S. Burns “Upper-Limb Powered Exoskeleton Design,” *IEEE/ASME Trans. on Mechatronics*, vol. 12, no. 4, pp. 408-417, 2007.
- [51] C.Perry Joel , J.Rosen, S.Burns, Upper-Limb Powered Exoskeleton Design, *IEEE Transactions on Mechatronics*, Volume 12, No. 4, pp. 408-417, August 2007
- [52] J. Peters, S.Schaal. Learning to Control in Operational Space, *International Journal of Robotics Research*, Vol.27, No.2, 197-212. 2008

- [53] P.K. Pook, D.H. Ballard, Recognizing teleoperated manipulations, in: Proceedings of the IEEE International Conference on Robotics and Automation, ICRA'93, 1993.
- [54] J.L.Pons, *Wearable Robots: biomechatronic exoskeletons*, San Francisco, CA: Wiley, 2007
- [55] Rüdiger Dillmann, Teaching and learning of Robot tasks via observation of human performance in *Robotics and Autonomous Systems* 47, pages 109–116 (2004)
- [56] J. Rosen, N. Brand, M. B. Fuchs, and M. Arcan, “A Myosignal-Based Powered Exoskeleton System,” *IEEE Trans. on Syst. Man Cybern. A*, vol. 31, no. 3, pp. 210-222, 2001.
- [57] Juang, B.-H. y Rabiner, L. R. The segmental k-means algorithm for estimating parameters of hidden markov models. En *IEEE Trans. on Acoustics, Speech, and Signal Processing*, tomo 38, páginas 1639–1641. Septiembre 1990
- [58] A.B. Segre and G. DeJong. Explanation-based manipulator learning: Acquisition of planning ability through observation. In *IEEE Conference on Robotics and Automation (ICRA)*, pages 555–560, March 1985.
- [59] T. G. Sugar, etc., Design and Control of RUPERT: A Device for Robotic Upper Extremity Repetitive Therapy, *IEEE Trans. Neural Syst. Rehabil. Eng.*, Vol. 15, No. 3, 336-346, 2007
- [60] Tuttle, T.D., “Understanding and modeling the behavior of a Harmonic Drive Gear Transmission”, MIT Technical Report 135, MIT Artificial intelligence laboratory (1992)
- [61] C.P. Tung and A.C. Kak. Automatic learning of assembly task using a DataGlove system. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 1, pages 1–8, 1995.

- [62] A. Steinfeld, T. Fong, D. Kaber, M. Lewis, J. Scholtz, A. Schultz, M. Goodrich, Common metrics for human-Robot interaction, in: Proceedings of the 1st ACM/IEEE International Conference on Human_Robot Interactions, HRI'06, 2006.
- [63] Stefan Schaal, Auke Ijspeert, & Aude Billard, Computational Approaches to Motor Learning by Imitation in: Philosophical Transaction of the Royal Society of London: Series B, Biological Sciences 358: 537-547 (2003)
- [64] Sutton, R. S. & Barto, A. G., Reinforcement learning : An introduction. Adaptive computation and machine learning. Cambridge: MIT Press, 1998
- [65] Yang, J., Xu, Y., Chen, C. S.: Hidden Markov Model Approach to Skill Learning and its Application to TeleRobotics. IEEE Trans. Robotics and Automation 10(5) (1994) 612-631
- [66] J. van , S. Miller, D. Duckworth, H. Hu, A. Wan, X. Fu, K. Goldberg, P. Abbeel, Superhuman Performance of Surgical Tasks by Robots using Iterative Learning from Human-Guided Demonstrations in IEEE International Conference on Robotics and Automation Anchorage Convention District May 3-8, 2010, Anchorage, Alaska, USA pages: 2074-2081 (2010)
- [67] Markus Schneider, Wolfgang Ertel, Robot Learning by Demonstration with Local Gaussian Process Regression in: Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference pages 255- 260.
- [68] Z.Wang, H. B.Amor, D.Vogt, J.Peters. Probabilistic movement modeling for intention inference in human-Robot interaction, *International Journal of Robotics Research*, Vol.32, No.7, 841-858. 2013
- [69] G. Schwarz, Estimating the dimension of a model, *The Annals of Statistics* 4, pages 461-464.(1978)

- [70] Sotirios P. Chatzis *, Dimitrios Korkinof, Yiannis Demiris, A nonparametric Bayesian approach toward Robot learning by demonstration in: *Robotics and Autonomous Systems* 60 pages 789–802, (2012)
- [71] Mark W.Spong and M.Vidyasagar, *Robot Dynamics and Control*, John Wiley & Sons Inc., Canada, 1989.
- [72] Li-Li Wei; Jing-Qiang Jiang, .^A hidden Markov model-based K-means time series clustering algorithm, "Intelligent Computing and Intelligent Systems (ICIS), 2010 IEEE International Conference on , vol.3, no., pp.135,138, 29-31 Oct. 2010
- [73] A. Vakanski, I. Mantegh, A. Irish, and F. Janabi-Sharifi. Trajectory Learning for Robot Programming by Demonstration Using Hidden Markov Model and Dynamic Time Warping", *IEEE Transactions on system, man, and cybernetics -part B: Cybernetics*, Vol. 42, No. 4, 1039-1052. 2012
- [74] H.Van der Loos, D.J.Reinkensmeyer, Rehabilitation and health care Robotics, *Handbook of Robotics*. New York, NY: Springer; p. 1223–1251, 2008.
- [75] Vapnik, V. , Principles of riskminimization for learning theory. InMoody, J. E., Hanson, S. J. & Lippmann, R. P. (eds.), *Advances In Neural Information Processing Systems* 4, pp. 831–838. Morgan Kaufman, San Mateo, CA.(1992)
- [76] Vapnik, V. N.: *The Nature of Statistiactal Learning Theory*. Springer-Verlag, New York (1995)
- [77] S. Vijayakumar and S. Schaal. Locally weighted projection regression: An $O(n)$ algorithm for incremental real time learning in high dimensional spaces. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 288–293, 2000.
- [78] A.J.Viterbi, Error bounds for convolucional codes and an asymptotically optimum decoding algorithm, *IEEE Transactions on Information Theory*, Vol.13, No.2, 260-269, 1967.

- [79] J. Yang, Y. Xu, and C.S. Chen. Human action learning via hidden Markov model. *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, Vol.27, No.1, 34–44, 1997
- [80] J. Yang, Y. Xu, and C.S. Chen. Human action learning via hidden Markov model. *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, Vol.27, No.1, 34–44. 1997
- [81] Amelia Zafra, Mykola Pechenizkiy, Sebastián Ventura. HyDR-MI: A hybrid algorithm to reduce dimensionality in multiple instance learning, *Information Sciences*, Volume 222, Pages 282-301. 2013
- [82] Zhang, Y., Wang, J., Xu, Y.: A Dual Neural Network for Bi-criteria Kinematic Control of Redundant Manipulators, *IEEE Trans. Robotics and Automation* 18(6) (2002) 923-931