



Universidad Autónoma del Estado
de Hidalgo

Instituto de Ciencias Básicas
e Ingeniería

Centro de Investigación en
Tecnologías de Información y
Sistemas

**"Seguimiento Adaptativo de Trayectorias con
Convergencia en Tiempo Finito de un
Robot Antropomórfico Virtual de
Tres grados de Libertad"**

Tesis para obtener el título de
Licenciados en Sistemas Computacionales

Presentan:

González Rodríguez Araceli

Pineda Ortega Manuel

Soberanes Leal Dely Madai

Asesores:

Dr. Omar Arturo Domínguez Ramírez

M.C.C. Herbert Lara Ordaz

L.C. Luis Ramón Macías Pulido

Agradecimientos

Dios me ha dado dos grandes regalos, el primero de ellos: mis padres Ma. Luisa y Remedios, ustedes que me han dado la herencia más grande que un padre puede darle a un hijo la educación, me dieron armas para poder desarrollarme profesionalmente no importando el gran esfuerzo que estaban haciendo para dárme las, ahora los comprendo y les agradezco infinitamente por todo su amor y apoyo a lo largo de mi vida y de mi carrera, quizá esto es poco en comparación de todo lo que ustedes me han dado pero se las dedico con mucho amor.

El segundo, mi esposo Agus y mi princesa Valeria quienes me han dado el ultimo empujoncito que necesitaba para cerrar este capitulo de mi vida y poderlo compartir con las personas que más amo.

"Siempre di lo que sientes y haz lo que piensas. Si supiera que hoy fuera la última vez que te voy a ver dormir, te abrazaría fuertemente y rezaría al Señor para poder ser el guardián de tu alma. Si supiera que esta fuera la última vez que te vea salir por la puerta, te daría un abrazo, un beso y te llamaría de nuevo para darte más. Si supiera que esta fuera la última vez que voy a oír tu voz, grabaría cada una de tus palabras para poder oír las una y otra vez indebidamente. Si supiera que estos son los últimos minutos que te veo diría "te quiero" y no asumiría, tontamente, que ya lo sabes".

Gabriel García Márquez

Araceli

Gracias a Dios por permitir cerrar un ciclo en mi vida, gracias a dios por dejarme estar en este mundo, por darme unos padres maravillosos que se han esforzado por darme esta herencia pero sobre todo por estar siempre conmigo en las buenas y en las malas.

A mis padres por el esfuerzo que realizaron por mi; A ti Mamá por demostrarnos que no importa que tan difícil es la prueba que nos pone dios sino el coraje y la fe para seguir adelante. A ti Papá por enseñarnos a luchar siempre por lo que se quiere y a ser hombres de bien. Este trabajo se los dedico a ustedes.

A mis hermanos les agradezco por su apoyo incondicional y por recordarme que siempre puedo contar con ellos.

Gracias abuelo por confiar siempre y por la fortaleza que demostraste toda la vida, siempre fuiste importante en nuestras vidas.

Al Dr. Ornar por esperar tanto tiempo y por estar siempre preocupado por que cerráramos esta etapa de nuestra vida, sin tu apoyo hubiese sido más pesado el camino.

A mis amigos por estar ahí en las buenas y en las malas, agradezco la paciencia y apoyo pero sobre todo su amistad a Dely y Araceli que son parte de este logro.

Manuel

Agradezco primeramente a Dios por ser mi mejor amigo, mi fortaleza, darme todo lo que tengo y levantarme tras cada caída.

A mi madre por ser la mejor, estar siempre conmigo y apoyarme, porque sin ella y sus enseñanzas no estaría aquí ni sería quien soy ahora; a mi mami le dedico esta tesis.

A mis amigos Elizabeth, Jorge, Sergio y Yanet porque son como mis hermanos y juntos hemos conocido la verdadera amistad, valor importante para mi vida, gracias por estar conmigo, por aconsejarme, regañarme, compartir risas y llanto todo este tiempo. A mi amiga Jaktziry por haber cambiado mi vida...

A mi amigo eterno de la universidad: Manuel, por apoyarme y estar conmigo a lo largo de la carrera y aún después ... A mis tíos por ser como mis padres y aunque algunos ya no están conmigo han contribuido para que yo haya llegado hasta aquí.
¡Gracias a todos! Los quiero mucho.

Dely

Resumen

Este trabajo de investigación se presenta como tesis para la obtención del título de Licenciado en Sistemas Computacionales; debido a la naturaleza del trabajo se realizó una revisión de la literatura sobre los robots manipuladores presentando así aspectos como antecedentes, características, clasificaciones y puntos generales sobre éstos, así como algunos de los proyectos existentes que abordan temas similares al que se expone.

Dado el propósito del integrar una plataforma virtual dinámica para la evaluación de estrategias de control, se presenta el desarrollo matemático de los modelos cinemático y dinámico, que nos ayudan a representar a los robots para la realización de simulaciones matemáticas, de igual forma se presenta estrategias de control clásicas y modernas, lo cual se aplica para el seguimiento de trayectorias.

El software propuesto por este trabajo permite la visualización en línea del robot, llevando acabo la solución en lazo cerrado de un robot de tres grados de libertad con un control no lineal por modos deslizantes de segundo grado, para el seguimiento de una trayectoria (circunferencia). Para esta solución se implementó el método de Runge - Kutta de orden superior para un sistema de ecuaciones diferenciales. Los modelos matemáticos mencionados se basan en la formulación de Euler - Lagrange para el modelo dinámico y la formulación de Denavit - Hartenberg para el modelo cinemático.

Se realiza un estudio comparativo entre dos controles clásicos: proporcional derivativo y proporcional integral derivativo; otro entre dos controles modernos: no lineal por modos deslizantes de segundo orden y adaptable pasivo robusto. Otra aportación de éste trabajo es la implementación de un generador de tiempo base que ayuda a la convergencia de robots manipuladores en aplicaciones que requieren de precisión en tiempo y espacio, el generador de tiempo base se aplica a los dos controles modernos antes mencionados para el seguimiento de trayectoria.

El software antes mencionado está diseñado para que el usuario experimente con el robot virtual y un control no lineal por modos deslizantes de segundo grado, al permitir la manipulación de las ganancias del controlador y así mejorar la sintonización del robot, el tiempo de simulación y en su caso el tiempo de convergencia, las herramientas utilizadas para el desarrollo del software son Visual C++, OpenGL para el robot virtual y para la interfaz de usuario Delphi 7.0, de igual forma se apoyó en Matlab para la realización de simulación fuera de línea y comprobación de resultados.

Índice general

1. Introducción	9
1.1. Objetivo General	9
1.2. Antecedentes	9
1.3. Justificación	10
1.4. Contribuciones	10
1.5. Estudio del estado del arte	10
1.5.1. Laboratorios virtuales de robótica	11
1.6. Organización de la tesis	16
2. Generalidades de modelado matemático de robots manipuladores	18
2.1. Introducción	18
2.2. Conceptos preliminares de robótica	18
2.2.1. Antecedentes	23
2.2.2. Clasificación de los robots	27
2.2.3. Morfología de los robots manipuladores	31
2.2.4. Subsistemas de un robot manipulador.	40
2.2.5. Aplicaciones de los robots	42
2.3. Modelo cinemático de robots manipuladores	44
2.3.1. Introducción	44
2.3.2. Descripciones: posición, orientación y marcos	46
2.3.3. Cadena cinemática	48
2.3.4. Marcos ortonormales	49
2.3.5. Parámetros Denavit-Hartenberg.	49
2.3.6. Matriz de transformación homogénea	50
2.3.7. Matrices elementales de un robot	50
2.3.8. Modelo cinemático directo de posición (MCDP)	51
2.3.9. Modelo cinemático inverso de posición (MCIP)	51
2.3.10. Modelo cinemático directo de velocidad (MCDV)	52
2.3.11. Modelo cinemático inverso de velocidad (MCIV)	53
2.3.12. Modelo cinemático directo de aceleración (MCDA)	53
2.3.13. Modelo cinemático inverso de aceleración (MCIA)	54
2.4. Modelo dinámico de robots manipuladores	54
2.4.1. Introducción	54
2.4.2. Conceptos preliminares de la dinámica de cuerpos rígidos	54
2.4.3. Formulación Euler-Lagrange	55
2.4.4. Propiedades dinámicas de un robot manipulador.	57
2.5. Conclusiones	58

3. Generalidades de control de robots manipuladores	59
3.1. Introducción	59
3.2. Conceptos preliminares de control	59
3.3. Tipos de control	61
3.4. Control Proporcional Derivativo (PD)	62
3.5. Control Proporcional Integral Derivativo (PID)	63
3.6. Control por modos deslizantes de segundo orden o PID no lineal.	64
3.7. Control adaptable P-R	66
3.7.1. Historia del control adaptable	66
3.7.2. Para qué sirve el control adaptable	67
3.7.3. Control adaptable.	67
3.8. Conclusiones.	67
4. Visualización virtual, análisis y diseño del sistema de “Seguimiento de trayectorias.”	68
4.1. Introducción	68
4.2. Conceptos preliminares de realidad virtual	68
4.2.1. Clasificación de la realidad virtual.	71
4.2.2. Características generales del ambiente virtual.	71
4.2.3. Herramientas de edición	72
4.2.4. Lenguajes	75
4.2.5. Visual C++	76
4.2.6. Delphi 7.0	77
4.3. Construcción del Ambiente virtual	80
4.3.1. Animación al ambiente virtual	80
4.3.2. Construcción del robot virtual	80
4.4. Comportamiento complejo	82
4.5. Análisis y diseño de la interfaz de “Seguimiento trayectorias ”.	82
4.5.1. Análisis.	82
4.5.2. Diseño.	95
4.6. Conclusiones	103
5. Modelado y control del robot antropomórfico virtual	104
5.1. Introducción	104
5.2. Modelo cinemático	104
5.2.1. Parámetros de Denavit-Hartenberg (DH).	104
5.2.2. Matrices elementales y de transformación homogénea.	104
5.2.3. Modelo cinemático directo de posición (MCDP)	108
5.2.4. Modelo cinemático inverso de posición (MCIP)	109
5.2.5. Modelo cinemático directo de velocidad (MCDV)	115
5.2.6. Modelo cinemático inverso de velocidad (MCIV)	117
5.2.7. Modelo cinemático directo de aceleración (MCDA)	118
5.2.8. Modelo cinemático inverso de aceleración (MCIA)	121
5.3. Modelo dinámico	123
5.3.1. Obtención de los módulos de velocidad y alturas de centros de masas	123
5.3.2. Determinación de energías cinéticas del robot	126
5.3.3. Determinación de energías potenciales del robot	127
5.3.4. Ecuaciones de movimiento (Lagrangiano)	128
5.3.5. Formulación Euler-Lagrange.	128
5.3.6. Representación matricial del MD.	131
5.3.7. Propiedades dinámicas	134
5.4. Control	135
5.4.1. Control PD	135
5.4.2. Control PID	139

5.4.3. PIDNL	144
5.4.4. Control adaptable pasivo robusto	146
5.5. Seguimiento exacto: Convergencia en Tiempo Finito.	156
5.5.1. PIDNL con TBG	157
5.5.2. Control adaptable pasivo robusto con TBG	160
5.6. Conclusiones	163
6. Conclusiones y perspectivas.	165
6.1. Conclusiones generales	165
6.2. Perspectivas y trabajo futuro	166
A. Abreviaturas	167
B. Glosario	170
C. Apéndice: Propiedades dinámicas	180
D. Apéndice: Manual de usuario del sistema.	184
D.1. Robot Virtual	186
D.2. Control PIDNL	187
D.3. Control Adaptable	191
E. Artículo: Visualización virtual del Sistema de Seguimiento de trayectorias (Realidad virtual).	197
F. Artículo: Control PID no Lineal con Convergencia en Tiempo Finito.	216
G. Artículo: Seguimiento Adaptativo de Trayectorias con Convergencia en Tiempo Finito de un Robot Antropomórfico Virtual.	232
H. Apéndice: Contenido del CD.	252
Bibliografía	253

Índice de figuras

1.1. Interfaz de COSIMIR[49].	11
1.2. Interfaz de Robolab[43].	12
1.3. Interfaz Robolab II [43].	12
1.4. Interfaz de Robolab con un robot PA-10 de Mitshubitshi[43].	13
1.5. Interfaz de LVRM [50].	14
1.6. Interfaz de “Simulación de robots articulados” [51].	15
2.1. Similitud con el brazo humano[31].	20
2.2. Robot montado sobre una base[30].	21
2.3. Vista de los componentes de un robot manipulador[31].	22
2.4. El brazo de Stanford 1969[37].	24
2.5. Shakey, el primer vehículo autoguiado controlado por inteligencia artificial (1970)[16].	25
2.6. Robot Qrio de Sony[39].	26
2.7. Componentes de un robot.	27
2.8. Robot que su forma de desplazamiento es con ruedas[35].	29
2.9. Robot Androide [36].	30
2.10. Tipos de articulaciones : las articulaciones β y θ son de rotación (R), mientras que Υ pertenece a una articulación prismática (P)[30].	31
2.11. Esquema de un robot cartesiano [30].	32
2.12. Espacio de un robot cartesiano [17].	32
2.13. Un robot cartesiano[57].	33
2.14. Esquema de un robot cilíndrico [30].	33
2.15. Espacio de un robot cilíndrico [17].	33
2.16. Ejemplo de un robot cilíndrico[57].	34
2.17. Esquema de un robot polar [30].	34
2.18. Espacio de un robot polar [17].	34
2.19. Ejemplo del robot polar[57].	35
2.20. Espacio de trabajo del robot tipo SCARA [17].	35
2.21. Ejemplo de robot SCARA[17].	35
2.22. Esquema de un robot de brazo articulado [30].	36
2.23. Espacio de trabajo de un robot articulado [17].	36
2.24. Ejemplo de robot de brazo articulado[57].	36
2.25. Articulaciones y elementos de un robot manipulador PUMA Unimation.	37
2.26. Estructura y cadena cinemática del manipulador Stanford/JPL [15].	38
2.27. Yasukawa Motoman L-3 [15].	39
2.28. Robot Da Vinci que puede operar pacientes [41].	43
2.29. Robot en la industria [60].	43
2.30. Relación entre cinemática directa e inversa	45
2.31. Sistema de coordenadas.	46
2.32. Localizando un objeto en posición y orientación.	47
2.33. Representación genérica de la cadena cinemática de un mecanismo de eslabones articulados [10][20].	49
2.34. Forma explícita de definir la cadena cinemática de un robot manipulador [10][20].	49

3.1. Un controlador de tipo proporcional, en el que la señal aplicada al motor es proporcional a la señal de error[53].	61
3.2. Esquema de un controlador PID[53].	62
4.1. Portada de AutoCAD 2006[44].	72
4.2. Portada de 3D Studio Max 7.0[45].	73
4.3. Portada de Maya [46].	74
4.4. Portada de VRML[47].	75
4.5. Portada de Visual C++.	76
4.6. Portada de Delphi 7.0.	78
4.7. Aproximación con rectángulos [25].	80
4.8. Paralelepípedo a partir de 6 rectángulos [25].	81
4.9. Configuraciones de eslabones con paralelepípedos [25].	81
4.10. Diagrama de flujo información del sistema “Seguimiento de trayectorias”.	83
4.11. Diagrama de flujo del procedimiento de solución.	84
4.12. Diagrama de flujo del procedimiento referencia.	85
4.13. Diagrama de flujo del procedimiento de control.	86
4.14. Diagrama de flujo del procedimiento rungetpp’.	87
4.15. Diagrama de flujo del procedimiento rungetpp’.	88
4.16. Diagrama de flujo del procedimiento rungetpp’.	89
4.17. Diagrama de flujo del procedimiento rungetpp’.	90
4.18. Diagrama de flujo de la función f.	91
4.19. Diagrama de flujo procedimiento rungeerror.	92
4.20. Diagrama de flujo procedimiento rungeerror.	93
4.21. Diagrama de flujo de la función ei.	94
4.22. Diagrama de flujo del procedimiento modelosrobot.	95
4.23. Diagrama de flujo del funcionamiento del sistema.	96
4.24. Diagrama de flujo del funcionamiento del sistema.	97
4.25. Diagrama de flujo del funcionamiento del sistema.	98
4.26. Menú principal de la herramienta propuesta.	99
4.27. Interfaz PHANToM Virtual TRAZO.	99
4.28. Control PIDNL.	100
4.29. Resultados en forma gráfica del control PIDNL.	100
4.30. Trayectoria realizada por PHANToM Virtual TRAZO implementando el control PIDNL.	100
4.31. Control PIDNL con TBG.	100
4.32. Resultados en forma gráfica del control PIDNL con TBG.	101
4.33. Trayectoria realizada por PHANToM Virtual TRAZO implementando el control PIDNL con TBG.	101
4.34. Demo control PIDNL con TBG realizado en MatLab.	101
4.35. Código de MatLab para el control PIDNL con TBG.	101
4.36. Resultados en forma gráfica del control PIDNL con TBG (Demo MatLab).	102
4.37. Trayectoria realizada por PHANToM Virtual TRAZO implementando el control PIDNL con TBG.	102
4.38. Demo control adaptable con TBG realizado en MatLab.	102
4.39. Código de MatLab para el control adaptable con TBG.	102
4.40. Resultados en forma gráfica del control adaptable con TBG (Demo MatLab).	103
4.41. Trayectoria realizada por PHANToM Virtual TRAZO implementando el control adaptable con TBG.	103
5.1. Cadena cinemática sintética	112
5.2. Plano z-x/y	113
5.3. Cadena cinemática con parámetros dinámicos	123
5.4. Descripción de comportamiento de variables articulares	124
5.5. Control PD. Coordenada Generalizada q1 y qd1 (Real y Deseada).	136
5.6. Control PD. Coordenada Generalizada q2 y qd2 (Real y Deseada).	136
5.7. Control PD. Coordenada Generalizada q3 y qd3 (Real y Deseada).	136

5.8. Control PD. Coordenada Operacional X y px (Real y Deseada).	136
5.9. Control PD. Coordenada Operacional Y y py (Real y Deseada).	137
5.10. Control PD. Coordenada Operacional Z y pz (Real y Deseada).	137
5.11. Control PD. Espacio operacional: (X vs Y) y (px vs py) (Real y Deseado).	137
5.12. Control PD. Error Articular 1 (eq1).	137
5.13. Control PD. Error Articular 2 (eq2).	138
5.14. Control PD. Error Articular 3 (eq3).	138
5.15. Control PD. Error Operacional $E_x=px-X$.	138
5.16. Control PD. Error Operacional $E_y=py-Y$.	138
5.17. Control PD. Error Operacional $E_z=pz-Z$.	139
5.18. Control PID, Coordenada Generalizada q1 y qd1 (Real y Deseada).	140
5.19. Control PID, Coordenada Generalizada q2 y qd2 (Real y Deseada).	140
5.20. Control PID, Coordenada Generalizada q3 y qd3 (Real y Deseada).	141
5.21. Control PID, Coordenada Operacional X y px (Real y Deseada).	141
5.22. Control PID, Coordenada Operacional Y y py (Real y Deseada).	141
5.23. Control PID, Coordenada Operacional Z y pz (Real y Deseada).	141
5.24. Control PID, Espacio operacional (X vs Y) y (px vs py)(Real y Deseado).	142
5.25. Control PID, Error Articular 1 (eq1)	142
5.26. Control PID, Error Articular 2 (eq2)	142
5.27. Control PID, Error Articular 3 (eq3)	142
5.28. Control PID, Error Operacional ($E_x= px - X$)	143
5.29. Control PID, Error Operacional ($E_y= py - Y$)	143
5.30. Control PID, Error Operacional ($E_z= pz - Z$)	143
5.31. Control PIDNL,Coordenada Generealizada q1 y qd1 (Real y Deseada).	145
5.32. Control PIDNL, Coordenada Generalizada q2 y qd3 (Real y Deseada).	145
5.33. Control PIDNL, Coordenada Generalizada q3 y qd3 (Real y Deseada).	145
5.34. Control PIDNL, Trayectorias en el espacio oparacional: (x,y) y (px,py), (Real y Deseada).	145
5.35. Control PIDNL, Error extendido S.	146
5.36. Control Adaptable, Coordenada Generalizada q1 y qd1 (Real y Deseada).	155
5.37. Control Adaptable, Coordenada Generalizada q2 y qd2 (Real y Deseada).	155
5.38. Control Adaptable, Coordenada Generalizada q3 y qd3 (Real y Deseada).	156
5.39. Control Adaptable, Trayectoria en el espacio operacional: (x,y) y (px,py), (Real y Deseada).	156
5.40. Control Adaptable, Error extendido S.	156
5.41. Control PIDNL con TBG, Coordenada Generalizada q1 y qd1 (Real y Deseada).	159
5.42. Control PIDNL con TBG, Coordenada Generalizada q2 y qd2 (Real y Deseada).	159
5.43. Control PIDNL con TBG, Coordenada Generalizada q3 y qd3 (Real y Deseada).	160
5.44. Control PIDNL con TBG, Trayectorias en el espacio operacional: (x,y) y (px,py), (Real y Deseada).	160
5.45. Control PIDNL con TBG, Error extendido S.	160
5.46. Control adaptable con TBG, Coordenada Generalizada q1 y qd1 (Real y Deseada).	162
5.47. Control adaptable con TBG, Coordenada Generalizada q2 y qd2 (Real y Deseada).	162
5.48. Control adaptable con TBG, Coordenada Generalizada q3 y qd3 (Real y Deseada).	163
5.49. Control adaptable con TBG, Trayectorias en el espacio operacional: (x,y) y (px,py), (Real y Deseada).	163
5.50. Control adaptable con TBG, Error extendido S.	163
C.1. Representación de las coordenadas generalizadas y las velocidades angulares	181
C.2. Representación de la posición del órgano terminal.	182
C.3. Representación de las propiedades del modelo dinámico del robot antropomórfico	183
D.1. Proceso de instalación Paso 1.	184
D.2. Proceso de instalación Paso 2.	184
D.3. Proceso de instalación Paso 3.	185
D.4. Proceso de instalación Paso 4.	185
D.5. Proceso de instalación Paso 5.	185

D.6. Proceso de instalación Paso 6.	185
D.7. Ubicación de la aplicación en el menú de inicio.	186
D.8. Menú principal.	186
D.9. Interfaz “PHANToM Virtual TRAZO” despliegue de menú de opciones.	187
D.10.Coordenadas generalizadas y operacionales del robot virtua.	187
D.11.Control PIDNL.	187
D.12.Simulación Terminada Control PIDNL.	187
D.13.Trayectoria realizada por el Robot virtual Control PIDNL.	188
D.14.Resultados de la simulación Control PIDNL.	188
D.15.Gráfica coordenada generalizada q1 y qd1 Control PIDNL.	189
D.16.Gráfica coordenada generalizada q2 y qd2 Control PIDNL.	189
D.17.Gráfica coordenada generalizada q3 y qd3 Control PIDNL.	189
D.18.Gráfica de errores Control PIDNL.	189
D.19.Simulación Terminada Control PIDNL con TBG.	190
D.20.Trayectoria realizada por el Robot virtual Control PIDNL con TBG.	190
D.21.Resultados de la simulación Control PIDNL con TBG.	190
D.22.Gráfica coordenada generalizada q1 y qd1 Control PIDNL con TBG.	191
D.23.Gráfica coordenada generalizada q2 y qd2 Control PIDNL con TBG.	191
D.24.Gráfica coordenada generalizada q3 y qd3 Control PIDNL con TBG.	191
D.25.Gráfica de errores Control PIDNL con TBG.	191
D.26.“Demo MatLab Control PIDNL con TBG”.	192
D.27.Trayectoria realizada por el robot virtual Control PIDNL con TBG.	192
D.28.Código MatLab Control PIDNL con TBG.	193
D.29.Gráficas MatLab Control PIDNL con TBG.	193
D.30.“Demo MatLab Control Adaptable con TBG”.	194
D.31.Trayectoria realizada por el robot virtual Control Adaptable con TBG.	194
D.32.Código MatLab Control Adaptable con TBG.	195
D.33.Gráficas MatLab Control Adaptable con TBG.	196

Índice de cuadros

2.1. Parámetros de Denavit-Hartenberg del PUMA 560 [14].	37
2.2. Parámetros de Denavit-Hartenberg del Brazo de Stanford/JPL [15].	38
2.3. Parámetros de Denavit-Hartenberg del Yasukawa Motoman L-3 [15]	39
3.1. Ajustes adecuados de un integrador Proporcional Integral para diferentes condiciones[36].	64
3.2. Tipos de procesos que pueden ser controlados satisfactoriamente por cada uno de los modos básicos de control[36].	65
5.1. Parámetros Denavit-Hartemberg [51]	104
5.2. Descripción de las funciones de la representación matricial del modelo dinámico	132
5.3. Descripción de los vectores de fricción.	133
5.4. Descripción de los elementos de la matriz asimétrica.	135

Capítulo 1

Introducción

1.1. Objetivo General.

Integración de una plataforma virtual dinámica para evaluación de estrategias de control avanzado no lineal con convergencia en tiempo finito en un robot antropomórfico, con propósitos de planificación de movimientos y trayectorias.

1.2. Antecedentes

La realidad virtual, en la última década ha contribuido de manera importante en diversas disciplinas de la ciencia y la tecnología. Son relevantes los logros que ha permitido como una herramienta auxiliar para la generación y el entendimiento del conocimiento puro y aplicado, sin embargo, son relativamente pocas las aplicaciones que permiten involucrar una inmersión virtual activa en la que un operador humano tenga percepción visual, auditiva y táctil de los fenómenos que ocurren en un ambiente sintetizado a partir de un mundo real. Es conocido que en la naturaleza como en los desarrollos tecnológicos diseñados y construidos por el mismo hombre, los comportamientos obedecen a funciones altamente no lineales, cuyas representaciones formales no son tan sencillas ni tan evidentes. Esto implica que la correspondencia entre un mundo virtual, sintetizado a partir de un acontecimiento real, empleando modelos lineales y representaciones puramente cinemáticas que intentan describir las evoluciones espaciales, desprecian de manera significativa los fenómenos inherentes a un sistema real. Por lo que el estímulo visual no corresponde al acontecimiento real que se desea representar.

De acuerdo a la revisión del estado del arte, relativa a simuladores y visualizadores virtuales, no consideran las propiedades dinámicas inherentes al sistema físico que se desea representar virtualmente, existen trabajos como el proyecto alemán COSIMIR que permite la interacción de robots virtuales cooperativos con propósitos educativos e industriales [49], sin embargo a pesar de que permite la programación real de trayectorias en robots industriales no emplea la dinámica del robot real, los movimientos virtuales están basados en las propiedades cinemáticas exclusivamente y los modelos son desarrollados mediante software para diseño CAD. Existe un laboratorio de robótica español denominado ROBOLAB [43], utilizado para la planificación de trayectorias para robots manipuladores, sin embargo la estrategia de planificación y ejecución virtual está basada en solo la cinemática del manipulador, teniendo como deficiencia la misma que el proyecto alemán COSIMIR. En México, los trabajos desarrollados en este ámbito son en el Tecnológico de Monterrey, ITESM, corresponde a un laboratorio virtual de robótica LVRM [50], cuyos propósitos son únicamente didácticos y permiten mediante una interfaz hombre-máquina virtual planificar cinemáticamente los movimientos de robots manipuladores como un robot SEIKO y un PUMA Unimation. En todos estos casos de actualidad, la limitación en el estímulo visual está basada exclusivamente a las propiedades cinemáticas, es decir, se desprecian características dinámicas, como las fuerzas inerciales, gravitatorias, tribológicas, centrípetas y de Coriolis. En el Centro de Investigación en Tecnologías de Información y Sistemas de la Universidad Autónoma del Estado de Hidalgo, se desarrolló un proyecto de investigación para visualizar los efectos de un controlador sobre la dinámica no lineal de un robot manipulador fuera de línea

[24,25]. En este trabajo se ejecutan fuera de línea los resultados del control de robots efectuados previamente en MATLAB, se permite apreciar de manera explícita las evoluciones dinámicas del robot y la trayectoria efectuada en línea. En este trabajo se evalúa un control PID NO LINEAL y un control adaptable utilizando un integrador numérico RUNGE-KUTTA de orden superior, y una contribución adicional que es la convergencia en tiempo finito, con el afán de verificar los efectos dinámicos en el ambiente virtual cuando se tiene como dato de entrada el tiempo de convergencia entre la condición inicial del robot y la referencia (regulación o seguimiento).

1.3. Justificación

Actualmente los sistemas de visualización virtual [43, 49, 50] tienen las siguientes limitaciones:

1. Los modelos matemáticos están basados en aspectos cinemáticos.
2. La cinemática es de posición, no consideran la cinemática diferencial (velocidad y aceleración).
3. Sus aplicaciones son exclusivamente didácticas o de planeación de movimientos de robots experimentales.
4. No consideran las fuerzas inerciales, gravitatorias, tribológicas, centrípetas y de Coriolis, como propiedades dinámicas inherentes a todos los sistemas robóticos.
5. Evalúan acciones de control lineal, cuando los robots son sistemas altamente no lineales.
6. La planificación de trayectorias no considera el tiempo de convergencia, situación que no permite estimar el tiempo de ejecución de la tarea industrial.

1.4. Contribuciones

Las contribuciones de este trabajo son:

1. Mundo virtual dinámico de un robot antropomórfico en línea con un control no lineal.
2. Estudio comparativo de un control adaptable, un control por modos deslizantes de segundo orden y dos controladores clásicos.
3. Solución al problema de convergencia de robots manipuladores con aplicaciones de precisión en tiempo y espacio.
4. Evaluación de un control adaptable para calculado pasivo robusto para seguimiento exacto de trayectorias con convergencia en tiempo finito en el robot virtual.
5. Evaluación de un control por modos deslizantes de segundo orden para seguimiento exacto de trayectorias con convergencia en tiempo finito en el robot virtual.

1.5. Estudio del estado del arte

En la investigación que se ha llevado a cabo para el desarrollo de este trabajo de tesis, se ha detectado que en algunos laboratorios de robótica y/o centros de investigación del extranjero así como en México existen trabajos similares como el que desarrollamos.

1.5.1. Laboratorios virtuales de robótica

- COSIMIR:

Es un antecedente más del desarrollo de laboratorios de robótica virtuales, este software fue diseñado por la compañía 3Festo, basándose en 3D[49].

Es un laboratorio virtual de robótica construido en Alemania para fines educativos e industriales dentro de cuyo ambiente se manipulan diferentes tipos de robots, así como las tareas que desempeñan.

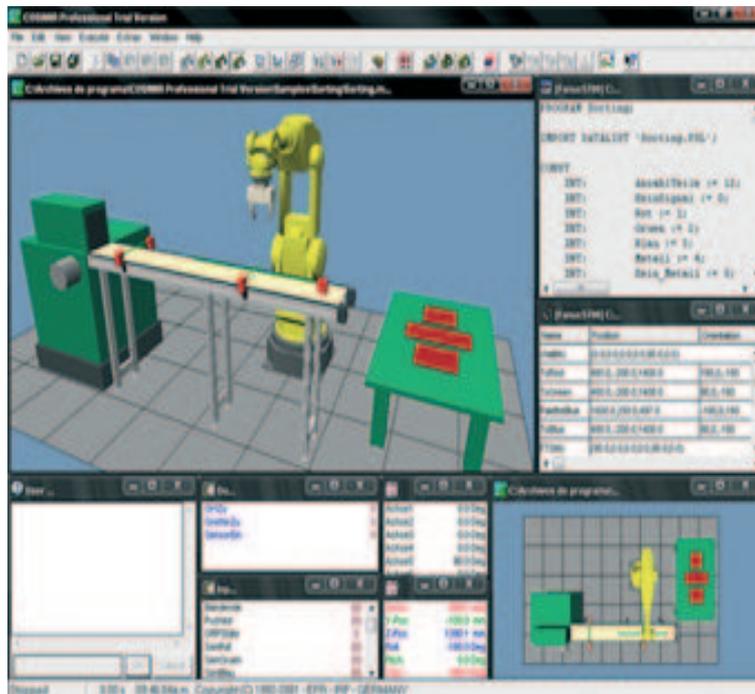


Figura 1.1: Interfaz de COSIMIR[49].

COSIMIR utiliza los idiomas nativos de la robustez (MELFABASIC o comando de Movemaster) para programar dentro del ambiente de la simulación. Los paquetes de COSIMIR y de COSIROP son herramientas de gran alcance para alcanzar eficacia y costo eficiencia máximos en la configuración y la operación de las soluciones robustas apoyadas de la automatización, y permiten planear y que funcionen los sistemas con un grado muy alto de confianza. El software hace distinción entre las necesidades de los usuarios, para lo cual hace la siguiente clasificación:

Usuarios educativos: es altamente educacional de los conocimientos técnicos básicos y una funcionalidad extensa, culminando en el diseño, programando y operación de un sistema virtual de la robustez.

Los programadores: benefician la simulación esencial de funcionalidad de 3D, y modelar la capacidad del paquete industrial de COSIMIR. Los usuarios pueden realizar fácilmente la programación on-line de los sistemas de la robustez de Mitsubishi, y hacen que “muestre una representación gráfica del funcionamiento” de su trabajo.

El profesional de COSIMIR: con el software profesional 3D para modelar, la simulación y programar de los varios sistemas de la robusteza industrial, y un nuevo “ambiente de aprendizaje” desarrollado por COSIMIR educativo.

El modelar con el profesional de COSIMIR se hace simple en virtud de las bibliotecas de robustezas, efectores terminales y componentes de la automatización.

El programa también importa varios formatos del CAD. Sin embargo dicho software es de tipo comercial ya que tiene un costo de adquisición ya sea para fines educativos o comerciales[49].

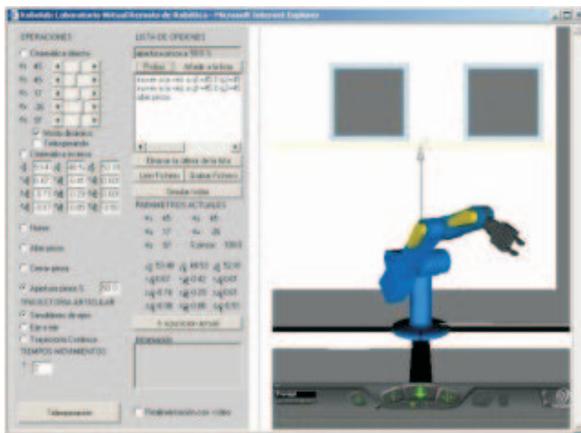


Figura 1.2: Interfaz de Robolab[43].

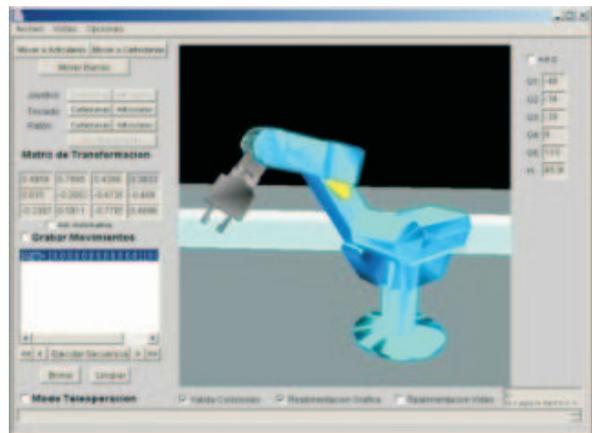


Figura 1.3: Interfaz Robolab II [43].

- Robolab:

En la Universidad de Alicante (Valencia, España) se diseñó un sistema llamado Robolab, el cual permite a los estudiantes de Robótica practicar comandos de posicionamiento en un robot industrial simulado desde cualquier parte del mundo, con la ayuda de este sistema aprenden aspectos básicos de robótica, cinemática y diseño de trayectorias. Lo mejor es que los comandos se pueden enviar a un robot real, situado en el laboratorio del grupo AUROVA en la Universidad de Alicante, para ver los resultados de forma on-line, a través de Internet[43].

Existen dos versiones de Robolab, en Robolab I ofrece una interfaz creada con Java y VRML, un software estándar de Internet para representar escenarios de realidad virtual. Se ofrece también una simulación de realidad virtual basado en VRML, que representa el entorno tridimensional con el robot (un Scorbot ER-IX de Intelitek) y su espacio de trabajo, como se puede ver en la Figura 1.2.

Posteriormente, se ha desarrollado una nueva versión (Robolab II) basada en Java y Java-3D. Los únicos medios requeridos por el estudiante son una computadora personal conectado a Internet, un navegador web y los componentes de software de la máquina virtual de Java y de VRML en su caso.

La principal diferencia de la nueva versión del sistema con respecto a la anterior, es que la representación virtual del robot se realiza mediante Java-3D. Con esto se consigue que la interfaz de usuario (cliente) esté contenida toda en un applet de forma compacta, solucionando algunos problemas de compatibilidad e instalación con determinados sistemas operativos o navegadores que presenta la versión basada en VRML.

Además de una interfaz gráfica de usuario mejorada, (mostrada en la Figura 1.4) ésta versión ofrece nuevas características, entre las que destacan las descritas a continuación:

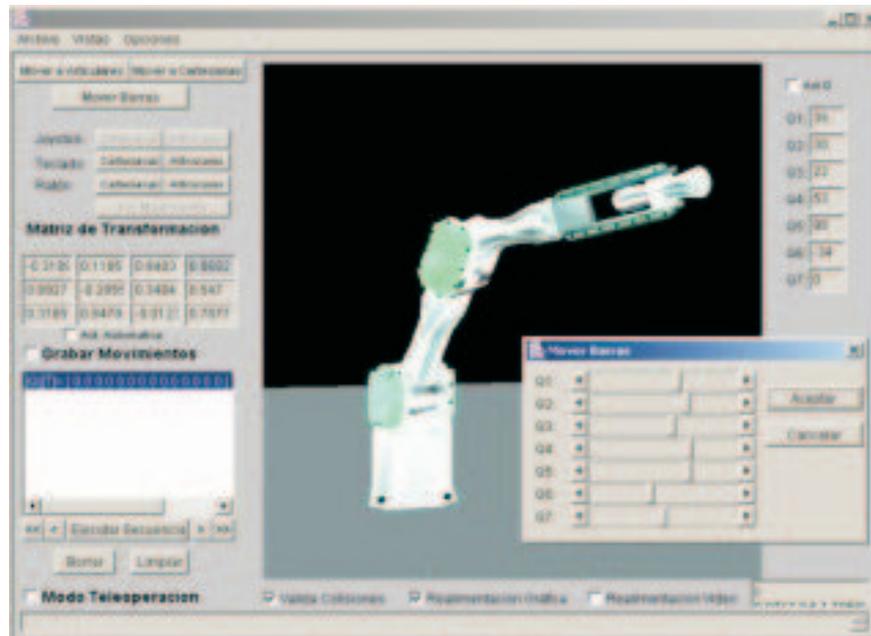


Figura 1.4: Interfaz de Robolab con un robot PA-10 de Mitshubitshi[43].

En primer lugar, el sistema utiliza una biblioteca de clases Java para el modelado de robots a partir de Java-3D desarrollada por el grupo AUROVA, que permite modelar, simular y tele-operar cualquier brazo robot, y no sólo el modelo Scorbot como la primera versión. La biblioteca permite especificar un robot definiendo sus articulaciones y eslabones, sus características cinemáticas, y su interfaz con el controlador correspondiente. La Figura 1.4 muestra la interfaz de usuario con otro robot: un PA-10 de Mitshubitshi presente en este laboratorio.

La interfaz de usuario soporta diferentes dispositivos para definir los movimientos del robot: teclado, ratón o joystick comercial para juegos. En caso de disponer de un joystick para juegos con realimentación de fuerza, cuando se detectan colisiones del robot con el entorno (o con síg mismo) en la simulación, el joystick reproduce las sensaciones de choque.

- Laboratorio de robótica y manufactura de ITESM:

La finalidad de este laboratorio es la de proveer educación a distancia generados a través de los mandos virtuales, debido a que se considera importante que el alumnado dependiente de esta institución tenga una aprendizaje continuo y no deje de practicar con este tipo de robot, claro sin dejar de lado la seguridad que esto tiene al hacerse de forma virtual[50].

El diseño y la implementación del Laboratorio Virtual de Robótica y Manufactura (LVRM) desarrollado en el campus Monterrey del ITESM, esta siendo en colaboración de los campus Morelos, Toluca, Edo. de México y Monterrey.

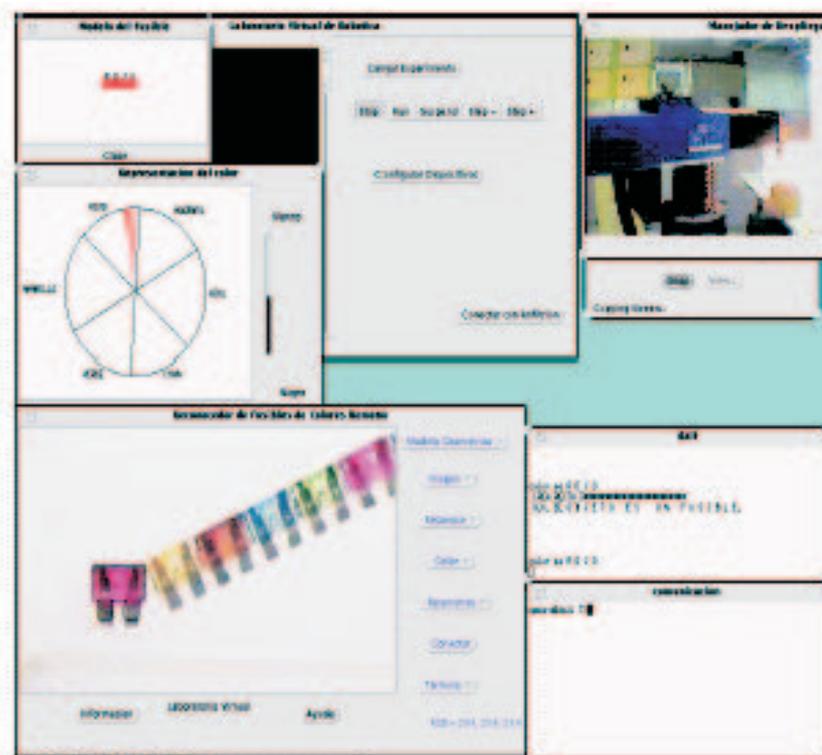


Figura 1.5: Interfaz de LVRM [50].

Componentes del LVRM][50]:

La versión actual del LVRM cuenta con los siguientes componentes:

Robot cilíndrico: el robot SEIKO con 4 grados de libertad (DOF) y una pinza de 2 posiciones, fue conectada a una computadora. Esta conexión permite controlar y programar las operaciones del robot desde la red.

Simulador del robot: se construyó un modelador de sólidos, que permite la definición de objetos compuestos y de mecanismos articulados mediante la introducción de restricciones cinemáticas. Al definir las variables articulares para un robot simulado, el modelador genera simultáneamente los “controladores” para el robot real; de forma que al aplicar desplazamientos sobre el modelo simulado, estos desplazamientos se aplican simultáneamente sobre el robot real.

Reconocedor de objetos por forma y color: el modelo del objeto a reconocer es alimentado como una lista de segmentos en 2D y una región para la búsqueda del color. Los segmentos son seleccionados y ordenados para guiar el reconocimiento.

El LVRM comprende dos entidades, residentes en diferentes lugares, comunicadas mediante una conexión Internet. El usuario demanda el servicio desde el “Invitado”, el cual se compone de un conjunto de programas que brindan el control para la operación y programación de los mecanismos remotos; el simulador es uno de tales programas. Los mecanismos, a excepción del simulador, residen físicamente en el “Anfitrión” y están conectados a sus respectivas computadoras para su control y operación.

El Invitado envía, a los mecanismos, comandos para su ejecución; enseguida recibe e ilustra las respuestas de estos mecanismos, sobre la ejecución de los comandos. Los comandos se refieren a actividades tales como “mover robot”, “adquirir y enviar secuencias de imágenes” o “reconocer objetos”. Por su parte, el Anfitrión recibe los comandos y los distribuye a los mecanismos respectivos para su ejecución; además supervisa la ejecución de los comandos y recibe las respuestas para reenviarlas al solicitante. Por lo cual, el Invitado y el Anfitrión cuentan con controladores que registran los comandos solicitados para realizar un experimento, tomando en consideración la aplicación que envía el comando, el dispositivo al cual se dirige y los parámetros de ejecución. Mediante este registro, se supervisa la ejecución completa del experimento.

- Simulación de robots articulados[51]:

Fue creado en el Centro Superior de Informática de la Universidad de la Laguna, España y sus autores son: “Benito José Cuesta Viera, Juan Lucio Cruz Méndez y Leopoldo Acosta Sánchez”.

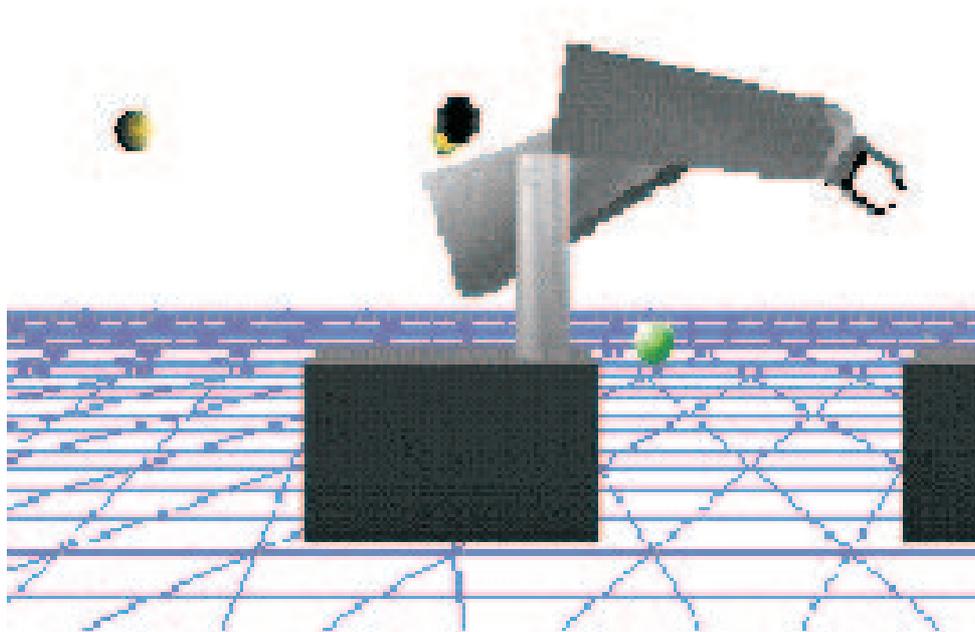


Figura 1.6: Interfaz de “Simulación de robots articulados” [51].

Este trabajo es un ambiente virtual del robot PUMA, realizado en Java 3D, tiene la particularidad de tener la opción de manipular un robot PUMA real y poder observar el desempeño a través de una cámara de video.

El proyecto está estructurado en cuatro grandes bloques estrechamente relacionados y que se presenta a continuación.

- Módulo de Simulación.- Herramienta de simulación del PUMA.

- Módulo de Acceso Remoto.- Permite interactuar con un robot PUMA real.
- Módulo de Monitorización.- Realiza un seguimiento del PUMA real.
- Complementos.- El WEB SITE, documentación del proyecto, videos, imágenes, etc.

El módulo de simulación está formado por un Applet que se encuentra ubicado en el servidor WEB del Centro Superior de Informática de la Universidad de la Laguna, aunque podría estar ubicado en cualquier servidor WEB. El módulo de acceso remoto y monitorización lo forman otros dos Applet que requieren un trato especial. El módulo de acceso remoto deberá estar ubicado un servidor WEB y además requiere el uso de un servidor RMI que utiliza un programa llamado rmiregistry que forma parte del Java Development Kit y permite el acceso a métodos remotos en Java. Además, este equipo servidor deberá estar conectado a un robot PUMA a través una PPI y un circuito. No entraremos en detalle de cómo funciona este circuito porque no forma parte de este proyecto, simplemente comentaremos que facilita la comunicación entre la PPI y el robot PUMA. El módulo de monitorización comparte el servidor WEB con el módulo de acceso remoto y requiere un dispositivo para adquirir las imágenes del robot.

1.6. Organización de la tesis

El trabajo de investigación se encuentra dividido en seis capítulos:

En el Capítulo 1 se plantea el objetivo general de este trabajo de tesis, así como los antecedentes, la justificación, contribuciones que se realizan a las diversas disciplinas relacionadas con la manipulación de robots, además de presentar un estudio del estado del arte.

En el Capítulo 2 se despliegan los conceptos básicos y definiciones importantes de robot y robótica de acuerdo con diversas asociaciones y organizaciones de prestigio dentro de la materia. Así como la descripción de los componentes y funciones de un robot manipulador. Se realiza un pequeño bosquejo histórico de la robótica, además de clasificaciones de robot de acuerdo a su aprendizaje, al control de sus trayectorias, a su autonomía, generación y la clasificaciones según diversas organizaciones.

Se menciona también la morfología de los robots manipuladores, sus elementos, configuraciones, y ejemplos de dispositivos destacados como el robot SCARA, PUMA de Unimation y el brazo de Stanford, el lector conocerá los subsistemas así como, las funciones que desempeñan y los componentes que integran a estos; para el desarrollo de los modelos matemáticos (los cuales se tratan en el Capítulo 5) se presenta un esbozo teórico de los modelos cinemático y dinámico.

En el Capítulo 3 una vez presentado lo necesario para comprender aspectos relacionados con los robots manipuladores, es turno de introducirse al control de estos dispositivos por lo se presenta un poco de teoría de estrategias de control clásicas y avanzadas, dichas estrategias son necesarias para proporcionar al robot estabilidad y precisión en los movimientos.

En el Capítulo 4 se hace mención a conceptos relevantes de realidad virtual esto con el fin del que el lector tenga una mejor percepción, también se presentan clasificación y características de la realidad virtual, así como algunas de las herramientas de edición y lenguajes existen, los cuales se pueden utilizar para de un ambiente virtual y la plataforma que se propone en este proyecto.

También se hace referencia a la construcción del robot virtual “PHANToM Virtual TRAZO” y lo concerniente al comportamiento complejo del mismo como es el desempeño visual cinemático y dinámico del robot virtual.

Otra parte importante que se aborda en este Capítulo es el análisis y diseño de la de la interfaz de “Seguimiento de trayectoria” que es la plataforma propuesta, se muestran los diagramas de flujo con información del sistema, diagramas de flujo de detalle de cada uno de los procedimientos y funciones que contiene el sistema, así como un diagrama de flujo que muestra el funcionamiento general de la aplicación y una descripción de los módulos que contiene el sistema.

En el Capítulo 5 se desarrollan los modelos matemáticos cinemático y dinámico de un robot antropomórfico de tres grados de libertad, empleando metodologías como la de Euler - Lagrange y la representación Denavit - Hartenberg. En este trabajo de tesis se tomaron en cuenta los aspectos dinámicos que influyen en un robot como pueden ser: fricción viscosa y seca, fuerza de Coriolis, gravedad entre otros, a partir de estos modelos se realizaron pruebas en específico del modelo dinámico para verificar la validez de dicho modelo, esto al evaluar sus propiedades matemáticas.

También se desarrollaron y analizaron los diferentes tipos de control como: PD, PID, PIDNL y Adaptable Pasivo Robusto, todos con seguimiento de trayectorias y los dos últimos con convergencia en tiempo finito, las simulaciones se llevaron a cabo en Matlab, además en este trabajo incluye una interfaz que implementa el control PIDNL en Visual C++ 6.0, el cual fue resuelto con el método de Runge Kutta de orden superior debido a la exigencia de precisión.

En el Capítulo 6 se exteriorizan las conclusiones, perspectivas y el trabajo a futuro.

Capítulo 2

Generalidades de modelado matemático de robots manipuladores

2.1. Introducción

Para comprender qué es un robot manipulador es preciso conocer algunos conceptos fundamentales de robótica y establecer su lugar en relación con otros tipos de automatización industrial. También es importante conocer componentes que lo integran y las funciones que realizan así como las clasificaciones que existen al respecto. En este capítulo se presenta una descripción de los conceptos importantes de la robótica, así como una clasificación.

Se sabe que el modelo matemático de un robot está basado en la cinemática y la dinámica, la primera está compuesta por los modelos cinemáticos tanto directos como inversos de posición, velocidad y aceleración, estos definidos por la cadena cinemática y empleando la metodología de “Denavit - Hartenberg” que proporcionan el movimiento del robot, y a lo que se refiere al segundo modelo es el que proporciona las capacidades de inercia y movimiento del robot manipulador que se obtiene a través de la formulación de “Euler - Lagrange” este tomando las propiedades dinámicas del robot.

Ahora bien en este capítulo se realiza una revisión de los conceptos que se mencionan, en forma general ya que el desarrollo de los modelos utilizados en la aplicación se describen en capítulos posteriores.

2.2. Conceptos preliminares de robótica

La robótica es un concepto de dominio público ya que en el mayor de los casos se sabe o se tiene la idea de lo que es la robótica, se sabe sus aplicaciones y el potencial que tiene. La robótica como hoy en día la conocemos, tiene sus orígenes hace miles de años. Nos basaremos en hechos registrados a través de la historia, y comenzaremos aclarando que antiguamente los robots eran conocidos con el nombre de autómatas, y la robótica no era reconocida como ciencia, es más, la palabra robot surgió hace mucho después del origen de los autómatas.

Los hombres creaban autómatas como un pasatiempo, eran creados con el fin de entretener a su dueño. Los materiales que se utilizaban se encontraban al alcance de todo el mundo, es decir, utilizaban maderas resistentes, metales como el cobre y cualquier otro material moldeable, esto es, que no necesitara o requiriera de algún tipo de transformación para poder ser utilizado en la creación de los autómatas.

Estos primeros autómatas utilizaban, principalmente, la fuerza bruta para poder realizar sus movimientos. A las primeras máquinas herramientas que ayudaron al hombre a facilitarle su trabajo no se les daba el nombre de autómatas, sino más bien se les reconocía como artefactos o simples máquinas[40].

Robótica

Conjunto de disciplinas que convergen hacia un nuevo objetivo: “cumplir las aspiraciones de suministrar al hombre un mecanismo que lo libere de actividades tediosas y/o peligrosas y que, como su nombre lo indica (*“Robotnik”*, del Checo siervo; *“Robota”*, del ruso trabajo), estén a su servicio con un buen grado de automatización y dependencia”[9].

Una definición más sencilla de la robótica puede ser:

“La robótica es la conexión inteligente de percepción para la acción”[12].

Una perspectiva diferente:

“La robótica es la disciplina que involucra:

- El diseño, manufactura, el control, y la programación de autómatas;
- El uso de autómatas para solucionar problemas;
- El estudio de los procesos de control, los sensores, y los algoritmos usados en humanos, animales y máquinas; y
- La aplicación de los procesos de control y algoritmos para el diseño de autómatas”[13].

Robot

Aunque la palabra robot fué introducida al mundo por Karel Capek, un dramaturgo Checo, no fué sino hasta 1941 que Issac Asimov emplea el término *“robot”* en su historia titulada *“Robot AL76 Goes Atray”*, y luego en 1942 en *“Runaround”* utiliza el término *“Robótica”* y enuncia las *“Tres leyes de la robótica”*[9].

Principios o leyes de un robot

1. Un robot no puede dañar a un ser humano o, por inacción, permitir que un ser humano resulte dañado.
2. Un robot debe obedecer las órdenes dadas por los seres humanos excepto cuando tales órdenes entren en conflicto con la primera ley.
3. Un robot debe proteger su propia existencia hasta donde ésta protección no entre en conflicto con la Primera o la Segunda ley.

Se encuentran definiciones de Robot como ingenio mecánico, controlado electrónicamente, capaz de moverse y ejecutar de forma automática acciones diversas, siguiendo un programa establecido.

Veamos otras definiciones:

- **Asociación Japonesa de Robots Industriales (JIRA: Japan Industrial Robots Association):**

“Es una máquina capaz de realizar movimientos versátiles parecidos a los de nuestras extremidades superiores, con cierta capacidad sensorial y de reconocimiento y capaces de controlar su comportamiento”.

En la Figura 2.1 se muestra la similitud de las extremidades de un robot y las del ser humano.

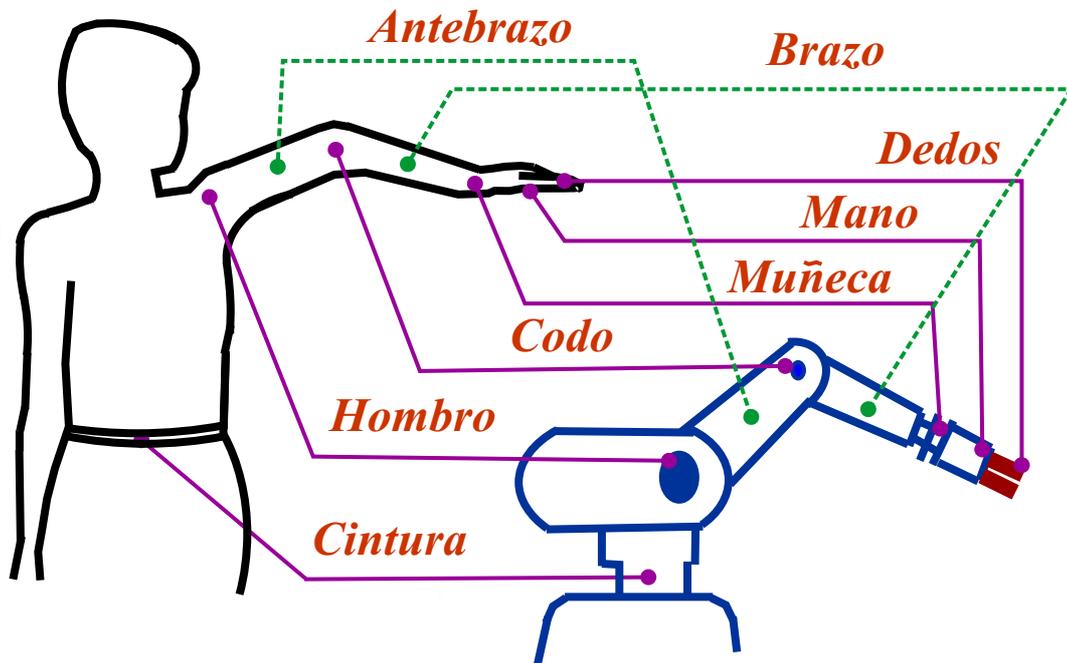


Figura 2.1: Similitud con el brazo humano[31].

- **Asociación de Robots Industriales (RIA: Robot Industrial Association or Robot Institute of America):**

“Es manipulador multifuncional y reprogramable diseñado para mover cargas, materiales, piezas, herramientas y dispositivos especiales según variadas trayectorias, programados para realizar diferentes tareas” [2, 32].

- **Organización Internacional de Estándares (ISO: International Standard Organization):**

“Un robot es una máquina formada por un mecanismo que posee varios grados de libertad; a menudo da la apariencia en uno o más brazos que terminan en una muñeca, una pieza de trabajo o un dispositivo de inspección. Posee una unidad de control con memoria y en algunos casos sensores y dispositivos de adaptación para extraer información de su medio ambiente. Estas máquinas de propósitos múltiples se diseñan para llevar a cabo funciones repetitivas fácilmente modificables” [32].

- **Nuevo diccionario Colegial de Webster (Webster’s New Collegiate Dictionary):**

“Es un aparato o dispositivo automático que ejecuta funciones atribuidas ordinariamente a humanos u operadores que aparentan tener inteligencia casi humana”.

- **Asociación Francesa de Normalización (AFNOR):**

“Es un manipulador controlado en posición, reprogramable, polivalente, de varios grados de libertad, capaz de manipular materiales, piezas, herramientas y dispositivos especializados, durante la ejecución de movimientos variables y programados para la realización de una variedad de tareas” [10].

Como se puede notar un robot no podría ponerse en marcha sin la colaboración de varias áreas del conocimiento como son la mecánica, el control y automatización, la electrónica y la computación.

Los robots actuales están constituidos por un brazo con varias articulaciones, a cuyo extremo se acopla el manipulador, con “n” grados de libertad, todo ello montado sobre una base que puede ser fija o móvil, (ver Figura 2.2). Dado el carácter inherente de automático, un robot debe ser capaz de realizar por sí mismo las operaciones que permitan localizar los objetos a manipular, sujetarlos con la presión requerida y volver a depositarlos en el sitio disponible.



Figura 2.2: Robot montado sobre una base[30].

Todo ello se consigue con el uso de sensores adecuados que permitan al sistema saber en todo momento dónde está el efector final, dónde y cómo está la pieza a manipular, cuál es la presión ejercida sobre la misma, velocidad de traslación relativa, etc., y además, controlar todos estos parámetros continuamente o con el tiempo de aceptable resolución. Por ello, la computadora forma parte integrante de los robots industriales desde su nacimiento.

Actualmente y con la ayuda del microprocesador ha surgido una segunda generación de robots inteligentes que se caracterizan por ser capaces de relacionarse con el entorno y tomar decisiones en tiempo real[28][29].

En las Figuras 2.3 y 2.25, se pueden observar algunos elementos de los que enseguida se describen.

- Articulaciones: es la unión de dos elementos que se mueven uno con respecto del otro.
- Gripper u órgano terminal: o efector final, dispositivo que se instala en el extremo del manipulador, el cual interactúa con los objetos de su entorno al efectuar una tarea. Consiste en un sistema mecánico capaz de sujetar y soltar objetos (pinza mecánica) o herramienta capaz de realizar alguna operación.
- Actuadores: son los dispositivos que producen la potencia mecánica que se requiere para mover los eslabones del manipulador. Pueden ser motores eléctricos, pistones y motores neumáticos e hidráulicos.

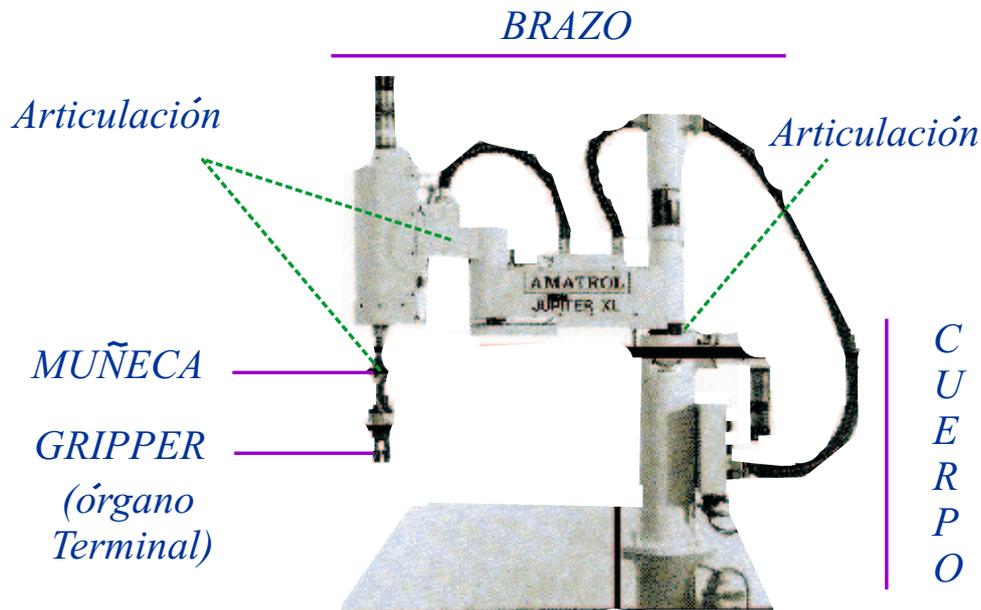


Figura 2.3: Vista de los componentes de un robot manipulador[31].

- Transmisiones de potencia: permiten la conducción del movimiento generado en los actuadores hasta cada eslabón del manipulador. Pueden ser del tipo de bandas dentadas, engranes, cadenas, etc.
- Grados de libertad: corresponde al número mínimo de variables independientes que es necesario definir para determinar la situación en el espacio de todos los eslabones del manipulador. La regla de *K. Kutzbach* permite determinar el número de grados de libertad de un robot con mecanismos especiales en términos del número de eslabones y de articulaciones.

$$G = 6 \times (e - 1) - 5 \times a \quad (2.1)$$

Donde:

G = grados de libertad.
 e = número de eslabones.
 a = número de articulaciones.

- Coordenadas generalizadas: representadas por los ángulos entre los eslabones del manipulador, definidos como: $q_1, q_2 \dots q_n$, donde n corresponde al número de grados de libertad. En la acción de control son las variables de control del sistema.
- Coordenadas operacionales: representadas por la posición cartesiana en x, y y z , y la orientación en λ, μ, ν del órgano terminal en el espacio. Durante la acción de control, las coordenadas operacionales, permiten conocer la posición instantánea que el robot adquiere con su movimiento.
- Espacio de configuraciones admisibles: o espacio de trabajo del robot, es la zona del espacio físico que puede ser alcanzada por un punto del órgano terminal.

Algunos otros conceptos de robótica pueden ser revisados en el glosario de ésta tesis.

Robótica y automatización

La automatización y la robótica son dos tecnologías estrechamente relacionadas. La automatización se puede definir como una tecnología que está relacionada con el empleo de sistemas mecánicos, electrónicos y basados en las computadoras en la operación y control de la producción. Y la robótica emplea sistemas electro-mecánicos, controlados por computadora para la ejecución de diversas tareas, incluidas las relacionadas con la producción.

2.2.1. Antecedentes

Los primeros trabajos que condujeron a la construcción de robots en la actualidad se remontan a los últimos años de la década de los cuarenta, después de la Segunda Guerra Mundial. Donde comenzaron programas de investigación en Oak Ridge y Argonne National Laboratories para desarrollar manipuladores mecánicos controlados de forma remota con el objetivo de manejar materiales radioactivos. Estos sistemas eran de tipo “maestro - esclavo”, diseñados para reproducir fielmente los movimientos de mano y brazo realizados por un operario humano. El manipulador maestro era guiado por el usuario a través de una secuencia de movimientos, mientras que el manipulador esclavo duplicaba la unidad maestra tan fidedignamente tal como lo era posible.

El trabajo sobre manipuladores “maestro-esclavo” fue seguido rápidamente por sistemas más sofisticados capaces de operaciones repetitivas autónomas.

A mediados de los años cincuenta, George C. Devol desarrolló un dispositivo que llamó “dispositivo de transferencia programada articulada”, un manipulador cuya operación podía ser programada (y, por tanto, cambiada) además podía seguir una secuencia de pasos de movimientos determinados por las instrucciones del programa. Trabajos posteriores de Devol y Joseph F. Engelberger condujeron al primer robot industrial, introducido por Unimation Inc., en 1959. La clave de este dispositivo era el uso de una computadora en conjunción con un manipulador para producir una máquina que podía ser “enseñada” para realizar una variedad de tareas de forma automática. Estos robots se podían reprogramar y cambiar de herramienta a un costo relativamente bajo para efectuar otros trabajos cuando cambiaban los requisitos de fabricación.

Aunque los robots programados ofrecían una herramienta de fabricación nueva y potente, en los años setenta se descubrió que la flexibilidad de estas máquinas se podía mejorar significativamente mediante el uso de una realimentación sensorial.

Al comienzo de esta década, H. A. Ernest publicó el desarrollo de una mano mecánica controlada por computadora con sensores táctiles. Este dispositivo, llamado el MH-1, podía “sentir” bloques y usar esta información para controlar la mano de manera que apilaba los bloques sin la ayuda de un operario. Este trabajo es uno de los primeros ejemplos de un robot capaz de conducta adaptativa en un entorno razonable no estructurado. Este programa de investigación posteriormente evolucionó como parte del proyecto MAC, y se le añadió una cámara de televisión para comenzar la investigación sobre la percepción en la máquina.

Durante el mismo periodo, Tomovic y Boni (1962) desarrollaron una mano prototipo provista con un sensor de presión que detectaba el objeto y proporcionaba una señal de alimentación de entrada a un motor para iniciar uno de dos modelos de aprehensión.

En 1963, la American Machine and Foundry Company (AMF) introdujo el robot comercial VERSATRAN. Comenzando en este mismo año, se desarrollaron diversos diseños de brazos para manipuladores, tales como el brazo Roehampton y el de Edinburgh.

A finales de los años sesenta, McCarthy y sus colegas en el Stanford Artificial Intelligence Laboratory publicaron el desarrollo de una computadora con manos, ojos y oídos (es decir, manipuladores, cámaras de TV y micrófonos). Demostraron un sistema que reconocía mensajes hablados, “veía” bloques distribuidos sobre una mesa, y los manipulaba de acuerdo con instrucciones. Durante este periodo, Pieper (1968) estudió el problema cinemático de un manipulador controlado por computadora.

Mientras tanto, otros países (en particular Japón) comenzaron a ver el potencial de los robots industriales. Ya que en 1968, la compañía japonesa Kawasaki Heavy Industries negoció una licencia con Unimation Inc., para sus robots.

Uno de los desarrollos poco usuales sucedió en 1969 cuando se desarrolló un camión experimental por la General Electric para la Armada Americana. El mismo año se desarrolló el brazo Boston y al año siguiente el brazo de Stanford (ver Figura 2.4), que estaba equipado con una cámara y controlado por computadora. Algunos de los trabajos más serios en robótica comenzaron cuando estos brazos se utilizaron como robots manipuladores[37].

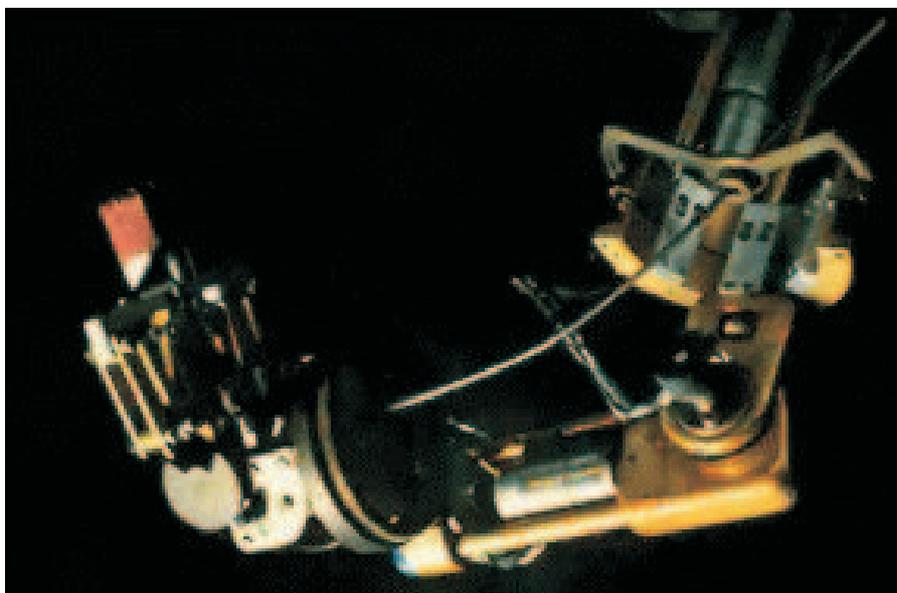


Figura 2.4: El brazo de Stanford 1969[37].

Las primeras aplicaciones industriales en Europa, de robots industriales en cadenas de fabricación de automóviles, datan de los años 1970 y 1971. En este último año, Kahn y Roth analizan el comportamiento dinámico y el control de un brazo manipulador.

Durante la década de los 70, la investigación en robótica se centra en gran parte en el uso de sensores externos para su utilización en tareas de manipulación. Es también en estos años cuando se consolida definitivamente la presencia de robots en las cadenas de montaje y plantas industriales en el ámbito mundial. Se lanza también Shakey, el primer vehículo autoguiado controlado por inteligencia artificial (ver Figura 2.5)[16].

En 1972 se desarrolló en la universidad de Nottingham, Inglaterra, el SIRCH, un robot capaz de reconocer y orientar objetos en dos dimensiones. Este mismo año, la empresa japonesa Kawasaki instala su primera cadena de montaje automatizada en Nissan, Japón, usando robots suministrados por Unimation, Inc.

En 1973, Bolles y Paul utilizan realimentación visual en el brazo Stanford para el montaje de bombas de agua

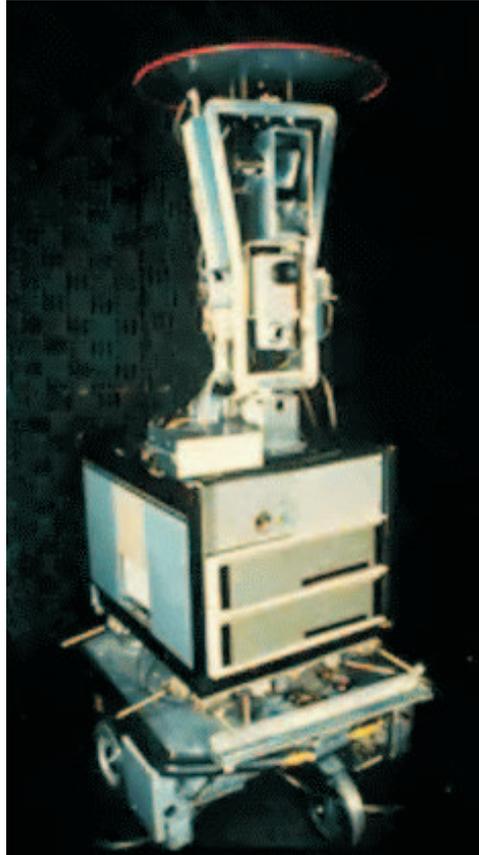


Figura 2.5: Shakey, el primer vehículo autoguiado controlado por inteligencia artificial (1970)[16].

de automóvil. También este mismo año, la compañía sueca ASEA (futura ABB), lanza al mercado su familia de robots IRB 6 e IRB 60, para funciones de perforación de piezas.

En 1974, Cincinnati Milacron introdujo su primer robot industrial controlado por computadora. Lo llamó *“The Tomorrow Tool”* (*“La herramienta del mañana”*) o *“T3”*, que podía levantar más de 100 libras así como seguir a objetos móviles en una línea de montaje.

En 1978 se introdujo el robot PUMA (Programmable Universal Machine for Assembly) para tareas de montaje por Unimation, basándose en diseños obtenidos en un estudio de la General Motors. En este mismo año el robot T3 se adaptó y programó para realizar operaciones de taladrado y circulación de materiales en componentes de aviones, bajo el patrocinio de Air Force[2][5].

En 1979 Japón introduce el robot SCARA (Selective Compliance Assembly Robot Arm), y la compañía italiana DEA (Digital Electric Automation), desarrolla el robot PRAGMA para la General Motors.

En la década de los 80 se avanza en las técnicas de reconocimiento de voz, detección de objetos móviles y factores de seguridad. También se desarrollan los primeros robots en el campo de la rehabilitación, la seguridad, con fines militares y para la realización de tareas peligrosas. Así por ejemplo, en 1982, el robot Pedesco, se usa para limpiar un derrame de combustible en una central nuclear. También se pone un gran énfasis en los campos de visión artificial, sensorización táctil y lenguajes de programación. Gracias a los primeros pasos dados por compañías como IBM o Intelledex Corporation, que introdujo en 1984 el modelo ligero de ensamblaje 695, basado

en el microprocesador Intel 8087 y con software Robot Basic, una modificación del Microsoft Basic, actualmente se tiende al uso de una interfaz (el ordenador) y diversos lenguajes de programación especialmente diseñados, que evitan el “cuello de botella” que se producía con la programación “clásica”. Esta puede ser ahora on-line u off-line, con interfaces gráficas (user-friendly interfaces) que facilitan la programación, y un soporte SW+HW que tiende a ser cada vez más versátil.

En 1986 HONDA, la empresa Japonesa inicia un proyecto para construir un robot humanoide, su evolución y sus numerosos problemas se mantienen en secreto. Pero para 1997 HONDA presenta P3 un enorme robot humanoide. Cuando los resultados se hicieron públicos muchos investigadores que trabajaban en proyectos similares se quedaron atónitos, sus proyectos estaban a años luz de los resultados obtenidos por HONDA[39].

En 1999 la empresa SONY lanza “Aibo” un perro-robot. Los resultados en forma de publicidad gratuita que HONDA ha obtenido provocan una carrera de marketing tecnológico entre empresas de tecnología en Japón. Y en el año 2000 SONY presenta un pequeño humanoide en la “Robodex 2000”. Mientras los EEUU miran a sus robots en Marte, Japón mira a sus robots a la cara. Cada uno en su terreno es el rey de la robótica[39].

Para el año 2003, aquel robot humanoide de SONY, Qrio (“*Quest for curiosity*”) (ver Figura 2.6) , se convierte en el primer humanoide comercial completamente autónomo capaz de correr. De esta forma HONDA sería el primero en caminar, pero SONY el primero en correr. La carrera está abierta y otras empresas anuncian su propósito de unirse[39].



Figura 2.6: Robot Qrio de Sony[39].

Y en 2004 la primera edición del “Darpa Grand Challenge”. La guerra de Irak de 2003 y sus numerosas bajas puso en evidencia la necesidad de reducir las bajas militares estadounidenses en futuros conflictos. Se multiplica la financiación de este tipo de proyectos. Y los robots forman ya parte importante para la guerra[39].

2.2.2. Clasificación de los robots

Robots manipuladores

Robots de servicio.

Los robots de servicio son dispositivos electromecánicos móviles o estacionarios, dotados de uno o varios brazos mecánicos, controlados por computadora (ver Figura 2.7) y que realizan tareas de servicio, no industriales [6].

Dentro de esta categoría están los robots especiales utilizados como ayuda a minusválidos, los robots de laboratorio, los robots didácticos, los robots que se utilizan en operaciones de alto riesgo, los robots de exploración submarina, entre otros.

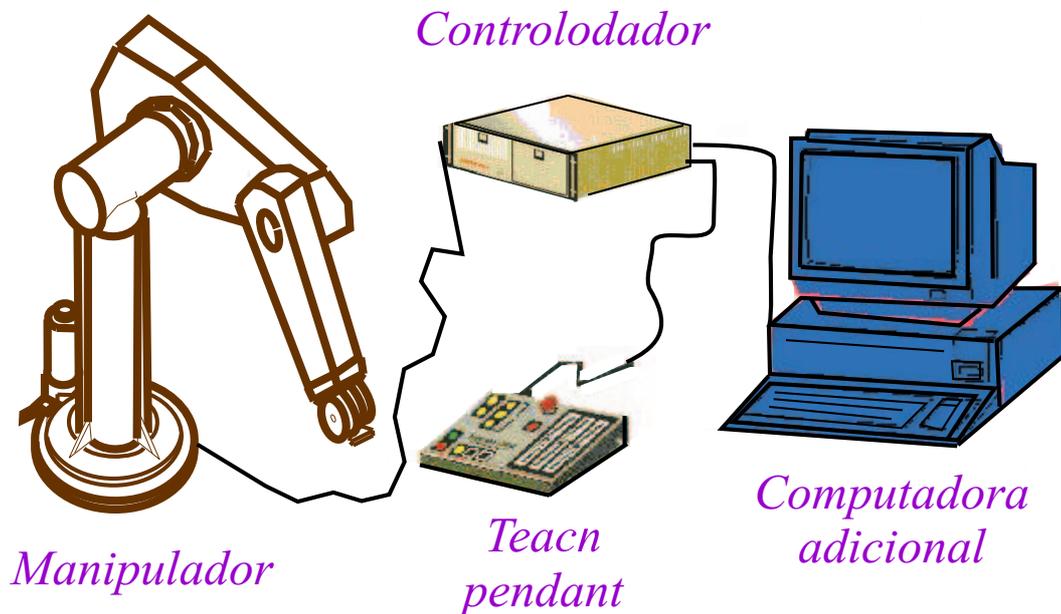


Figura 2.7: Componentes de un robot.

Robots industriales.

Se pueden clasificar con base en diferentes criterios, entre ellos está la clasificación de la AFRI (Asociación Francesa de Robótica Industrial) y de la IFR (Federación Internacional de Robótica)[6].

Clasificación de los robots según la AFRI(Asociación Francesa de Robótica)[6]:

- TIPO A: manipulador con control manual o telemando.
- TIPO B: manipulador automático con ciclos preajustados; regulación mediante fines de carrera o topes; accionamiento neumático, eléctrico o hidráulico.
- TIPO C: robot programable con trayectoria continua o punto a punto. Carece de conocimientos sobre su entorno.

- TIPO D: robot capaz de adquirir datos de su entorno, readaptando su tarea en función de estos.

Clasificación según la IFR (Federación Internacional de Robótica)[6]:

Distingue entre cuatro tipos de robots:

- Robot secuencial.
- Robot de trayectoria controlable.
- Robot adaptativo.
- Robot telemanipulado.

Clasificación en generaciones:

- 1ª Generación: repite la tarea programada secuencialmente. No toma en cuenta las posibles alteraciones del entorno.
- 2ª Generación: adquiere información limitada de su entorno y actúa en consecuencia. Puede localizar, clasificar (visión), detectar esfuerzos y adaptar sus movimientos en consecuencia.
- 3ª Generación: su programación se realiza mediante el empleo de un lenguaje natural. Posee capacidad para la planificación automática de tareas.

Clasificación de acuerdo a su aprendizaje[6][16]:

- Robots Manuales: exigen que el movimiento de sus elementos sea controlado por el hombre.
- Robots autónomos de secuencia fija: permiten gobernar el movimiento de sus elementos en forma autónoma pero no es posible alterar sus operaciones o secuencia de operaciones.
- Robots autónomos de secuencia variable: permite gobernar el movimiento de sus elementos en forma autónoma y además, es posible alterar algunas de las características de los ciclos de trabajo.
- Robots programables: pueden ser programados mediante lenguajes específicos o mediante un proceso de aprendizaje en el cual un operador enseña inicialmente una secuencia de movimientos, el sistema los memoriza y se limita a repetirla indefinidamente.
- Robots inteligentes: son manipuladores o sistemas mecánicos multifuncionales controlados por computadora, capaces de relacionarse con su entorno a través de sensores y de tomar decisiones en tiempo real. Pueden analizar su entorno y reaccionar ante sus modificaciones o alteraciones, para ello requiere del concurso de sus sistemas de análisis y control de sensores adecuados, de control de esfuerzos, de seguimiento de trayectorias, de reconocimiento de imágenes, etc..

Clasificación de acuerdo al control de sus trayectorias[6][16]:

Existen dos formas básicas de controlar la trayectoria de un robot:

- Control de trayectoria punto a punto: con el control punto a punto, el robot es programado para realizar una pausa en cada punto para planear el próximo paso en una trayectoria predeterminada. A pesar del hecho de que el movimiento no es controlado entre los set points, ésta generalmente sucede a lo largo de una trayectoria natural, dependiendo de la configuración geométrica del robot. Por otro lado el robot controlado punto a punto ofrece mayor precisión en términos de repetibilidad.

- Control de trayectoria continua: el resultado de un control de trayectoria continua es un movimiento más suave a lo largo de la trayectoria definida pero se disminuye a la máxima velocidad posible en los recorridos, lo cual es una función del tamaño de los pasos calculados por la computadora maestra en tiempo real, usando métodos de interpolación. En algunos modelos (diseñados para finales de los 80's), la eficiencia será mas baja comparada con el mismo robot operando en modo de control punto a punto.



Figura 2.8: Robot que su forma de desplazamiento es con ruedas[35].

Robots móviles

Los robots móviles, pueden desplazarse usando ruedas, pistas o patas, y usualmente se mueven en dos dimensiones; sin embargo, los robots que bucean o navegan en el agua, así como los que vuelan, también están clasificados como robots móviles, y se mueven en tres dimensiones, siendo por lo tanto más difíciles de controlar. Por lo tanto se pueden clasificar de la siguiente manera [6]:

De acuerdo al medio en el cual se desempeñan:

- Robots acuáticos: robots submarinos de investigación de las fuerzas navales y de los investigadores científicos.
- Robots terrestres: o de navegación sobre terreno sólido, como los diseñados para la NASA.
- Robots voladores: tales como el efigenia, o el helicóptero robot mejorado con sistemas difusos.

De acuerdo a su sistema de locomoción:

- Deslizante.
- Rodante.
- Caminadora. Bípeda, múltipeda.
- Rulante. Orugas, cadenas.
- Otras

Según su forma de desplazamiento.

Su desplazamiento puede ser a su vez, robots con patas, ruedas (ver Figura 2.8, orugas, turbinas de baja profundidad, insectos, exploradores, de transporte, cooperativos, guardianes, etc.).



Figura 2.9: Robot Androide [36].

Según el grado de Autonomía

Autonomía: la facultad de autonomía de los robots, se puede considerar bajo diversos aspectos; la capacidad sensorial, la capacidad de decisión, y las capacidades de adaptación, aprendizaje y cooperación. En términos generales, se puede definir como la capacidad del robot para sentir la situación actual y actuar apropiadamente con respecto a ella[6].

- Robots Móviles Autónomos: tienen la capacidad para responder a situaciones cambiantes, ambiguas o impredecibles, sin necesidad de la intervención humana.
- Robots Teleoperados: operados a distancia. Las tareas de percepción del entorno, planificación y manipulación compleja son realizadas por humanos. Las mayores dificultades radican en las limitaciones del hombre en la capacidad de procesamiento numérico y precisión y, sobre todo, en el acoplamiento y coordinación entre el hombre y el robot. El retraso de transmisión de información juega un papel importante y su consideración resulta fundamental en el diseño del sistema de control.

Según el sistema de guiado.

- Vehículos de Guiado Automático (AGV): vehículos que se desplazan por rutas prefijadas, marcadas con cables guías, imanes, patrones ópticos. Su uso principal se encuentra en entornos industriales en los que se

requiere desplazar materiales. También se emplean en hospitales, oficinas y otros entornos donde cumplen la función de repartidores de comida, insumos, documentos, etc.

- Vehículos Autónomos (AV): generan automáticamente sus movimientos ayudados por la información que sus sensores proporcionan.

Otros Robots[6]

- Androides: robots con los cuales se intenta reproducir total o parcialmente la forma y el comportamiento cinemático del ser humano. En la actualidad son poco evolucionados, destinados fundamentalmente al estudio y experimentación. En la Figura 2.9 se puede ver un robot Androide.
- Zoomórficos: los robots zoomórficos que considerados en sentido no restrictivo podrían incluir también a los androides, constituyen una amplia clase caracterizada fundamentalmente por sus sistemas de locomoción que imitan a los diversos seres vivos.
- Híbridos: son aquellos cuya estructura se sitúa entre las ya expuestas bien sea por conjunción o yuxtaposición.

2.2.3. Morfología de los robots manipuladores

Elementos y articulaciones

Los elementos o eslabones son los cuerpos rígidos que conforman un manipulador, mientras que las articulaciones son las partes que permiten el movimiento de los elementos que lo conforman.

En principio, las articulaciones pueden ser de diferentes tipos en la robótica (ver Figura 2.10) [17]:

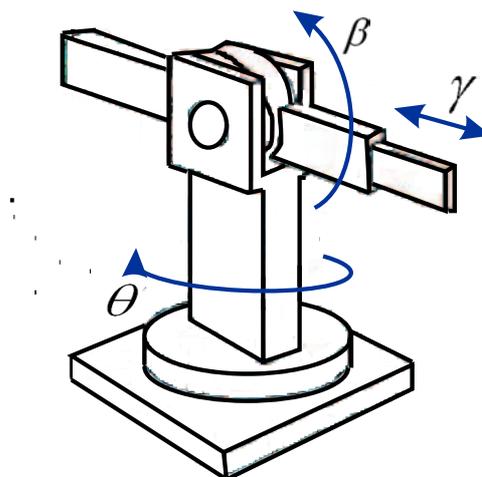


Figura 2.10: Tipos de articulaciones : las articulaciones β y θ son de rotación (R), mientras que γ pertenece a una articulación prismática (P)[30].

- De revolución (de revoluta, rotativas): el elemento asociado a la articulación rota con respecto al anterior a lo largo de un eje que atraviesa a ambos elementos. Se denotan con la letra R.
- Prismáticas (deslizantes): son aquellas en las que un elemento se desplaza con respecto al precedente a lo largo de un eje. Se denotan con la letra P.

El número de articulaciones determina los grados de libertad (Degrees Of Freedom, DOF) del manipulador. Por ejemplo, el robot que se muestra en la Figura 2.10 posee tres grados de libertad. Típicamente un manipulador posee seis grados de libertad independientes, tres de posición y tres de orientación.

Configuraciones

Aunque en principio un manipulador es un dispositivo de propósito general, en la práctica los manipuladores son usualmente diseñados con al menos una clase extensa de aplicaciones en mente, como soldadura, manipulación de materiales, y ensamble.

Los manipuladores usualmente son clasificados cinemáticamente basándose en el brazo. La mayoría de los manipuladores caen dentro en uno de estos cinco tipos geométricos:

- Cartesiano o rectangular: de configuración PPP (ejes prismáticos). Posee tres ejes perpendiculares y el movimiento lineal se realiza a lo largo de dichos ejes. Este tipo de manipuladores presentan la desventaja de necesitar un espacio muy grande para trabajar y un gran porcentaje de este espacio no está incluido dentro de su volumen de trabajo, es decir, no es posible acceder a todo este espacio con el efector final del robot. En este tipo de configuración los algoritmos de control son más simples. En las Figuras 2.11, 2.12 y 2.13, se muestra el esquema, espacio de trabajo y un ejemplo de un robot cartesiano [17].

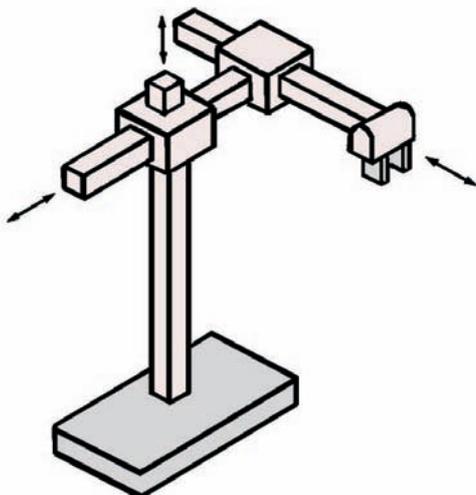


Figura 2.11: Esquema de un robot cartesiano [30].

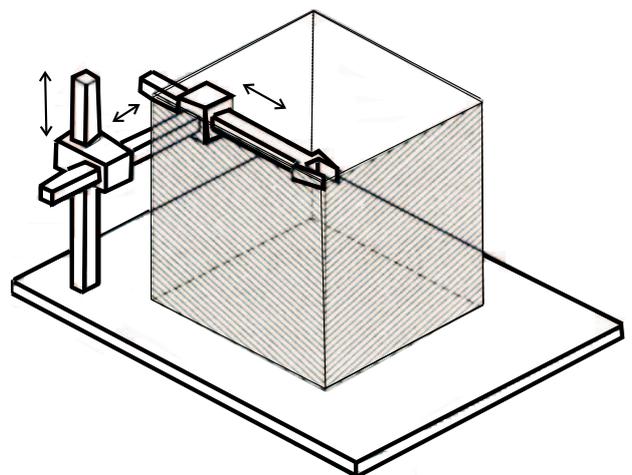


Figura 2.12: Espacio de un robot cartesiano [17].



Figura 2.13: Un robot cartesiano[57].

- Cilíndrico: de configuración RPP (primer eje de revolución, segundo y tercer eje prismáticos).

Consisten de dos ejes lineales que se deslizan ortogonalmente entre sí y están montados sobre una base rotatoria, tal como se aprecia en la Figura 2.15.

Por sus características, el volumen o entorno de trabajo corresponde a una porción de cilindro del cual deriva su nombre. Estos robots tienen un volumen de acceso restringido [17]. Ver Figuras 2.14, 2.15 y 2.16.

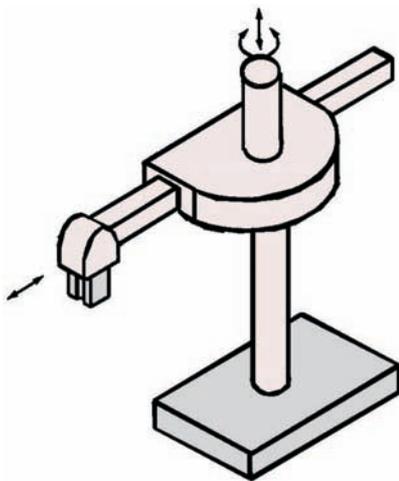


Figura 2.14: Esquema de un robot cilíndrico [30].

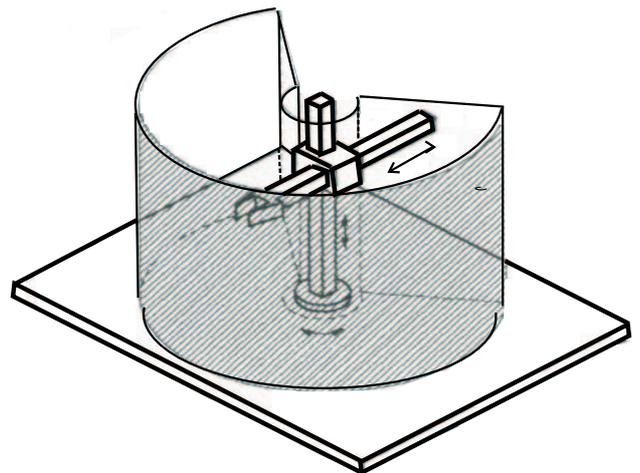


Figura 2.15: Espacio de un robot cilíndrico [17].

- Polar o esférico: este tipo de robot manipulador es de configuraciones RRP (primer y segundo eje revolución y el tercer eje primático).



Figura 2.16: Ejemplo de un robot cilíndrico[57].

El alcance de un punto en el espacio se logra mediante un brazo telescópico que se extiende y también se retrae a lo largo de su eje, este se encuentra montado sobre una base giratoria con dos ejes rotacionales que son independientes.

Su principal inconveniente es que presenta grandes vacíos en su volumen de trabajo[17]. Ver Figuras 2.17, 2.18 y 2.19.

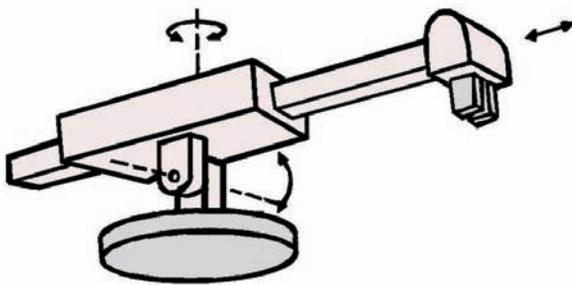


Figura 2.17: Esquema de un robot polar [30].

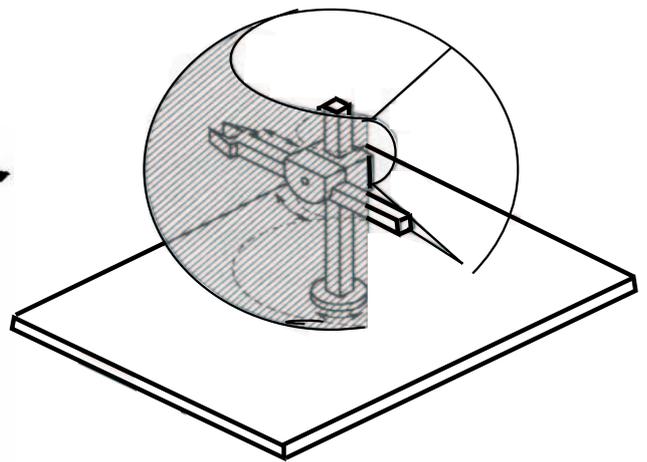


Figura 2.18: Espacio de un robot polar [17].



Figura 2.19: Ejemplo del robot polar[57].

- Robot SCARA (Selective Compliance Assembly Robot Arm): es una versión especial del brazo articulado, o mas bien mejor aún, ya que es una combinación de las configuraciones de revolución y cilíndrica.

Sus articulaciones rotativas de hombro y de codo giran alrededor de ejes verticales, permitiendo que se efectúen giros en un plano horizontal.

Esta configuración proporciona rigidez para el robot en la dirección vertical, pero flexibilidad en el plano horizontal, lo cual le hace ideal para tareas de montaje.

Las cargas gravitacionales, coriolis y fuerzas centrífugas, no someten a tensiones las estructuras tanto como lo harían si los ejes fueran horizontales [17]. Esta ventaja es muy importante a altas velocidades y altos requerimientos de precisión en el robot. Ver Figuras 2.20 y 2.21.

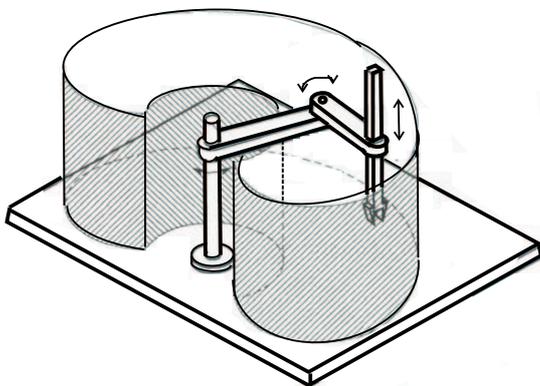


Figura 2.20: Espacio de trabajo del robot tipo SCARA [17].

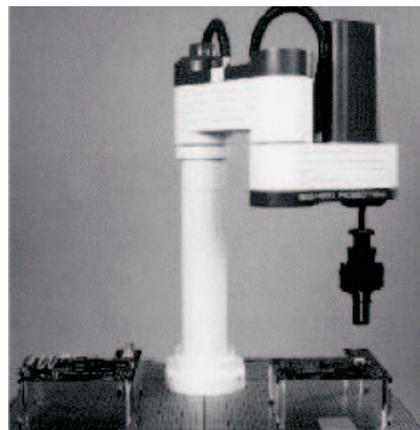


Figura 2.21: Ejemplo de robot SCARA[17].

- Robot de brazo articulado (con articulaciones rotacionales): de configuración RRR. Es el que más se asemeja al brazo humano, por ello se le denomina Antropomórfico. Aunque presentan algunos vacíos en su volumen de trabajo, son los que tienen mejores características de acceso con el efector final a la mayoría de los puntos comprendidos en su universo de trabajo. Al ser todas sus articulaciones de rotación, presentan mayores inconvenientes en la compensación de cargas e inercias de rotación.

En este tipo de configuración, los errores de posicionamiento tienden a aumentar a medida que se extiende el brazo. Su precisión es inherentemente mayor en el entorno de la base. La posición de los actuadores, requiere una mayor compensación de cargas, ya que están ubicados directamente sobre las articulaciones [17]. Ver Figuras 2.22, 2.23 y 2.24.

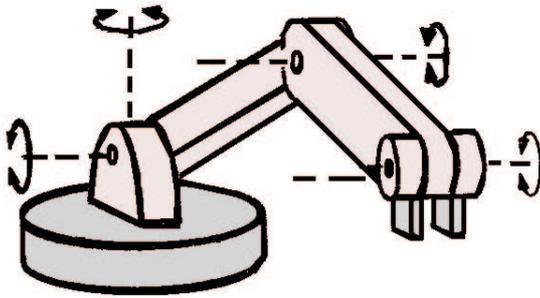


Figura 2.22: Esquema de un robot de brazo articulado [30].

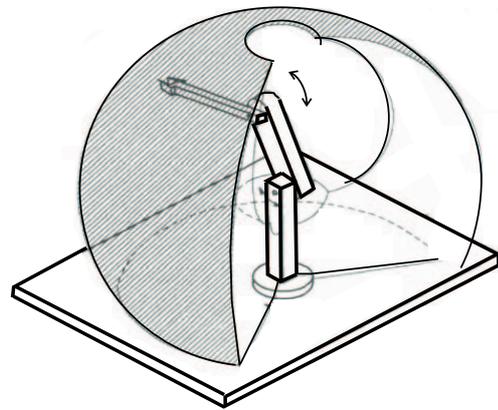


Figura 2.23: Espacio de trabajo de un robot articulado [17].

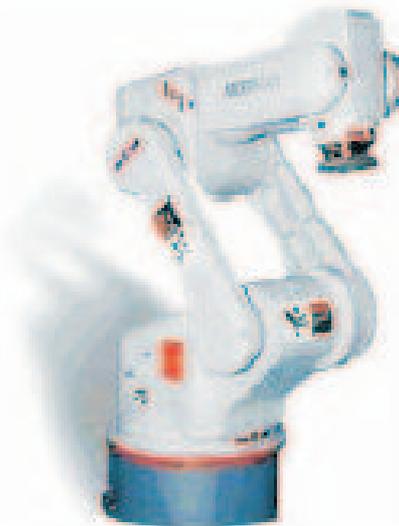


Figura 2.24: Ejemplo de robot de brazo articulado[57].

A continuación se muestran algunos tipos de robots y sus características:

PUMA Unimation

El manipulador industrial PUMA (Programmable Universal Machine for Assembly) es uno de los robots más populares dentro de la industria. Fue diseñado en 1978 por Unimation para realizar tareas de montaje. Cuenta con seis articulaciones, todas ellas de revolución.

La Tabla 2.1 muestra los Parámetros Denavit - Hartenberg (PDH) para un manipulador PUMA modelo 560 [14].

i	$\alpha_i - 1$	$a_i - 1$	d_i	θ_i
1	0	0	0	θ_1
2	-90°	0	0	θ_2
3	0	a_2	d_3	θ_3
4	-90°	a_3	d_4	θ_4
5	-90°	0	0	θ_5
6	-90°	0 </td <td>0</td> <td>θ_6</td>	0	θ_6

Cuadro 2.1: Parámetros de Denavit-Hartenberg del PUMA 560 [14].

Los tres primeras articulaciones o elementos de un robot PUMA definen la posición del órgano terminal, mientras que las tres últimas definen la orientación como se muestra en la Figura 2.25.

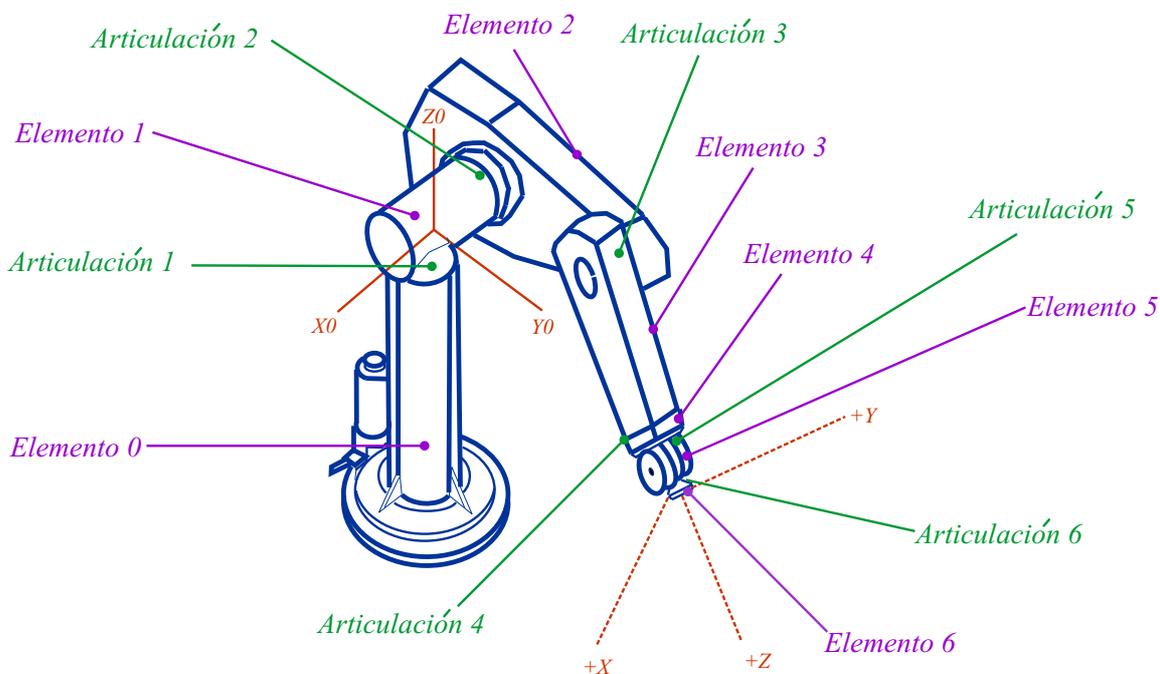


Figura 2.25: Articulaciones y elementos de un robot manipulador PUMA Unimation.

Brazo de Stanford

El brazo de Stanford es uno de los manipuladores recientes que ha sido estudiado en gran detalle, diseñado por V. Scheinman. Este manipulador cuenta con seis articulaciones, cinco de las cuales son rotacionales o de revolución y una, la tercera, es prismática.

En la Tabla 2.2 se muestran los PDH de un manipulador Stanford/JPL [15].

Articulación	d_i	a_i	α_i	θ_i
1	d_1	0	-90°	θ_1
2	d_2	0	90°	θ_2
3	d_3	0	0	-90°
4	d_4	0	-90°	θ_4
5	d_5	0	90°	θ_5
6	d_6	0	0	θ_6

Cuadro 2.2: Parámetros de Denavit-Hartenberg del Brazo de Stanford/JPL [15].

La Figura 2.26, muestra la estructura, así como la cadena cinemática del brazo de Stanford/JPL.

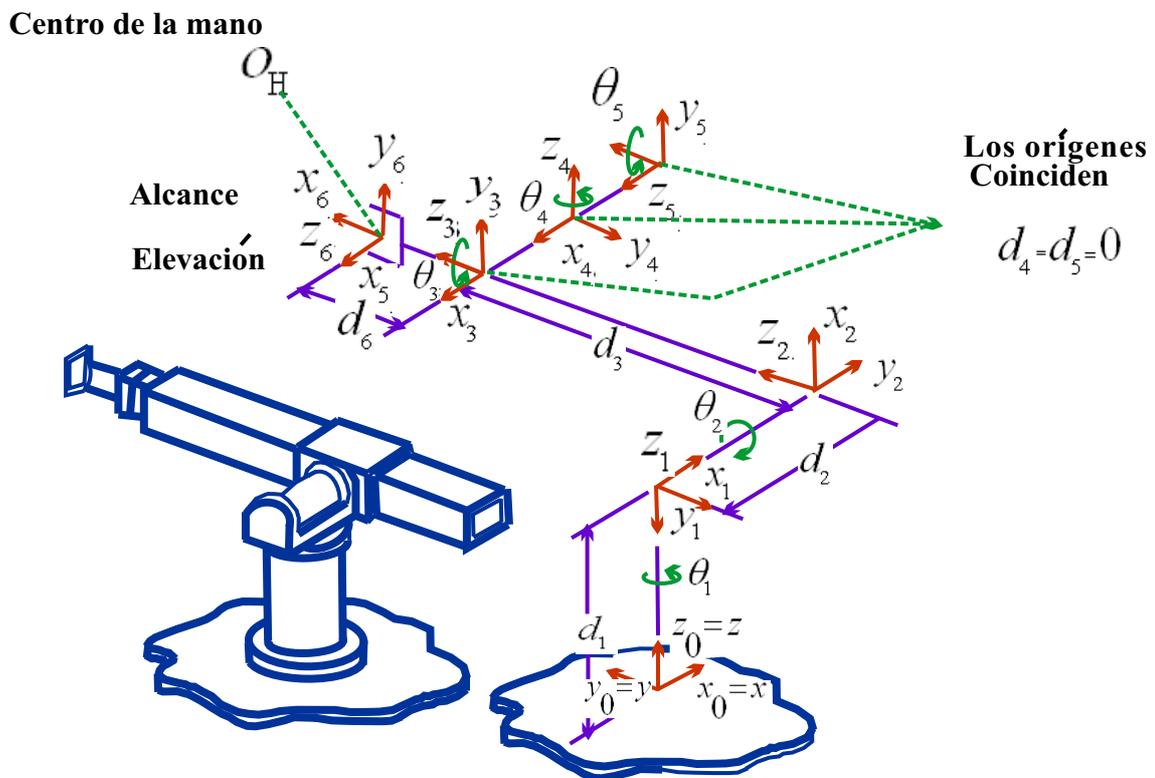


Figura 2.26: Estructura y cadena cinemática del manipulador Stanford/JPL [15].

Yasukawa Motoman L-3

Este es un manipulador industrial popular con cinco grados de libertad. La Tabla 2.3 muestra los parámetros D-H para este manipulador[15].

Articulación	d_i	$a_i - 1$	$\alpha_i - 1$	θ_i
1	0	0	0	θ_1
2	0	0	-90°	θ_2
3	0	L_2	0	θ_3
4	0	L_3	0	θ_4
5	0	0	90°	θ_5

Cuadro 2.3: Parámetros de Denavit-Hartenberg del Yasukawa Motoman L-3 [15]

En la Figura 2.27 se muestra la estructura del robot Motoman L-3 en su posición de descanso.

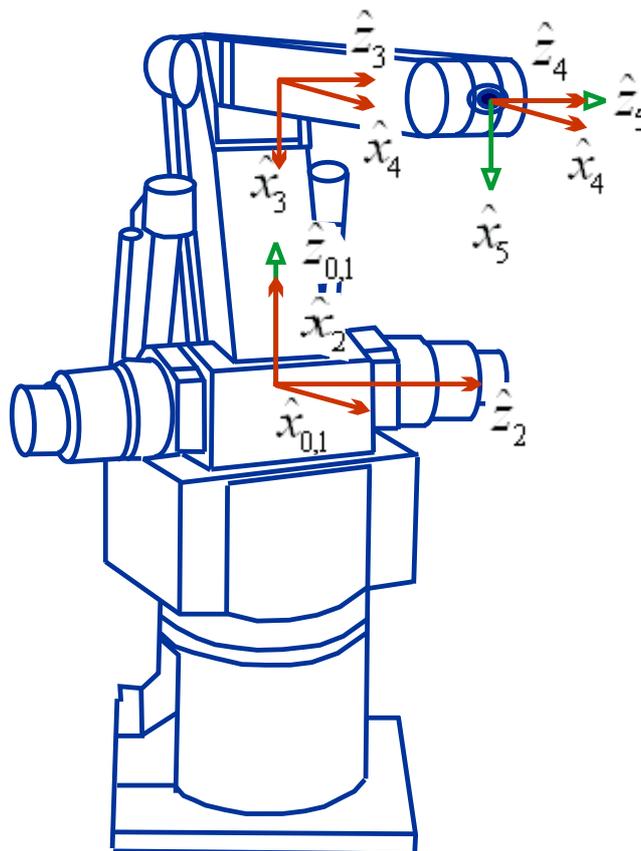


Figura 2.27: Yasukawa Motoman L-3 [15].

2.2.4. Subsistemas de un robot manipulador.

En todos los sistemas automáticos de manipulación de piezas u objetos (robots) podemos distinguir cinco subsistemas estructurales bien definidos, que son las siguientes: subsistema electromecánico, comunicación, decisión, percepción y de control

1. Subsistema electro-mecánico

Es toda la parte mecánica que la constituye, en la cual se incluyen los actuadores y el sistema de agarre o sujeción de piezas, incluyendo también el sistema de engranes para los robots que los requieren. Y la parte electrónica que comprende básicamente la Interfaz Electrónica de Potencia (IEP), dispositivos de acondicionamiento de señales, Interfases de Comunicación, etc..

Los dispositivos de transmisión de energía y actuadores proporcionan la fuerza para mover el brazo del robot. Los actuadores incluyen dispositivos hidráulicos (fluidos a presión), eléctricos (electricidad) y neumáticos (aire comprimido) correspondientes a los tres sistemas básicos de control de robot.

Actuadores neumáticos e hidráulicos

Estos actuadores obtienen su energía mediante el movimiento de fluidos, para los actuadores neumáticos el fluido es aire comprimido y para los actuadores hidráulicos el fluido suele ser aceite a presión.

Motores eléctricos

Los motores proporcionan una excelente controlabilidad requiriendo un mínimo de mantenimiento. Los motores más utilizados en robótica son los motores de Corriente Directa (CD) , los motores de fase y los motores de Corriente alterna (CA).

La potencia desarrollada por estos actuadores debe transmitirse hasta la articulación del robot. Los mecanismos de transmisión, entre los que se encuentran los sistemas de polea, engranajes y tornillos, se emplean para este propósito.

Por otro lado el sistema de coordenadas empleado para el control depende del tipo de estructura mecánica adoptada en la construcción del mismo, y también influye el tipo de movimiento o desplazamiento que deba realizar.

Cabe aclarar que el conocimiento de los materiales ha podido disponer de elementos más adecuados; los actuadores son más pequeños, potentes y precisos; los sensores son mucho más exactos; los medios de comunicación, más fiables; los métodos de fabricación, más seguros; sus estructuras, más completas, y la programación es más sencilla y eficaz.

En la electrónica del robot, la interfase electrónica de potencia (IEP) es la parte que permite acondicionar (ajustar) una señal eléctrica que proviene del subsistema de control, en una señal eléctrica con la potencia y características adecuadas para excitar al actuador.

Por lo general, la señal eléctrica cuya función es controlar al actuador, proviene de un microprocesador con características de voltaje y corriente que por sí solos no son capaces de excitar al actuador. Así que una

IEP está compuesta por circuitos electrónicos que permiten amplificar una señal pequeña y acondicionarla para cumplir los requerimientos que el actuador demanda.

La interfaz electrónica de potencia se diseña de forma tal que se pueda controlar otras características para el control del actuador como son el sentido de giro para el caso de un motor o la dirección en un pistón hidráulico.

Hay una gran variedad de circuitos electrónicos, algunos comerciales provistos en un chip, que tiene la función de acondicionar una señal con gran potencia. Entre los más importantes están del tipo de los CMOS, de los tiristores, de los triacs, los transistores de potencia, etc..

2. Subsistema de comunicación

Constituido básicamente por una computadora o microprocesador, que permite al usuario comunicarle al manipulador la tarea a realizar, y conocer la situación de este y de su entorno durante la ejecución de la misma.

Para intercambiar información de la computadora y el manipulador, se utiliza generalmente una interfase de comunicación como es el puerto de impresión o mejor aún una tarjeta de adquisición de datos (uso industrial) para el caso de una computadora personal (PC).

Actualmente el avance de la microelectrónica y por ende el abaratamiento de costos en el desarrollo de microprocesadores ha permitido que cada vez más, estos se utilicen en el control de robots industriales (manipuladores).

3. Subsistema de decisión

Es un sistema intermedio, entre la parte de comunicación y la de control, que consiste generalmente en un programa de computadora, cuya función básica es la de interpretar las instrucciones del programa para definir una tarea, y convertir éstas en consignas de movimiento de cada eslabón del manipulador.

La programación de un robot debe ser sencilla y para ello se emplean básicamente dos métodos: el aprendizaje manual, en el cual el robot repite los movimientos efectuados directamente por una mano maestra y los graba en una memoria, repitiéndolos a continuación, y programación mediante el uso de un lenguaje de alto nivel, para lo que existen actualmente numerosos lenguajes creados específicamente para el control de robot, aunque también pueden emplearse sistemas mixtos.

4. Subsistema de percepción

Conjunto de instrumentos captadores o sensores, que a través de las señales físicas generan señales eléctricas que permiten a los subsistemas de control y decisión conocer la situación instantánea del manipulador y de los objetos del ambiente durante la ejecución de una tarea o desarrollo de una trayectoria.

Los sensores también son llamados transductores debido a la capacidad de convertir una señal física en una señal eléctrica. Los sensores pueden ser propioceptivos y exteroceptivos.

Propioceptivos:

Constituye a todos aquellos instrumentos de sensibilidad que son propios del robot, es decir que obtienen señales de sí mismos, por ejemplo un potenciómetro lineal acoplado a la flecha de un motor, puede obtener señales que permitan saber la posición angular que tiene actualmente la flecha del motor, entre otros ejemplos, se encuentran: potenciómetros, tacómetros, sensores táctiles, codificadores ópticos, etc..

Exteroceptivos:

Son los sensores que obtienen señales del medio o del exterior, con respecto al robot, por ejemplo un sistema de visión artificial constituido por una cámara de TV, proporciona información del medio en el que está inmerso el manipulador.

5. Subsistema de control

Procesa la información recibida de los subsistemas de decisión (la consigna original) y de percepción (errores a corregir) y genera las señales físicas necesarias para que, después de ser amplificadas, los actuadores del manipulador ejecuten finalmente los movimientos requeridos por sus eslabones en la realización de la tarea.

El sistema de control y el lenguaje de programación conforman la sección de toma de decisiones automática, que incluye la planificación, el control de movimientos y la interpretación de los datos aportados por los sensores.

El controlador suele estar formado por una unión de suma donde se comparan las señales de entrada y salida, un dispositivo de control que determina la acción de control, los amplificadores de potencia necesarios y los dispositivos de hardware asociados para realizar la acción de control en la instalación.

2.2.5. Aplicaciones de los robots

¿Para qué son usados los robots?. Es por todos conocido que los robots están en cualquier parte. Pero, algunos son más complicados y algunos tienen usos más interesantes. Una de las áreas de mayor uso de los robots es en producción, el cual tiene que ver con cualquier producto que es producido en grandes cantidades. Carros, aviones, computadoras y miles de otros productos son creados de esta manera.

Los robots ponen juntas las partes necesarias, pintan carros con determinado color, empaacan productos y hacen otro tipo de trabajos que podrían ser aburridos cuando una persona tiene que hacerlos una y otra vez. Los robots pueden trabajar más rápido que una persona y con más precisión por lo que no es tan fácil cometer errores y más productos pueden ser construidos de esta manera[42].

Otra área en donde los robots son muy necesarios es cuando las personas tienen que trabajar en “ambientes de alto peligro” o cualquier tipo de lugar en que puede ser peligroso para los seres humanos. Esto puede significar checar las bases de una construcción donde el edificio está por derrumbarse o lugares como el océano o el ártico donde las condiciones del ambiente no son favorables. Otros ambientes peligrosos incluyen lugares con derrames químicos o lugares donde se tienen que localizar minas [20].

Una de las aplicaciones también interesantes de los robots está en el área de medicina. Como por ejemplo se ha desarrollado un robot llamado Da Vinci (ver Figura 2.28) que puede operar en pacientes[18].



Figura 2.28: Robot Da Vinci que puede operar pacientes [41].



Figura 2.29: Robot en la industria [60].

Dentro de las múltiples aplicaciones que tienen los robots, se enuncian las siguientes:

- Alimentación de máquinas herramienta.
- Pintura.
- Soldadura de puntos.
- Soldadura de arcos.
- Manejo de materiales.
- Limpieza de piezas fundidas y pulido.
- Inspección.
- Montaje.
- Robots para la enseñanza.
- Seguridad.
- Manipulación de sustancias tóxicas.
- Investigación aeroespacial.
- Robots de recreación.
- En la medicina, realizando operaciones a pacientes.

2.3. Modelo cinemático de robots manipuladores

2.3.1. Introducción

La cinemática del brazo robot trata con el estudio analítico de la geometría del movimiento de un robot con respecto a un sistema de coordenadas de referencia fija como una función del tiempo, sin considerar las fuerzas/momentos que originan dicho movimiento. Así pues, trata con la descripción analítica del desplazamiento espacial del robot como función del tiempo, en particular las relaciones entre las variables espaciales de tipo articulación y la posición y orientación del efector final del robot.

En la cinemática del robot, se plantean dos problemas fundamentales descritos a continuación:

Problema cinemático directo:

Para un manipulador determinado, dado el vector de ángulos de las articulaciones (ecuación) y los parámetros geométricos del elemento, donde (n) es el número de grados de libertad. Se desea conocer cuál es la orientación y la posición del efector final del manipulador con respecto a un sistema de coordenadas de referencia.

Problema cinemático inverso (o solución del brazo):

Dada una posición y orientación deseada del efector final del manipulador y los parámetros geométricos de los elementos con respecto a un sistema de coordenadas de referencia, se desea saber si el manipulador puede alcanzar la posición y la orientación de la mano que se desea. Y si puede, cuántas configuraciones diferentes del manipulador van a satisfacer la misma condición. Como las variables independientes en un brazo de robot son las variables de articulación y una tarea se suele dar en términos de las coordenadas de referencia, el problema cinemático inverso se utiliza de forma más frecuente.

En resumen, este análisis permite conocer las relaciones que identifican la ubicación cartesiana del órgano terminal en términos de los ángulos existentes entre los eslabones (coordenadas articulares o generalizadas) y viceversa, así mismo sus variaciones en términos del tiempo (cinemática diferencial).

El modelo cinemático de un robot puede ser:

1. Modelo cinemático de posición (MCP).
 - Modelo cinemático directo de posición (MCDP).
 - Modelo cinemático inverso de posición (MCIP).
2. Modelo cinemático de velocidad (MCV).
 - Modelo cinemático directo de velocidad (MCDV).
 - Modelo cinemático inverso de velocidad (MCIV).
3. Modelo cinemático de aceleración (MCA).
 - Modelo cinemático directo de aceleración (MCDA).
 - Modelo cinemático inverso de aceleración (MCIA).

La representación matemática que define la modelación cinemática de un robot, permite controlar el movimiento del robot dada una referencia, comunmente una trayectoria planificada como en gran diversidad de aplicaciones industriales.

La observación existente, que posiblemente logre un bajo desempeño en la realización de una tarea, es el hecho de no considerar las propiedades dinámicas del robot.

Para entender la relación de la Cinemática Directa y la Inversa se presenta a continuación la Figura 2.30.

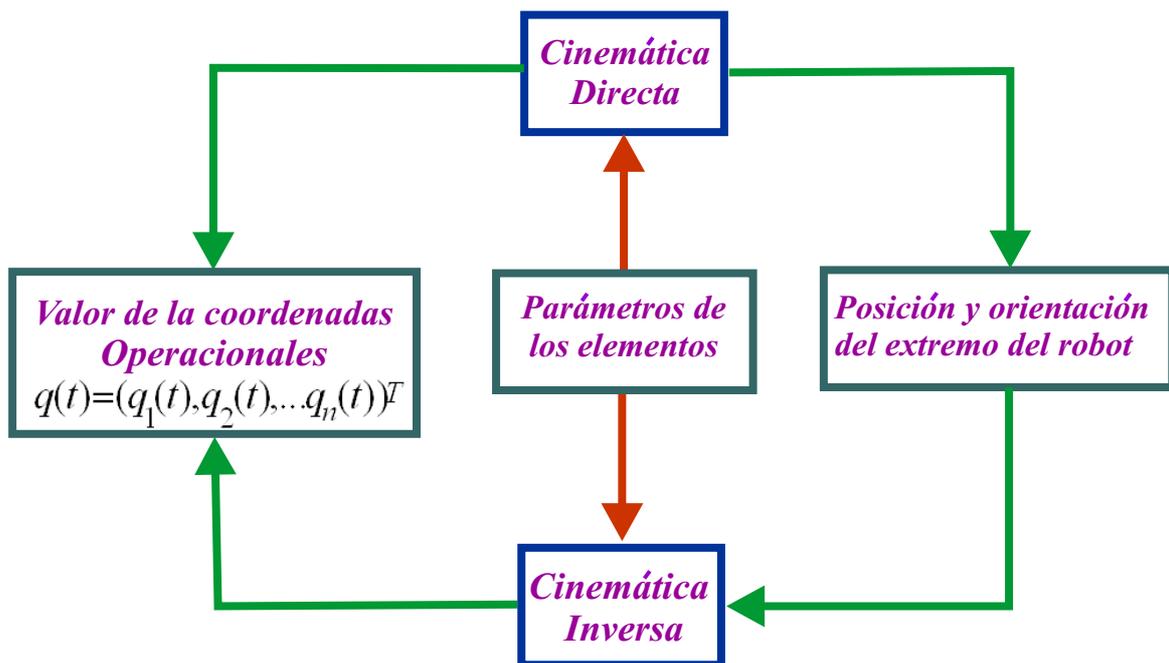


Figura 2.30: Relación entre cinemática directa e inversa

Algunas aplicaciones del modelo cinemático de un robot son:

1. Planificación de trayectorias.
2. Desarrollo de simuladores físicos.
3. Soldaduras.
4. Posicionamiento de objetos en el espacio.
5. Manufactura de circuitos impresos, entre muchos otros.

Para obtener el modelo cinemático se requiere identificar la cadena cinemática del robot, y a partir de ella situar mecánicamente los marcos ortonormales por articulación. Normalmente y dadas las especificaciones anteriores se definen los parámetros de Denavit-Hartenberg que contribuyen para conocer las matrices elementales del robot, el productos de ellas constituyen una matriz de transformación homogénea que define una matriz de rotación y un vector columna de posición. A partir de ella es posible conocer la posición y orientación del órgano terminal del robot en términos de las variables articulares, concepto asociado con el MCDP.

A partir del MCDP es posible conocer la situación inversa (MCIP) y la cinemática diferencial (MCDV, MCIV, MCDA y MCIA) como a continuación se describe de manera general.

2.3.2. Descripciones: posición, orientación y marcos

Una descripción es usada para especificar atributos de varios objetos relacionados con los sistemas manipuladores. Estos objetos son partes, herramientas o incluso el mismo manipulador.

A continuación se mostrará la descripción de la posición, orientación y de una entidad que contiene a las dos descripciones anteriores, los marcos[14].

Descripción de posición

Una vez establecido un sistema de coordenadas es posible localizar cualquier punto en ese universo utilizando un vector de posición de $\vec{V} \in \mathbb{R}^{3 \times 1}$. Como comúnmente se definen varios sistemas de coordenadas además del sistema de coordenadas del universo, los vectores deben estar marcados con información para identificar con respecto a qué sistema de coordenadas están definidos. Una forma de hacerlo es utilizar un superíndice que indique a qué sistema de referencia están asignados. Esto significa que los componentes tienen valores numéricos que indican distancias a lo largo de los ejes de A. Así, cada una de esas distancias a lo largo de un eje representan la proyección del vector sobre el eje correspondiente[14].

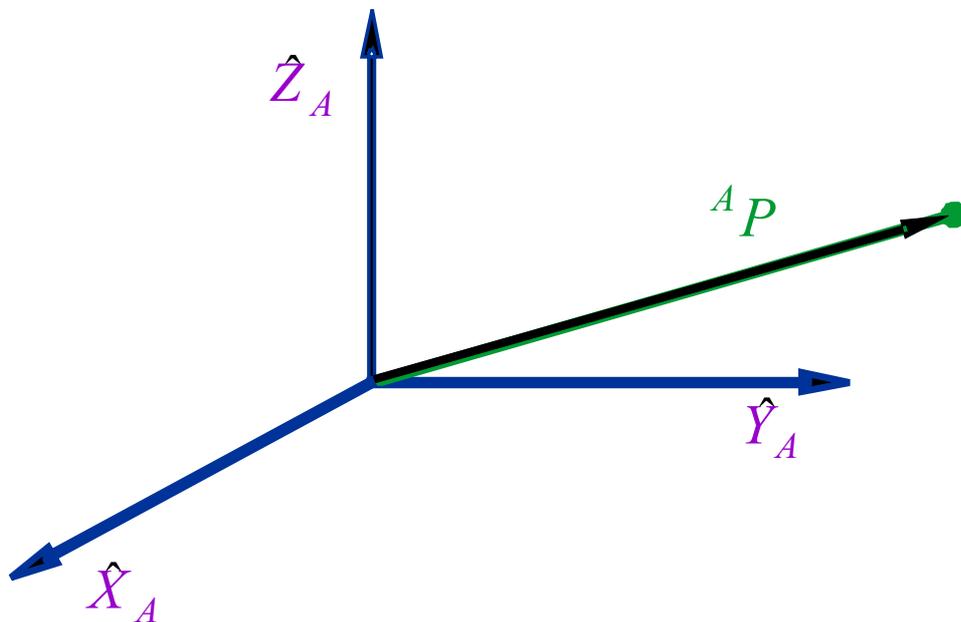


Figura 2.31: Sistema de coordenadas.

La Figura 2.31 representa a un sistema de coordenadas A, con tres vectores unitarios ortogonales. Un punto está representado con un vector y puede entenderse como una posición en el espacio. A cada elemento del vector se le asigna los sufijos x, y o z:

$${}^A P = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} \quad (2.2)$$

En resumen, la posición de un punto en el espacio se describe mediante un vector de posición (Ecuación 2.2).

Descripción de orientación

A menudo es necesario representar no sólo un punto en el espacio, sino también se hace necesario describir la orientación de un cuerpo en el espacio. Por ejemplo, si el vector ${}^A P$ en la Figura 2.32 localiza el punto entre la pinza del efector final del manipulador, la localización completa del efector final no está especificada hasta que su orientación sea también especificada. Para describir la orientación de un cuerpo, le será asignado un sistema de coordenadas y entonces se dará una descripción de este sistema de coordenadas con respecto al sistema de referencia. En la Figura 2.32, el sistema de coordenadas B ha sido asignado al cuerpo. La descripción de B con respecto a A es ahora suficiente para dar una orientación al cuerpo.

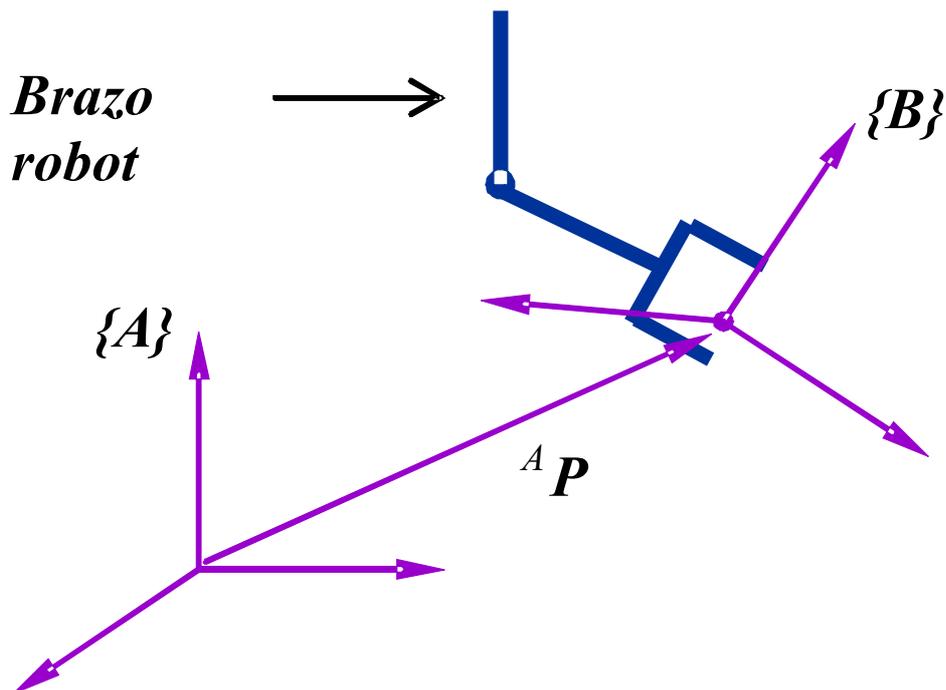


Figura 2.32: Localizando un objeto en posición y orientación.

De esta forma, la posición de los puntos se describe mediante vectores mientras que la orientación de los cuerpos se describe con la asignación de un sistema de coordenadas. Una forma de describir al sistema de coordenadas asignado, B, es escribiendo los vectores unitarios de sus tres ejes principales en términos del sistema de coordenadas A.

Los vectores unitarios que dan las direcciones principales del sistema de coordenadas B se denotan como \hat{X}_B , \hat{Y}_B y \hat{Z}_B . Cuando se escriben en términos del sistema de coordenadas A se denotan como ${}^A \hat{X}_B$, ${}^A \hat{Y}_B$ y ${}^A \hat{Z}_B$. Puede ser conveniente colocar estos tres vectores juntos como columnas de una matriz de 3×3 en el orden ${}^A \hat{X}_B$,

${}^A\hat{Y}_B$ y ${}^A\hat{Z}_B$. Llamaremos a esta matriz una matriz de rotación, y como esta matriz en particular describe a B con respecto de A, se denota ${}^A_B R$ (ver Ecuación 2.3)[14].

$${}^A_B R = [{}^A\hat{X}_B \quad {}^A\hat{Y}_B \quad {}^A\hat{Z}_B] = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{34} \end{bmatrix} \quad (2.3)$$

En resumen, un juego de tres vectores pueden ser usados para especificar una orientación. Así, para representar la posición de un punto se usa un vector, mientras que para representar la orientación se usa una matriz.

Podemos reemplazar los escalares t_{ij} con expresiones notando que los componentes de cualquier vector son simples proyecciones de ese vector sobre las direcciones unitarias de su marco de referencia. De esta forma, cada componente de ${}^A_B R$ puede ser escrito como el producto punto de un par de vectores unitarios:

$${}^A_B R = [{}^A\hat{X}_B \quad {}^A\hat{Y}_B \quad {}^A\hat{Z}_B] = \begin{bmatrix} \hat{X}_B \cdot \hat{X}_A & \hat{Y}_B \cdot \hat{X}_A & \hat{Z}_B \cdot \hat{X}_A \\ \hat{X}_B \cdot \hat{Y}_A & \hat{Y}_B \cdot \hat{Y}_A & \hat{Z}_B \cdot \hat{Y}_A \\ \hat{X}_B \cdot \hat{Z}_A & \hat{Y}_B \cdot \hat{Z}_A & \hat{Z}_B \cdot \hat{Z}_A \end{bmatrix} \quad (2.4)$$

Como el producto punto de dos vectores unitarios nos lleva al coseno del ángulo entre ellos, es claro el por qué los componentes de una matriz de rotación sean algunas veces llamados cosenos directores.

Descripción de un marco

La situación de determinar la posición y orientación de un objeto resulta tan común en robótica que se ha definido una entidad llamada marco, que es un conjunto de cuatro vectores que dan información sobre la posición y orientación. Por ejemplo, en la Figura 2.32 un vector indica la posición del efector final y otros tres describen su orientación. Por lo tanto, la descripción de un marco está dada por un vector de posición y una matriz de rotación. Nótese que un marco es un sistema de coordenadas, donde además de la orientación contiene un vector de posición que localiza su origen con relación a otro marco.

En la Figura 2.32 se introdujo la representación gráfica de un marco. Un marco se representa con tres flechas que representan los vectores unitarios que definen los ejes principales del marco. La otra flecha representa un vector que va desde un origen a otro. Este vector representa la posición del origen al final de la flecha en términos del marco al inicio de la flecha. La dirección de esta flecha nos indica, por ejemplo, en la Figura 2.32, que el marco B es relativo a A y no viceversa.

En conclusión, un marco puede ser usado como la descripción de un sistema de coordenadas en relación con otro. Un marco representa tanto la posición como la orientación por lo que es una generalización de ambas ideas[14].

2.3.3. Cadena cinemática

Cadena cinemática:

Conjunto de elementos mecánicos que soportan la herramienta útil del robot (base, armadura, muñeca, etc.).

La cadena cinemática derivada del mecanismo de eslabones articulados se muestra en la representación genérica como se ve en la Figura 2.33. Y se define de manera explícita la cadena cinemática de un robot manipulador en la Figura 2.34[10][20].

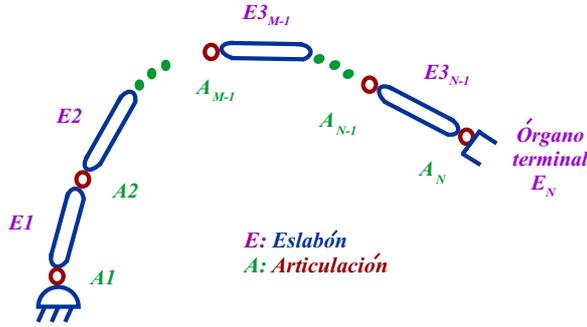


Figura 2.33: Representación genérica de la cadena cinemática de un mecanismo de eslabones articulados [10][20].

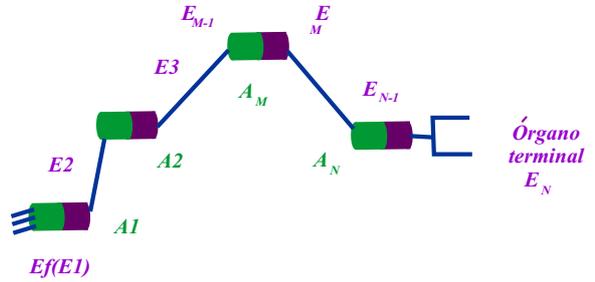


Figura 2.34: Forma explícita de definir la cadena cinemática de un robot manipulador [10][20].

2.3.4. Marcos ortonormales

Existe una regla clásica para definir los marcos ortonormales para cada articulación del robot, estas son:

1. El origen 0_{i+1} es el punto de intersección del eje de la articulación A y A_{i+1} con la normal común a los ejes de las articulaciones A y A_{i+1} . Si los ejes de las articulaciones son paralelos o están alineadas, la perpendicular común se selecciona arbitrariamente. Es recomendable seguir un criterio de simplicidad.
2. El vector \hat{x}_{i+1} se define sobre la perpendicular común a los ejes de las articulaciones A y A_{i+1} , orientado del eje de A_i al de A_{i+1} . Si los ejes de las dos articulaciones se interceptan o están alineadas, la orientación de \hat{x}_{i+1} es arbitraria.
3. El vector \hat{z}_{i+1} se define de manera tal, que coincida con el eje de la articulación A_{i+1} orientado arbitrariamente.

\hat{y}_{i+1} se define de tal manera, que se completa un marco de mano derecha, es decir: $\hat{y}_{i+1} = \hat{z}_{i+1} + \hat{x}_{i+1}$ [19].

2.3.5. Parámetros Denavit-Hartenberg.

Una vez asignado un marco de referencia a cada eslabón, es necesario definir la posición y orientación de cada marco respecto al precedente. Esto se puede hacer aplicando cuatro parámetros geométricos asociados con cada elemento ya que nos permiten describir completamente cualquier articulación prismática o de revolución.

La regla que permite identificar los Parámetros de Denavit Hertenberg (PDH) para cada grado de libertad de un robot es la siguiente [1][17]:

1. θ_i : es el ángulo de la articulación del eje \hat{x} al eje \hat{x}_{i+1} , medido respecto del eje \hat{z}_i . Utilizando la regla de la mano derecha.
2. d_i : es la distancia entre \hat{x} y \hat{x}_{i+1} , medida sobre \hat{z}_i .
3. a_i : es la longitud de la perpendicular común a los ejes de las articulaciones A_i y A_{i+1} .
4. α_i : es el ángulo de separación del eje \hat{z}_i al eje \hat{z}_{i+1} , medido respecto a \hat{x}_{i+1} . Utilizando la regla de la mano derecha[1].

2.3.6. Matriz de transformación homogénea

La Matriz de transformación homogénea (MTH) es una matriz de 4 x 4 que transforma un vector de posición expresado en coordenadas homogéneas desde un sistema de coordenadas hasta otro sistema de coordenadas.

Dada una matriz de transformación homogénea generalizada para cualquier robot manipulador (RM), se describen a las matrices elementales del RM en términos de los PDH [1][2][10][15].

La matriz de transformación homogénea (MTH) es la Ecuación 2.5:

$${}^i_{i-1}T = \begin{bmatrix} C\theta_i & -S\theta_i S\alpha_i & S\theta_i S\alpha_i & a_i C\theta_i \\ S\theta_i & C\theta_i C\alpha_i & -C\theta_i S\alpha_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

2.3.7. Matrices elementales de un robot

Cada matriz elemental contiene información de traslación y rotación de la cadena cinemática del RM.

Del producto de las matrices elementales se obtiene la MTH:

$${}^1_2T \cdot {}^2_3T \cdot {}^3_4T = {}^1_4T \quad (2.6)$$

Donde:

$${}^1_4T = \begin{bmatrix} R_{3 \times 3} & P_{3 \times 1} \\ f_{1 \times 3} & 1 \end{bmatrix}$$

$R_{3 \times 3}$: matriz de rotación.

$P_{3 \times 1}$: vector de posición.

$f_{1 \times 3}$: transformación de perspectiva.

1: escalado.

Así que con la aplicación de identidades trigonométricas para la suma de ángulos se obtiene la siguiente Ecuación 2.7:

$${}^1_4T = \begin{bmatrix} C_1 C_{23} & -C_1 S_{23} & -S_1 & L_3 C_1 C_{23} + L_2 C_1 C_2 \\ S_1 C_{23} & -S_1 S_{23} & C_1 & L_3 S_1 C_{23} + L_2 S_1 C_2 \\ -S_{23} & -C_{23} & 0 & -L_3 S_{23} - L_2 S_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.7)$$

Donde:

$$\begin{aligned}\cos(\theta_1) &= C_1 \\ \cos(\theta_2 + \theta_3) &= C_{23} \\ \text{sen}(\theta_1) &= S_1 \\ \text{sen}(\theta_2 + \theta_3) &= S_{23}\end{aligned}$$

Por lo tanto la MTH permite conocer las coordenadas de posición y orientación del extremo final del RM (órgano terminal ficticio) respecto a la base.

2.3.8. Modelo cinemático directo de posición (MCDP)

El MCDP es el conjunto de ecuaciones que proporcionan las coordenadas operacionales a partir de los ángulos de las articulaciones y se describe a partir de la siguiente ecuación:

$$x = f(q) \tag{2.8}$$

Como las ecuaciones que definen a las coordenadas operacionales (cartesianas para posición y ángulos de Bryant o Euler para orientación) en función de las coordenadas generalizadas o variables articulares (ángulos entre eslabones)[1][2][10][15].

El MCDP permite conocer la posición instantánea del órgano terminal respecto a la base fija del RM. El MCDP está definido para la posición por el vector $P_{3 \times 1}$ y para la orientación por la matriz $R_{3 \times 3}$. Por tal motivo se define lo siguiente:

$$\begin{aligned}x &= Px \\ y &= Py \\ z &= Pz\end{aligned} \tag{2.9}$$

Este modelo se desarrolla matemáticamente en el capítulo 5 de ésta tesis.

2.3.9. Modelo cinemático inverso de posición (MCIP)

El MCIP es el conjunto de ecuaciones que permiten conocer las variables articulares o coordenadas generalizadas (ángulos entre eslabones) en términos de las coordenadas operacionales de posicionamiento (x, y, z)[1][2][10][15].

La representación matemática formal es la siguiente:

$$q = f^{-1}(x) \tag{2.10}$$

es decir:

$$\begin{bmatrix} \theta_1 \\ \theta_2 \\ M \\ \theta_n \end{bmatrix} = f^{-1}(x, y, z) = \begin{bmatrix} f_1^{-1}(x, y, z) \\ f_2^{-1}(x, y, z) \\ M \\ f_n^{-1}(x, y, z) \end{bmatrix} \quad (2.11)$$

El MCIP permite controlar cinemáticamente a un RM, su obtención parte del conocimiento de las matrices elementales, la MTH y la cadena cinemática. Así tenemos lo siguiente:

$$\begin{aligned} {}^1_4T &= {}^1_2T \cdot {}^2_3T \cdot {}^3_4T \\ {}^1_2T^{-1} &= {}^1_4T \cdot {}^2_3T \cdot {}^3_4T \end{aligned} \quad (2.12)$$

Donde:

1_4T : MTH (matriz de transformación homogénea).

2.3.10. Modelo cinemático directo de velocidad (MCDV)

La ecuación del MCDV permite conocer la velocidad instantánea que el efector final adquiere durante la ejecución de una tarea. La velocidad que adquiere el efector final basándose en leyes convencionales (física) de la cinemática de un móvil (cuerpo), es:

$$V = \frac{d}{dt}x = \frac{d}{dt}f(q) \quad (2.13)$$

por lo tanto dado el MCDP (ecuación 2.8) obtenemos MCDV [1][2][12][20].

Sin embargo el MCDV puede ser representado por la siguiente expresión:

$$\dot{x} = J\dot{q} \quad (2.14)$$

Donde:

\dot{x} : vector de velocidad del órgano terminal del robot

\dot{q} : vector de velocidad de las articulaciones del robot

J : matriz jacobiana del robot manipulador, cuyos elementos son J_{ij}

2.3.11. Modelo cinemático inverso de velocidad (MCIV)

Las ecuaciones del MCIV de un RM, permiten controlar la velocidad del mismo. La expresión que permite encontrar el MCIV, es la siguiente:

$$\dot{x} = \frac{d}{dt}f(q) = J\dot{q} \quad (2.15)$$

Despejando a \dot{q} del MCDV, tenemos:

$$\dot{x} = J\dot{q}$$

$$J^{-1}\dot{x} = J^{-1}J\dot{q}$$

$$\dot{q} = J^{-1}\dot{x} \quad (2.16)$$

donde:

$$J^{-1} = \frac{adj J}{\det J}$$

para evitar singularidad, es decir indeterminación:

$$\det J \neq 0$$

de este modo es posible identificar las singularidades del RM, es decir las configuraciones inadmisibles, en otras palabras los arreglos de $(\theta_1, \theta_2, \theta_3)$ deseados que no puede adquirir el RM.

En el capítulo 5 de ésta tesis se encuentra el desarrollo matemático más detallado.

2.3.12. Modelo cinemático directo de aceleración (MCDA)

El MCDA se obtiene de la derivada con respecto al tiempo del MCDV, de la manera siguiente [1][2][10][15]:

$$\dot{x} = J\dot{q}$$

$$\frac{d}{dt}[\dot{x}] = \frac{d}{dt}[J\dot{q}]$$

$$\ddot{x} = J\ddot{q} + \dot{J}\dot{q} \quad (2.17)$$

Donde:

\ddot{x} : corresponde al vector de aceleración del órgano terminal del robot.

\ddot{J} : corresponde al vector de aceleración de las articulaciones.

\dot{J} : representa la primera derivada respecto al tiempo de la matriz Jacobiana.

2.3.13. Modelo cinemático inverso de aceleración (MCIA)

El MCIA se obtiene despejando a $J\ddot{q}$ del MCDA, de la manera siguiente[1][2][12][20]:

$$\ddot{x} = J\ddot{q} + \dot{J}\dot{q}$$

$$J\ddot{q} = \ddot{x} - \dot{J}\dot{q}$$

$$\ddot{q} = J^{-1}[\ddot{x} - \dot{J}\dot{q}] \quad (2.18)$$

2.4. Modelo dinámico de robots manipuladores

2.4.1. Introducción

La dinámica del robot trata con las formulaciones matemáticas de las ecuaciones de movimiento del brazo. Las ecuaciones de movimiento de un manipulador son un conjunto de ecuaciones matemáticas que describen su conducta dinámica. Tales ecuaciones son útiles para la simulación en computadora del movimiento del robot, el diseño de ecuaciones de control apropiadas para el robot y la evaluación del diseño y estructura del brazo. El objetivo del control de un manipulador basado en computadora es mantener la respuesta dinámica del mismo de acuerdo con algún rendimiento del sistema previamente especificado y objetivos deseados. En general, el rendimiento dinámico de un manipulador depende directamente de la eficacia de los algoritmos de control y de su modelo dinámico. El problema de control consiste en obtener modelos dinámicos del brazo del robot físico y a continuación especificar leyes o estrategias de control correspondientes para conseguir la respuesta y rendimiento del sistema deseado.

El modelo dinámico de un robot se puede obtener a partir de leyes físicas conocidas tales como las leyes de la mecánica newtoniana y lagrangiana. Este conduce al desarrollo de las ecuaciones de movimiento dinámico para las diversas articulaciones del manipulador en términos de los parámetros geométricos e inerciales de los elementos. Métodos convencionales como las formulaciones de Lagrange - Euler (L-E) y Newton - Euler (N-E) se pueden aplicar entonces sistemáticamente para desarrollar las ecuaciones de movimiento del robot.

La obtención del modelo dinámico de un manipulador basado en la formulación de L-E es simple y sistemática.

Suponiendo el movimiento del cuerpo rígido, las ecuaciones de movimiento resultante, excluyendo la dinámica de los dispositivos de control electrónico, huelgo y el rozamiento de los engranajes, son un conjunto de ecuaciones diferenciales no lineales acopladas de segundo orden.

En ambos casos se puede necesitar calcular sus coeficientes dinámicos. Desgraciadamente, el cálculo de estos coeficientes requiere una relativa cantidad de operaciones aritméticas. Así las ecuaciones de L-E son muy difíciles de utilizar con fines de control en tiempo real a menos que se simplifiquen.

La eficacia computacional se consigue a partir de una formulación compacta utilizando matrices de transformación de Euler (o matrices de rotación) y vectores de posición relativos entre articulaciones.[2]

2.4.2. Conceptos preliminares de la dinámica de cuerpos rígidos

■ Energía cinética

Es una forma de energía debida al movimiento de los cuerpos. Equivale al trabajo que es necesario realizar para que el cuerpo pase del estado de reposo ($v = 0$) al estado de desplazamiento con una velocidad v . En mecánica clásica un cuerpo de masa m desplazándose a una velocidad v posee una energía cinética.

- **Energía potencial**

Es una función de las coordenadas tal que la diferencia entre sus valores en las posiciones inicial y final es igual al trabajo efectuado sobre la partícula para moverla de su posición inicial a la final.

- **Fuerza de Coriolis**

Es la fuerza producida por la rotación de la tierra en el espacio, que tiende a desviar la trayectoria de los objetos que se desplazan sobre la superficie terrestre; a la derecha en el hemisferio norte y a la izquierda, en el sur.

Esta fuerza no sólo aparece durante la rotación de la tierra sino, en general, para cualquier objeto con masa que se desplaza a una determinada velocidad sobre otro objeto en rotación.

- **Fuerza de centrípeta**

Fuerza dirigida hacia un centro, que hace que un objeto se desplace en una trayectoria circular.

- **Fricción**

Fuerza que resiste el movimiento encontrada entre dos cuerpos, bajo la acción de una fuerza externa en la cuál un cuerpo tiende a moverse sobre la superficie del otro.

La fricción es una característica ineludible en los sistemas que se encuentran en movimiento, habitualmente es el efecto que lleva al reposo. En un robot, la fricción se presenta en las articulaciones y se refleja como una fuerza opuesta al movimiento, este fenómeno físico depende de factores como la temperatura, el tipo de materiales utilizados cuyas superficies están en contacto, etc. .

- **Seca**

Si se trata de mover un bloque sobre una superficie rugosa por aplicación de una fuerza, éste no se moverá si la fuerza no sobrepasa cierto valor. Esto quiere decir que si el bloque no se mueve, la superficie debe ejercer una fuerza igual a la aplicada.

- **Viscosa**

La fricción viscosa presenta una relación lineal entre la fuerza aplicada y la velocidad del movimiento: $f(t) = Bv(t)$, donde Bv es la constante de viscosidad.

2.4.3. Formulación Euler-Lagrange

Las ecuaciones de movimiento L-E proporcionan ecuaciones de estado explícitas para la dinámica del robot y se pueden utilizar para analizar y diseñar estrategias de control avanzadas en el espacio de las variables de articulación. En una menor medida se está utilizando para resolver el problema dinámico directo, esto es, dadas las fuerza/pares deseadas, se utilizan las ecuaciones dinámicas para resolver las aceleraciones de las articulaciones, que se integran a continuación para obtener las coordenadas y velocidades generalizadas; o para el problema dinámico inverso, esto es, dadas las coordenadas generalizadas deseadas y sus primeras dos derivadas respecto del tiempo, se calculan las fuerzas/pares generalizados. [2]

La formulación de Lagrange-Euler, está definida por la siguiente expresión:

$$\tau = \frac{d}{dt} \frac{\partial}{\partial \dot{q}} L - \frac{\partial}{\partial q} L \tag{2.19}$$

Donde:

L : lagrangiano.

$$L = \sum_{i=1}^n (K_i - P_i) \quad (2.20)$$

donde:

τ : par torsor para cada eslabón del manipulador.

K : energía cinética (debida a la velocidad):

$$K = \frac{1}{2} m v_i^2 \quad (2.21)$$

P : energía potencial (debido a la altura):

$$P_i = m_i g h_i \quad (2.22)$$

Por lo tanto la ecuación 2.20, se puede representar así:

$$L = \sum_{i=1}^n \left\{ \frac{1}{2} m_i V_i^2 - m_i g h_i \right\} \quad (2.23)$$

n : número de grados de libertad del robot.

La representación matricial del modelo dinámico de un robot manipulador considerando fricción viscosa y seca, cuyos elementos se obtienen a partir de la formulación de Lagrange-Euler (ecuación 2.13), se basa en la siguiente expresión:

$$\tau = H(q)\ddot{q} + c(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) \quad (2.24)$$

donde:

τ : pares o fuerzas generalizadas.

$H(q)$: matriz de inercia.

$c(q, \dot{q})$: matriz de fuerzas de Coriolis.

$G(q)$: vector de fuerzas gravitatorias.

$F(\dot{q})$: vector de fuerzas de fricción viscosa y seca.

q : articulación del robot.

\dot{q} : vector de velocidad de las articulaciones del robot.

\ddot{q} : vector de aceleración articular.

$$F(\dot{q}) = Fv(\dot{q}) + Fs(\dot{q}) \quad (2.25)$$

donde:

$Fv(\dot{q})$: Vector de fricción viscosa.

$Fs(\dot{q})$: Vector de fricción seca.

$$F(\dot{q}) = \left\{ \begin{array}{l} \sum_{i=1}^n (b_i \dot{\theta}_i) \quad Fv \\ \sum_{i=1}^n (k_i Sgn(\dot{\theta}_i) \approx k_i Tanh(\beta_i \dot{\theta}_i)_{\beta \rightarrow \infty}) \quad Fs \end{array} \right\} \quad (2.26)$$

2.4.4. Propiedades dinámicas de un robot manipulador.

Estabilidad

1. Hace referencia a la ausencia de oscilación en el movimiento de la herramienta.
2. Las oscilaciones son perjudiciales porque:
 - Producen deterioros adicionales sobre las partes mecánicas e hidráulicas del brazo.
 - Puede que, en repeticiones sucesivas del mismo movimiento, la herramienta siga diferentes caminos, obligando a una mayor distancia entre la trayectoria deseada y los objetos del entorno.
 - Puede incrementar el tiempo necesario para que la herramienta se posicione en una posición precisa.
 - Pueden causar una colisión con algún elemento del entorno si se sobrepasa la posición de parada.
3. Tipos de oscilaciones:
 - Amortiguadas.
 - No amortiguadas[52].

Resolución espacial

1. Capacidad del sistema para regresar la punta de la herramienta a un punto predeterminado del espacio.
2. El grado de precisión es la diferencia entre la posición alcanzada y la originalmente especificada.
3. Factores que influyen en la precisión:

- Impresión de los componentes mecánicos.
- Resolución de los componentes de control.
- La posición arbitraria fijada como objetivo[52].

Repetibilidad

1. Capacidad del robot para situarse de nuevo sobre el mismo punto que había sido indicado previamente[52].
2. Viene representada por la superficie esférica que encierra los puntos de varianza del extremo del robot después de múltiples órdenes de posicionamiento, desde el mismo punto inicial al mismo punto final de un programa en idénticas condiciones de carga[52].

Las propiedades del modelo dinámico son:

1. Simetría de la matriz de inercia.

$$H(q) = H(q)^T \quad (2.27)$$

2. La matriz de inercia es definida positiva.

$$\det \{H(q)\} > 0 \quad (2.28)$$

3. Matriz antisimétrica.

$$y^T (\dot{H}(q) - 2C(q, \dot{q})) y \cong 0 \quad (2.29)$$

2.5. Conclusiones

En la actualidad los términos robot y robótica, son tan controvertidas debido a lo que representa en muchas áreas; es por ello que no se puede determinar aún una clasificación ya que los significados que se le dan son distintos y variados, como se pudo observar en este capítulo se muestra una clasificación más general tratando de abarcar las más importantes.

Ya que se han definido los conceptos básicos de robot así como su clasificación de igual manera las formulas básicas de la modelación cinemática y dinámica de los robots manipuladores, el lector puede comprender el desarrollo de ésta tesis y el desarrollo matemático que se presenta en capítulos posteriores.

Para continuar con el desarrollo de este trabajo y para una mejor comprensión del lector se presenta la modelación analítica de la geometría de un robot, así como la descripción analítica del desplazamiento espacial del mismo, lo que se puede generalizar como cinemática, de igual forma se presenta la dinámica del robot que son las formulaciones matemáticas del movimiento del brazo en lo que intervienen leyes físicas como mecánica Newtoniana y Lagrangiana.

Capítulo 3

Generalidades de control de robots manipuladores

3.1. Introducción

Los sistemas de control para robots manipuladores, requieren del desarrollo de modelos matemáticos obtenidos a través de las leyes físicas que rigen la dinámica del robot. Estos modelos definen y predicen el comportamiento presente y futuro del mismo, sin contar con su presencia física. Un robot industrial, requiere de ciertos elementos sensoriales que le permitan actuar y reaccionar para alcanzar una determinada posición, es decir, el robot reacciona ante la presencia de algún obstáculo y corrige su trayectoria. Estas acciones y reacciones del robot, resultan del diseño de sistemas de control capaces de compensar cualquier perturbación o variación en los parámetros del robot.

Para los constructores y diseñadores de robots una de las áreas de mayor interés es el diseño de controladores para robots, por lo tanto el estudio del control es de gran importancia para los grandes retos teóricos que ofrece la investigación ya que ésta es indispensable en aplicaciones específicas que pueden ser llevadas a cabo mediante los robots comerciales que se estén utilizando en la actualidad.

En este capítulo se tratan las generalidades acerca de algunos métodos y conceptos básicos de control. Los controles básicos que se abordan son: el control Proporcional Derivativo (PD), control Proporcional Integral Derivativo (PID), control de modos deslizantes de segundo orden (PID no lineal) y el control Adaptable Pasivo Robusto. Estos controles brindan a los robots, estabilidad en sus movimientos y de igual forma precisión. Por lo tanto la revisión y estudio que se realiza en este capítulo nos sirve como base para una mejor comprensión de las pruebas que se realizan en el Capítulo 5.

3.2. Conceptos preliminares de control

El controlador

Recibe este nombre el dispositivo que se encarga de regular el movimiento de los elementos del manipulador y todo tipo de acciones, cálculos y procesamiento de información, que se realiza. La complejidad del control varía según los parámetros que se gobiernan, pudiendo existir las siguientes categorías[53]:

- Controlador de posición. Sólo interviene en el control de la posición del elemento terminal. Puede actuar en modo punto a punto, o bien, en modo continuo, en cuyo caso recibe el nombre de control continuo de trayectoria.
- Control cinemático. Cuando además de la posición se regula la velocidad.

- Control dinámico. Se tienen en cuenta, también, las propiedades dinámicas del manipulador, motores y elementos asociados.
- Control adaptativo. Además de lo indicado en los anteriores controles, también se considera la variación de las características del manipulador al variar la posición.

Sin embargo la metodología de estudio de diseño de los sistemas de control puede resumirse a través de las siguientes etapas [61]:

1. Familiarización con el sistema físico a controlarse: deben considerarse las variables físicas del sistema cuyo comportamiento se desea controlar, como lo son velocidad, posición y aceleración, tales variables son llamadas salidas del sistema. Aunque también se debe tomar en cuenta las entradas de sistema que en este caso de estudio es la fuerza de par o Torque.
2. Modelado: modelo matemático se obtiene a través de dos técnicas la analítica y la experimental; la primera es basada en ecuaciones de la física que rige el comportamiento del sistema y la segunda requiere de datos experimentales del sistema.
3. Especificaciones de control: se especifican las características que se desean para el sistema. Así se puede apreciar que el control es un tema sobresaliente de la intersección que se da entre la robótica y la ingeniería eléctrica y en especial el el área del control automático.
4. Controlador.

Refiriéndose a otro aspecto, el control puede llevarse a cabo en lazo abierto o en lazo cerrado. En el caso del control en lazo abierto, se produce una señal de consigna que determina el movimiento, pero no se analiza si se ha realizado con exactitud o se ha producido un error, al efectuarse en la realidad. El control en lazo abierto es típico en los motores paso a paso (PAP), en los cuales las señales que generan un paso del mismo, dan lugar al giro de un determinado ángulo del eje, es decir, son motores con movimiento cuantificado. Existen sistemas con salidas temporizadas o movimientos predeterminados, que actúan en lazo abierto[53].

El control en lazo abierto tiene muchas causas de error (inercia, interferencias, fricciones, desplazamientos, etc.), y si bien es muy simple y económico, no se admite en las aplicaciones industriales, donde es fundamental la exactitud en la repetibilidad de los movimientos. Sin embargo, en robots dedicados a la enseñanza y el entrenamiento, este tipo de control está muy extendido.

La mayoría de los sistemas de robots industriales poseen un control en lazo cerrado, con realimentación. Este control hace uso de un transductor o sensor de la posición real de la articulación o del elemento terminal, cuya información se compara con el valor de la señal de mando o consigna, que indica la posición deseada. El error entre estas dos magnitudes, se trata de diversas formas para obtener una señal final, que aplicada a los elementos motrices, varíe la posición real hasta hacerla coincidir con la deseada. En la siguiente Figura 3.1 se muestra un esquema por bloques de un controlador en bucle cerrado, en el que la señal que actúa sobre el motor es proporcional a la señal de error (posición deseada menos posición real), de forma que la relación entre la salida del controlador y la señal de error es constante.

Además del controlador de tipo proporcional, hay otros como el “integral”, cuya salida varía proporcionalmente a la señal de error y el “derivativo”, en el que la salida varía proporcionalmente con la velocidad de variación de la señal de error. La combinación de los tres tipos de controladores descritos, dan lugar al que se denomina PID (Proporcional - Integral - Derivativo). La siguiente Figura (3.2) presenta el esquema general de un controlador PID, con los circuitos característicos implementados a base de Amplificadores Operacionales[53].

Los sensores empleados para la determinación de la posición de los ejes de los motores motrices, pueden ser de carácter analógico o digital, como discos de plástico transparentes con rayas negras, que al girar a través de detectores ópticos, cortan el haz de luz entre emisor y detector y generan una serie de impulsos eléctricos que

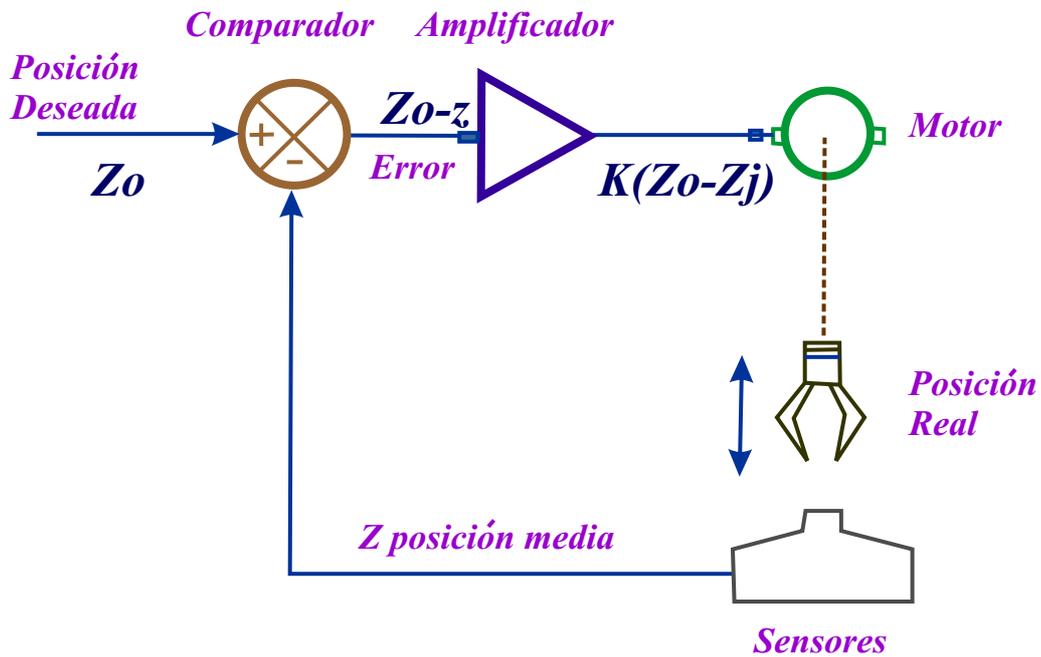


Figura 3.1: Un controlador de tipo proporcional, en el que la señal aplicada al motor es proporcional a la señal de error[53].

sirven para calcular el ángulo desplazado[53].

Los modernos controladores de robots son ordenadores, en los que el programa correspondiente se encarga de calcular las señales aplicadas a los actuadores, tras el procesado de la señal de consigna y la que procede de los transductores de posición[53].

3.3. Tipos de control

Para el control de Robots existen cuatro acciones básicas de control que se utilizan por separado, o en combinación, para proporcionar los seis tipos más comunes de controladores. Las acciones son: “todo o nada”, control proporcional, control derivativo y control integral. Los siete tipos de control son:

- “Todo o nada”.
- Proporcional.
- Integral.
- Proporcional más integral (PI).
- Proporcional más derivativo (PD).
- Proporcional integral más derivativo (PID).
- Adaptable.

Cada uno de estos tipos de controladores es el mejor para determinadas situaciones [2].

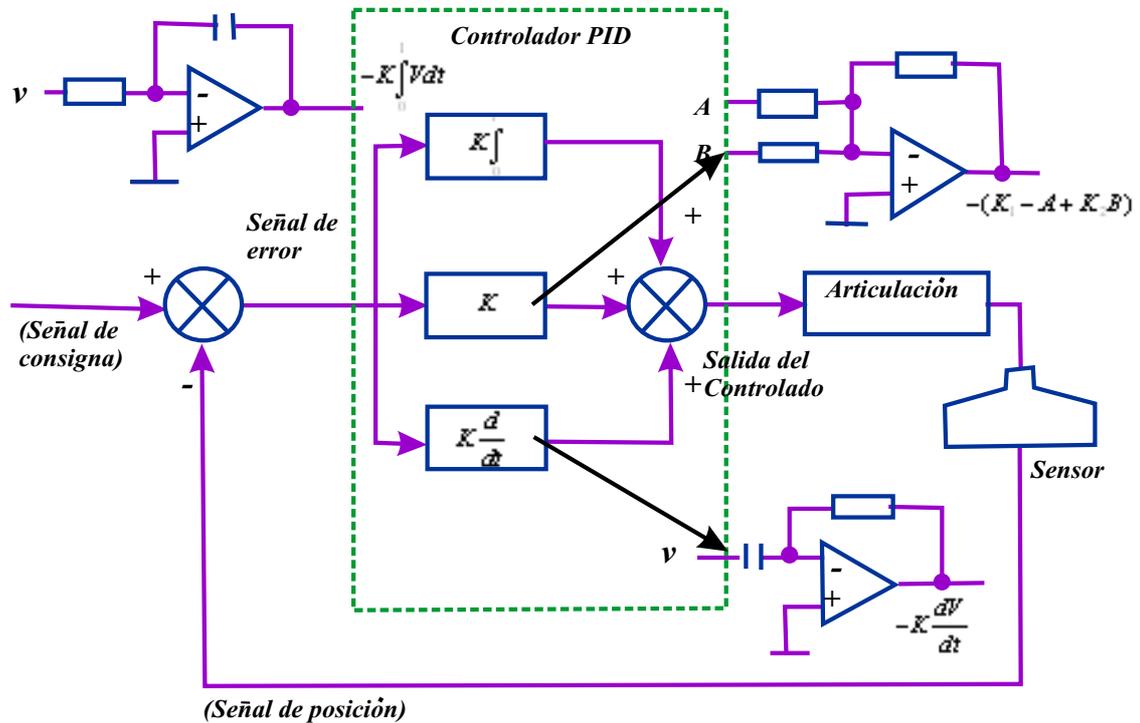


Figura 3.2: Esquema de un controlador PID[53].

3.4. Control Proporcional Derivativo (PD)

La acción del control derivativo proporciona una señal de control proporcional a la velocidad de cambio de la señal de error. Puesto que ésta no genera ninguna salida a menos que el error sea cambiado, en raras ocasiones se utiliza sola. El controlador PD se representa por [2]:

$$\tau = k_p e(t) + k_d \frac{de(t)}{dt} \quad (3.1)$$

donde:

k_p : constante proporcional.

k_d : constante derivativa.

τ : es la fuerza final aplicada (torque).

$e(t)$: es el error general con respecto al tiempo.

$\frac{de}{dt}$: derivada del error con respecto a t.

t: tiempo.

Mientras que:

$$k_d = 2\sqrt{k_p} \quad (3.2)$$

El efecto de la acción del control derivativo es anticipar cambios en el error y proporcionar una respuesta más rápida a los cambios [2].

3.5. Control Proporcional Integral Derivativo (PID)

Tres de las acciones de control se pueden combinar para formar el controlador PID se representa mediante [2]:

$$\tau = k_p e(t) + k_i \int e(t) dt + k_d \frac{de}{dt} \quad (3.3)$$

siempre que las condiciones se cumplan:

k_d debe de ser mucho menor que k_p y este a su vez mucho menor que k_i .

$$k_i \ll k_d \ll k_p$$

donde:

k_d : constante derivativa.

k_p : constante proporcional.

por lo tanto se puede utilizar una proporción estándar como a continuación se presenta:

$$k_i \ll 2\sqrt{k_p}$$

$$k_p = Wn^2$$

$$k_d = 2 * Wn$$

donde:

Wn : es la frecuencia natural.

El control PID es el tipo de control más general y, con toda probabilidad, es el tipo de controlador más utilizado. Proporciona una respuesta rápida, un buen control de la estabilidad del sistema y un bajo error de régimen permanente. Como se indicó con anterioridad, en los modernos controladores de robot, los cálculos asociados con alguno de los controladores anteriores suelen ejecutarse por microcomputadoras. [3][4]

El análisis de un sistema de control suele dividirse en dos partes: respuesta transitoria y respuesta de régimen permanente. La respuesta transitoria de un sistema es el comportamiento del sistema durante la transición desde

RAZÓN DE REACCIÓN DEL PROCESO	RETARDO TOTAL	BANDA PROPORCIONAL	RAZÓN DE REAJUSTE (RECÍPROCO DE LA CONSTANTE DE TIEMPO INTEGRAL)
Lenta	Corto	Angosta	Rápida
Lenta	Moderado	Mediana	Lenta
Rápida	Corto	Mediana	Rápida
Rápida	Moderado	Amplia	Lenta

Cuadro 3.1: Ajustes adecuados de un integrador Proporcional Integral para diferentes condiciones[36].

algún estado inicial al estado final. La respuesta de régimen permanente es el comportamiento del sistema cuando el tiempo se aproxima a “infinito”. [5]

En la Tabla 3.1 se resume la relación entre las características del proceso y el modo de control. Como es natural[21].

En la Tabla 3.1 se muestran los ajustes relativos de la banda proporcional y la razón de reajuste para un control proporcional más integral. Se supone que el control proporcional más integral funciona bajo varias condiciones de razón de reacción del proceso y retardo de transferencia y tiempo muerto. El retardo de transferencia y el tiempo muerto (retardo de transporte) han sido agrupados bajo el término retardo en la Tabla 3.2[21].

3.6. Control por modos deslizantes de segundo orden o PID no lineal.

En aplicaciones de control de robots, con tareas de regulación a una coordenada no singular del espacio de trabajo del manipulador, el control PD con compensación de gravedad y el control PID lineal tienen excelente desempeño. Sin embargo, estos controladores no tienen eficiente desempeño en tareas de seguimiento de trayectorias. Un control con eficiente desempeño en tareas de seguimiento, sin conocimiento de los parámetros dinámicos del robot, es el control PIDNL.

Si se sabe que S es el error y que éste contiene el error de posición (también llamado error extendido) y velocidad, entonces tenemos que la formulación es la siguiente:

$$S = e + \alpha \frac{de}{dt} \quad (3.4)$$

Y si entendemos que e está definida por:

$$e = \theta_{referencia} - \theta_{real} \quad (3.5)$$

MÉTODO DE CONTROL	RETARDO DE REACCIÓN DEL PROCESO (MÍNIMO)	RETARDO DE TRANSFERENCIA MÁXIMO	TIEMPO MUERTO	TAMAÑO DE LA PERTURBACIÓN DE LA CARGA (MÁXIMO)	VELOCIDAD DE LA PERTURBACIÓN DE LA CARGA (MÁXIMO)
Encendido-apagado	Largo solamente (no puede ser corto)	Muy corto	Muy corto	Pequeña	Lenta
Sólo proporcional	Largo o moderado (no puede ser corto)	Moderado	Moderado	Pequeña	Lenta
Proporcional mas integral	Cualquiera	Moderado	Moderado	Cualquiera	Lenta
Proporcional más derivativo	Largo o moderado (no puede ser muy corto)	Moderado	Moderado	Pequeña	Cualquiera
Proporcional más integral mas derivativo	Cualquiera	Cualquiera	Cualquiera	Cualquiera	Cualquiera

Cuadro 3.2: Tipos de procesos que pueden ser controlados satisfactoriamente por cada uno de los modos básicos de control[36].

Entonces derivando el error tenemos la siguiente ecuación:

$$\frac{de}{dt} = \frac{d}{dt}(\theta_{referencia} - \theta_{real}) \quad (3.6)$$

De esta forma el método es evaluado con el fin de determinar un momento que es próximo menor a los eslabones de referencia en comparación a los métodos anteriores.

Con esto nos puede llevar a un mayor control al momento de la ejecución. Sustituyendo en S los valores calculados se obtiene lo siguiente:

$$S = (\theta_{referencia} - \theta_{real}) + \alpha(\dot{\theta}_{referencia} - \dot{\theta}_{real}) \quad (3.7)$$

Por lo tanto la ecuación final es:

$$\tau = K_d * S + K_i \int \text{sgn}(s) dt \quad (3.8)$$

Donde:

$$\lim_{\beta \rightarrow \infty} \tanh(\beta S) = \text{Sgn}(S) \quad (3.9)$$

Por lo tanto:

$$\tau = K_d * S + K_i \int \tanh(\beta S) dt \quad (3.10)$$

Teniendo las siguientes condiciones:

$$S = e + \alpha \dot{e} \quad (3.11)$$

Donde la constante $\beta = 100$

Y las ganancias estarán definidas de la siguiente manera:

$$K_d = Wn^2 \quad (3.12)$$

$$\alpha = \frac{2}{Wn} \quad (3.13)$$

$$K_i \ll 2Wn^3 \quad (3.14)$$

Donde hay que mencionar que al referirse a Wn es la frecuencia natural que todo cuerpo posee y se tiene como constante, dentro del modelo es una constante que determina los valores iniciales a las constantes de proporción y derivación.

3.7. Control adaptable P-R

3.7.1. Historia del control adaptable

La técnica del Control Adaptable surge a partir de 1950 con el fin de dar solución a problemas de control en los que las características del sistema a controlar son variables o poco conocidas.

Una de las tantas definiciones que se han dado al control adaptable es la siguiente, según un foro realizado referente al tema, en el año de 1961: “Un sistema Adaptativo es aquel sistema físico que ha sido diseñado desde el punto de vista Adaptativo”. En otras palabras es un sistema del cual se espera se “adapte” a los cambios en la dinámica del proceso a controlar, esperando con ello resultados óptimos.

Otra definición, según el texto de control adaptable y robusto del profesor Francisco R. Rubio[56], control adaptable es un tipo de control no lineal en el que el estado del proceso puede ser separado en dos escalas de tiempo que evolucionan a diferente velocidad. La escala lenta corresponde a los cambios de los parámetros y por consiguiente a la velocidad con la cual los parámetros del regulador son modificados y la escala rápida que corresponde a la dinámica del bucle ordinario de realimentación.

El sistema de control adaptable debe proporcionar información continua sobre el estado actual del proceso. Es decir en todo momento realizar una IDENTIFICACIÓN. El sistema adaptable debe comparar el desempeño actual con el deseado y después de esto tomar la ”mejor“ decisión. En este caso los parámetros han experimentado un cambio, se han ADAPTADO.

Existe una diferencia muy clara entre el control clásico y el control adaptativo.

El control clásico está diseñado y estructurado para sistemas lineales y con parámetros invariantes en el tiempo. En caso de que se presenten alteraciones en el funcionamiento del sistema, es posible que el control no responda de manera eficiente.

El control adaptativo presenta una alternativa diferente. Este responde a cambios inesperados en la dinámica del sistema y a perturbaciones. Es un control con parámetros variantes en el tiempo.

Temas tales como sistemas lineales y no lineales, diseño de controladores, estabilidad, control estocástico, procesamiento de señales, estimación e identificación de sistemas, están íntimamente ligados con el control adaptativo.

3.7.2. Para qué sirve el control adaptable

Un regulador adaptativo está diseñado para controlar sistemas en los cuales se tiene parámetros variantes con el tiempo. Esta variación de los parámetros está ligada con envejecimiento de las componentes del sistema, con alinealidades de los actuadores, así como a cambios súbitos en el proceso (perturbaciones).

3.7.3. Control adaptable.

Para el diseño del control, es necesario diseñar una referencia nominal q_r tal que en lazo cerrado, el robot tenga un comportamiento estable, se propone la siguiente

$$\dot{q}_r = \dot{q}_d - \alpha \Delta q \quad (3.15)$$

sustituyendo Ecuación 3.15 en Ecuación 3.3 se obtiene el error extendido,

$$S = \dot{\Delta}q + \alpha \Delta q, \quad (3.16)$$

donde $\Delta q = q - q_d$, y q_d representa la trayectoria o coordenada de referencia.

El control adaptable τ propuesto es el siguiente

$$\tau = Y_r(q, \dot{q}, \ddot{q}_r) \hat{\Theta} - K_d S \quad (3.17)$$

$$\dot{\hat{\Theta}} = -\Gamma Y_r(q, \dot{q}, \ddot{q}_r)^T S \quad (3.18)$$

donde $K_d = K_d^T \in \mathfrak{R}^{n \times n}$ y $\Gamma = \Gamma^T \in \mathfrak{R}^{p \times p}$ son matrices diagonales definidas positivas, y $\hat{\Theta}$ representa el vector de parámetros estimados. La ecuación de error en lazo cerrado definida a partir de Ecuación 3.18 y 3.17 es

$$H(q) \dot{S} = -\{C(q, \dot{q}) + B\} S - Y_r(q, \dot{q}, \ddot{q}_r) \Theta + Y_r(q, \dot{q}, \ddot{q}_r) \hat{\Theta} - K_d S \quad (3.19)$$

simplificando,

$$H(q) \dot{S} = -\{C(q, \dot{q}) + B + K_d\} S + Y_r(q, \dot{q}, \ddot{q}_r) \Delta \Theta \quad (3.20)$$

donde $\Delta \Theta = \hat{\Theta} - \Theta$.

3.8. Conclusiones.

Se puede concluir que el control “es un algoritmo que permite la regulación de la energía aplicada a los motores que generan el movimiento del robot, de manera que alcancen los parámetros deseados con el fin de disminuir el error”.

Existen diversos tipos de control desde los que controlan sólo la posición, posición - velocidad, los que toman en cuenta las propiedades dinámicas del sistema los que consideran las variaciones de las características del robot, los controles presentados permiten observar las mejoras que presentan uno con respecto al otro de forma teórica.

Capítulo 4

Visualización virtual, análisis y diseño del sistema de “Seguimiento de trayectorias.”

4.1. Introducción

El desarrollo de nuevas tecnologías es cada vez más acelerado. Algunas producen grandes impactos mientras otras no logran superar las expectativas iniciales más simples. En este contexto, ¿Por qué el desarrollo de la Realidad Virtual se presupone importante?. Hay varias razones de orden sociológico, económico y práctico que permiten suponer que ésta tecnología alcanzará una posición relevante: un aumento explosivo de la información y de la complejidad del mundo, que requiere de herramientas y personas con mayor destreza en el manejo de variables múltiples, y con una rápida capacidad de reacción. Pero al mismo tiempo, se requiere que el riesgo sea mínimo para el individuo expuesto.

Un elemento indispensable para los hombres es la comunicación. Dado el gran auge que han tenido las computadoras, aparece la necesidad de mejorar la comunicación entre ellas y sus usuarios, es decir, intentar una comunicación más natural con la máquina. Es en este escenario que surge el concepto de realidad virtual y los ambientes virtuales, inicialmente como una interfaz hombre-máquina más eficiente.

En el presente capítulo se hace un estudio de forma comparativa de ciertas herramientas así como de lenguajes utilizados en el presente para el desarrollo y diseño de ambientes virtuales, en este proyecto se pueden apreciar algunas características que contienen cada uno de ellos y de ésta forma definir cuál ha de ser el óptimo lenguaje y/o herramienta que se utiliza para este trabajo a lo que se refiere la construcción del robot virtual.

De igual forma se hace una descripción de cómo funciona el modelo cinemático y dinámico en el ambiente virtual, así como las diferencias principales en cuanto a lo visual entre ambos modelos.

4.2. Conceptos preliminares de realidad virtual

Definir “Realidad virtual” (RV, o simplemente VR de su término inglés Virtual Reality) es difícil. Existen tantas definiciones como investigadores en un área, que dada su reciente y rápida evolución no han permitido tener una definición única. De este modo, no resulta extraño que la realidad virtual resulte ser altamente relativa para las diferentes personas y en las diferentes situaciones. Realidad Virtual, cada una de las palabras que componen este término resultan ser conflictivas a la hora de buscarles una definición única y exacta. No cabe duda que la realidad resulta ser altamente relativa para las diferentes entidades que la viven; y que el término virtual (que

tiene existencia aparente y no real), puede abarcar demasiado. Más que tratar de dar “una definición” veremos un cúmulo de definiciones, donde cada una entrega una idea de lo que se debería entender por realidad virtual:

- Es la experiencia de telepresencia, donde telepresencia es la sensación de presencia utilizando un medio de comunicación.
- Es una manera mediante la cual los humanos visualizan, manipulan e interactúan con computadoras y datos extremadamente complejos.
- Es un paso más allá de lo que sería la simulación por computadoras, tratándose más bien de una simulación interactiva, dinámica y en tiempo real de un sistema.
- Consiste en simulaciones tridimensionales interactivas que reproducen ambientes y situaciones reales.
- Es un entorno de tres dimensiones sintetizado por computadora en el que participantes acoplados de forma adecuada pueden atraer y manipular elementos físicos simulados en el entorno y, de alguna manera, relacionarse con las representaciones de otras personas pasadas, presentes o ficticias, o con criaturas inventadas.
- Es un ambiente altamente interactivo donde el usuario participa a través del uso de una computadora en un mundo virtualmente real. Es una simulación tridimensional por computadora durante la cual el usuario resulta inmerso tan completamente que ésta realidad, de origen artificial, aparenta ser real.
- Es simulación por computadora, dinámica y tridimensional, con alto contenido gráfico, acústico y táctil, orientada a la visualización de situaciones y variables complejas, durante la cual el usuario ingresa, a través del uso de sofisticados dispositivos de entrada, a mundos que aparentan ser reales, resultando inmerso en ambientes altamente participativos, de origen artificial. Una nueva y sorprendente forma de navegar información.
- Es el medio que proporciona una visualización participativa en tres dimensiones y la simulación de mundos virtuales, siendo dichos mundos el elemento fundamental de un sistema de realidad virtual. La realidad virtual es un entorno generado por computadora en el que los participantes pueden entrar físicamente e interactuar con él desplazándose por su interior o modificándolo de cualquier forma. En su forma más simple, un mundo virtual podría estar compuesto por un edificio tridimensional por el que podríamos desplazarnos, aunque sin modificar nada. Sin embargo, con el equipamiento adecuado, los usuarios podrían ver, desplazarse e interactuar a través de estos entornos gráficos generados por computadora.

Existen diversos términos para referirse al mismo concepto, tal es el caso de “realidad sintética”, “mundos virtuales o ficticios” y “ciberespacio”, palabras diferentes, mismos significados, idénticas indefiniciones.

Por regla general se reconoce que el término Realidad Virtual fue acuñado originalmente por Jaron Lanier, ex director de VPL Research, en 1989 (Myron Krueger fue el primero en atribuírselo), para distinguir entre simuladores y la experiencia de inmersión de realidad virtual, aún que otros autores difieren de esta posición planteando que ya en 1987 David Zelzer (Media Lab) utilizaba dicha expresión. Aunque el origen exacto del uso de la expresión “realidad virtual” es ambiguo, y más bien anecdótico, la cronología del desarrollo de la misma es bastante más clara.

Aunque los dispositivos tradicionales de input-output con la computadora (teclado, mouse, monitor, joystick) ofrecen grandes facilidades para el manejo de información, están limitados cuando las aplicaciones van ganando en complejidad y sofisticación, como es el caso de la realidad virtual que requiere de otros dispositivos para poder aprovechar todo su potencial.

En 1958, la Philco Corporation desarrolla un sistema basado en un dispositivo visual del casco, controlado por los movimientos de la cabeza del usuario; algunos años después, Iván Sutherland y otros crean el Casco Visor

(Head Mounted Display, HMD) mediante el cual el usuario podía examinar, moviendo la cabeza, un ambiente gráfico. Morton Heiling inventa y opera el Sensorama.

En 1969, Myron Krueger crea ambientes interactivos que permiten la participación de cuerpo entero, en eventos apoyados por computadoras. Por aquel entonces, diversos artistas ofrecían espectáculos que incluían imágenes, sonidos, vibraciones y hasta olores, que junto con cámaras que captaban sus movimientos sumergían al espectador en un mundo irreal, reproducido en una pantalla gigante. Paralelamente, Frederick Brooks, de la Universidad de Carolina del Norte en Chapel Hill, trabajaba en un proyecto para lograr que los usuarios muevan objetos gráficos mediante un manipulador mecánico.

Para fines de los 70, en el Media Lab del Instituto Tecnológico de Massachusetts (MIT), se obtiene el Mapa Filmado de Aspen, EE.UU. una simulación de vídeo de un paseo a través de la ciudad, donde el usuario puede recorrer sus calles y edificios. Tom de Fanti fue el inventor del Guante de Datos en 1976; aunque su diseño fue mejorado posteriormente por Tom Zimmerman, dando origen al DataGlove, diseñado originalmente para poder tocar una guitarra virtual o imaginaria. Aunque fue Jason Lanier quien comenzó a crear y a divulgar las posibilidades de las nuevas tecnologías y el que se hizo famoso en todo el mundo al presentar en 1984 una versión mejorada del guante de datos para la NASA. Ese mismo año, Michael McGreevy y sus colegas de la NASA desarrollan Lentes de Datos, con los cuales el usuario puede mirar el interior de un mundo gráfico mostrado en una computadora[54].

En los inicios de los 90, los sistemas de realidad virtual salen de los ambientes de laboratorios en busca de aplicaciones comerciales, apareciendo las cabinas de entrenamiento para pilotos de guerra, los conocidos simuladores de vuelo, los recorridos virtuales, entre otras muchas aplicaciones.

En la actualidad, estamos aún en presencia del crecimiento y consolidación de las técnicas y recursos de la realidad virtual, el cual ha sido posible gracias al esfuerzo e interés combinado de científicos, militares y visionarios, y por que no decirlo, al dinero de las empresas que ven en ella una nueva y prometedora máquina de hacer dinero.

Para poder evaluar a un sistema realidad virtual como tal, es necesario que cumpla con ciertos elementos, entre los que se destacan[54]:

- **Simulación:** simulación para modelar un sistema, cabe decir que ésta tiene que ser lo suficientemente realista para convencer al usuario. Es decir una simulación del modelo o del mundo a experimentar, donde regirán una serie de reglas, no necesariamente iguales a las de la vida real.
- **Interacción:** interacción para tener control de la exploración del sistema creado; de no existir ésta interacción el sistema no dejaría de ser una película o recorrido fijado a priori. Para lograr la interacción existen diversas interfaces hombre-máquina, que van desde teclado y mouse hasta guantes o trajes sensoriales. La interactividad con el mundo virtual supone que el usuario pueda mover objetos (además de sí mismo) y modificarlos, y que tales acciones, produzcan cambios en ese mundo artificial.
- **Percepción (Inmersión):** la percepción viene a ser el factor más importante, algunos sistemas de realidad virtual se dirigirán principalmente a los sentidos (visual, auditivo, táctil) por medio de elementos externos (cascos de visualización, guantes de datos, cabinas, etc.); otros tratarán de llegar directamente al cerebro, evitando así las interfaces sensoriales externas; y otros, los más simples recurrirán a toda la fuerza de la imaginación del hombre para experimentar una realidad virtual relativa. De este modo el usuario puede creer que realmente está viviendo las situaciones artificiales que el sistema de realidad virtual genera. La característica de percepción es asociada con el grado de inmersión que el usuario siente.
- **Generación de imágenes:** en un sistema de realidad virtual las imágenes mostradas al usuario no se encuentran almacenadas en ningún sitio, sino que son generadas dependiendo de la perspectiva que pretenda observarse. Esto es debido a la total libertad de movimientos que disfruta el usuario, y que hace imposible tener guardadas todas las imágenes correspondientes a todos los posibles puntos de vista. Los sistemas de

realidad virtual poseen una base de datos con todos los elementos que componen el mundo virtual, a partir de la cual, éstos generan la información que se mostrará al usuario.

- **Tridimensionalidad:** debe existir realmente una dimensión de profundidad, similar a la que ofrecen los simuladores de vuelo. Para conseguir este efecto de tridimensionalidad los objetos del mundo virtual deben tener asociada una tercera dimensión que marque su profundidad en la pantalla[54].

Los tres primeros elementos, son imprescindibles en cualquier sistema de realidad virtual, aunque algunos investigadores plantean que las tres bases de la realidad virtual son las “3i” (Interacción, Inmersión e Imaginación)[54].

4.2.1. Clasificación de la realidad virtual.

El hardware de realidad virtual es bastante variado. Así que en función de estos elementos involucrados, se pueden clasificar cuatro grandes grupos de sistemas que se proclaman como realidad virtual[54]:

- *Sistemas Desktop de Realidad Virtual:* o realidad virtual de escritorio, donde se engloban todas aquellas aplicaciones que muestran una imagen 2D ó 3D en un monitor en lugar de proyectarla en un “casco”. Los ejemplos típicos son los simuladores de vuelos y la mayoría de los juegos para PC’s. Algunos tendrán interfaces sofisticadas, pero tienen en común el concepto de 3D en 2D.
- *Realidad Virtual en Segunda Persona:* son aquellos sistemas en los cuales el usuario sabe que está en el mundo virtual porque se ve a sí mismo dentro de la escena. Es decir es un integrante “visible” del mundo virtual, por medio de la proyección de su imagen en un fondo o ambiente, las cuales son visualizadas en la pantalla. Aplica la idea de “ver para creer” para inducir la sensación de presencia.
- *Sistemas de Telepresencia:* utilizan generalmente elementos como cámaras, micrófonos, dispositivos táctiles y de fuerza con elementos de retroalimentación, ligados a elementos de control remoto para permitir al usuario manipular robots u otros dispositivos ubicados en lugares remotos mientras se está experimentando en forma virtual.
- *Sistemas de Inmersión de Realidad Virtual:* son aquellos que literalmente sumergen al usuario en el mundo virtual, mediante la utilización de sistemas visuales del tipo HMD, equipos sensores de posición y movimientos así como procesadores de sonido. Quedando el usuario estrechamente relacionado con el ambiente virtual, y aislado hasta cierto punto del mundo real. Los usuarios ven al mundo virtual como si estuvieran viendo al mundo real[54].

4.2.2. Características generales del ambiente virtual.

En el ambiente virtual desarrollado el modelo tridimensional del robot se presenta como la unión de diferentes superficies dibujadas en un espacio común determinado por los ejes X, Y, y Z.

Este ambiente posibilita el movimiento gradual del robot a partir de rotaciones en cada una de sus articulaciones, habilitadas utilizando barras de deslizamiento cuyos valores mínimo y máximo corresponden a los límites reales de movimiento. De manera simultánea permite visualizar la posición del efector final, calculada a partir de la rotación efectuada en cada articulación, según el sistema de referencia ubicado en la base del modelo que corresponde al sistema de referencia real del robot.

Las opciones principales de manejo de gráficos ofrecidas por Matlab, son utilizadas para facilitar la visualización del modelo. El volumen de trabajo, es presentado por el programa a petición del usuario, después de ser calculado según los límites de las articulaciones del robot[55].



Figura 4.1: Portada de AutoCAD 2006[44].

4.2.3. Herramientas de edición

1. AutoCAD.

AutoCAD de Autodesk es una herramienta dentro del campo denominado CAD (Computer Aided Design) o Diseño Asistido por Computadora. Para ser más exacto AutoCAD. es un programa de diseño en CAD analítico (frente a otros sistemas de CAD paramétrico). La versatilidad del sistema lo ha convertido en un estándar general[44].

Características:

- Dibujar de una manera ágil, rápida y sencilla, con acabado perfecto y sin las desventajas que encontramos si se ha de hacer a mano.
- Permite intercambiar información no solo por papel, sino mediante archivos, y esto representa una mejora en rapidez y efectividad a la hora de interpretar diseños, sobretodo en el campo de las tres dimensiones.

Con herramientas para gestión de proyectos podemos compartir información de manera eficaz e inmediata. Esto es muy útil sobretodo en ensamblajes, contrastes de medidas, etc..

- Es importante en el acabado y la presentación de un proyecto o plano, ya que tiene herramientas para que el documento en papel sea perfecto, tanto en estética, como, lo más importante, en información, que ha de ser muy clara. Para esto tenemos herramienta de acotación, planos en 2D a partir de 3D, cajetines, textos, colores, etc. aparte de métodos de presentación fotorrealísticos.
- Un punto importante para AutoCAD es que se ha convertido en un estándar en el diseño por computadora debido a que es muy versátil, pudiendo ampliar el programa base mediante programación (Autolisp, DCL, Visual Basic, etc.)[44].

2. 3D Studio.



Figura 4.2: Portada de 3D Studio Max 7.0[45].

3D Studio Max es una aplicación basada en el entorno Windows (9x/NT) que permite crear tanto modelados como animaciones en tres dimensiones (3D) a partir de una serie de vistas o visores (planta y alzados).

La utilización de 3D Studio Max permite al usuario la fácil visualización y representación de los modelos, así como su exportación y salvado en otros formatos distintos del que utiliza el propio programa.

Características:

- Es reescrito en un lenguaje de programación orientado a objetos.
- Funcionalidad reservada a costosas estaciones de trabajo gráficas.
- Integración de los entornos de modelado y animación.
- Manipulación en tiempo real con o sin presencia de tarjeta de aceleración de gráficos en 3D.
- Posee una función deshacer multinivel.
- 3D Studio MAX es, además un entorno de desarrollo excepcional gracias a su arquitectura “plug-in”.
- 3D Studio MAX está totalmente abierto a desarrollos externos.

3. Maya.



Figura 4.3: Portada de Maya [46].

Maya es un software de modelado 3D especialmente diseñado para la animación de personajes y la creación de efectos especiales[46].

Características:

Entre las principales características de animación de personajes en 3D implementadas en Maya se encuentran:

- Deformaciones que se pueden agrupar, reordenar y animar.
- Un conjunto de herramientas de IK (Inverse Kinematics) para tener el control preciso sobre el movimiento de los personajes.
- Control de la animación facial a partir de un conjunto de Blend Shapes.
- Herramientas de modelado de la piel.
- Control de las expresiones de los personajes.
- Soporte de captura de movimiento en vivo.
- Sincronización sonora integrada.
- Maya también incorpora un avanzado conjunto de herramientas para la creación de efectos visuales, como la caída de agua sobre un molino o la simulación de la colisión de dos galaxias.

Las aplicaciones de Maya son:

- Diseño Arquitectónico.
- Animación en general.
- Animación de personajes.
- Dinámica.
- Modelado.
- Renderizado.

4.2.4. Lenguajes

VRML



Figura 4.4: Portada de VRML[47].

Es un acrónimo para Virtual Reality Modeling Language (Lenguaje para Modelado de Realidad Virtual). Técnicamente hablando, VRML no es un lenguaje para programar realidad virtual inmersiva ni tampoco un lenguaje de modelado.

La realidad virtual inmersiva implica una experiencia tridimensional inmersiva y dispositivos externos como cascos o guantes digitales para lograr capturar otros sentidos diferentes al oído y a la vista. VRML no prevé una inmersión sensorial total[47].

Características:

- Es un lenguaje estándar, y por consiguiente, universalmente utilizado en Internet como el lenguaje para simulaciones interactivas dentro de la Web.
- El VRML es un lenguaje para descripción de escenas no un lenguaje de programación.
- Provee un conjunto básico de primitivas para el modelaje geométrico tridimensional y tiene la capacidad de dar comportamiento a los objetos y asignar diferentes animaciones que pueden ser activadas por eventos generados por diferentes usuarios.
- Capaz de realizar rutas preestablecidas, como el movimiento del brazo de un robot, el giro de un ave en pleno vuelo, etc.
- Representación de los objetos en 3D.
- Recubriendo con Texturas un objeto.
- Manejo de cámaras.
- Manejo de luces y sombreados.
- Movimiento.
- Guión (Script)[49].

OpenGL

OpenGL es una API (Application Programming Interface) creada por Sillicon Graphics Inc. para simplificar la programación de aplicaciones gráficas en 2D ó 3D. OpenGL pone a nuestra disposición un pequeño conjunto de primitivas geométricas, a la par de estas primitivas, también brinda comandos para manipular la forma en que se verán los objetos. Una cosa importante de OpenGL es que es independiente del sistema operativo y del sistema de ventanas en el que se ejecute[48].

Características:

- Tecnología que facilitan la manipulación de los modelos 3D, e incluso puede ayudar a crearlos.
- No hay preocuparse por el hardware a la hora de programar, por lo tanto la programación no se hace tan complicada, de igual manera el código que se escribe con openGL será independiente de la plataforma y del sistema operativo.
- Sigue una serie de reglas para nombres de funciones y uso de variables.
- OpenGL permite al desarrollador escribir aplicaciones que se puedan desplegar en varias plataformas fácilmente.
- Primitivas geométricas: permiten construir descripciones matemáticas de objetos. Las actuales primitivas son: puntos, líneas, polígonos, imágenes y bitmaps.
- El doble buffering ayuda a eliminar el parpadeo de las animaciones. Cada fotograma consecutivo en una animación se construye en un buffer separado de memoria y mostrado sólo cuando está completo[48].

4.2.5. Visual C++

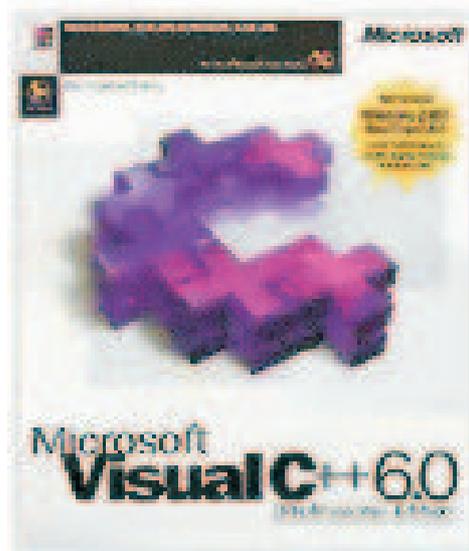


Figura 4.5: Portada de Visual C++.

Como sabemos, Windows es el entorno más popular de interfaz gráfico de usuario (GUI). Desde este punto de vista, Windows es un entorno multitarea basado en ventanas, que representan programas, y que permite ejecución concurrente.

Para desarrollar programas, Windows provee una librería de rutinas y funciones (SDK - Kit de desarrollo de software) que permiten gestionar componentes como menús, diálogos, ventanas, etc..

Visual C++ es un entorno integrado de desarrollo que permite la programación orientada a objetos (POO) conjuntamente con el sistema de desarrollo SDK (también denominado API) de Windows.

Características:

- Interfaz de dispositivo gráfico de Windows. Se introdujo un nivel de abstracción llamado la interfaz de dispositivo gráfico (GDI) que proporcionan controladores de video e impresora conectadas al sistema; en vez de dirigirse al hardware, el programa llama a las funciones de GDI que hacen referencia a una estructura de datos llamada contexto de dispositivo, Windows relaciona el contexto de dispositivo con el dispositivo físico y emite las instrucciones de entrada/salida adecuadas.
- Bibliotecas de enlace dinámico. Permite el enlazado dinámico, lo que significa que pueden cargar y enlazar en tiempo de ejecución bibliotecas construidas de una forma especial. Múltiples aplicaciones pueden compartir bibliotecas de enlace dinámico (DLL), lo que ahorra memoria y espacio en disco.
- Procesamiento de mensajes. El sistema operativo llama a main cuando el usuario ejecuta un programa.
- Programación basada en recursos. Para realizar una programación dirigida por datos en MS-DOS, es necesario codificar los datos como constantes de inicialización o proporcionar archivos de datos aparte para que los lea el programa.
- Gestión de memoria. Algunas de las funciones de gestión de memoria de Win16, como el GlobalAlloc, se trasladaron a Win32, pero se hizo para permitir a los desarrolladores modificarán el código fuente con rapidez al pasar de una versión a otra.
- Entre sus principales componentes se encuentran:
 - El AppWizard. Es un generador de código que crea un esqueleto de trabajo de una aplicación de Windows con características, nombres de clases y nombres de archivos código que se especifican a través de cuadros de diálogo. El código del AppWizard es mínimo; la funcionalidad se encuentra en las clases de base del armazón de la aplicación.
 - El classWizard. Es un programa implementado como DLL, simplifica el mantenimiento del código de clase de Visual C++; en caso de requerir una fusión virtual nueva o una función de gestión de mensajes, el ClassWizard escribe los prototipos, cuerpos de de la función y si es necesario , el código para enlazar el mensaje de Windows a la función.

4.2.6. Delphi 7.0

En el pasado las opciones de lenguajes de programación eran más o menos las siguientes: si se buscaba facilidad de aprendizaje, existía BASIC. Para programar sistemas comerciales existía variedad en versiones de COBOL. Cuando se manejaban cantidades de datos relativamente grandes se pensaba en programar DBASE III+. No se descarta C y Assembler si se requería de velocidad y acceso a los recursos básicos del Sistema Operativo.

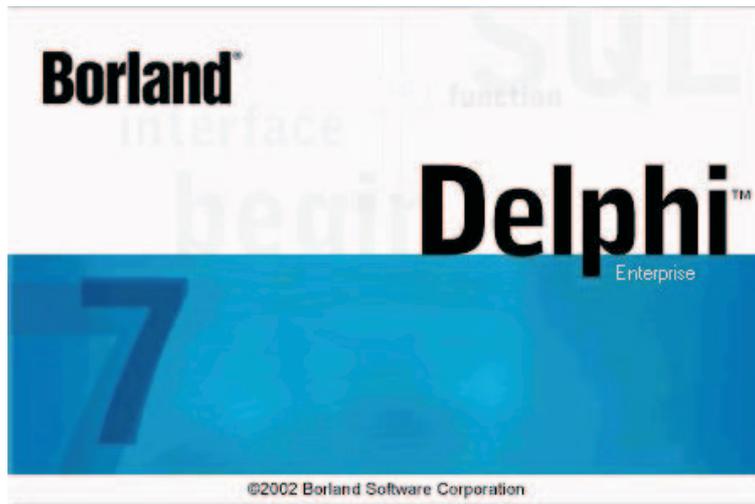


Figura 4.6: Portada de Delphi 7.0.

Borland crea Turbo Pascal, lenguaje creado con fines académicos, para enseñar las bases de la programación estructurada. El Borland ponía en las manos de los programadores un lenguaje poderoso, un compilador sumamente rápido y un Entorno Integrado de Desarrollo desde el que se podía editar el código, compilarlo e incluso ejecutarlo sin necesidad de utilizar programas externos.

El desarrollo del lenguaje y el entorno fue rápido, siempre ajustándose a las nuevas tecnologías, incorporó los conceptos de Programación Orientada a Objetos en cuanto apareció el nuevo modelo. Las extensiones realizadas al Pascal original seguían la filosofía del lenguaje, haciéndolas fáciles de incorporar para los programadores. A partir de la aparición de los objetos se conoce al lenguaje como Object Pascal.

En las primeras versiones para Windows, Borland ofrecía junto con el compilador una biblioteca de objetos muy bien construida -Object Windows Library, OWL- que hacía de capa intermedia entre la Application Programming Interface (API) de Windows y el programador. No obstante, todavía era necesario escribir muchas líneas de código y conocer bastante de Programación Orientada a Objetos.

Y apareció Visual Basic de Microsoft, que revolucionó el mercado. A partir de ese momento, quedó claro que los entornos de desarrollo bajo Windows tenían que ser parecidos al de ese producto. Miles de programadores se pasaron a las filas de Visual Basic. Borland no se hizo esperar y lanzó al mercado Delphi, basado en el lenguaje Pascal pero con un entorno de programación que hace más fácil el desarrollo de aplicaciones en entorno Windows.

¿Por qué Delphi?. La respuesta a esta pregunta puede no ser fácil. Se podría decir que la herramienta adecuada para un trabajo es aquella que permite la consecución correcta y a tiempo del mismo, además de estas condiciones mínimas, cada lenguaje se desempeña mejor en ciertas áreas o desarrollos; por ejemplo, Clipper fue concebido para manejar bases de datos con formato dBase, y es ese su fuerte.

Delphi “lenguaje de propósito general”, se puede definir como tal debido a que no tiene un lugar muy bien definido; esto significa que se comporta bien ante tipos diferentes de problemas, y Borland ha puesto mucho ahínco

en lograr que el rendimiento sea el mejor posible. Delphi se puede usar para casi cualquier tipo de programa, obteniéndose un rendimiento excelente con facilidad.

Características de Delphi:

- Posee un Entorno de Desarrollo Integrado para Windows (IDE, incluye un editor de texto ASCII, un visor de propiedades y eventos y un potente debugger.), con características de programación visual. Esto significa que la mayor parte del programa se hace gráficamente, lo hace ahorrar tiempo y esfuerzo.
- El lenguaje Pascal ha sido ampliado y mejorado, sobre todo para adaptarlo a la programación con objetos.
- La programación se hace más intuitiva y sencilla con el uso de componentes.
- Incluye un soporte de Bases de Datos poderoso y fácil de usar.
- Genera ejecutables nativos, sin necesidad de librerías de run-time (salvo el caso de los programas que usan Bases de Datos).
- Se pueden crear componentes nuevos que se integran en el entorno de la misma manera que los nativos.
- Igualmente Delphi utiliza la RAD (Rapid Application Development), que permite el desarrollo de programas de forma rápida y visual.
- Delphi controla por si solo el sistema de mensajes de Windows, con lo cual el programador necesita preocuparse de capturar los mensajes que dicta Windows para desencadenar un procedimiento.
- Delphi utiliza una librería de componentes llamada VCL (Visual Components Library) que brinda una serie de objetos y que nos da la posibilidad de utilizar los objetos sin necesidad de llamar a las distintas API de Windows para se dibuje en pantalla el componente, además de poder manipular al antojo el componente, así como poder controlar de una manera muy fácil los distintos eventos que podría generar.

Proyección Delphi:

Si bien Delphi no ha sido un lenguaje muy difundido o utilizado bajo entorno Windows como lo es Visual Basic, Linux lo vio con otros ojos. Su nueva versión del clásico sistema operativo, llamada Kilix, incluye gran parte de código de Delphi para las tareas de administración de procesos y de memoria. Los programadores y expertos en Linux aseguran que “un buen programador debe saber programar en Delphi”.

Entre otras cosas, Delphi se está orientando a la programación de sitios web dinámicos, lo que le abre el paso para competir con los lenguajes que están dominando esta área, como PHP, Pearl o ASP. Por lo visto, ahora es muy necesario comenzar a entender Delphi, ya que parece ser un lenguaje prometedor.

4.3. Construcción del Ambiente virtual

4.3.1. Animación al ambiente virtual

4.3.2. Construcción del robot virtual

El robot PHANToM está constituido por tres eslabones y tres articulaciones (sin tomar a la base como el eslabón cero) 4.7.[25].

Primera aproximación (con rectángulos).

Se construyó un rectángulo para representar al primer eslabón, como se muestra en la Figura 5.3), para dibujar el segundo eslabón, primero se trasladó el origen del píxel de trazo: `glTranslatef(0.0f, longitud eslabon2, 0.0f);` y se dibujó otro rectángulo para representar al segundo eslabón, lo mismo se hizo para el tercer eslabón, con respecto del segundo eslabón[25].

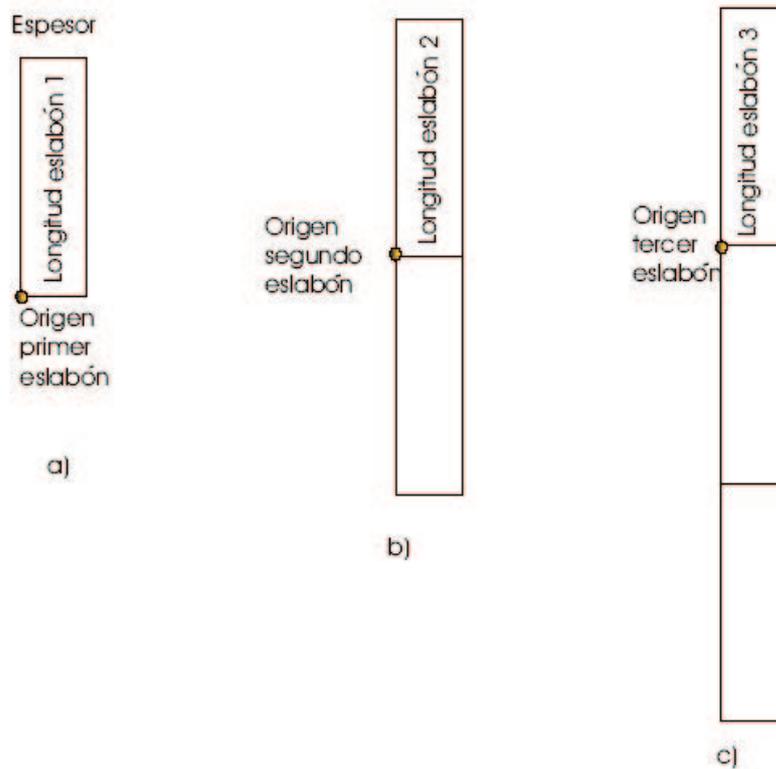


Figura 4.7: Aproximación con rectángulos [25].

Las funciones para dibujar un eslabón con un rectángulo son:

```
Begin (GL_QUADS); glVertex3f(0.0f, 0.0f, 0.0f);  
glVertex3f(ESP,0.0f, 0.0f); glVertex3f(ESP, LongEsl1, 0.0f);  
glVertex3f(0.0f,LongEsl1, 0.0f); glEnd();
```

Segunda aproximación (con paralelepípedos).

Para construir un paralelepípedo se necesitan 6 rectángulos para cada una de las caras como se muestra en la Figura 4.8.

Para dibujar un paralelepípedo se implementó una función que dibuje los 6 rectángulos cuando se le pasan 3 parámetros que son: esq1 (ancho), esq2 (alto), esq3 (largo; así mismo para dar movimiento a cada uno de los eslabones se emplea también el criterio de la Figura 4.8, en la Figura 4.9 se muestra la generación de movimiento entre los eslabones.

Construcción especial para el primer eslabón e incorporación de esferas en las articulaciones El primer eslabón se construye mas grande colocándole una esfera en la parte superior, estas dos características proporcionan una mejor apariencia al robot. El primer eslabón se construye a partir de un cubo, es decir, que el valor de altura (longitud del eslabón) es el mismo valor para el largo y ancho.

La esfera se construye con un radio con valor en porcentaje a la longitud del primer eslabón. Un porcentaje que proporciona una buena apariencia es de 75 %, es decir, que el radio de la circunferencia es el 75 % de la longitud del eslabón. Para mejorar la apariencia de la segunda y tercera articulación, se dibuja una esfera en ellas.

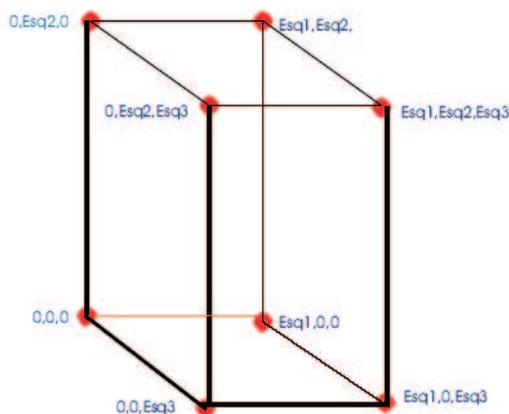


Figura 4.8: Paralelepípedo a partir de 6 rectángulos [25].

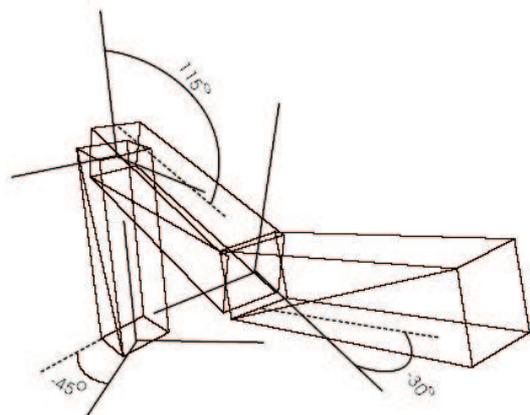


Figura 4.9: Configuraciones de eslabones con paralelepípedos [25].

4.4. Comportamiento complejo

Después de realizarse las comparaciones entre el desempeño cinemático y dinámico visualmente se observa que en el cinemático no presenta problema alguno al realizar el movimiento dado que sólo requiere de que se le de un punto de referencia, ahora bien debido a la involucración de las propiedades dinámicas se observa que realiza pequeñas pero significantes oscilaciones en el desempeño visual dinámico, esto favorece ya que los movimientos se acercan aún más a la realidad.

4.5. Análisis y diseño de la interfaz de “Seguimiento trayectorias ”.

Considerando el propósito primordial de éste proyecto, que consiste en desarrollar una herramienta que permita la simulación, visualización del robot en el seguimiento de trayectorias (que en esta caso de estudio es una circunferencia con un radio de 0.1 y con centro en (0.3,0.2)) así como la visualización de los resultados numéricos y su representación en forma gráfica, en ésta sección se presenta el análisis y diseño de ésta interfaz.

4.5.1. Análisis.

Para el desarrollo del sistema se consideraron los siguientes puntos:

- Visualización del Robot virtual.
- Simulación de un control PIDNL.
 - Control PIDNL.
 - Control PIDNL con convergencia en tiempo finito (TBG).
 - Carga de Archivo de Matlab para la visualización de un PIDNL con convergencia en tiempo finito (TBG).
- Carga de Archivo de Matlab para la visualización de un control adaptable con convergencia en tiempo finito (TBG).

Nota: los puntos relacionados con control PIDNL así como adaptable la trayectoria que siguen es una circunferencia.

La visualización del robot se podrá realizar para cada uno de los puntos anteriores, la simulación se resuelve a través del método de solución de ecuaciones diferenciales de Runge Kutta, el método empleado es de orden superior debido a la exigencia de precisión en la tarea.

Para la visualización del robot virtual se retomó la aplicación “PHANToM Virtual TRAZO” [Herbert Lara Ordaz, “Análisis y diseño de una interfaz de visualización virtual del dispositivo háptico PHANToM 1.0 basada en la dinámica de Euler-Lagrange”, Universidad Autónoma del Estado de Hidalgo, Pachuca, Hidalgo, 2005.], desarrollada bajo el lenguaje C++ con OpenGL, es por esto que la aplicación desarrollada se debió adaptarse al funcionamiento de la interfaz de visualización ya existente.

La carga de los archivos de Matlab se justifica debido al desarrollo de programas que resuelven el modelo matemático del robot a través de Matlab, éste equipo de trabajo desarrolló programas en Matlab para resolver

un control adaptable, un control por modos deslizantes de segundo orden y dos controladores clásicos, es por esto que en la aplicación se muestra los resultados obtenidos en los primeros dos mencionados.

A continuación se muestran los diagramas de flujo utilizados, en primer lugar se apoyó en un diagrama de flujo de información de sistema, el cual sirve para describir el flujo de información entre los distintos soportes físicos del sistema, refleja las operaciones normales para el desarrollo del proceso que realizan los componentes del sistema; de igual forma se muestran diagramas de flujo de detalle que abordan la solución del problema planteado.

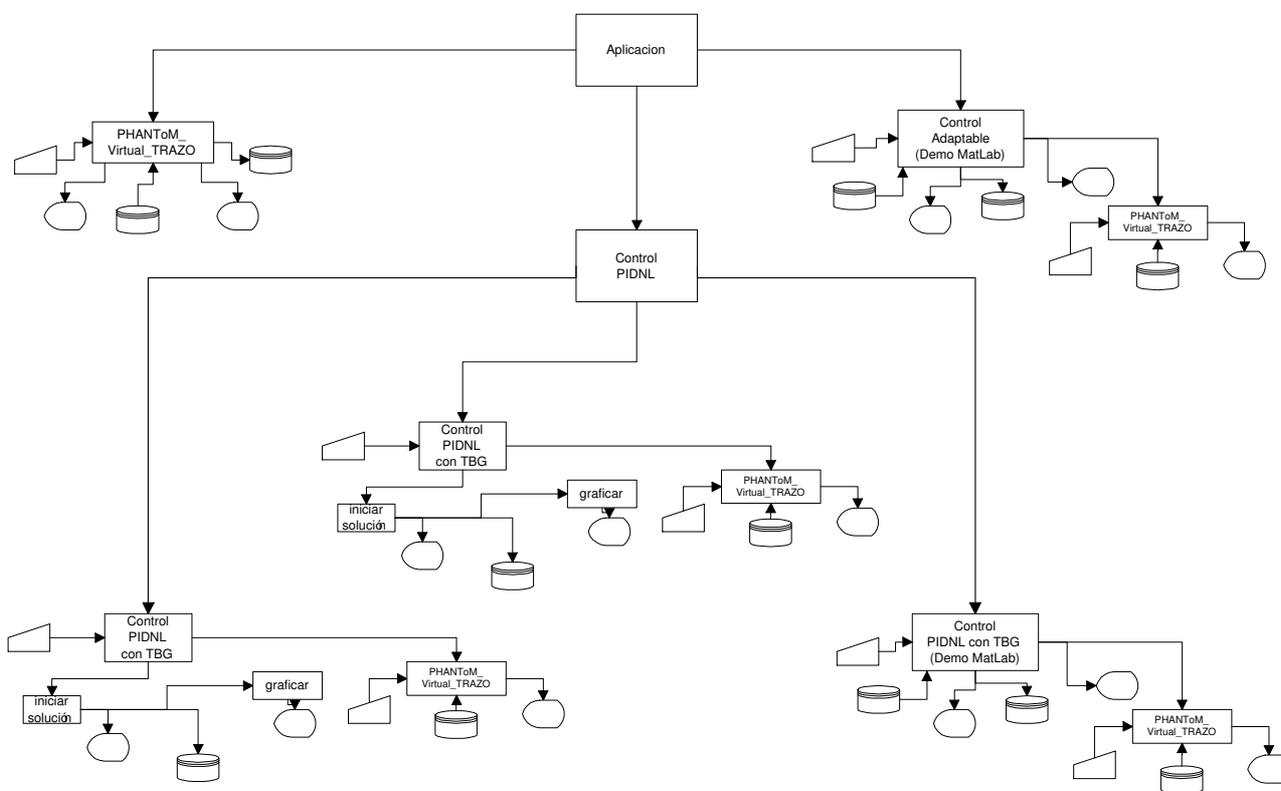


Figura 4.10: Diagrama de flujo información del sistema “Seguimiento de trayectorias”.

El diagrama presentado en la Figura 4.11 se puede observar el procedimiento que se lanza o mejor dicho se inicia para dar paso al proceso de solución del problema de seguimiento de trayectoria con un control PIDNL ya sea con o sin TBG, en dicho procedimiento se inicializa al robot en una posición inicial, así como el primer punto a seguir que será cero para x, y y z, de acuerdo al tiempo especificado de simulación se calcula el número de iteraciones que se realizarán ya que el tiempo de simulación podrá ser cambiado por el usuario, en este procedimiento es donde se va generando el archivo de trayectoria que será leído por la interfaz “PHANToM Virtual TRAZO” de igual forma se generan las gráficas de las variables articulares y los errores de las mismas, para llevar acabo la solución se requiere de llamar a otros procedimientos y funciones.

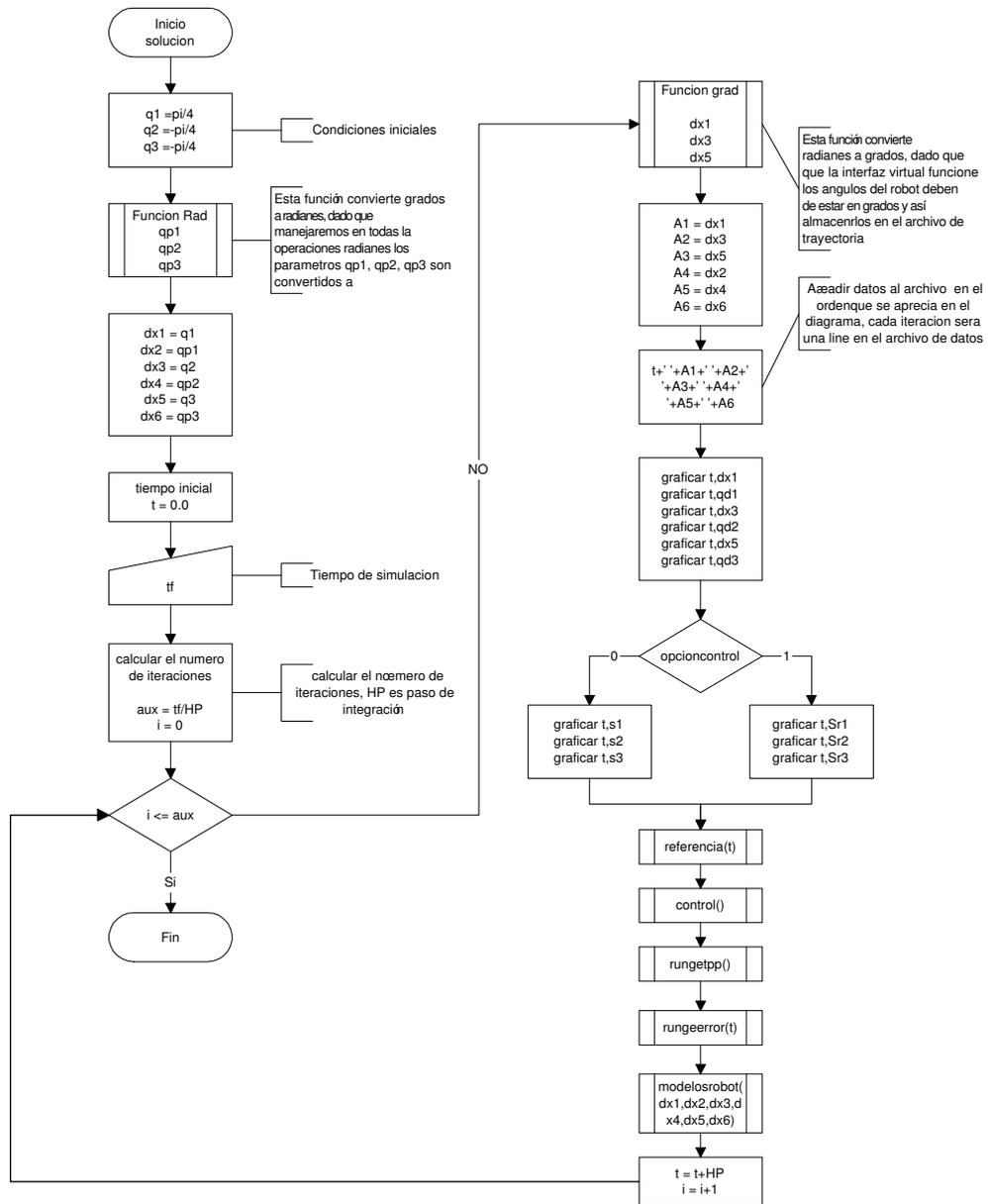


Figura 4.11: Diagrama de flujo del procedimiento de solución.

Para llevar a cabo el seguimiento se requiere saber el punto a seguir en cada una de las iteraciones, de lo que se encarga el procedimiento Referencia, el cálculo de dichos puntos se hace por medio de la ecuación de la circunferencia, así también se hace uso de la primera y segunda derivada de la referencia, de igual forma es necesario calcular el modelo cinemático directo, cinemático inverso de velocidad y aceleración, el diagrama de éste procedimiento se puede ver en la Figura 4.12 .

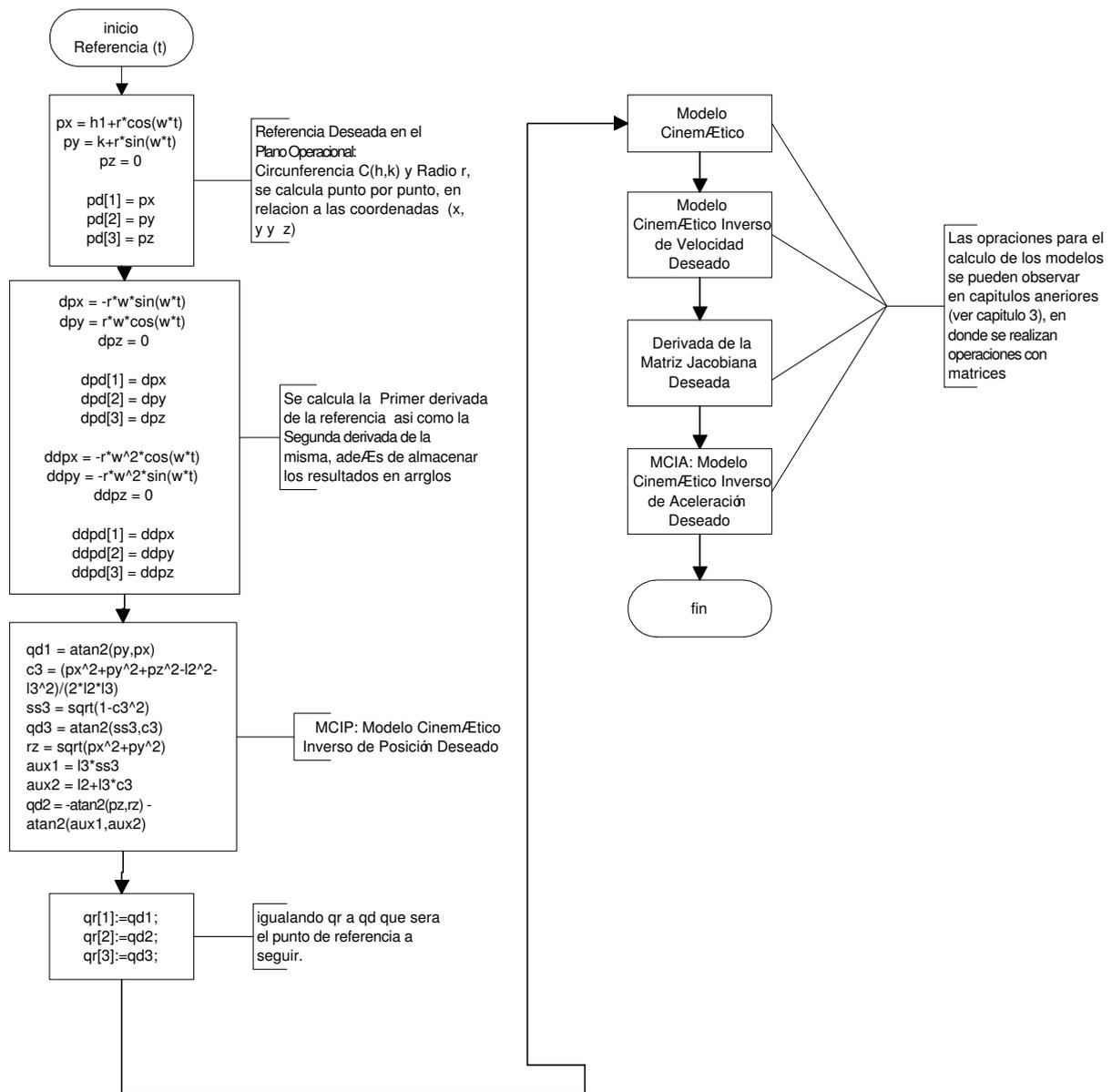


Figura 4.12: Diagrama de flujo del procedimiento referencia.

El diagrama que se observa en la Figura 4.13 es el del procedimiento de control, en el cual se logra observar que se puede elegir un control PIDNL sin TBG y otro con TBG, en este procedimiento se pone en práctica el Control PIDNL propuesto en el Capítulo 3 específicamente en la sección 3.6 y así como la implementación del generador de tiempo base que se propone en el Capítulo 5 específicamente en la sección 5.5.1.

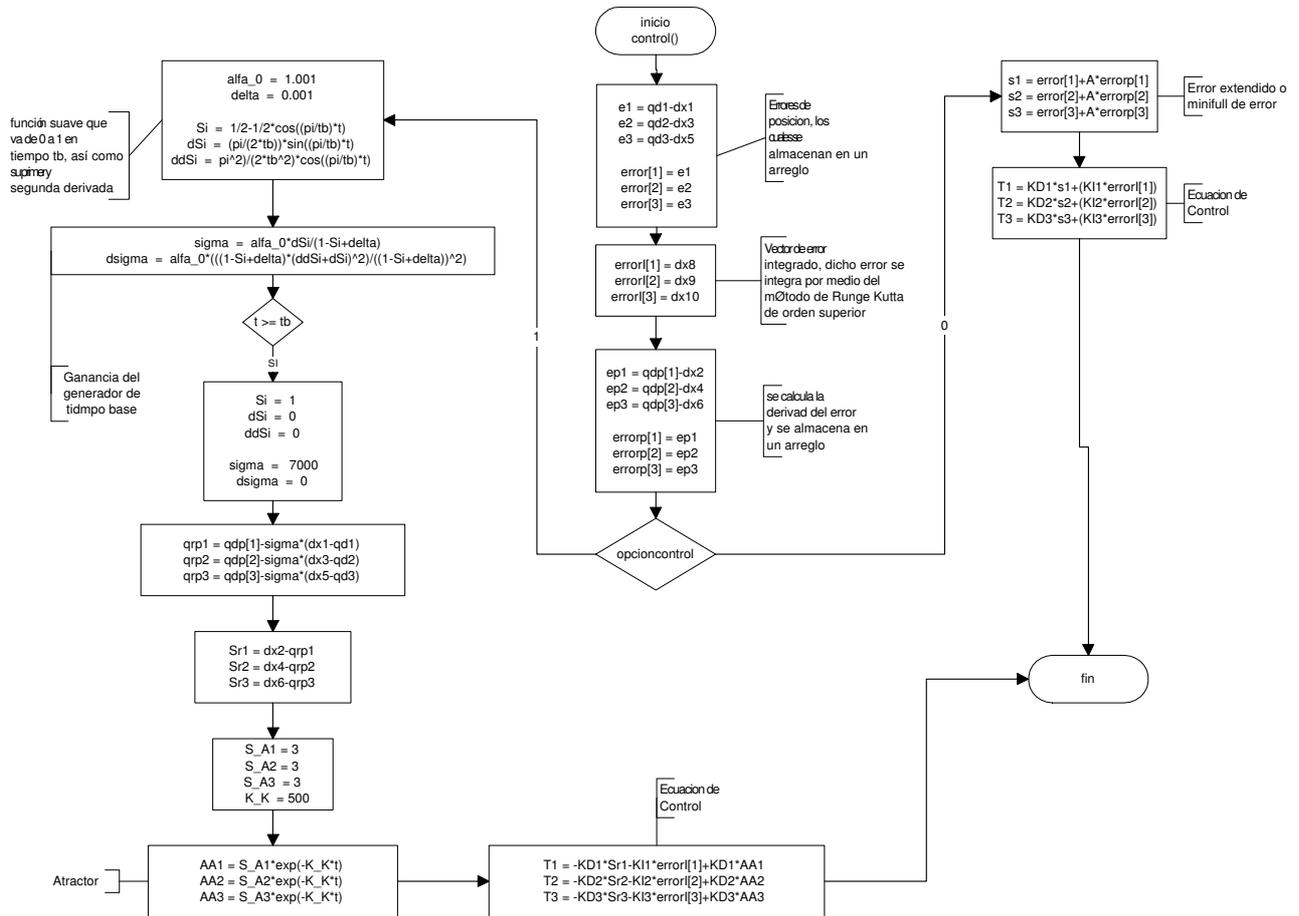


Figura 4.13: Diagrama de flujo del procedimiento de control.

El diagrama del procedimiento rungetpp se muestra en la Figura 4.14, este procedimiento es el encargado de llevar a cabo el método de Runge Kutta de orden superior para llevar a cabo la integración de qpp y qp para así obtener de cada integración anterior qp y q, como se puede ver qp se obtiene de la integración qpp, el valor obtenido de qp se integra nuevamente para obtener q, en los diagramas se muestran las operaciones necesarias para el cálculo del método de Runge Kutta de Orden superior (Figuras 4.14, 4.15, 4.16 y 4.17).

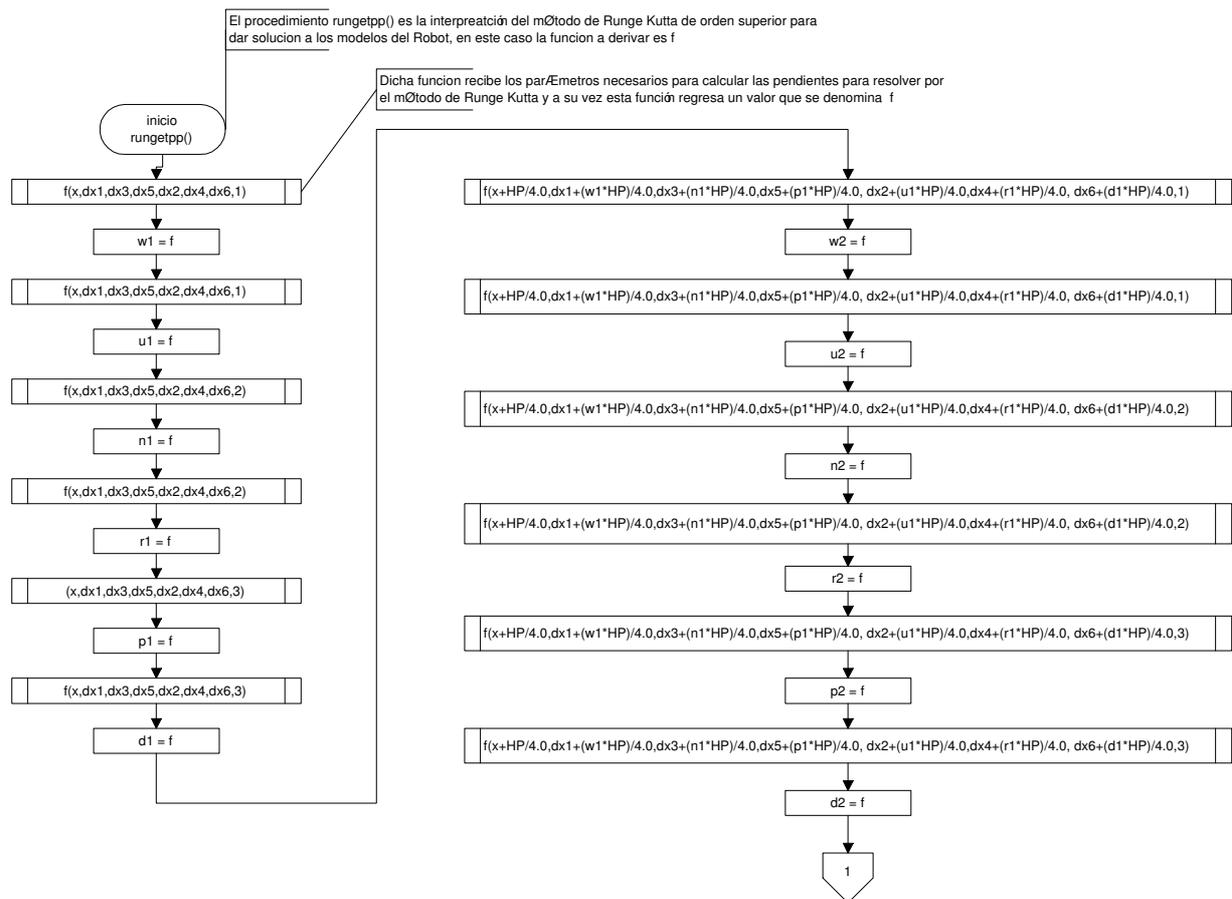


Figura 4.14: Diagrama de flujo del procedimiento rungetpp'.

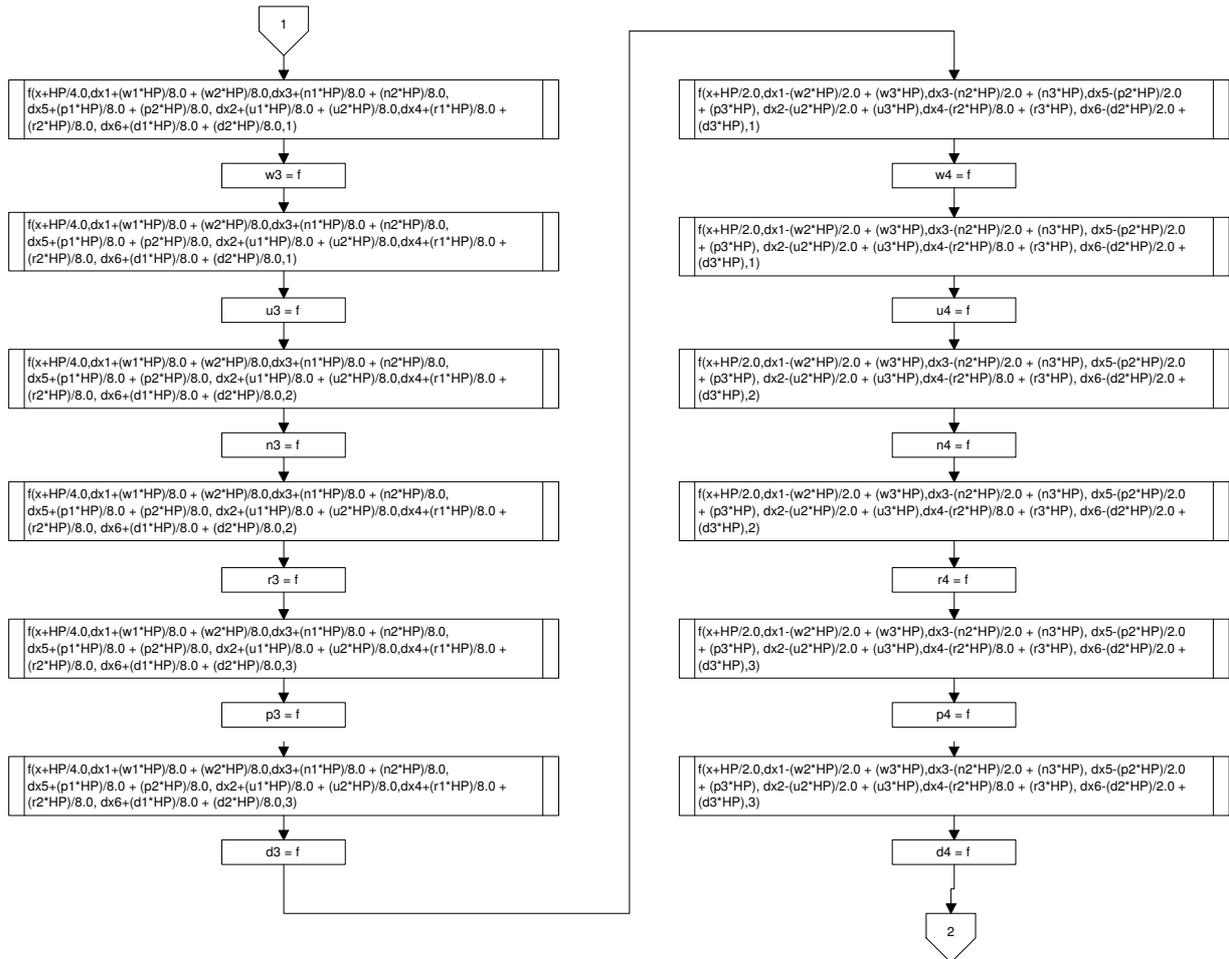


Figura 4.15: Diagrama de flujo del procedimiento rungetpp'.

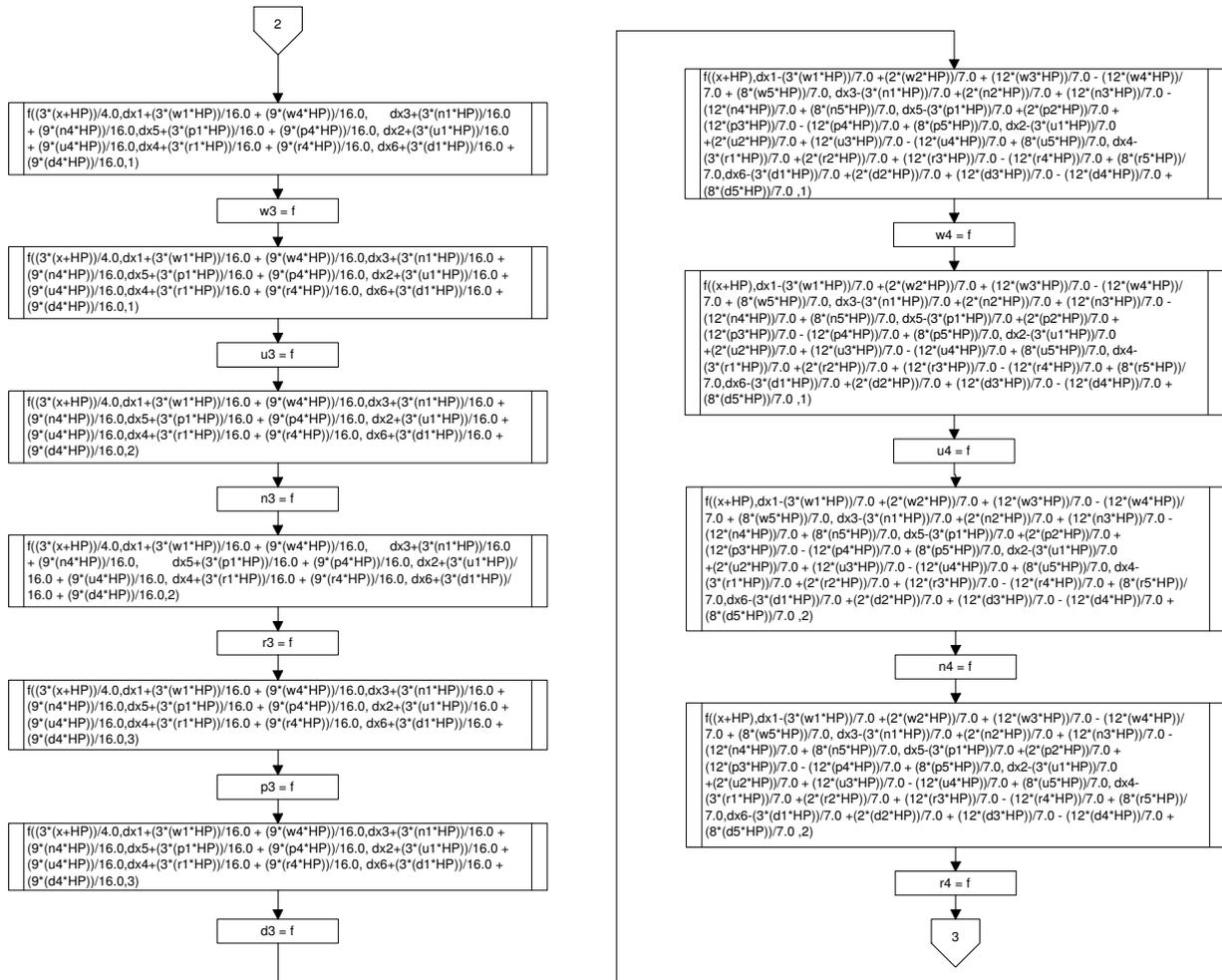


Figura 4.16: Diagrama de flujo del procedimiento rungetpp'.

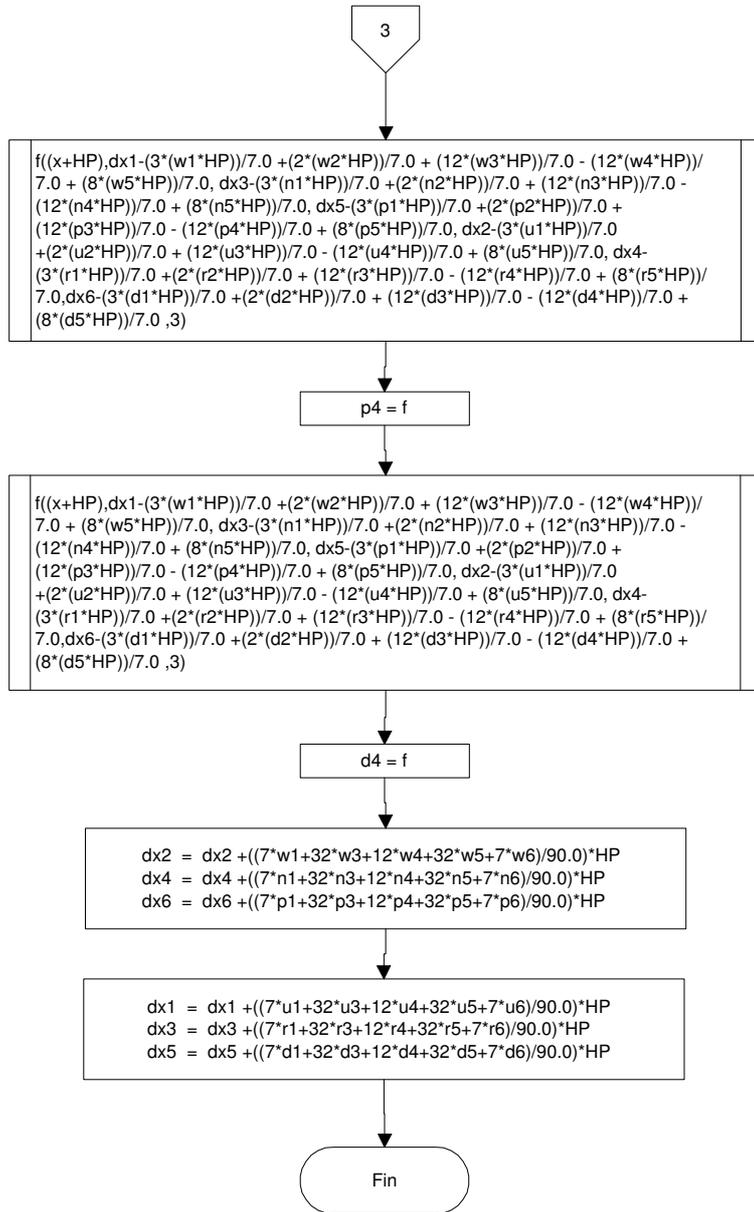


Figura 4.17: Diagrama de flujo del procedimiento rungetpp'.

En los diagramas anteriores se puede notar que el procedimiento rungetpp hace una llamada a una función denominada f, la cual retorna un valor, el diagrama de ésta función se puede observar en la Figura 4.18, en la función se realiza el cálculo de la matriz de inercia, del vector de fuerzas de coriolis y centrípetas, así como del vector de fuerza de gravedad y del vector de fricción seca y viscosa.

El valor que regresa al procedimiento rungetpp pueden ser tres opciones, dependiendo al grado de libertad que nos estemos refiriendo al llamar a la función y este valor es usado en el siguiente llamado a la función f por el procedimiento rungetpp.

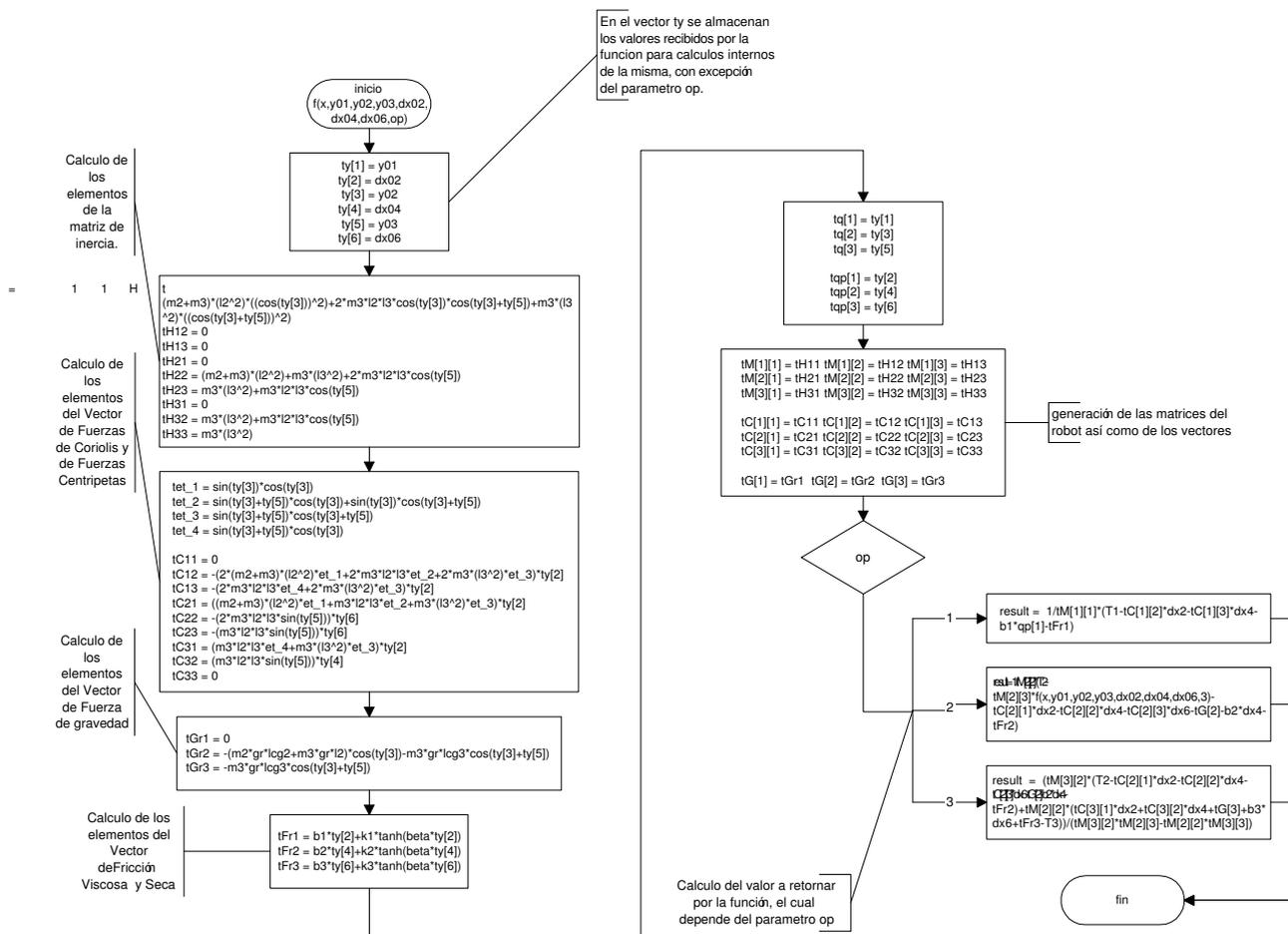


Figura 4.18: Diagrama de flujo de la función f.

Otro procedimiento utilizado es el de rungeerror Figuras 4.19 y 4.20 que es de nuevo la interpretación del método de Runge Kutta de orden superior, pero esta vez es para la integración de una función denominada ei Figura 4.21, la cual regresa un valor al procedimiento rungeerror.

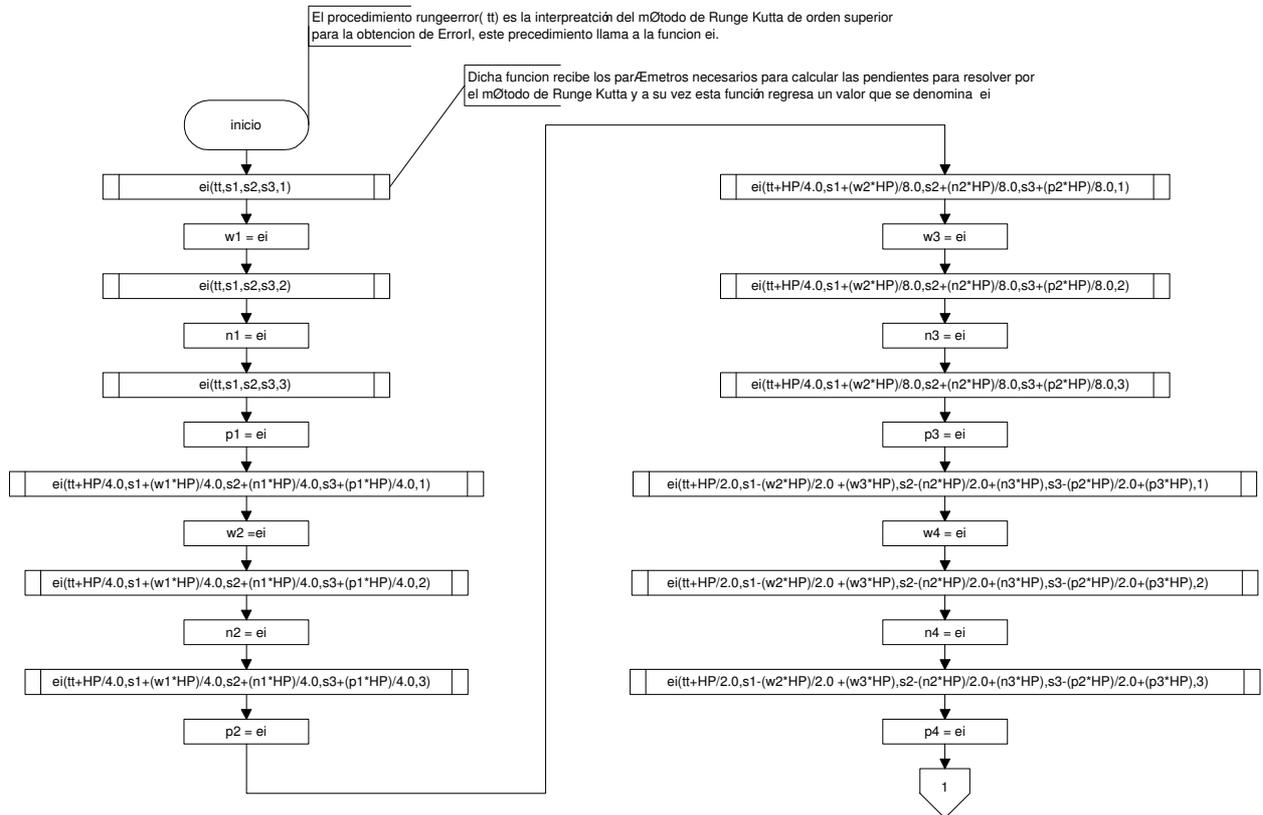


Figura 4.19: Diagrama de flujo procedimiento rungeerror.

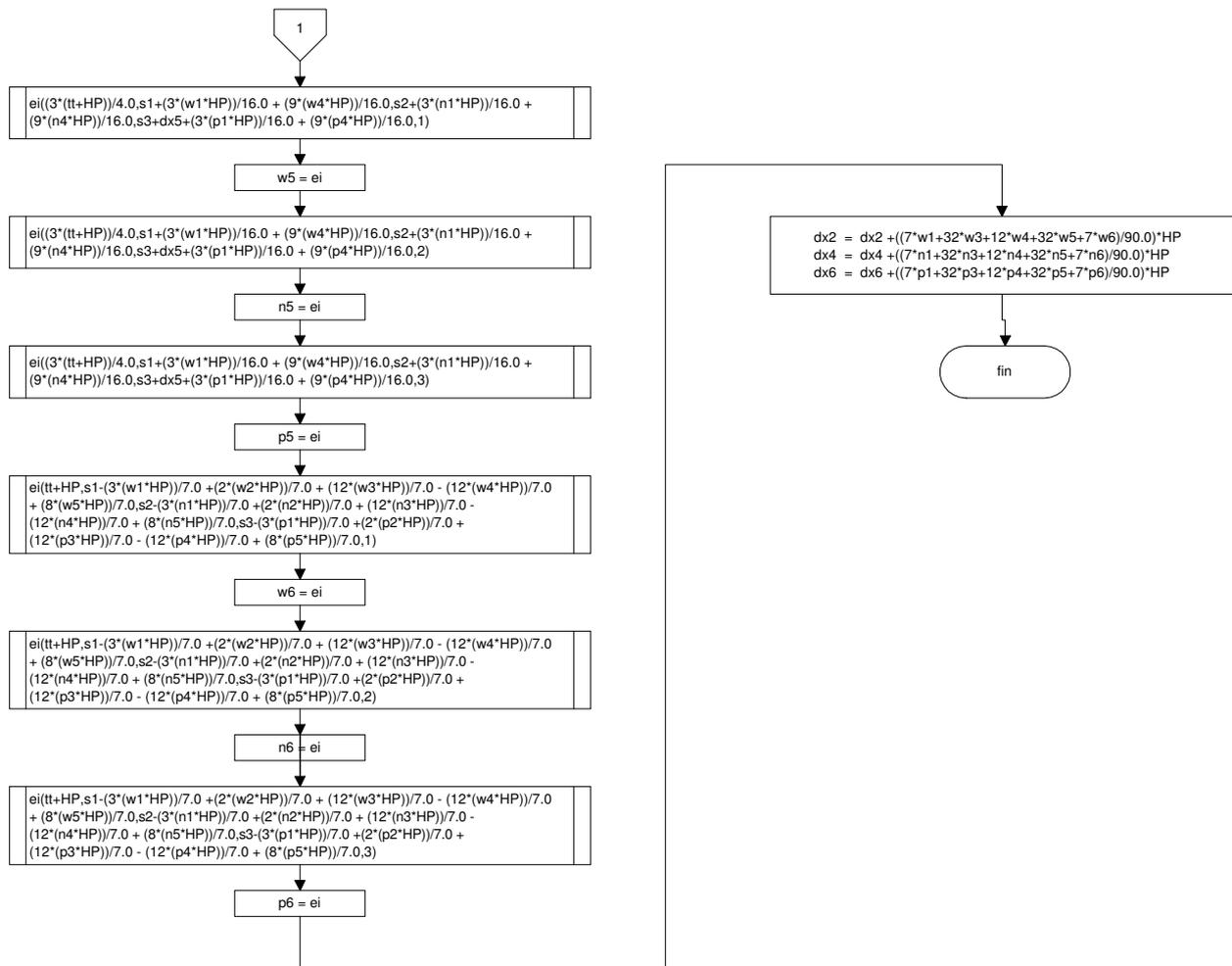


Figura 4.20: Diagrama de flujo procedimiento rungeerror.

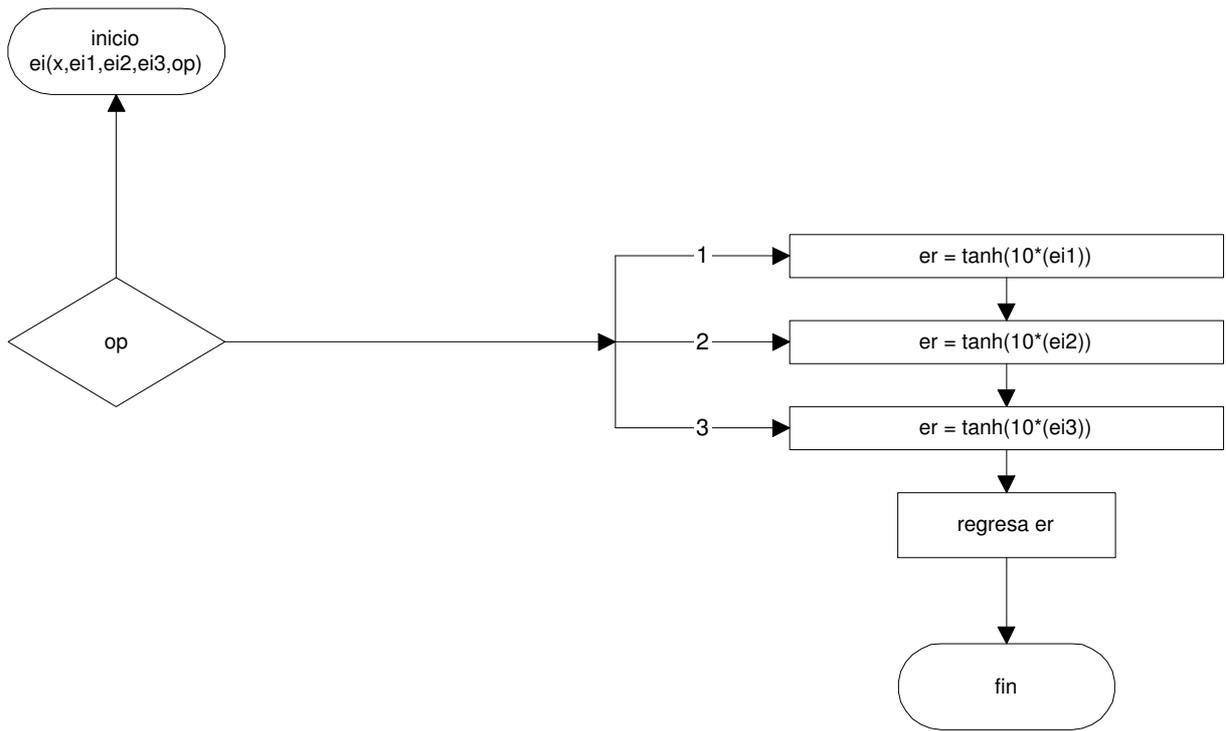


Figura 4.21: Diagrama de flujo de la función ei.

Por último se muestra el procedimiento modelosrobot el cual recibe parámetros que son las variables articulares, calculadas y así actualizar los modelos (dinámico y cinemático) del robot para las siguientes iteraciones, ver Figura 4.22.

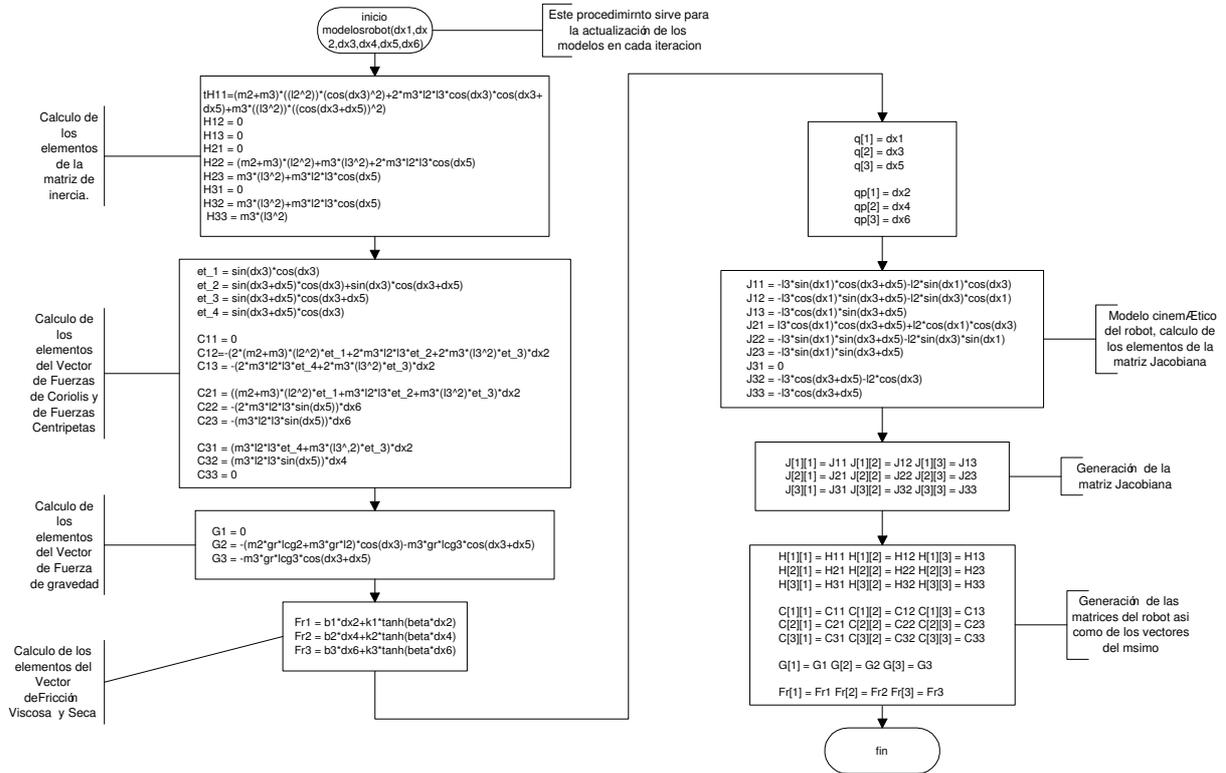


Figura 4.22: Diagrama de flujo del procedimiento modelosrobot.

4.5.2. Diseño.

En esta sección se presenta el diseño de la herramienta propuesta, “Modelado de Sistemas de control de un robot manipulador”, el cual está desarrollado en Delphi 7 bajo la plataforma Windows XP; está formado por varios módulos que fueron desarrollados y probados independientemente antes de desarrollar un solo sistema. Los requerimientos del sistema desarrollado son:

- Sistema operativo win9x o superior.
- Procesador
- Teclado
- Mouse

Nota: El comportamiento de la aplicación al solucionar el control PINL en cualquiera de las dos opciones resueltas por dicha aplicación, en cuanto a la velocidad de respuesta dependerá de la velocidad del procesador del equipo de cómputo sobre el cual se esté trabajando; si se deseara que ésta aplicación se ejecutara sobre un SO

UNIX se tendría que compilar con en Kilyx que es lenguaje Pascal para UNIX como Delphi es para Windows.

En ésta sección se muestra un diagrama de flujo de detalle del funcionamiento de la herramienta propuesta ver Figuras 4.23, 4.24 y 4.25

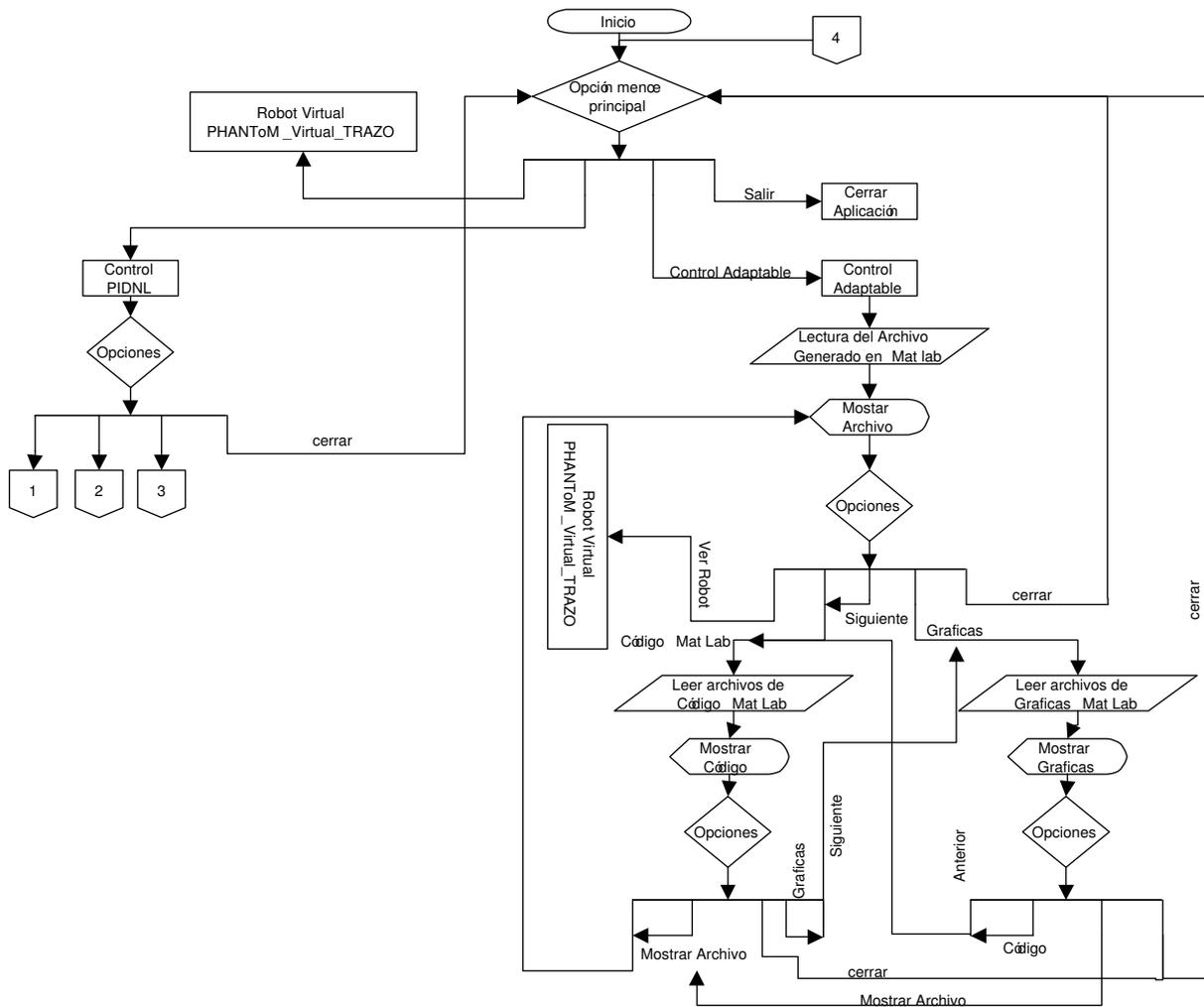


Figura 4.23: Diagrama de flujo del funcionamiento del sistema.

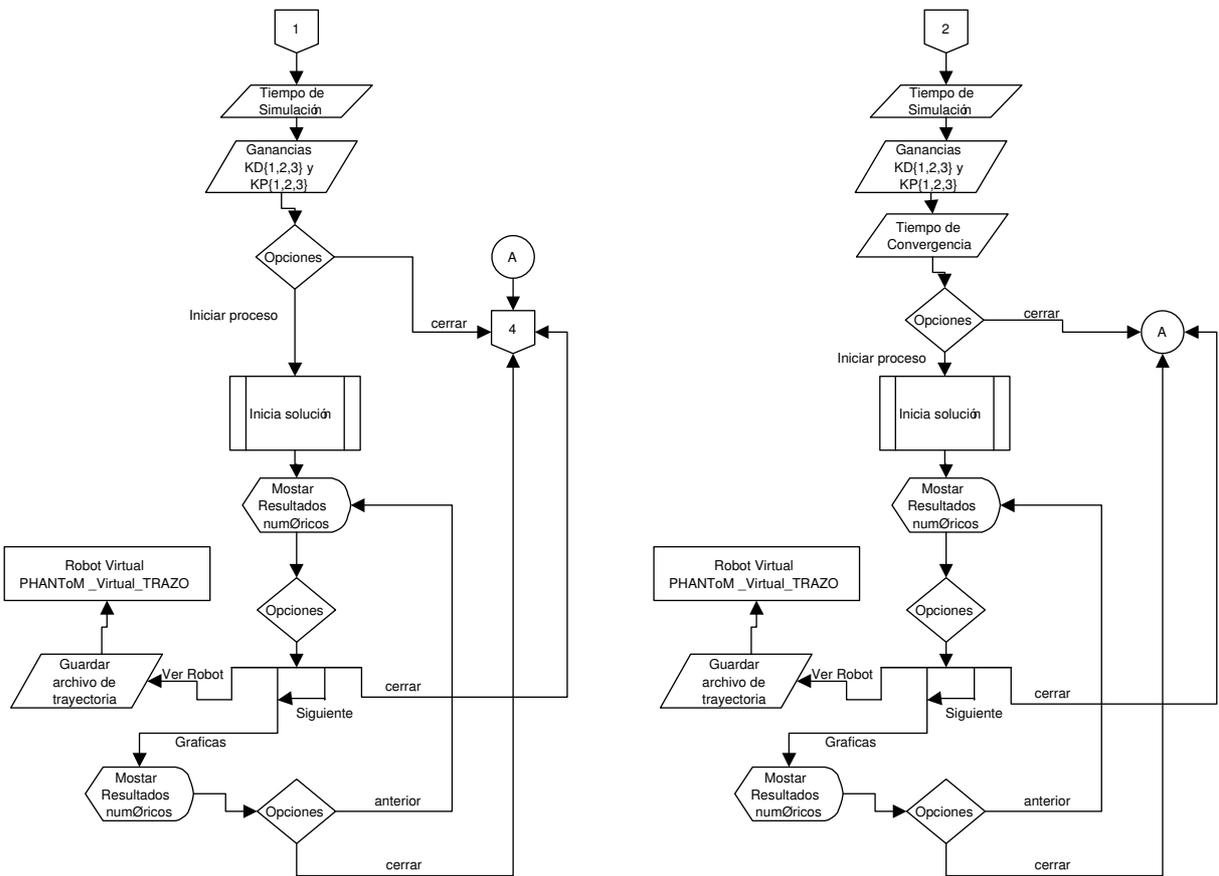


Figura 4.24: Diagrama de flujo del funcionamiento del sistema.

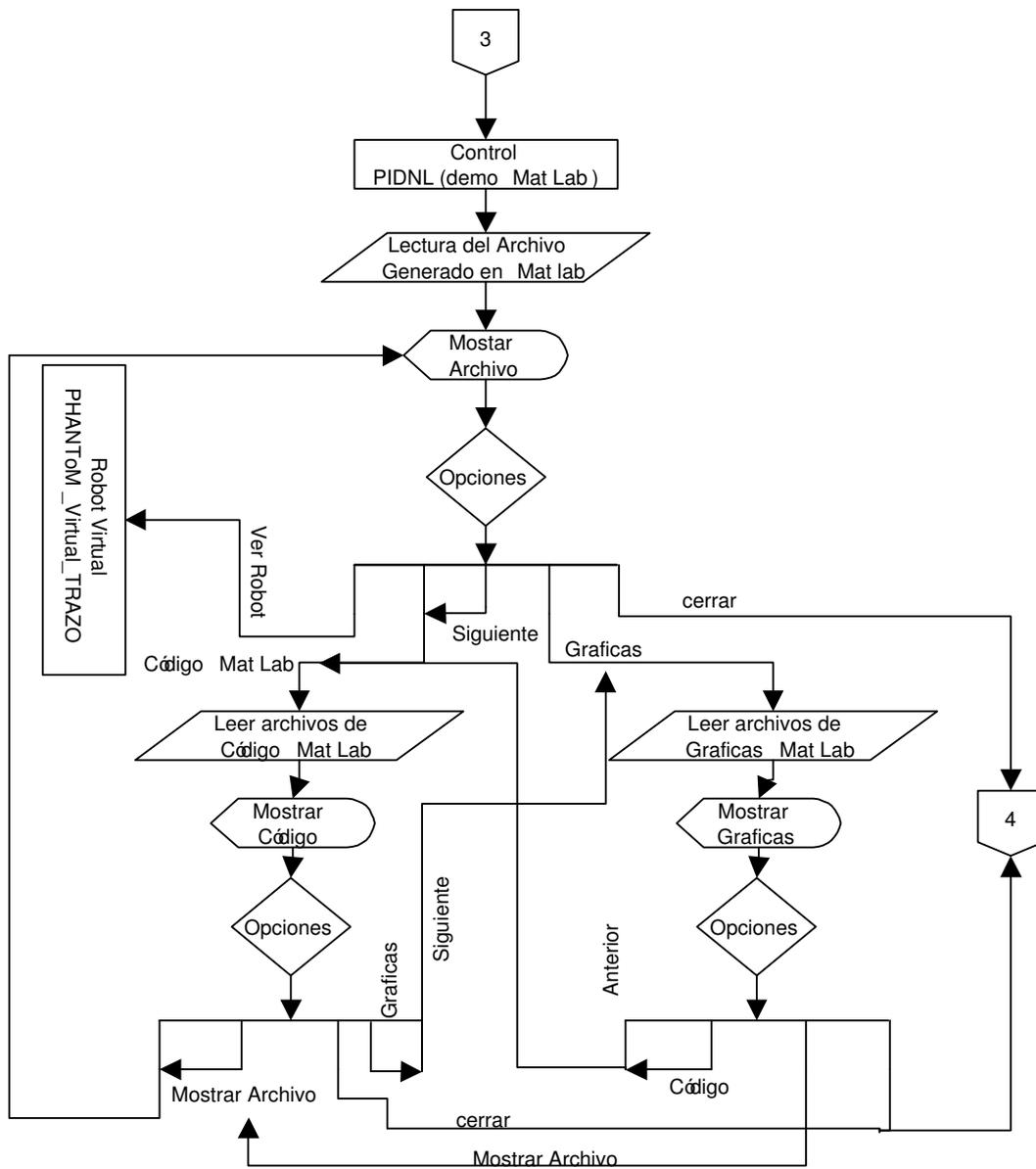


Figura 4.25: Diagrama de flujo del funcionamiento del sistema.

En la aplicación se encuentran los siguientes módulos (ver Figura 4.26):

- Robot virtual.
- Control PIDNL.
- Control adaptable.



Figura 4.26: Menú principal de la herramienta propuesta.



Figura 4.27: Interfaz PHANToM Virtual TRA-ZO.

En el primer módulo se puede realizar la visualización del robot así como interactuar con el (Ver Figura 4.27).

En el segundo módulo se puede realizar la simulación del seguimiento de trayectoria ya mencionada con un control PIDNL con o sin generador de tiempo base (Figuras 4.28 y 4.31 respectivamente), las dos opciones son resueltas en línea por mediante la aplicación del método de Runge Kutta antes mencionado, los valores que la aplicación despliega en referencia a las ganancias y tiempo de simulación y tiempo de convergencia en el caso del control PIDNL con generador de tiempo base, son valores de prueba mediante los cuales éste equipo obtuvo buenos resultados sin suponer que sean los mejores dado que ésta aplicación es para experimentar con el robot, además la herramienta propuesta muestra las gráficas de las coordenadas generalizadas las opciones resueltas en línea Figuras 4.29 referente al control PIDNL y 4.32 referenta al control PIDNL con TBG, este módulo presenta también un “demo” Figura 4.27 el cual realiza la carga de un archivo obtenido por Matlab ver Figura 4.34, ya que también se trabajó con este software en donde se resolvió el mismo modelo matemático implementado en la aplicación; en este demo se muestra el código de MatLab Figura 4.35 así como las gráficas obtenidas en MatLab Figura 4.36.

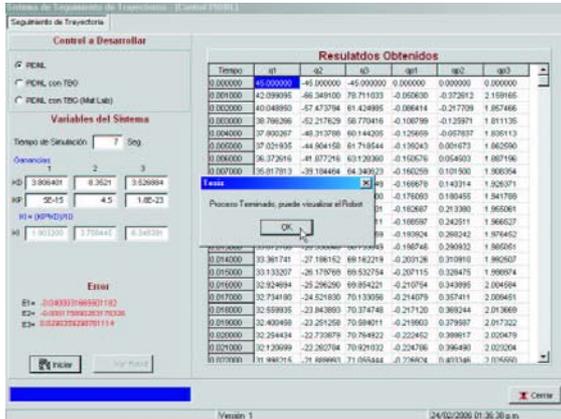


Figura 4.28: Control PIDNL.



Figura 4.29: Resultados en forma gráfica del control PIDNL.



Figura 4.30: Trayectoria realizada por PHANTOM Virtual TRAZO implementando el control PIDNL.

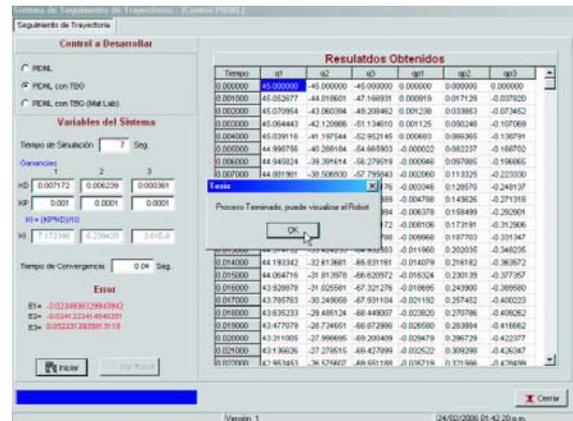


Figura 4.31: Control PIDNL con TBG.



Figura 4.32: Resultados en forma gráfica del control PIDNL con TBG.

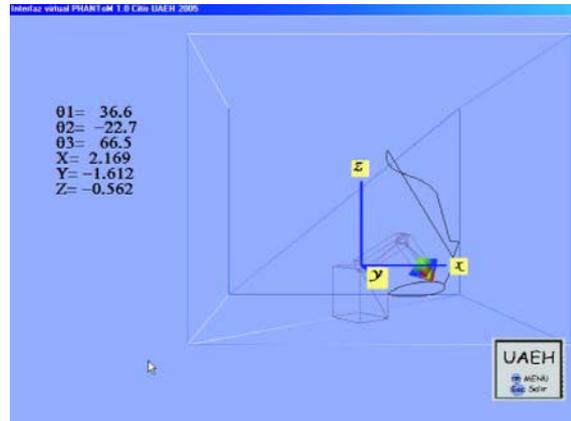


Figura 4.33: Trayectoria realizada por PHANTOM ToM Virtual TRAZO implementando el control PIDNL con TBG.

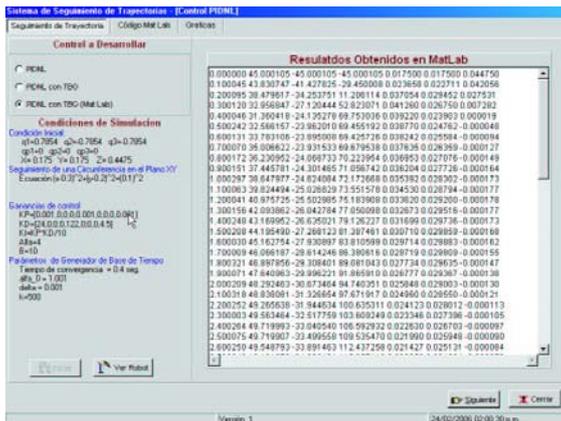


Figura 4.34: Demo control PIDNL con TBG realizado en MatLab.

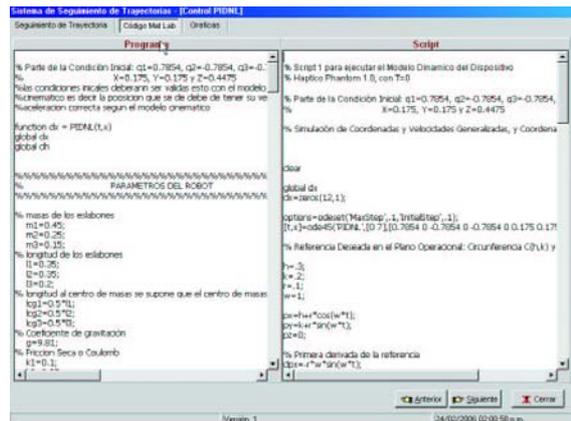


Figura 4.35: Código de MatLab para el control PIDNL con TBG.

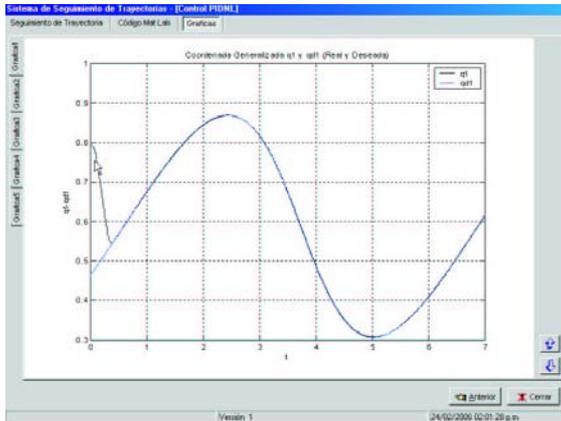


Figura 4.36: Resultados en forma gráfica del control PIDNL con TBG (Demo MatLab).

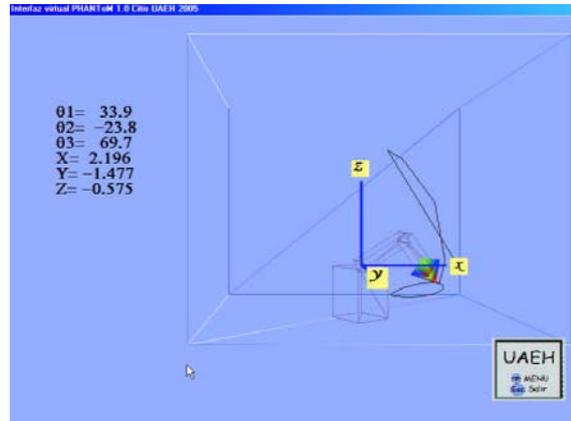


Figura 4.37: Trayectoria realizada por PHANTOM Virtual TRAZO implementando el control PIDNL con TBG.

El tercer módulo presenta un “demo” el cual realiza la carga de un archivo obtenido por Matlab, ya que también se trabajó con este software en donde se resolvió el mismo modelo matemático implementado en la aplicación; en cada una de las opciones se muestran los valores obtenidos numérica y gráficamente.

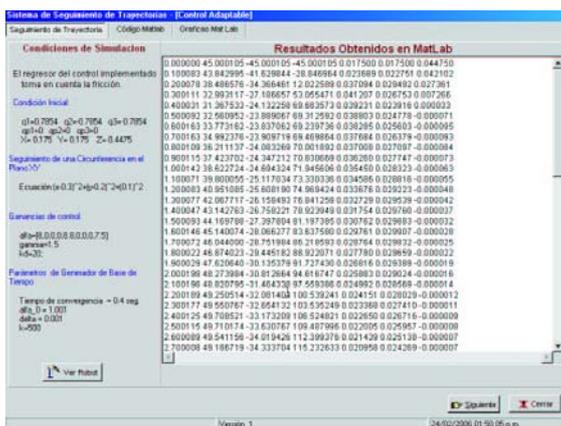


Figura 4.38: Demo control adaptable con TBG realizado en MatLab.

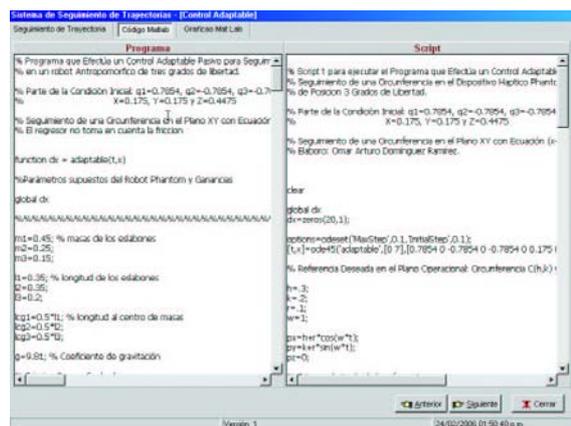


Figura 4.39: Código de MatLab para el control adaptable con TBG.

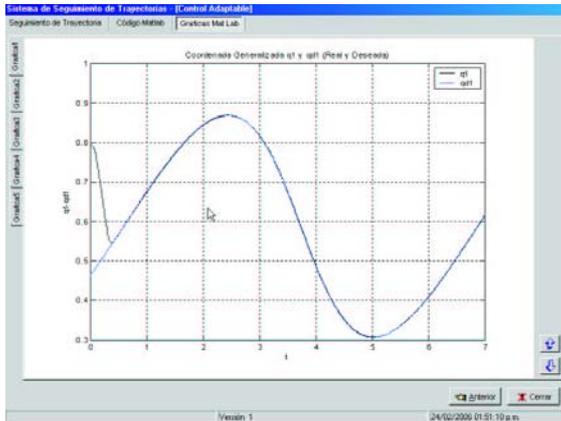


Figura 4.40: Resultados en forma gráfica del control adaptable con TBG (Demo MatLab).



Figura 4.41: Trayectoria realizada por PHANToM Virtual TRAZO implementando el control adaptable con TBG.

4.6. Conclusiones

Hoy en día la realidad virtual es un tema muy común o mejor dicho conocido, pero es necesario conocer todo lo que involucra la realidad virtual es por eso que se presenta una reseña, que incluye conceptos, clasificación y características generales de la realidad virtual. Existen diversas herramientas que se pueden usar para generar un ambiente virtual como los pueden ser los lenguajes y herramientas de edición, como AutoCad, 3D Studio, Maya, VRML, OpenGL, Visual C++, Delphi, entre otros.

Cabe señalar que al inicio de este proyecto la interfaz virtual se creó a partir de 3D Studio Max, se exportó a código Visual C++ con OpenGL y se comenzó a trabajar sobre Visual C++, mientras tanto se liberó la aplicación "PHANToM Virtual TRAZO" desarrollada en el Centro de Investigación en Tecnologías de Información y Sistemas Ingeniería en Electrónica y Telecomunicaciones de esta Universidad, dicha aplicación está creada en lenguaje C++ con OpenGL y es una interfaz virtual que permite la interacción (movimientos y seguimiento de trayectorias) con un robot antropomórfico de tres grados de libertad, lo que se adapta a este trabajo de tesis, se retomo esta aplicación en este capítulo se hace mención a la creación de esta aplicación y del comportamiento complejo de la misma.

Este equipo de trabajo desarrolló una interfaz de usuario que permite el seguimiento de trayectorias (en este caso una circunferencia), que inicialmente se trabajó en Visual C++ 6.0 y se termino en Delphi 7.0, debido a que este lenguaje (object pascal) y software es muy noble en aprendizaje, aplicación y colaboración con otras aplicaciones externas como "PHANToM Virtual TRAZO", además de que puede trabajar con OpenGL de igual forma que Visual C++, si en un futuro se desea incorporar el robot a la aplicación propuesta.

Capítulo 5

Modelado y control del robot antropomórfico virtual

5.1. Introducción

5.2. Modelo cinemático

5.2.1. Parámetros de Denavit-Hartenberg (DH).

Dado que en el Capítulo 2 de este trabajo de tesis se hace un explicación donde se definen los parámetros de Denavit - Hartenberg para un robot antropomórfico de tres grados de libertad Denavit-Hartenberg, se presentan en la siguiente tabla (5.1):

Eslabón i	α	a	d	θ
1	$-\frac{\pi}{2}$	0	0	θ_1
2	0	L_2	0	θ_2
3	0	L_3	0	θ_3

Cuadro 5.1: Parámetros Denavit-Hartenberg [51]

5.2.2. Matrices elementales y de transformación homogénea.

Una matriz de transformación homogénea permite definir la posición y orientación de un marco de referencia respecto al otro.

De ésta forma, para los marcos ortonormales de referencia definidos en la cadena cinemática, es posible determinar una matriz de transformación homogénea general, que defina en función de los parámetros de Denavit-Hartenberg la posición y orientación de un marco de referencia i respecto al marco de referencia $i - 1$.

Una matriz de transformación homogénea con estas características es la siguiente:

$${}^i_{i-1}T = \begin{bmatrix} C\theta_i & -S\theta_i S\alpha_i & S\theta_i S\alpha_i & a_i C\theta_i \\ S\theta_i & C\theta_i C\alpha_i & -C\theta_i C\alpha_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.1)$$

Sustituyendo en esta matriz los parámetros de Denavit-Hartenberg correspondientes a cada eslabón, se obtiene para tres grados de libertad de la estructura de posición del robot, tres matrices de transformación homogénea llamadas matrices elementales del robot.

Por lo tanto las matrices elementales de la estructura de posición del robot de 3 grados de libertad son las matrices 5.2, 5.3 y 5.4:

$${}^1_2T = \begin{bmatrix} C_1 & 0 & -S_1 & 0 \\ S_1 & 0 & C_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.2)$$

$${}^2_3T = \begin{bmatrix} C_2 & -S_2 & 0 & L_2 C_2 \\ S_2 & C_2 & 0 & L_2 S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.3)$$

$${}^3_4T = \begin{bmatrix} C_3 & -S_3 & 0 & L_3 C_3 \\ S_3 & C_3 & 0 & L_3 S_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.4)$$

La matriz de transformación homogénea que describe el marco ortonormal de referencia en el extremo final del último eslabón, respecto al primer marco de referencia y de esta información de posición en coordenadas cartesianas y orientación en ángulos de Euler, está definido realizando el siguiente producto sucesivo de sus matrices elementales como se muestra a continuación:

$${}^1_4T = {}^1_2T \cdot {}^2_3T \cdot {}^3_4T \quad (5.5)$$

Cuyo desarrollo es el siguiente:

Primero obteniendo la matriz 1_3T

$${}^1_3T = {}^1_2T \cdot {}^2_3T \quad (5.6)$$

Donde el producto de las matrices 1_2T y 2_3 es el siguiente:

$${}^1_3T = \begin{bmatrix} C_1 & 0 & -S_1 & 0 \\ S_1 & 0 & C_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} C_2 & -S_2 & 0 & L_2C_2 \\ S_2 & C_2 & 0 & L_2S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} C_1C_2 & -S_2C_1 & -S_1 & L_2C_1C_2 \\ S_1C_2 & -S_1S_2 & C_1 & L_2S_1C_2 \\ -S_2 & -C_2 & 0 & -L_2S_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.7)$$

Por último

$${}^1_4T = {}^1_3T \cdot {}^3_4T \quad (5.8)$$

$${}^1_4T = \begin{bmatrix} C_1C_2 & -S_2C_1 & -S_1 & L_2C_1C_2 \\ S_1C_2 & -S_1S_2 & C_1 & L_2S_1C_2 \\ -S_2 & -C_2 & 0 & -L_2S_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} C_3 & -S_3 & 0 & L_3C_3 \\ S_3 & C_3 & 0 & L_3S_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$${}^1_4T = \begin{bmatrix} C_1C_2C_3 - S_2S_3C_1 & -S_3C_1C_2 - S_2C_1C_3 & -S_1 & L_3C_1C_2C_3 - L_3S_2S_3C_1 - L_2C_1C_2 \\ S_1C_2C_3 - S_1S_2S_3 & -S_1S_3C_2 - S_1S_2C_3 & C_1 & L_3S_1C_2C_3 + L_3S_1S_2S_3 + L_2S_1C_2 \\ -S_2C_3 - S_2S_3C_2 & S_2S_3 - C_2C_3 & 0 & -L_3S_2C_3 - L_3S_3C_2 - L_2S_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Simplificando los elementos de la matriz se tiene la Ecuación 5.9:

$${}^1_4T = \begin{bmatrix} C_1C_{23} & -C_1S_{23} & -S_1 & L_3C_1C_{23} + L_2C_1C_2 \\ S_1C_{23} & -S_1S_{23} & C_1 & L_3S_1C_{23} + L_2S_1C_2 \\ -S_{23} & -C_{23} & 0 & -L_3S_{23} - L_2S_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.9)$$

Donde :

$$\begin{aligned}\cos(\theta_1) &= C_1 \\ \cos(\theta_2 + \theta_3) &= C_{23} \\ \text{sen}(\theta_1) &= S_1 \\ \text{sen}(\theta_2 + \theta_3) &= S_{23}\end{aligned}$$

Las coordenadas de posición cartesiana son[23]:

$$x_p = [x_1, x_2, x_3]^T = [l, m, n]^T \quad (5.10)$$

$$\begin{aligned}l &= E_{14} = x \\ m &= E_{24} = y \\ n &= E_{34} = z\end{aligned}$$

$$x = L_3 C_1 C_{23} + L_2 C_1 C_2 \quad (5.11)$$

$$y = L_3 S_1 C_{23} + L_2 S_1 C_2 \quad (5.12)$$

$$z = -L_3 S_{23} - L_2 S_2 \quad (5.13)$$

Las coordenadas de orientación, en términos de los ángulos de Bryant, son[23]:

$$x_0 = [x_4, x_5, x_6]^T = [\lambda, \mu, \nu]^T \quad (5.14)$$

Los ángulos λ, μ, ν corresponden a los giros sucesivos que necesitaría aplicarse al marco R_i en el orden x - y - z para hacerlo coincidir con el marco R_f .

Donde R_i corresponde a la condición inicial del marco ortonormal en el extremo final del último eslabón y R_f la deseada o final del mismo.

$$\begin{aligned}x_4 = \lambda &= \text{atan2}(-E_{23}, E_{33}) \\ x_5 = \mu &= \text{atan2}(E_{13}, E_{33}/\cos \lambda) \\ x_6 = \nu &= \text{atan2}(-E_{12}, E_{11})\end{aligned} \quad (5.15)$$

Donde E_{ij} corresponden a los elementos $i - j$ de la matriz de transformación homogénea 1_4T .

Finalmente, se concluye que las coordenadas operacionales, en coordenadas cartesianas para posición y ángulos de Bryant para orientación, constituyen w_l modelo cinemático de posición.

5.2.3. Modelo cinématico directo de posición (MCDP)

El MCDP de un robot manipulador es la relación que permite determinar las coordenadas operacionales del robot en función de las variables articulares o coordenadas generalizadas que del mismo.

$$x = f(q) \quad (5.16)$$

La obtención del MCDP, se obtiene a continuación:

$$x = [x_1, x_2, x_3, x_4, x_5, x_6]^T \quad (5.17)$$

$$x = [x, y, z, \lambda, \mu, v]^T \quad (5.18)$$

Si

$$\lambda = \text{atan2}(-C_1, 0) = \text{atan}(-C_1, 0) = 270^\circ (3\pi/2 \text{rad}) \quad (5.19)$$

$$\mu = \text{atan2}(-S_1, 0/\cos 270^\circ) = 270^\circ (3\pi/2 \text{rad}) \quad (5.20)$$

$$v = \text{atan2}(-(-C_1), S_{23}), C_1 C_{23} = \text{tg}^{-1}(S_{23}/C_{23}) = \theta_2 + \theta_3 \quad (5.21)$$

Por lo tanto

$$x = \begin{bmatrix} x \\ y \\ z \\ \lambda \\ \mu \\ v \end{bmatrix} = \begin{bmatrix} L_3 C_1 C_{23} + L_2 C_1 C_2 \\ L_3 S_1 C_{23} + L_2 S_1 C_2 \\ -L_3 S_{23} - L_2 S_2 \\ 270^\circ \\ 270^\circ \\ \theta_2 + \theta_3 \end{bmatrix} \quad (5.22)$$

que define finalmente, el MCDP.

Entonces x, y, z, λ, μ, v son de la siguiente manera:

$$x = L_3 C_1 C_{23} + L_2 C_1 C_2 \quad (5.23)$$

$$y = L_3 S_1 C_{23} + L_2 S_1 C_2 \quad (5.24)$$

$$z = -L_3 S_{23} - L_2 S_2 \quad (5.25)$$

$$\lambda = 270^\circ \quad (5.26)$$

$$\mu = 270^\circ \quad (5.27)$$

$$v = \theta_2 + \theta_3 \quad (5.28)$$

5.2.4. Modelo cinemático inverso de posición (MCIP)

El MCIP de un robot manipulador es la función inversa si esta existe, que permite obtener las coordenadas generalizadas o variables articulares (ángulos entre eslabones, para el caso de estudio) en términos de las coordenadas operacionales. En otras palabras conocer los ángulos articulares que el robot debe adquirir para lograr una situación deseada del órgano terminal (en posición y orientación) en el plano o espacio de trabajo.

$$q = f^{-1}(x) \quad (5.29)$$

La obtención del MCIP, se proporciona a continuación:

dato que:

$${}^1_4T = {}^1_2T \cdot {}^2_3T \cdot {}^3_4T \quad (5.30)$$

es posible que:

$${}^1_2T^{-1} \cdot {}^1_4T = {}^2_3T \cdot {}^3_4T \quad (5.31)$$

Obteniendo la matriz inversa de 1_2T (Ecuación 5.2)

Si

$${}^1_2T = \begin{bmatrix} C_1 & 0 & -S_1 & 0 \\ S_1 & 0 & C_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.32)$$

$${}^1_2T^{-1} = \begin{bmatrix} C_1 & S_1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -S_1 & C_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.33)$$

Para la matriz de transformación homogénea 1_4T se consideran elementos E_{ij} , tal que[51]:

$${}^1_4T = \begin{bmatrix} E_{11} & E_{12} & E_{13} & E_{14} \\ E_{21} & E_{22} & E_{23} & E_{24} \\ E_{31} & E_{32} & E_{33} & E_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.34)$$

Y sus elementos son:

$$\begin{aligned} E_{11} &= C_1 C_{23} \\ E_{12} &= -C_1 S_{23} \\ E_{13} &= -S_1 \\ E_{14} &= L_3 C_1 C_{23} + L_2 C_1 C_2 \\ E_{21} &= S_1 C_{23} \\ E_{22} &= -S_1 S_{23} \\ E_{23} &= C_1 \\ E_{24} &= L_3 S_1 C_{23} + L_2 S_1 C_2 \\ E_{31} &= -S_{23} \\ E_{32} &= -C_{23} \\ E_{33} &= 0 \\ E_{34} &= -L_3 S_{23} - L_2 S_2 \end{aligned} \quad (5.35)$$

Por lo tanto la multiplicación de las matrices ${}^1_2T^{-1} \cdot {}^1_4T$ (Ecuaciones 5.33, 5.34) es como a continuación se presenta:

$${}^1_2T^{-1} * {}^1_4T = \begin{bmatrix} C_1 & S_1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -S_1 & C_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} E_{11} & E_{12} & E_{13} & E_{14} \\ E_{21} & E_{22} & E_{23} & E_{24} \\ E_{31} & E_{32} & E_{33} & E_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^1_2T^{-1} \cdot {}^1_4T = \begin{bmatrix} E_{11}C_1 + E_{21}S_1 & E_{12}C_1 + E_{22}S_1 & E_{13}C_1 + E_{23}S_1 & E_{14}C_1 + E_{24}S_1 \\ -E_{31} & -E_{32} & -E_{33} & -E_{34} \\ -E_{11}S_1 + E_{21}C_1 & -E_{12}S_1 + E_{22}C_1 & -E_{13}S_1 + E_{23}C_1 & -E_{14}S_1 + E_{24}C_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.36)$$

A continuación se realiza el producto de las matrices 2_3T y 3_4T (ecuaciones 5.3, 5.4) que dan como resultado a la matriz 2_4T

$$\begin{aligned}
 {}^2_3T \cdot {}^3_4T &= \begin{bmatrix} C_2 & -S_2 & 0 & L_2C_2 \\ S_2 & C_2 & 0 & L_2S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} C_3 & -S_3 & 0 & L_3C_3 \\ S_3 & C_3 & 0 & L_3S_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\
 {}^2_4T &= \begin{bmatrix} C_{23} & -S_{23} & 0 & L_3C_{23} + L_2C_2 \\ S_{23} & C_{23} & 0 & L_3S_{23} + L_2S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.37}
 \end{aligned}$$

Efectuando una comparación entre elementos de la matriz 2_4T con los elementos de la matriz resultante ${}^1_2T^{-1} * {}^1_4T$ y observando la factibilidad de hacerlo con el elemento (3,4), tenemos:

$$-E_{14}S_1 + E_{24}C_1 = 0 \tag{5.38}$$

De acuerdo al MCDP

$$-xS_1 + yC_1 = 0 \tag{5.39}$$

Despejando

$$\frac{y}{x} = \frac{S_1}{C_1} \tag{5.40}$$

Por lo tanto

$$\tan \theta_1 = \frac{y}{x} \tag{5.41}$$

Obteniendo la expresión que resuelve la primera variable articular (Ecuación 5.42)

$$\theta_1 = \text{Atan2}(y, x) \tag{5.42}$$

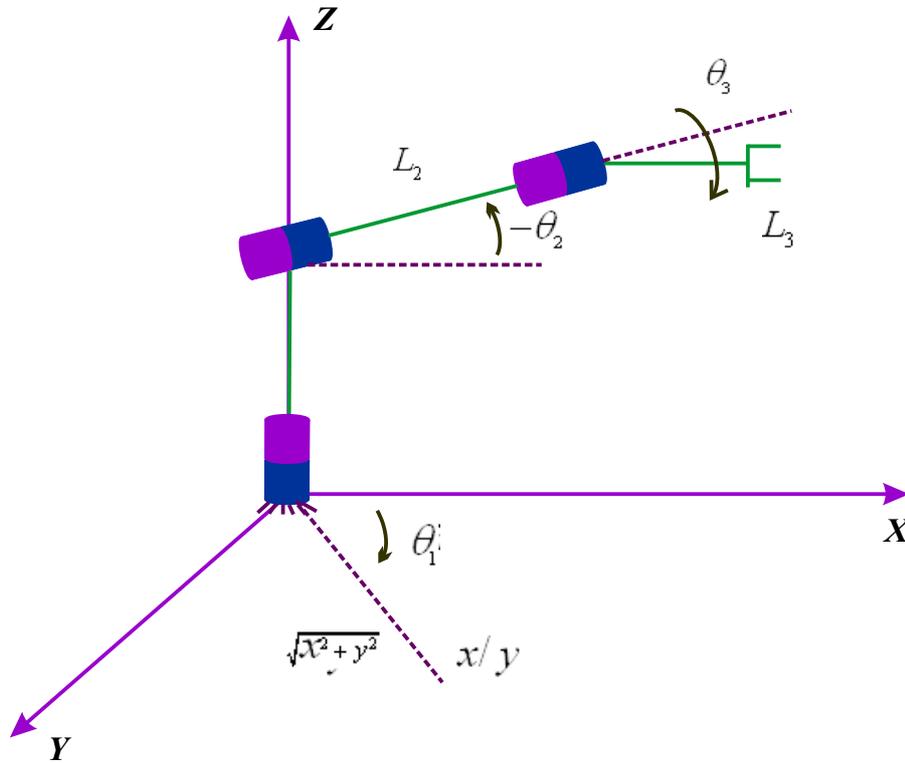


Figura 5.1: Cadena cinemática sintética

Haciendo referencia a la cadena cinemática sintética siguiente Figura 5.1:

De la Figura 5.1, analizando el plano z-y/x tenemos tal como se muestra en la Figura 5.2

De la ley de cosenos, aplicando al triángulo que forman los puntos ΔABC se tiene que:

$$\cos(180^\circ - \theta_3) = -\cos \theta_3 = \frac{L_2^2 + L_3^2 - (x^2 + y^2 + z^2)}{2L_2L_3} \quad (5.43)$$

Por lo tanto

$$\cos \theta_3 = \frac{x^2 + y^2 + z^2 - L_2^2 - L_3^2}{2L_2L_3} \quad (5.44)$$

Aplicando identidades trigonométricas

$$\text{sen} \theta_3 = \sqrt{1 - \cos^2 \theta_3} \quad (5.45)$$

$$\theta_3 = \text{atan2}(\text{sen} \theta_3, \cos \theta_3) \quad (5.46)$$

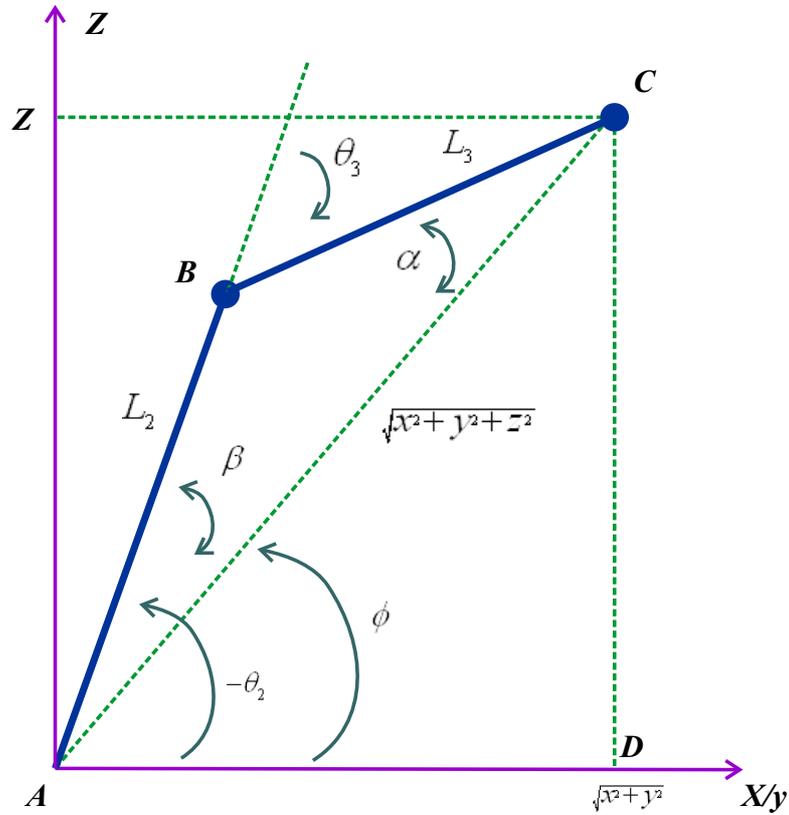


Figura 5.2: Plano z-x/y

Y se obtiene:

$$\theta_3 = \text{atan2}(\sqrt{1 - \cos^2 \theta_3}, \cos \theta_3)$$

Y

$$\cos \theta_3 = \frac{x^2 + y^2 + z^2 - L_2^2 - L_3^2}{2L_2L_3}$$

Y de la misma Figura 5.2 se obtiene también lo siguiente:

$$-\theta_2 = \phi + \beta \tag{5.47}$$

$$\theta_2 = -\phi - \beta$$

Considerando el triángulo rectángulo ΔACD

$$\phi = \text{atan2}(z, \sqrt{x^2 + y^2}) \tag{5.48}$$

Por la ley de cosenos para el triángulo ΔABC

$$\frac{\text{sen}\beta}{L_3} = \frac{\text{sen}\alpha}{L_2} \quad (5.49)$$

$$\alpha = 180^\circ - \beta - (180^\circ - \theta_3) = -\beta + \theta_3 \quad (5.50)$$

$$\alpha = \theta_3 - \beta$$

Por lo tanto

$$\frac{\text{sen}\beta}{L_3} = \frac{\text{sen}(\theta_3 - \beta)}{L_2}$$

$$\frac{\text{sen}\beta}{L_3} = \frac{\text{sen}\theta_3 \cos \beta - \text{sen}\beta \cos \theta_3}{L_2}$$

$$\frac{L_2}{L_3} = \frac{\text{sen}\theta_3 \text{ctg}\beta - \cos \theta_3}{L_2}$$

$$\frac{L_2 + L_3 \cos \theta_3}{L_3} = \text{sen}\theta_3 \text{ctg}\beta$$

$$\tan \beta = \frac{L_3 \text{sen}\theta_3}{L_2 + L_3 \cos \theta_3}$$

$$\beta = \text{atan2}(L_3 \text{sen}\theta_3, L_2 + L_3 \cos \theta_3) \quad (5.51)$$

Y bien entonces obtenemos:

$$\theta_2 = -\phi - \beta = -\text{atan2}(z, \sqrt{x^2 + y^2}) - \text{atan2}(L_3 \text{sen}\theta_3, L_2 + L_3 \cos \theta_3)$$

Por consiguiente el MCIP de la estructura de posición del robot es la siguiente:

Para θ_1

$$\theta_1 = \text{atan2}(x, y) \quad (5.52)$$

Para θ_2

$$\theta_2 = -\text{atan2}(z, \sqrt{x^2 + y^2}) - \text{atan2}(L_3 \text{sen}\theta_3, L_2 + L_3 \cos \theta_3) \quad (5.53)$$

Para θ_3

$$\theta_3 = \text{atan2}(\sqrt{1 - \cos^2 \theta_3}, \cos \theta_3) \quad (5.54)$$

$$\cos \theta_3 = \frac{x^2 + y^2 + z^2 - L_2^2 - L_3^2}{2L_2L_3} \quad (5.55)$$

5.2.5. Modelo cinemático directo de velocidad (MCDV)

Dada la expresión que identifica al MCDP (Ecuación 5.16)

$$x = f(q)$$

Aplicando su primera derivada

$$\dot{x} = \frac{d}{dt}f(q) \quad (5.56)$$

Y de esta forma obtenemos el MCDV

$$\dot{x} = J\dot{q} \quad (5.57)$$

Donde:

\dot{x} : vector de velocidad del órgano terminal del robot

$$\dot{x} = [\dot{x}, \dot{y}, \dot{z}, \dot{\lambda}, \dot{\mu}, \dot{v}]^T \quad (5.58)$$

\dot{q} : vector de velocidad de las articulaciones del robot

$$\dot{q} = [\dot{q}_1, \dot{q}_2, \dot{q}_3]^T \quad (5.59)$$

J : matriz jacobiana del robot, cuyos elementos son J_{ij}

$$J_{ij} = \frac{\partial f_i}{\partial q_j} \quad (5.60)$$

Dado el MCDP (Ecuaciones: 5.23, 5.24, 5.25, 5.26, 5.27, 5.28) derivando con respecto al tiempo, se tiene lo siguiente:

$$x = L_3C_1C_{23} + L_2C_1C_2$$

$$y = L_3S_1C_{23} + L_2S_1C_2$$

$$z = -L_3S_{23} - L_2S_2$$

$$\lambda = 270^\circ$$

$$\mu = 270^\circ$$

$$v = \theta_2 + \theta_3$$

Sus derivadas son:

Para \dot{x} :

$$\begin{aligned} \frac{d}{dt}(L_3 C_1 C_{23} + L_2 C_1 C_2) &= [C_1(-S_{23}(\dot{\theta}_2 + \dot{\theta}_3)) + C_{23}(-S_1 \dot{\theta}_1)] + L_2[C_1(-S_2)\dot{\theta}_2 + C_2(-S_1 \dot{\theta}_1)] \\ &= -L_3 C_1 S_{23} \dot{\theta}_2 - L_3 C_1 S_{23} \dot{\theta}_3 - L_3 C_{23} S_1 \dot{\theta}_1 - L_2 S_1 C_1 \dot{\theta}_2 - L_2 S_1 C_2 \dot{\theta}_1 \\ \dot{x} &= -\{L_3 S_1 C_{23} + L_2 S_1 C_2\} \dot{\theta}_1 - \{L_3 S_{23} C_1 + L_2 S_2 C_1\} \dot{\theta}_2 - \{L_3 S_{23} C_1\} \dot{\theta}_3 \end{aligned} \quad (5.61)$$

Para \dot{y} :

$$\begin{aligned} \frac{d}{dt}(L_3 S_1 C_{23} + L_2 S_1 C_2) &= L_3[S_1(-S_{23}(\dot{\theta}_2 + \dot{\theta}_3)) + C_{23} C_1 \dot{\theta}_1] + L_2[S_1 S_2 \dot{\theta}_2 + C_2 C_1 \dot{\theta}_1] \\ &= -L_3 S_1 S_{23} \dot{\theta}_2 - L_3 S_1 S_{23} \dot{\theta}_3 + L_3 C_1 C_{23} \dot{\theta}_1 - L_2 S_1 S_2 \dot{\theta}_2 - L_2 C_1 C_2 \dot{\theta}_1 \\ \dot{y} &= -\{L_3 C_1 C_{23} + L_2 C_1 C_2\} \dot{\theta}_1 - \{L_3 S_1 S_{23} + L_2 S_1 S_2\} \dot{\theta}_2 - \{L_3 S_1 S_{23}\} \dot{\theta}_3 \end{aligned} \quad (5.62)$$

Para \dot{z} :

$$\begin{aligned} \frac{d}{dt}(-L_3 S_{23} - L_2 S_2) &= -L_3 C_{23}(\dot{\theta}_2 + \dot{\theta}_3) - L_2 C_2 \dot{\theta}_2 \\ \dot{z} &= -\{L_3 C_{23} + L_2 C_2\} \dot{\theta}_2 - \{L_3 C_{23}\} \dot{\theta}_3 \end{aligned} \quad (5.63)$$

Para $\dot{\lambda}$ y $\dot{\mu}$:

$$\dot{\lambda} = 0 \quad (5.64)$$

$$\dot{\mu} = 0 \quad (5.65)$$

Para \dot{v}

$$\begin{aligned} \frac{d}{dt}(\theta_2 + \theta_3) &= \dot{\theta}_2 + \dot{\theta}_3 \\ \dot{v} &= \dot{\theta}_2 + \dot{\theta}_3 \end{aligned} \quad (5.66)$$

Dado que el modelo cinemático directo de velocidad puede ser representado en términos de una matriz jacobiana.

$$\dot{x} = J\dot{q}$$

Se calcula de la siguiente manera:

$$J = \begin{bmatrix} \frac{\partial x}{\partial q_1} & \frac{\partial x}{\partial q_2} & \frac{\partial x}{\partial q_3} \\ \frac{\partial y}{\partial q_1} & \frac{\partial y}{\partial q_2} & \frac{\partial y}{\partial q_3} \\ \frac{\partial z}{\partial q_1} & \frac{\partial z}{\partial q_2} & \frac{\partial z}{\partial q_3} \\ \frac{\partial \lambda}{\partial q_1} & \frac{\partial \lambda}{\partial q_2} & \frac{\partial \lambda}{\partial q_3} \\ \frac{\partial \mu}{\partial q_1} & \frac{\partial \mu}{\partial q_2} & \frac{\partial \mu}{\partial q_3} \\ \frac{\partial v}{\partial q_1} & \frac{\partial v}{\partial q_2} & \frac{\partial v}{\partial q_3} \end{bmatrix} \quad (5.67)$$

Por lo tanto, la matriz jacobiana es la siguiente:

$$J = \begin{bmatrix} -L_3 S_1 C_{23} - L_2 S_1 C_2 & -L_3 S_{23} C_1 - L_2 S_2 C_1 & -L_3 S_{23} C_1 \\ L_3 C_1 C_{23} + L_2 C_1 C_2 & -L_3 S_1 S_{23} - L_2 S_1 S_2 & -L_3 S_1 S_{23} \\ 0 & -L_3 C_{23} - L_2 C_2 & -L_3 C_{23} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} \quad (5.68)$$

5.2.6. Modelo cinemático inverso de velocidad (MCIV)

A partir del MCDV se define la ecuación representativa del MCIV, tal como se presenta a continuación:

$$\dot{q} = J^{-1}\dot{x} \quad (5.69)$$

Dado el jacobiano a su representación reducida (Ecuación 5.68)

$$J_* = \begin{bmatrix} -L_3 S_1 C_{23} - L_2 S_1 C_2 & -L_3 S_{23} C_1 - L_2 S_2 C_1 & -L_3 S_{23} C_1 \\ L_3 C_1 C_{23} + L_2 C_1 C_2 & -L_3 S_1 S_{23} - L_2 S_1 S_2 & -L_3 S_1 S_{23} \\ 0 & -L_3 C_{23} - L_2 C_2 & -L_3 C_{23} \end{bmatrix} \quad (5.70)$$

La matriz inversa del jacobiano reducido es:

$$J_*^{-1} = \begin{bmatrix} -\frac{S_1}{L_2 C_2 + L_3 C_{23}} & \frac{C_1}{L_2 C_2 + L_3 C_{23}} & 0 \\ \frac{C_1 C_{23}}{L_2 S_3} & \frac{S_1 C_{23}}{L_2 S_3} & -\frac{S_{23}}{L_2 S_3} \\ -\frac{L_2 C_1 C_2 + L_3 C_1 C_{23}}{L_2 L_3 S_3} & -\frac{L_2 S_1 C_2 + L_3 S_1 C_{23}}{L_2 L_3 S_3} & \frac{L_2 S_2 + L_3 S_{23}}{L_2 L_3 S_3} \end{bmatrix} \quad (5.71)$$

El MCIV una vez obtenido J_*^{-1} es el siguiente:

$$\dot{\theta}_1 = \frac{1}{L_2 C_2 + L_3 C_{23}} \{-S_1 \dot{x} + C_2 \dot{y}\} \quad (5.72)$$

$$\dot{\theta}_2 = \frac{1}{L_2 S_3} \{C_1 C_{23} \dot{x} + S_1 C_{23} \dot{y} - S_{23} \dot{z}\} \quad (5.73)$$

$$\dot{\theta}_3 = \frac{1}{L_2 L_3 S_3} \{-(L_2 C_1 C_2 + L_3 C_1 C_{23}) \dot{x} - (L_2 S_1 C_2 + L_3 S_1 C_{23}) \dot{y} + (L_2 S_2 + L_3 S_{23}) \dot{z}\} \quad (5.74)$$

5.2.7. Modelo cinemático directo de aceleración (MCDA)

Dado el MCDV del robot

$$\dot{x} = J \dot{q}$$

Ya obtenido el MCDV, se procede a obtener su derivada con respecto al tiempo, obteniendo así el modelo cinemático directo de aceleración, este se define como:

$$\ddot{x} = J \ddot{q} + \dot{J} \dot{q} \quad (5.75)$$

Donde:

\ddot{x} : corresponde al vector de aceleración del órgano terminal del robot.

\ddot{q} : corresponde al vector de aceleración de las articulaciones.

\dot{J} : representa la primera derivada respecto al tiempo de la matriz jacobiana.

La derivada de cada uno de los elementos de la matriz jacobiana es la siguiente:

$$\frac{d}{dt}(-L_3S_1C_{23} - L_2S_1C_2) = -L_3(S_1(-S_{23}(\dot{\theta}_2 + \dot{\theta}_3)) + C_{23}C_1\dot{\theta}_1) - L_2(S_1(-S_2)\dot{\theta}_2 + C_2C_1\dot{\theta}_1)$$

$$\dot{J}_{11} = L_3S_1S_{23}(\dot{\theta}_2 + \dot{\theta}_3) - L_3C_1C_{23}\dot{\theta}_1 + L_2S_1S_2\dot{\theta}_2 - L_2C_1C_2\dot{\theta}_1 \quad (5.76)$$

$$\frac{d}{dt}(-L_3S_{23}C_1 - L_2S_2C_1) = -L_3S_{23}(-S_1\dot{\theta}_1) - L_3C_1C_{23}(\dot{\theta}_2 + \dot{\theta}_3) - L_2S_2(-S_1\dot{\theta}_1) - L_2C_1C_2\dot{\theta}_2$$

$$\dot{J}_{12} = L_3S_1S_{23}\dot{\theta}_1 - L_3C_1C_{23}(\dot{\theta}_2 + \dot{\theta}_3) + L_2S_1S_2\dot{\theta}_1 - L_2C_1C_2\dot{\theta}_2 \quad (5.77)$$

$$\frac{d}{dt}(-L_3S_{23}C_1) = -L_3S_{23}(-S_1\dot{\theta}_1) - L_3C_1C_{23}(\dot{\theta}_2 + \dot{\theta}_3)$$

$$\dot{J}_{13} = L_3S_1S_{23}\dot{\theta}_1 - L_3C_1C_{23}(\dot{\theta}_2 + \dot{\theta}_3) \quad (5.78)$$

$$\frac{d}{dt}(L_3C_1C_{23} + L_2C_1C_2) = L_3C_1(-S_{23}(\dot{\theta}_2 + \dot{\theta}_3)) + L_3C_{23}(-S_1\dot{\theta}_1) + L_2C_1(-S_2\dot{\theta}_2) + L_2C_2(-S_1\dot{\theta}_1)$$

$$\dot{J}_{21} = -L_3S_{23}C_1(\dot{\theta}_2 + \dot{\theta}_3) - L_3S_1C_{23}\dot{\theta}_1 - L_2S_2C_1\dot{\theta}_2 - L_2S_1C_2\dot{\theta}_1 \quad (5.79)$$

$$\frac{d}{dt}(-L_3S_1S_{23} - L_2S_1S_2) = -L_3S_1C_{23}(\dot{\theta}_2 + \dot{\theta}_3) - L_3S_{23}C_1\dot{\theta}_1 - L_2S_1C_2\dot{\theta}_2 - L_2S_2C_1\dot{\theta}_1$$

$$\dot{J}_{22} = -L_3S_1C_{23}(\dot{\theta}_2 + \dot{\theta}_3) - L_3S_{23}C_1\dot{\theta}_1 - L_2S_1C_2\dot{\theta}_2 - L_2S_2C_1\dot{\theta}_1 \quad (5.80)$$

$$\frac{d}{dt}(-L_3S_1S_{23}) = -L_3S_1C_{23}(\dot{\theta}_2 + \dot{\theta}_3) - L_3S_{23}C_1\dot{\theta}_1$$

$$\dot{J}_{23} = -L_3S_1C_{23}(\dot{\theta}_2 + \dot{\theta}_3) - L_3S_{23}C_1\dot{\theta}_1 \quad (5.81)$$

$$\frac{d}{dt}(0) = 0$$

$$\dot{J}_{31} = 0 \quad (5.82)$$

$$\frac{d}{dt}(-L_3C_{23} - L_2C_2) = -L_3S_{23}(\dot{\theta}_2 + \dot{\theta}_3) + L_2S_2\dot{\theta}_2$$

$$\dot{J}_{32} = -L_3S_{23}(\dot{\theta}_2 + \dot{\theta}_3) + L_2S_2\dot{\theta}_2 \quad (5.83)$$

$$\frac{d}{dt}(-L_3 C_{23}) = -L_3(-S_{23}(\dot{\theta}_2 + \dot{\theta}_3))$$

$$\dot{J}_{33} = L_3 S_{23}(\dot{\theta}_2 + \dot{\theta}_3) \quad (5.84)$$

Por lo tanto la matriz jacobiana queda de la siguiente manera:

$$\dot{J}_* = \begin{bmatrix} \dot{J}_{11} & \dot{J}_{12} & \dot{J}_{13} \\ \dot{J}_{21} & \dot{J}_{22} & \dot{J}_{23} \\ \dot{J}_{31} & \dot{J}_{32} & \dot{J}_{33} \end{bmatrix} \quad (5.85)$$

Por consiguiente J y \dot{J} obetemos el MCDA.

Para \ddot{x} :

$$\frac{d}{dt}(-L_3 S_1 C_{23} \dot{\theta}_1 - L_2 S_1 C_2 \dot{\theta}_1 - L_3 S_{23} C_1 \dot{\theta}_2 - L_2 S_2 C_1 \dot{\theta}_2 - L_3 S_{23} C_1 \dot{\theta}_3) =$$

$$\begin{aligned} \ddot{x} = & -L_3 S_1 C_{23} \ddot{\theta}_1 + L_3 S_1 S_{23} \dot{\theta}_1 (\dot{\theta}_2 + \dot{\theta}_3) - L_3 C_1 C_{23} \dot{\theta}_1 \dot{\theta}_1 - L_3 S_1 C_2 \ddot{\theta}_1 + L_2 S_1 S_2 \dot{\theta}_1 \dot{\theta}_2 - L_2 C_1 C_2 \dot{\theta}_1 \dot{\theta}_1 - L_3 S_{23} C_1 \ddot{\theta}_2 \\ & + L_3 S_1 S_{23} \dot{\theta}_1 \dot{\theta}_2 - L_3 C_1 C_{23} \dot{\theta}_2 (\dot{\theta}_2 + \dot{\theta}_3) - L_2 S_2 C_1 \ddot{\theta}_2 - L_2 S_1 S_2 \dot{\theta}_1 \dot{\theta}_2 - L_2 C_1 C_2 \dot{\theta}_2 \dot{\theta}_2 - L_3 S_{23} C_1 \ddot{\theta}_3 + L_3 S_1 S_{23} \dot{\theta}_1 \dot{\theta}_3 \\ & - L_3 C_1 C_{23} \dot{\theta}_3 (\dot{\theta}_2 + \dot{\theta}_3) \end{aligned}$$

Simplificando y agrupando

$$\begin{aligned} \ddot{x} = & (-L_3 S_1 C_{23} - L_2 S_1 C_2) \ddot{\theta}_1 + (-L_3 S_{23} C_1 - L_2 S_2 C_1) \ddot{\theta}_2 - (L_3 S_{23} C_1) \ddot{\theta}_3 + (L_3 S_1 S_{23} (\dot{\theta}_2 + \dot{\theta}_3) - L_3 C_1 C_{23} \dot{\theta}_1 \\ & + L_2 S_1 S_2 \dot{\theta}_2 - L_2 C_1 C_2 \dot{\theta}_1) \dot{\theta}_1 + (L_3 S_1 S_{23} \dot{\theta}_1 - L_3 C_1 C_{23} (\dot{\theta}_2 + \dot{\theta}_3) + L_2 S_1 S_2 \dot{\theta}_1 - L_2 C_1 C_2 \dot{\theta}_2) \dot{\theta}_2 \\ & + (L_3 S_1 S_{23} \dot{\theta}_1 - L_3 C_1 C_2 (\dot{\theta}_2 + \dot{\theta}_3)) \dot{\theta}_3 \end{aligned} \quad (5.86)$$

Para \ddot{y}

$$\frac{d}{dt}(L_3 C_1 C_{23} \dot{\theta}_1 + L_2 C_1 C_2 \dot{\theta}_1 - L_3 S_1 S_{23} \dot{\theta}_2 - L_2 S_1 S_2 \dot{\theta}_2 - L_3 S_1 S_{23} \dot{\theta}_3) =$$

$$\begin{aligned} \ddot{y} = & L_3 C_1 C_{23} \ddot{\theta}_1 + L_3 C_1 S_{23} \dot{\theta}_1 (\dot{\theta}_2 + \dot{\theta}_3) - L_3 S_1 C_{23} \dot{\theta}_1 \dot{\theta}_1 + L_2 C_1 C_2 \ddot{\theta}_1 - L_2 S_2 C_2 \dot{\theta}_1 \dot{\theta}_2 - L_2 S_1 C_2 \dot{\theta}_1 \dot{\theta}_1 - L_3 S_1 S_{23} \ddot{\theta}_2 \\ & - L_3 S_1 C_{23} \dot{\theta}_2 (\dot{\theta}_2 + \dot{\theta}_3) - L_3 S_{23} C_1 \dot{\theta}_1 \dot{\theta}_2 - L_2 S_1 S_2 \ddot{\theta}_2 - L_3 S_1 C_{23} \dot{\theta}_2 \dot{\theta}_2 - L_2 S_1 C_1 \dot{\theta}_1 \dot{\theta}_2 - L_3 S_1 S_{23} \ddot{\theta}_3 - L_3 S_1 C_{23} \dot{\theta}_3 (\dot{\theta}_2 \\ & + \dot{\theta}_3) - L_3 S_{23} C_1 \dot{\theta}_1 \dot{\theta}_3 \end{aligned}$$

Simplificando y agrupando

$$\begin{aligned} \ddot{y} = & (L_3C_1C_{23} + L_2C_1C_2)\ddot{\theta}_1 + (-L_3S_1S_{23} - L_2S_1S_2)\ddot{\theta}_2 - (L_3S_1S_{23})\ddot{\theta}_3 + (-L_3S_{23}C_1(\dot{\theta}_2 + \dot{\theta}_3) - L_3S_1C_{23}\dot{\theta}_1 - \\ & L_2S_2C_1\dot{\theta}_2 - L_2S_1C_2\dot{\theta}_1)\dot{\theta}_1 + (-L_3S_1C_{23}(\dot{\theta}_2 + \dot{\theta}_3) - L_3S_{23}C_1\dot{\theta}_1 - L_2S_1C_2\dot{\theta}_2 - L_2S_2C_1\dot{\theta}_1)\dot{\theta}_2 + (-L_3S_1C_{23}(\dot{\theta}_2 + \dot{\theta}_3) \\ & - L_3S_{23}C_1\dot{\theta}_1)\dot{\theta}_3 \end{aligned} \quad (5.87)$$

Para \ddot{z}

$$\frac{d}{dt}(-L_3C_{23}\dot{\theta}_2 - L_2C_2\dot{\theta}_2 - L_3C_{23}\dot{\theta}_3) =$$

$$\ddot{z} = -L_3C_{23}\ddot{\theta}_2 + L_3S_{23}\dot{\theta}_2(\dot{\theta}_2 + \dot{\theta}_3) - L_2C_2\ddot{\theta}_2 + L_2S_2\dot{\theta}_2\dot{\theta}_2 + L_3C_{23}\ddot{\theta}_3 + L_3S_{23}\dot{\theta}_3(\dot{\theta}_2 + \dot{\theta}_3)$$

Simplificando y agrupando

$$\ddot{z} = (-L_3C_{23} - L_2C_2)\ddot{\theta}_2 - (L_3C_{23})\ddot{\theta}_3 - (L_3S_{23}(\dot{\theta}_2 + \dot{\theta}_3) + L_2S_2\dot{\theta}_2)\dot{\theta}_2 + (L_3S_{23}(\dot{\theta}_2 + \dot{\theta}_3))\dot{\theta}_3 \quad (5.88)$$

y para

$$\ddot{\lambda} = 0 \quad (5.89)$$

$$\ddot{\mu} = 0 \quad (5.90)$$

$$\ddot{v} = \ddot{\theta}_1 + \ddot{\theta}_2 \quad (5.91)$$

5.2.8. Modelo cinemático inverso de aceleración (MCIA)

El modelo cinemático inverso de aceleración se deduce a partir del MCDA obtenido anteriormente y este es:

$$\ddot{q} = J^{-1}[\ddot{x} - \dot{J}\dot{q}] \quad (5.92)$$

La inversa de jacobiano reducido es

$$J_*^{-1} = \begin{bmatrix} -\frac{S_1}{L_2C_2+L_3C_{23}} & \frac{C_1}{L_2C_2+L_3C_{23}} & 0 \\ \frac{C_1C_{23}}{L_2S_3} & \frac{S_1C_{23}}{L_2S_3} & -\frac{S_{23}}{L_2S_3} \\ -\frac{L_2C_1C_2+L_3C_1C_{23}}{L_2L_3S_3} & -\frac{L_2S_1C_2+L_3S_1C_{23}}{L_2L_3S_3} & \frac{L_2S_2+L_3S_{23}}{L_2L_3S_3} \end{bmatrix}$$

Por lo tanto aplicando $\ddot{q} = J_*^{-1}[\ddot{x} - \dot{J}\dot{q}]$

$$\begin{aligned} \ddot{\theta}_1 = & -\frac{S_1}{L_2C_2+L_3C_{23}}[\ddot{x} - ((L_3S_1S_{23}(\dot{\theta}_2 + \dot{\theta}_3) - L_3C_1C_{23}\dot{\theta}_1 + L_2S_1S_2\dot{\theta}_2 - L_2C_1C_2\dot{\theta}_1)\dot{\theta}_1 + (L_3S_1S_{23}\dot{\theta}_1 \\ & - L_3C_1C_{23}(\dot{\theta}_2 + \dot{\theta}_3) + L_2S_1S_2\dot{\theta}_1 - L_2C_1C_2\dot{\theta}_2)\dot{\theta}_2] + \frac{C_1}{L_2C_2+L_3C_{23}}[\ddot{y} - ((-L_3S_{23}C_1(\dot{\theta}_2 + \dot{\theta}_3) - L_3S_1C_{23}\dot{\theta}_1 \\ & - L_2S_2C_1\dot{\theta}_2 - L_2S_1C_2\dot{\theta}_1)\dot{\theta}_1 + (-L_3S_1C_{23}(\dot{\theta}_2 + \dot{\theta}_3) - L_3S_{23}C_1\dot{\theta}_1 - L_2S_1C_2\dot{\theta}_2 - L_2S_2C_1\dot{\theta}_1)\dot{\theta}_2 \\ & + (-L_3S_1C_{23}(\dot{\theta}_2 + \dot{\theta}_3) - L_3S_{23}C_1\dot{\theta}_1)\dot{\theta}_3] \end{aligned} \quad (5.93)$$

$$\begin{aligned} \ddot{\theta}_2 = & -\frac{C_1C_{23}}{L_2S_3}[\ddot{x} - ((L_3S_1S_{23}(\dot{\theta}_2 + \dot{\theta}_3) - L_3C_1C_{23}\dot{\theta}_1 - L_3C_1C_{23}\dot{\theta}_1 + L_2S_1S_2\dot{\theta}_2 - L_2C_1C_2\dot{\theta}_1)\dot{\theta}_1 \\ & + L_3S_1S_{23}\dot{\theta}_1 - L_3C_1C_{23}(\dot{\theta}_2 + \dot{\theta}_3) + L_2S_1S_2\dot{\theta}_1 - L_2C_1C_2\dot{\theta}_2)\dot{\theta}_2 + (L_3S_1S_{23}\dot{\theta}_1 - L_3C_1C_{23}(\dot{\theta}_2 + \dot{\theta}_3))\dot{\theta}_3] \\ & + \frac{S_1C_{23}}{L_2S_3}[\ddot{y} - ((-L_3S_{23}C_1(\dot{\theta}_2 + \dot{\theta}_3) - L_3S_1C_{23}\dot{\theta}_1 - L_2S_2C_1\dot{\theta}_2 - L_2S_1C_2\dot{\theta}_1)\dot{\theta}_1 + (-L_3S_1C_{23}(\dot{\theta}_2 + \dot{\theta}_3) \\ & - L_3S_{23}C_1\dot{\theta}_1 - L_2S_1C_2\dot{\theta}_2 - L_2S_2C_1\dot{\theta}_1)\dot{\theta}_2 + (-L_3S_1C_{23}(\dot{\theta}_2 + \dot{\theta}_3) - L_3S_{23}C_1\dot{\theta}_1)\dot{\theta}_3] - \frac{S_{23}}{L_2S_3}[\ddot{z} \\ & - ((L_3S_{23}(\dot{\theta}_2 + \dot{\theta}_3) + L_2S_2\dot{\theta}_2 + (L_3S_{23}(\dot{\theta}_2 + \dot{\theta}_3))\dot{\theta}_3)] \end{aligned} \quad (5.94)$$

$$\begin{aligned} \ddot{\theta}_3 = & -L_2C_1C_2 + L_3C_1C_{23}[\ddot{x} - ((L_3S_1S_{23}(\dot{\theta}_2 + \dot{\theta}_3) - L_3C_1C_{23}\dot{\theta}_1 + L_2S_1S_2\dot{\theta}_2 - L_2C_1C_2\dot{\theta}_1)\dot{\theta}_1 + (L_3S_1S_{23}\dot{\theta}_1 \\ & - L_3C_1C_{23}(\dot{\theta}_2 + \dot{\theta}_3) + L_2S_1S_2\dot{\theta}_1 - L_2C_1C_2\dot{\theta}_2)\dot{\theta}_2 + (L_3S_1S_{23}\dot{\theta}_1 - L_3C_1C_{23}(\dot{\theta}_2 + \dot{\theta}_3))\dot{\theta}_3] \\ & - \frac{L_2S_1C_2+L_3S_1C_{23}}{L_2L_3S_2}[\ddot{y} - ((-L_3S_{23}C_1(\dot{\theta}_2 + \dot{\theta}_3) - L_3S_1C_{23}\dot{\theta}_1 - L_2S_2C_1\dot{\theta}_2 - L_2S_1C_2\dot{\theta}_1)\dot{\theta}_1 + (-L_3S_1C_{23}(\dot{\theta}_2 + \dot{\theta}_3) \\ & - L_3S_{23}C_1\dot{\theta}_1 - L_2S_1C_2\dot{\theta}_2 - L_2S_2C_1\dot{\theta}_1)\dot{\theta}_2 + (-L_3S_1C_{23}(\dot{\theta}_2 + \dot{\theta}_3) - L_3S_{23}C_1\dot{\theta}_1)\dot{\theta}_3] \\ & + \frac{L_2S_2+L_3S_{23}}{L_2L_3S_3}[\ddot{z} - ((L_3S_{23}(\dot{\theta}_2 + \dot{\theta}_3) + L_2S_2\dot{\theta}_2)\dot{\theta}_2 + (L_3S_{23}(\dot{\theta}_2 + \dot{\theta}_3))\dot{\theta}_3)] \end{aligned} \quad (5.95)$$

5.3. Modelo dinámico

Representación de la cadena cinemática

El modelo dinámico del robot antropomórfico en su estructura se define a continuación, dada la representación gráfica de la cadena cinemática (Figura 5.3), considerando los parámetros dinámicos, nos permitirá en forma auxiliar derivar el MD.

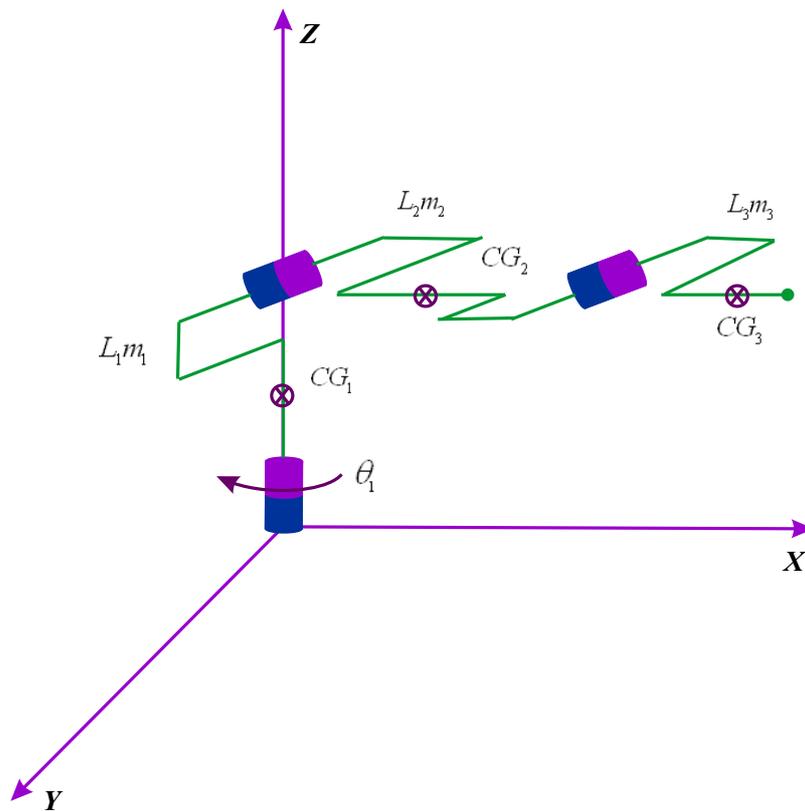


Figura 5.3: Cadena cinemática con parámetros dinámicos

La representación gráfica sintética, considerando las variables articulares y coordenadas operacionales es definida en la Figura 5.4.

5.3.1. Obtención de los módulos de velocidad y alturas de centros de masas

Las coordenadas operacionales del eslabón uno son:

$$\begin{aligned}x_1 &= 0 \\y_1 &= 0 \\z_1 &= -L_1\end{aligned}\tag{5.96}$$

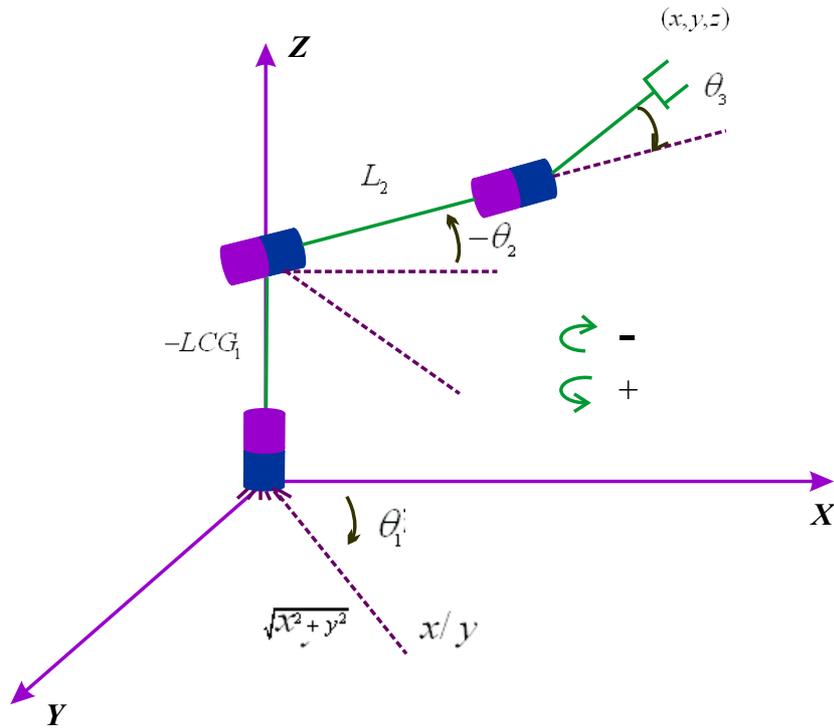


Figura 5.4: Descripción de comportamiento de variables articulares

y sus derivadas con respecto del tiempo:

$$\begin{aligned} \dot{x}_1 &= 0 \\ \dot{y}_1 &= 0 \\ \dot{z}_1 &= 0 \end{aligned} \quad (5.97)$$

El módulo de la velocidad y altura al centro de masas para el eslabón 1 es:

A partir de:

$$v_1^2 = \dot{x}_1^2 + \dot{y}_1^2 + \dot{z}_1^2 \quad (5.98)$$

Sustituyendo en la ecuación se observa que:

$$v_1^2 = 0^2 + 0^2 + 0^2$$

Por lo tanto, la velocidad de acuerdo al modelo cinemático de velocidad es:

$$v_1^2 = 0 \quad (5.99)$$

Y la altura para el eslabón 1 es:

$$h_1 = -LCG_1 \quad (5.100)$$

donde:

LCG : longitud al centro de masas.

Las coordenadas operacionales (x_2, y_2, z_2) y sus derivadas respecto del tiempo, para el eslabón 2 son:

$$\begin{aligned}x_2 &= L_2 C_1 C_2 \\y_2 &= L_2 C_1 C_2 \\z_2 &= -L_2 C_2\end{aligned}\quad (5.101)$$

Sus derivadas con respecto al tiempo

$$\begin{aligned}\dot{x}_2 &= -L_2 S_1 C_2 \dot{\theta}_1 - L_2 S_2 C_1 \dot{\theta}_2 \\ \dot{y}_2 &= L_2 C_1 C_2 \dot{\theta}_1 - L_2 S_1 S_2 \dot{\theta}_2 \\ \dot{z}_2 &= -L_2 C_2 \dot{\theta}_2\end{aligned}\quad (5.102)$$

Continuando, el siguiente módulo es del cuadrado de la velocidad y altura al centro de masas para el eslabón 2 a partir de:

$$v_2^2 = \dot{x}_2^2 + \dot{y}_2^2 + \dot{z}_2^2 \quad (5.103)$$

Substituyendo en la ecuación anterior.

$$v_2^2 = \left(-L_2 S_1 C_2 \dot{\theta}_1 - L_2 S_2 C_1 \dot{\theta}_2\right)^2 + \left(L_2 C_1 C_2 \dot{\theta}_1 - L_2 S_1 S_2 \dot{\theta}_2\right)^2 + \left(-L_2 C_2 \dot{\theta}_2\right)^2$$

Desarrollando.

$$\begin{aligned}v_2^2 &= L_2^2 S_1^2 C_2^2 \dot{\theta}_1^2 + 2L_2^2 S_1 S_2 C_1 C_2 \dot{\theta}_1 \dot{\theta}_2 + L_2^2 S_2^2 C_1^2 \dot{\theta}_2^2 + L_2^2 C_1^2 C_2^2 \dot{\theta}_1^2 - 2L_2^2 S_1 S_2 C_1 C_2 \dot{\theta}_1 \dot{\theta}_2 + L_2^2 S_1^2 S_2^2 \dot{\theta}_2^2 + L_2^2 C_2^2 \dot{\theta}_2^2 \\ v_2^2 &= L_2^2 C_2^2 \dot{\theta}_1^2 + L_2^2 S_2^2 \dot{\theta}_2^2 + L_2^2 C_2^2 \dot{\theta}_2^2 \\ v_2^2 &= L_2^2 C_2^2 \dot{\theta}_1^2 + L_2^2 \dot{\theta}_2^2\end{aligned}\quad (5.104)$$

La altura del eslabón 2 se define en la siguiente ecuación.

$$h_2 = -LCG_2 S_2 \quad (5.105)$$

Las coordenadas operacionales (x_3, y_3, z_3) y sus derivadas respecto del tiempo, para el eslabón 3 son las siguientes:

$$\begin{aligned}x_3 &= L_2 C_1 C_2 + L_3 C_1 C_{23} \\ y_3 &= L_2 S_1 C_2 + L_3 S_1 C_{23} \\ z_3 &= -L_2 S_2 + L_3 S_{23}\end{aligned}\quad (5.106)$$

$$\begin{aligned}
\dot{x}_3 &= -L_2S_1C_2\dot{\theta}_1 - L_2S_2C_1\dot{\theta}_2 - L_3S_1C_{23}\dot{\theta}_1 - L_3S_{23}C_1(\dot{\theta}_2 + \dot{\theta}_3) \\
\dot{y}_3 &= -L_2S_1S_2\dot{\theta}_2 + L_2C_1C_2\dot{\theta}_1 + L_3C_1C_{23}\dot{\theta}_1 - L_3S_1C_{23}(\dot{\theta}_2 + \dot{\theta}_3) \\
\dot{z}_3 &= -L_2C_2\dot{\theta}_2 - L_3C_{23}(\dot{\theta}_2 + \dot{\theta}_3)
\end{aligned} \tag{5.107}$$

Sustituyendo en la ecuación de velocidad del modelo dinámico directo de velocidad:

$$\begin{aligned}
v_3^2 &= \dot{x}_3^2 + \dot{y}_3^2 + \dot{z}_3^2 \\
v_3^2 &= \left\{ -L_2S_1C_2\dot{\theta}_1 - L_2S_2C_1\dot{\theta}_2 - L_3S_1C_{23}\dot{\theta}_1 - L_3S_{23}C_1(\dot{\theta}_2 + \dot{\theta}_3) \right\}^2 + \\
&\left\{ -L_2S_1S_2\dot{\theta}_2 + L_2C_1C_2\dot{\theta}_1 + L_3C_1C_{23}\dot{\theta}_1 - L_3S_1C_{23}(\dot{\theta}_2 + \dot{\theta}_3) \right\}^2 + \\
&\left\{ -L_2C_2\dot{\theta}_2 - L_3C_{23}(\dot{\theta}_2 + \dot{\theta}_3) \right\}^2
\end{aligned}$$

Simplificando:

$$v_3^2 = L_2^2\dot{\theta}_2^2 + L_3^2(\dot{\theta}_2 + \dot{\theta}_3)^2 + 2L_2L_3C_3\dot{\theta}_2(\dot{\theta}_2 + \dot{\theta}_3) + (L_2C_2\dot{\theta}_1 + L_3C_{23}\dot{\theta}_1)^2 \tag{5.108}$$

La altura del eslabón 3 se define en la siguiente ecuación.

$$h_3 = -L_2S_2 - L_3C_{23}S_{23} \tag{5.109}$$

5.3.2. Determinación de energías cinéticas del robot

El cálculo de las energías cinéticas para cada uno de los eslabones se presentan a continuación, a partir de la ecuación siguiente:

$$K = \frac{1}{2}mv^2 \tag{5.110}$$

Para el eslabón 1.

$$K_1 = \frac{1}{2}m_1v_1^2 = \frac{1}{2}m_1(0) = 0 \tag{5.111}$$

$$K_1 = 0$$

Para el eslabón 2.

$$K_2 = \frac{1}{2}m_2v_2^2 = \frac{1}{2}m_2 \left\{ L_2^2C_2^2\dot{\theta}_1^2 + L_2^2\dot{\theta}_2^2 \right\} \tag{5.112}$$

$$K_2 = \frac{1}{2}m_2L_2^2C_2^2\dot{\theta}_1^2 + \frac{1}{2}m_2L_2^2\dot{\theta}_2^2$$

Para el eslabón 3.

$$K_3 = \frac{1}{2}m_3v_3^2$$

$$K_3 = \frac{1}{2}m_3 \left\{ L_2^2\dot{\theta}_2^2 + L_3^2(\dot{\theta}_2 + \dot{\theta}_3)^2 + 2L_2L_3C_3\dot{\theta}_2(\dot{\theta}_2 + \dot{\theta}_3) + (L_2C_2\dot{\theta}_1 + L_3C_{23}\dot{\theta}_1)^2 \right\}$$

$$K_3 = \frac{1}{2}m_3L_2^2\dot{\theta}_2^2 + \frac{1}{2}m_3L_3^2(\dot{\theta}_2 + \dot{\theta}_3)^2 + m_3L_2L_3C_3\dot{\theta}_2(\dot{\theta}_2 + \dot{\theta}_3) + \frac{1}{2}m_3(L_2C_2\dot{\theta}_1 + L_3C_{23}\dot{\theta}_1)^2 \quad (5.113)$$

5.3.3. Determinación de energías potenciales del robot

El cálculo de las energías potenciales para cada uno de los eslabones se presentan a continuación, a partir de la ecuación siguiente:

$$P_i = m_i g h_i \quad (5.114)$$

Para el eslabón 1

$$P_1 = m_1 g h_1 = m_1 g \{-LCG_1\} \quad (5.115)$$

$$P_1 = m_1 g LCG_1$$

Para el eslabón 2

$$P_2 = m_2 g h_2 = \{-LCG_2 S_2\} \quad (5.116)$$

$$P_2 = -m_2 g LCG_2 S_2$$

Para el eslabón 3

$$P_3 = m_3 g h_3 = m_3 g \{-L_2 S_2 - LCG_3 S_{23}\} \quad (5.117)$$

$$P_3 = -m_3 g L_2 S_2 - m_3 g LCG_3 S_{23}$$

5.3.4. Ecuaciones de movimiento (Lagrangiano)

El lagrangiano está definido por la siguiente expresión

$$L = \sum_{i=1}^{n=3} (K_i - P_i) \quad (5.118)$$

A partir de las energías cinéticas y potenciales para el caso de estudio (3 grados de libertad) es:

$$L = (K_1 + K_2 + K_3) - (P_1 + P_2 + P_3) \quad (5.119)$$

Desarrollando dicha ecuación se tiene:

$$\begin{aligned} L &= \frac{1}{2}m_2L_2^2C_2^2\dot{\theta}_1^2 + \frac{1}{2}m_2L_2^2C_2^2\dot{\theta}_2^2 + \frac{1}{2}m_3L_2^2\dot{\theta}_2^2 + \frac{1}{2}m_3L_3^2(\dot{\theta}_2 + \dot{\theta}_3)^2 + m_3L_2L_3C_3\dot{\theta}_2(\dot{\theta}_2 + \dot{\theta}_3) \\ &+ \frac{1}{2}m_3(L_2C_2\dot{\theta}_1 + L_3C_{23}\dot{\theta}_1)^2 - m_1gLCG_1 + m_2gLCG_2S_2 + m_3gL_2S_2 + m_3gLCG_3S_{23} \\ L &= \frac{1}{2}m_2L_2^2C_2^2\dot{\theta}_1^2 + \frac{1}{2}m_2L_2^2\dot{\theta}_2^2 + \frac{1}{2}m_3L_2^2\dot{\theta}_2^2 + \frac{1}{2}m_3L_3^2\dot{\theta}_2^2 + m_3L_3^2\dot{\theta}_2\dot{\theta}_3 + \frac{1}{2}m_3L_3^2\dot{\theta}_3^2 + m_3L_2L_3C_2\dot{\theta}_2^2 \\ &+ m_3L_2L_3C_2\dot{\theta}_2\dot{\theta}_3 + \frac{1}{2}m_3L_2^2C_2^2\dot{\theta}_1^2 + m_3L_2L_3C_2C_{23}\dot{\theta}_1^2 + \frac{1}{2}m_3L_3^2C_{23}^2\dot{\theta}_1^2 - m_1gLCG_1 + m_2gLCG_2S_2 \\ &+ m_3gL_2S_2 + m_3gLCG_3S_{23} \end{aligned}$$

Simplificando y agrupando se obtiene la ecuación del Lagrangiano para el caso de estudio

$$\begin{aligned} L &= \left\{ \frac{1}{2}m_2L_2^2C_2^2 + \frac{1}{2}m_3L_2^2C_2^2 + m_3L_2L_3C_2C_{23} + \frac{1}{2}m_3L_3^2C_{23}^2 \right\} \dot{\theta}_1^2 + \left\{ \frac{1}{2}m_2L_2^2 + \frac{1}{2}m_3L_2^2 + \frac{1}{2}m_3L_3^2 + m_3L_2L_3C_3 \right\} \dot{\theta}_2^2 \\ &+ \left\{ \frac{1}{2}m_3L_3^2 \right\} \dot{\theta}_3^2 + \left\{ m_3L_3^2 + m_3L_2L_3C_3 \right\} \dot{\theta}_2\dot{\theta}_3 - m_1gLCG_1 + m_2gLCG_2S_2 + m_3g \{ L_2S_2 + LCG_3S_{23} \} \end{aligned} \quad (5.120)$$

5.3.5. Formulación Euler-Lagrange.

Dada la formulación de Euler-Lagrange definida por la siguiente expresión

$$\tau_i = \frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_i} - \frac{\partial L}{\partial \theta_i} \quad (5.121)$$

Para el caso de estudio obtendremos una ecuación de par torsor por cada eslabón del robot, es decir tendremos

$$\tau_1 = \frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_1} - \frac{\partial L}{\partial \theta_1} \quad (5.122)$$

$$\tau_2 = \frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_2} - \frac{\partial L}{\partial \theta_2} \quad (5.123)$$

$$\tau_3 = \frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_3} - \frac{\partial L}{\partial \theta_3} \quad (5.124)$$

Obteniendo la ecuación 1

Derivando:

Para $\frac{\partial L}{\partial \theta_1}$

$$\begin{aligned} \frac{\partial}{\partial \theta_1} \left(\frac{1}{2} m_2 L_2^2 C_2^2 + \frac{1}{2} m_3 L_2^2 C_2^2 + m_3 L_2 L_3 C_2 C_{23} + \frac{1}{2} m_3 L_3^2 C_{23}^2 \right) \dot{\theta}_1^2 \\ \frac{\partial L}{\partial \theta_1} = \{ m_2 L_2^2 C_2^2 + m_3 L_2^2 C_2^2 + 2m_3 L_2 L_3 C_2 C_{23} + m_3 L_3^2 C_{23}^2 \} \dot{\theta}_1 \end{aligned} \quad (5.125)$$

$$\frac{\partial L}{\partial \dot{\theta}_1} = (m_2 L_2^2 C_2^2 \dot{\theta}_1 + m_3 L_2^2 C_2^2 \dot{\theta}_1 + 2m_3 L_2 L_3 C_2 C_{23} \dot{\theta}_1 + m_3 L_3^2 C_{23}^2 \dot{\theta}_1)$$

Para $\frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_1}$

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_1} (m_2 L_2^2 C_2^2 \dot{\theta}_1 + m_3 L_2^2 C_2^2 \dot{\theta}_1 + 2m_3 L_2 L_3 C_2 C_{23} \dot{\theta}_1 + m_3 L_3^2 C_{23}^2 \dot{\theta}_1)$$

Derivando la ecuación anterior por partes

$$\begin{aligned} \frac{d}{dt} (m_2 L_2^2 C_2^2 \dot{\theta}_1) &= m_2 L_2^2 C_2^2 \ddot{\theta}_1 - 2m_2 L_2^2 S_2 \dot{\theta}_1 \dot{\theta}_2 \\ \frac{d}{dt} (m_3 L_2^2 C_2^2 \dot{\theta}_1) &= m_3 L_2^2 C_2^2 \ddot{\theta}_1 - 2m_3 L_2^2 S_2 \dot{\theta}_1 \dot{\theta}_2 \\ \frac{d}{dt} (2m_3 L_2 L_3 C_2 C_{23} \dot{\theta}_1) &= 2m_3 L_2 L_3 C_2 C_{23} \ddot{\theta}_1 - 2m_3 L_2 L_3 C_2 S_{23} (\dot{\theta}_2 + \dot{\theta}_3) - 2m_3 L_2 L_3 C_{23} S_2 \dot{\theta}_1 \dot{\theta}_2 \\ \frac{d}{dt} (m_3 L_3^2 C_{23}^2 \dot{\theta}_1) &= -2m_3 L_3^2 C_{23} S_{23} \dot{\theta}_1 \dot{\theta}_2 - 2m_3 L_3^2 C_{23} S_{23} \dot{\theta}_1 \dot{\theta}_3 + m_3 L_3 C_{23}^2 \ddot{\theta}_1 \end{aligned}$$

Por lo tanto la ecuación $\frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_1}$, queda de la siguiente forma:

$$\begin{aligned} \frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_1} &= ((m_2 + m_3) L_2^2 C_2^2 + 2m_3 L_2 L_3 C_2 C_{23} + m_3 L_3^2 C_{23}^2) \ddot{\theta}_1 - (2(m_2 + m_3) L_2^2 S_2 C_2 \\ &+ 2m_3 L_2 L_3 (S_{23} C_2 + S_2 C_{23}) + 2m_3 L_3^2 S_{23} C_{23}) \dot{\theta}_1 \dot{\theta}_2 - (2m_3 L_2 L_3 S_{23} C_2 + 2m_3 L_3^2 S_{23} C_{23}) \dot{\theta}_1 \dot{\theta}_3 \end{aligned} \quad (5.126)$$

y para $\frac{\partial L}{\partial \theta_1}$

$$\frac{\partial L}{\partial \theta_1} = 0 \quad (5.127)$$

Sustituyendo las Ecuaciones 5.125, 5.126, 5.127 en la ecuación 5.122 para obtener la ecuación par torsor 1:

$$\tau_1 = \frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_1} - \frac{\partial L}{\partial \theta_1}$$

$$\begin{aligned} \tau_1 = & ((m_2 + m_3)L_2^2 C_2^2 + 2m_3 L_2 L_3 C_2 C_{23} + m_3 L_3^2 C_{23}) \ddot{\theta}_1 - (2(m_2 + m_3)L_2^2 S_2 C_2 \\ & + 2m_3 L_2 L_3 (S_{23} C_2 + S_2 C_{23}) + 2m_3 L_3^2 S_{23} C_{23}) \dot{\theta}_1 \dot{\theta}_2 - (2m_3 L_2 L_3 S_{23} C_2 + 2m_3 L_3^2 S_{23} C_{23}) \dot{\theta}_1 \ddot{\theta}_3 \end{aligned} \quad (5.128)$$

Obteniendo la ecuación 2

$$\tau_2 = \frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_2} - \frac{\partial L}{\partial \theta_2}$$

Para $\frac{\partial L}{\partial \theta_2}$

$$\frac{\partial}{\partial \theta_2} \left(\frac{1}{2} m_2 L_2^2 \dot{\theta}_2^2 + \frac{1}{2} m_3 L_2^2 \dot{\theta}_2^2 + \frac{1}{2} m_3 L_3^2 \dot{\theta}_2^2 + m_3 L_2 L_3 C_2 \dot{\theta}_2^2 + m_3 L_3^2 \dot{\theta}_2 \dot{\theta}_3 + m_3 L_2 L_3 C_2 \dot{\theta}_2 \dot{\theta}_3 \right)$$

$$\frac{\partial L}{\partial \theta_2} = \{m_2 L_2^2 + m_3 L_2^2 + m_3 L_3^2 + 2m_3 L_2 L_3 C_3\} \dot{\theta}_2 + \{m_3 L_3^2 + m_3 L_2 L_3 C_3\} \dot{\theta}_3 \quad (5.129)$$

Para $\frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_2}$

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_2} (m_2 L_2^2 \dot{\theta}_2 + m_3 L_2^2 \dot{\theta}_2 + m_3 L_3^2 \dot{\theta}_2 + 2m_3 L_2 L_3 C_2 \dot{\theta}_2 + m_3 L_3^2 \dot{\theta}_3 + m_3 L_2 L_3 C_3 \dot{\theta}_3) =$$

Derivando la ecuación obtenemos

$$\begin{aligned} \frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_2} = & \{(m_2 + m_3)L_2^2 + m_3 L_3^2 + 2m_3 L_2 L_3\} \ddot{\theta}_2 + \{m_3 L_3^2 + m_3 L_2 L_3 C_3\} \ddot{\theta}_2 - \{2m_3 L_2 L_3 S_3\} \ddot{\theta}_2 \ddot{\theta}_3 \\ & - \{m_3 L_2 L_3 S_3\} \dot{\theta}_3 \end{aligned} \quad (5.130)$$

Para $\frac{\partial L}{\partial \theta_2}$

$$\begin{aligned} \frac{\partial L}{\partial \theta_2} = & \{-(m_2 + m_3)L_2^2 S_2 C_2 - m_3 L_3 L_2 (C_2 S_{23} + S_2 C_{23}) - m_3 L_3^2 C_{23} S_{23}\} \dot{\theta}_1^2 + m_2 g L C G_2 C_2 \\ & + m_3 g \{L_2 C_2 + L C G_3 C_3\} \ddot{\theta}_2 \end{aligned} \quad (5.131)$$

La ecuación de par torsor 2 es:

$$\begin{aligned} \tau_2 = & \{ (m_2 + m_3)L_2^2 + m_3L_3^2 + 2m_3L_2L_3C_3 \} \ddot{\theta}_2 + \{ m_3L_3^2 + m_3L_2L_3C_3 \} \ddot{\theta}_3 - \\ & \{ 2m_3L_2L_3S_3 \} \dot{\theta}_2\dot{\theta}_3 + \{ (m_2 + m_3)L_2^2S_2C_2 + m_3L_2L_3(C_2S_{23} + C_{23}S_2) + m_3L_3^2C_{23}S_{23} \} \dot{\theta}_1^2 - \\ & \{ m_3L_3^2C_{23}S_{23} \} \dot{\theta}_3^2 + m_2gLCG_2C_2 + m_3g \{ L_2C_2 + LCG_3C_{23} \} \end{aligned} \quad (5.132)$$

Obteniendo la ecuación 3

$$T_3 = \frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_3} - \frac{\partial L}{\partial \theta_3}$$

$$\frac{\partial L}{\partial \theta_3} = \{ m_3L_3^2 \} \dot{\theta}_3 + \{ m_3L_3^2 + m_3L_2L_3C_3 \} \dot{\theta}_2 \quad (5.133)$$

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_3} = \{ m_3L_3^2 \} \ddot{\theta}_3 + \{ m_3L_3^2 + m_3L_2L_3C_3 \} \ddot{\theta}_2 - \{ m_3L_2L_3S_3\dot{\theta}_3 \} \dot{\theta}_2 \quad (5.134)$$

$$\frac{\partial}{\partial \theta_3} = - \{ m_3L_2L_3C_2S_{23} + m_3L_3^2S_{23}C_{23}^2 \} \dot{\theta}_1 - \{ m_3L_2L_3S_3 \} \dot{\theta}_2 - \{ m_3L_2L_3S_3 \} \dot{\theta}_2^2 \dot{\theta}_3 + m_3gLCG_3C_{23} \quad (5.135)$$

la ecuación de par torsor 3 es:

$$\begin{aligned} T_3 = & - \{ m_3L_3^2 + m_3L_2L_3C_3 \} \ddot{\theta}_2 - \{ m_3L_3^2 \} \ddot{\theta}_3 + \{ m_3L_2L_3C_2S_{23} + m_3L_3^2S_{23}C_{23} \} \dot{\theta}_1^2 + \{ m_3L_2L_3S_3 \} \dot{\theta}_2^2 \\ & + m_3gLCG_3C_{23} \end{aligned} \quad (5.136)$$

5.3.6. Representación matricial del MD.

La representación matricial del modelo dinámico se basa en la siguiente expresión

$$T = H(q) \ddot{q} + C(q, \dot{q}) \dot{q} + G(q) + F(\dot{q}) \quad (5.137)$$

Donde (Ver Cuadro 5.2):

Matriz de inercia

$$H(q) = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix} \quad (5.138)$$

Función	Descripción
$H(q)\ddot{q}$	Matriz de inercia
$C(q,\dot{q})\dot{q}$	Matriz de coriolis y centrípetas
$G(q)$	Vector de fuerzas gravitatorias
$F(\dot{q})$	Vector de fuerzas de fricción seca y viscosa

Cuadro 5.2: Descripción de las funciones de la representación matricial del modelo dinámico

Los elementos de la matriz de inercia son

$$\begin{aligned}
H_{11} &= (m_2 + m_3) L_2^2 C_2^2 + 2m_3 L_2 L_3 C_2 C_{23} + m_3 L_3^2 C_{23}^2 \\
H_{12} &= 0 \\
H_{13} &= 0 \\
H_{21} &= 0 \\
H_{22} &= (m_2 + m_3) L_2^2 + m_3 L_2^2 + 2m_3 L_2 L_3 C_2 \\
H_{23} &= m_3 L_3^2 + m_3 L_2 L_3 C_3 \\
H_{31} &= 0 \\
H_{32} &= m_3 L_3^2 + m_3 L_2 L_3 C_3 \\
H_{33} &= m_3 L_3^2
\end{aligned} \tag{5.139}$$

La matriz de fuerzas de coriolis y centrípetas, así como sus elementos se definen a continuación

$$C(q, \ddot{q}) = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \tag{5.140}$$

En donde:

$$\begin{aligned}
C_{11} &= 0 \\
C_{12} &= -\{2(m_2 + m_3) L_2^2 S_2 C_2 + 2m_3 L_2 L_3 (S_{23} C_2 + S_2 C_{23}) + 2m_3 L_3^2 S_{23} C_{23}\} \dot{\theta}_1 \\
C_{13} &= \{2m_3 L_2 L_3 s_{23} c_2 + 2m_3 L_3^2 s_{23} c_{23}\} \dot{\theta}_1 \\
C_{21} &= \{(m_2 + m_3) L_2^2 S_2 C_2 + m_3 L_2 L_3 (S_{23} C_2 + S_2 C_{23}) + m_3 L_3^2 S_{23} C_{23}\} \\
C_{22} &= -\{2m_3 L_2 L_3 s_3\} \dot{\theta}_3 \\
C_{23} &= -\{m_3 L_2 L_3 S_3\} \dot{\theta}_3 \\
C_{31} &= \{m_3 L_2 L_3 S_{23} C_2 + m_3 L_3^2 S_{23} C_{23}\} \dot{\theta}_1 \\
C_{32} &= \{m_3 L_2 L_3 S_3\} \dot{\theta}_2 \\
C_{33} &= 0
\end{aligned} \tag{5.141}$$

El vector de fuerzas gravitatorias se define a continuación

$$G(q) = \begin{bmatrix} G_{11} \\ G_{21} \\ G_{31} \end{bmatrix} \quad (5.142)$$

y sus elementos son:

$$\begin{aligned} G_{11} &= 0 \\ G_{21} &= -m_2 L_{CG2} C_2 - m_3 \{L_2 C_2 - L_{CG3} C_{23}\} \\ G_{31} &= -m_3 L_{CG3} C_{23} \end{aligned} \quad (5.143)$$

El MD descrito en la ecuación 5.137 ha tomado en cuenta la fricción viscosa y seca, este vector fue definido como $F(\dot{q})$ el cual es la suma del vector de fricción seca y el vector de fricción viscosa que a continuación se describe:

$$F(\dot{q}) = Fv(\dot{q}) + Fs(\dot{q}) \quad (5.144)$$

en donde (ver Cuadro 5.3):

Vector	Descripción
$Fv(\dot{q})$	Fricción viscosa
$Fs(\dot{q})$	Fricción seca

Cuadro 5.3: Descripción de los vectores de fricción.

$Fv(\dot{q})$ se describe a continuación

$$Fv(\dot{q}) = \begin{bmatrix} Fv_1(\dot{q}) \\ Fv_2(\dot{q}) \\ Fv_3(\dot{q}) \end{bmatrix} = \begin{bmatrix} b_1 \dot{\theta}_1 \\ b_2 \dot{\theta}_2 \\ b_2 \dot{\theta}_2 \end{bmatrix} \quad (5.145)$$

en donde

b_i : es el coeficiente de fricción viscosa o rozamiento dinámico.

$\dot{\theta}_1$: es la velocidad angular.

$F s(\dot{q})$ se describe a continuación

$$F s(\dot{q}) = \begin{bmatrix} F s_1(\dot{q}) \\ F s_2(\dot{q}) \\ F s_3(\dot{q}) \end{bmatrix} = \begin{bmatrix} K_1 \operatorname{sgn}(\dot{\theta}_1) \\ K_2 \operatorname{sgn}(\dot{\theta}_2) \\ K_2 \operatorname{sgn}(\dot{\theta}_3) \end{bmatrix} = \begin{bmatrix} K_1 \tanh(\beta_1 \dot{\theta}_1) \\ K_2 \tanh(\beta_2 \dot{\theta}_2) \\ K_2 \tanh(\beta_3 \dot{\theta}_3) \end{bmatrix} \quad (5.146)$$

en donde

K_i : es el coeficiente de fricción seca que depende del grado de lubricación de la articulación.

$\operatorname{sgn}(\cdot)$: es la función signo

$\dot{\theta}_i$: es la velocidad angular.

5.3.7. Propiedades dinámicas

Las propiedades del modelo dinámico son:

1. Simetría de la matriz de inercia

$$H(q) = H(q)^T \quad (5.147)$$

$$H(q)^T = \begin{bmatrix} H_{11} & H_{21} & H_{31} \\ H_{12} & H_{22} & H_{32} \\ H_{13} & H_{23} & H_{33} \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix} \quad (5.148)$$

2. La matriz de inercia es definida positiva

$$\det \{H(q)\} > 0 \quad (5.149)$$

3. Matriz antisimétrica

$$y^T (\dot{H}(q) - 2C(q, \dot{q})) y \cong 0 \quad (5.150)$$

Elemento	Descripción
y	$y \in \mathbb{R}^{3*1}$ cualquier vector y
y^T	Transpuesta del vector y
$\dot{H}(q)$	Derivada de la matriz de inercia
$C(q, \dot{q})$	Matriz de fuerzas de coriolis y centrífugas

Cuadro 5.4: Descripción de los elementos de la matriz asimétrica.

5.4. Control

5.4.1. Control PD

La simulación digital en MatLab, consiste en desarrollar el seguimiento de una trayectoria en el espacio operacional del robot virtual, la cinemática y dinámica esta expuesta en el capítulo 5 sección 2 y 3 respectivamente, las trayectorias deseadas de posición son 5.151 y 5.152:

$$x(t) = h + r \cos(wt) \quad (5.151)$$

$$y(t) = k + r \sen(wt) \quad (5.152)$$

Donde:

$$w = 0,4\pi$$

$$h = 0,3$$

$$k = 0,2$$

$$r = 0,1$$

Para dicha simulación se asume que el dispositivo (robot virtual) parte de un estado de reposo con una condición inicial de posición, que esta definida en:

$$q_1 = 45^\circ$$

$$q_2 = -45^\circ$$

$$q_3 = -45^\circ$$

las ganancias del control son:

$$kp = [144, 0, 0; 0, 144, 0; 0, 0, 144]$$

$$kd = [24, 0, 0; 0, 24, 0; 0, 0, 24]$$

Los resultados de las simulaciones en MatLab del control PD se presentan en una serie de de graficas que a continuación se describen:

En la Figura 5.5 se puede observar los valores del ángulo de la primera articulación al efectuar la tarea de seguimiento, además se puede ver que al inicio de la simulación dicha articulación trata de alcanzar los valores deseados lográndolo en menos de 1 seg. Y en el resto de la simulación los valores obtenidos por el robot son aproximados a los valores deseados.

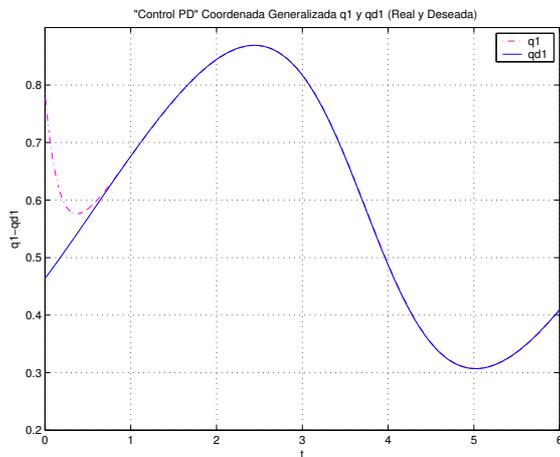


Figura 5.5: Control PD. Coordenada Generalizada q_1 y q_{d1} (Real y Deseada).

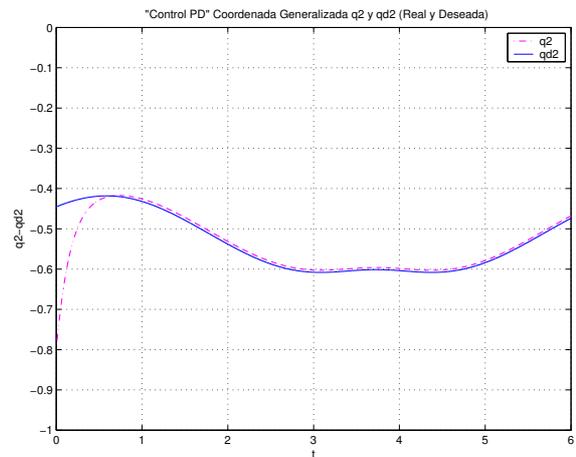


Figura 5.6: Control PD. Coordenada Generalizada q_2 y q_{d2} (Real y Deseada).

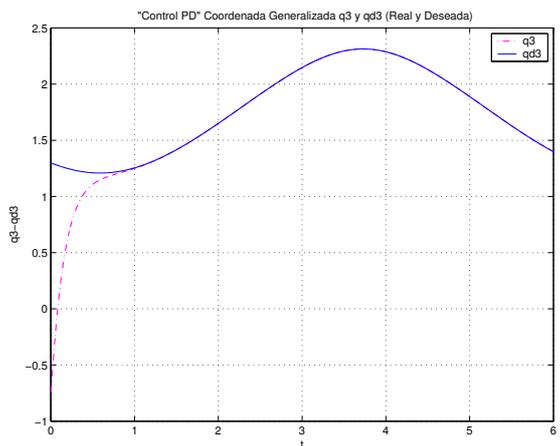


Figura 5.7: Control PD. Coordenada Generalizada q_3 y q_{d3} (Real y Deseada).

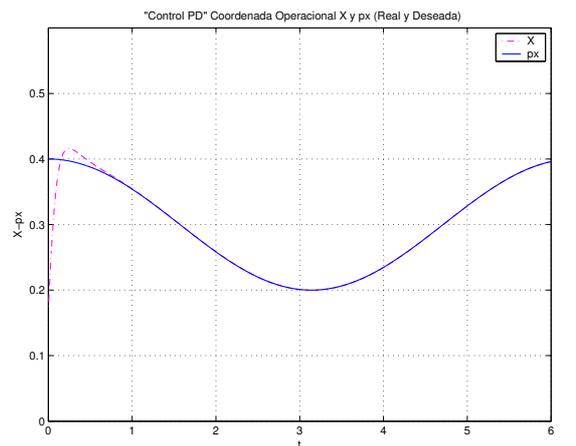


Figura 5.8: Control PD. Coordenada Operacional X y p_x (Real y Deseada).

Los valores del ángulo de la segunda articulación se pueden apreciar en la Figura 5.6, se logra notar que la gráfica de los valores deseados y los valores obtenidos por el robot están desfasadas durante toda la simulación, lo se traduce en un error.

Para la tercera articulación los valores del ángulo se pueden observar en la Figura 5.7, los valores obtenidos por la simulación se aproximan a los valores deseados en casi 1 seg. Y estos se mantienen semejantes el resto de la simulación.

Así como se presentan los ángulos de las articulaciones, también se muestran los valores de posición del efector final durante la simulación con respecto a los ejes cartesianos (x, y y z).

En la Figura 5.8, se pueden observar los valores que se obtuvieron en dicha simulación para el eje x , nótese que en la mayor parte de la simulación los valores obtenidos por el robot son aproximadamente iguales a los valores deseados.

Para el eje y, los valores obtenidos por el robot presentan una oscilación antes de alcanzar los valores deseados, una vez que estos son aproximadamente iguales se mantiene esta condición en el resto de la simulación (Ver Figura 5.9).

Debido a que el robot parte de una condición inicial, se puede apreciar que los valores obtenidos por este en la simulación alcanza los valores deseados que en toda la simulación son igual a cero (Figura 5.10).

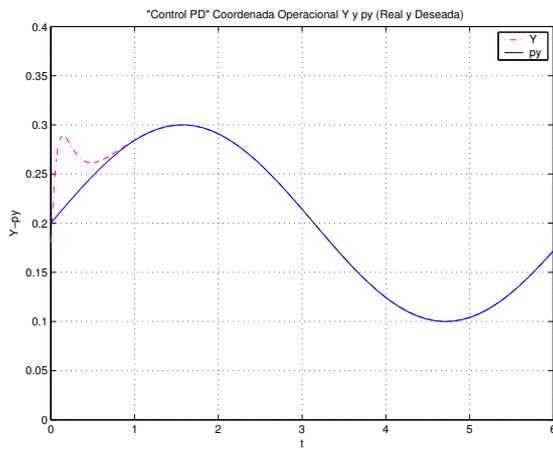


Figura 5.9: Control PD. Coordenada Operacional Y y py (Real y Deseada).

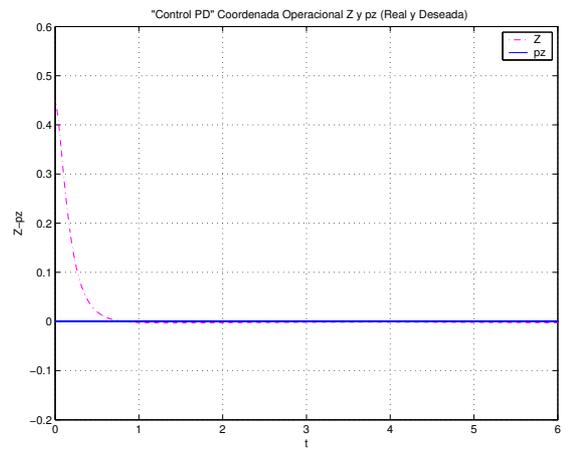


Figura 5.10: Control PD. Coordenada Operacional Z y pz (Real y Deseada).

Las trayectorias deseada y obtenida por el robot se pueden apreciar en la Figura 5.11 en donde se nota la semejanza entre ellas.

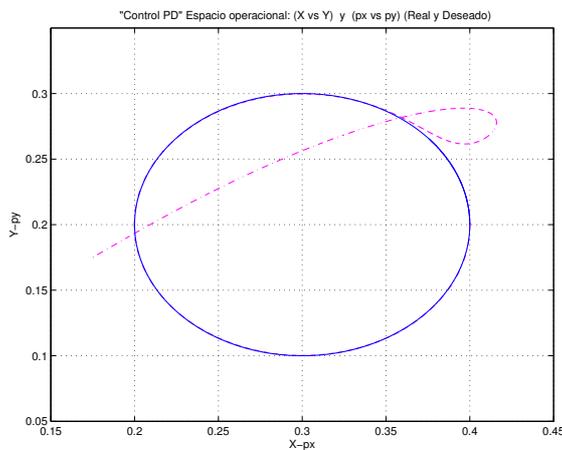


Figura 5.11: Control PD. Espacio operacional: (X vs Y) y (px vs py) (Real y Deseado).

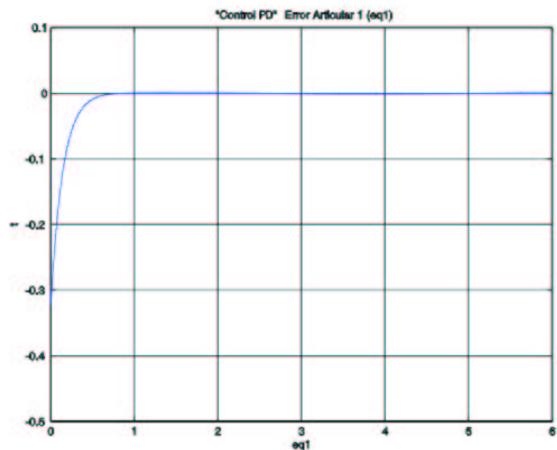


Figura 5.12: Control PD. Error Articular 1 (eq1).

Como se menciona anteriormente en la segunda articulación se aprecia un error (Figura 5.6). Dicho error se muestra de manera gráfica en la Figura 5.13 y dicho error es menor a 0.1.

La gráfica para el error de articulación 3 muestra que en la mayor parte de la simulación dicho error es 0 (Figura 5.14).

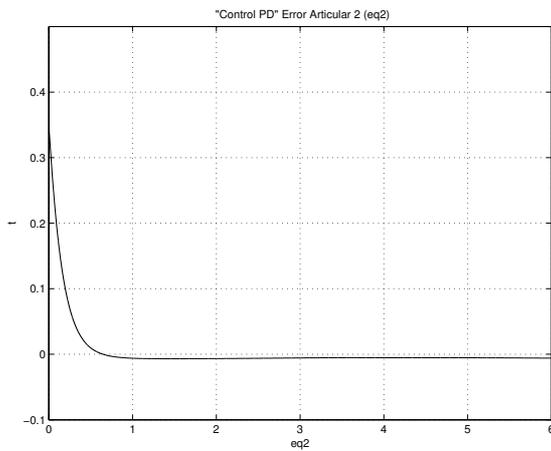


Figura 5.13: Control PD. Error Articular 2 (eq2).

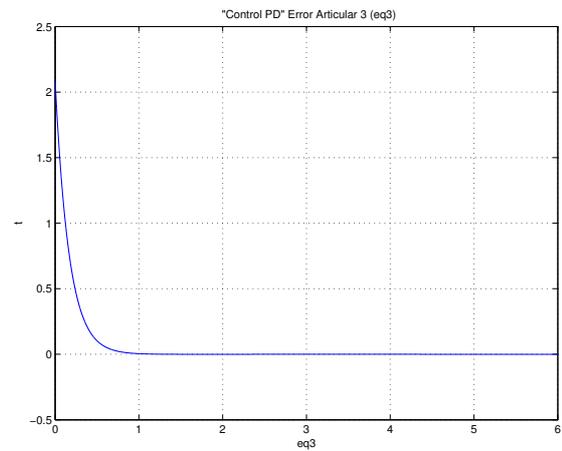


Figura 5.14: Control PD. Error Articular 3 (eq3).

Los errores operacionales en los tres ejes cartesianos son valores muy próximos a 0 (Figuras 5.15, 5.16, 5.17).

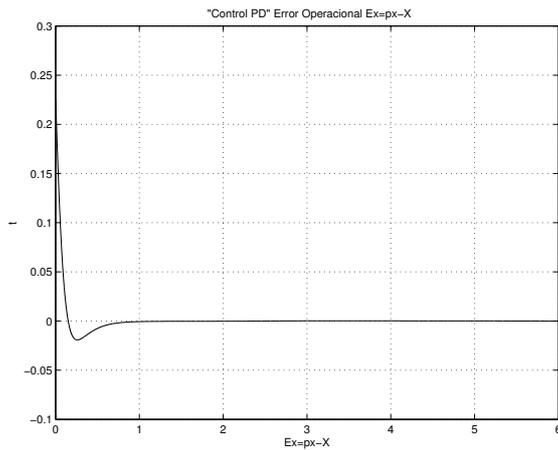


Figura 5.15: Control PD. Error Operacional $E_x=px-X$.

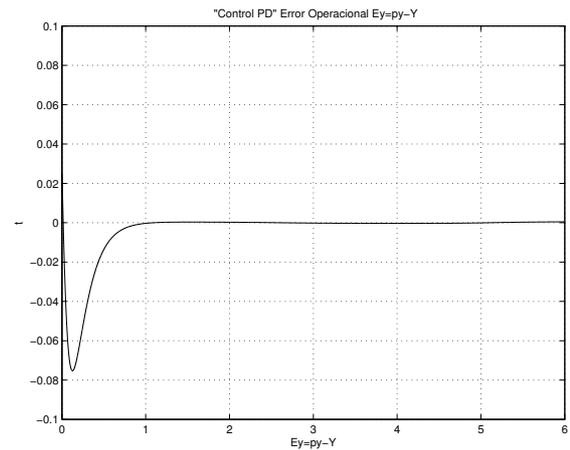


Figura 5.16: Control PD. Error Operacional $E_y=py-Y$.

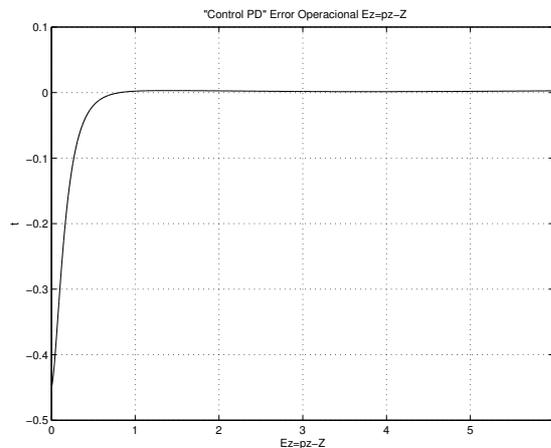


Figura 5.17: Control PD. Error Operacional $Ez=pz-Z$.

5.4.2. Control PID

La simulación digital en MatLab, consiste en desarrollar el seguimiento de una trayectoria en el espacio operacional del robot virtual, la cinemática y dinámica está expuesta en el Capítulo 5 sección 2 y 3 respectivamente, las trayectorias deseadas de posición son 5.151 y 5.152:

$$x(t) = h + r \cos(wt)$$

$$y(t) = k + r \sin(wt)$$

Donde:

$$w = 0,4\pi$$

$$h = 0,3$$

$$k = 0,2$$

$$r = 0,1$$

Para dicha simulación se asume que el dispositivo (robot virtual) parte de un estado de reposo con una condición inicial de posición, que esta definida en:

$$q_1 = 45^\circ$$

$$q_2 = -45^\circ$$

$$q_3 = -45^\circ$$

las ganancias del control son:

$$kp = [196, 0, 0; 0, 196, 0; 0, 0, 196]$$

$$kd = [28, 0, 0; 0, 28, 0; 0, 0, 28]$$

$$k_i = [2, 0, 0; 0, 16, 0; 0, 0, 2] / 200$$

Los resultados de las simulaciones en Matlab del control PID se presentan en una serie de de graficas que a continuación se describen:

Los valores del ángulo de la primera articulación se pueden observar en la Figura 5.18, dado que el robot parte de una condición inicial se puede ver como intenta alcanzar los valores deseados, y una vez que alcanza estos valores se mantienen aproximadamente iguales en el resto de la simulación.

Para la segunda articulación los valores del ángulo de dicha articulación se pueden observar en la Figura 5.19, así como en el control PD se observa un error (Figura 5.6) en el control PID también se nota, con la diferencia de que en este ultimo es menor.

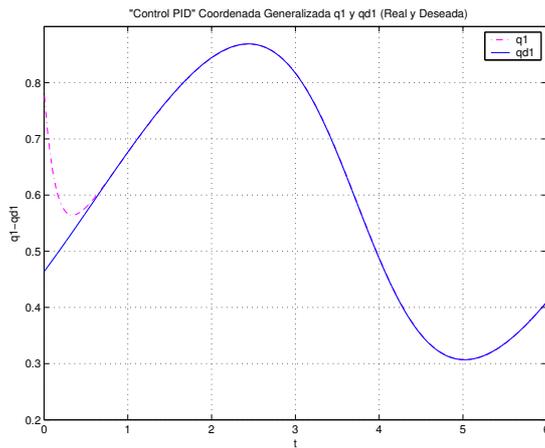


Figura 5.18: Control PID, Coordenada Generalizada q_1 y q_{d1} (Real y Deseada).

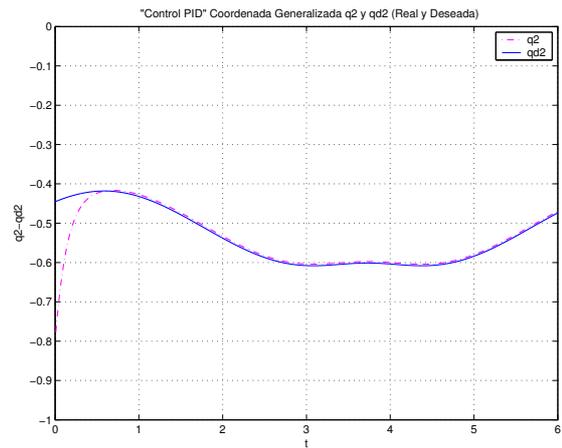


Figura 5.19: Control PID, Coordenada Generalizada q_2 y q_{d2} (Real y Deseada).

Los valores del ángulo de la tercera articulación se aprecian en la Figura 5.20, los valores obtenidos por el robot en la simulación se aproximan a los valores deseados en casi 1 seg. Y estos se mantienen semejantes el resto de la simulación.

De igual forma se presentan los valores de posición del efector final durante la simulación de la tarea de seguimiento, con respecto a los ejes x, y y z .

En la Figura 5.21, se pueden observar los valores que se obtuvieron en la simulación para el eje x, nótese que en la mayor parte de la simulación los valores obtenidos por el robot son aproximadamente iguales a los valores deseados, solo difieren al principio de la simulación debido a la condición inicial del robot y es por eso que se aprecia una pequeña oscilación.

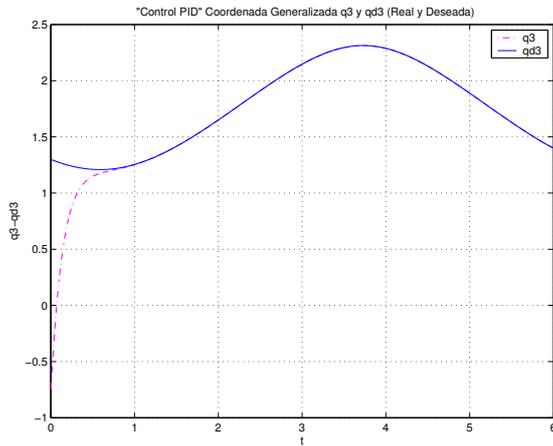


Figura 5.20: Control PID, Coordenada Generalizada q_3 y q_{d3} (Real y Deseada).

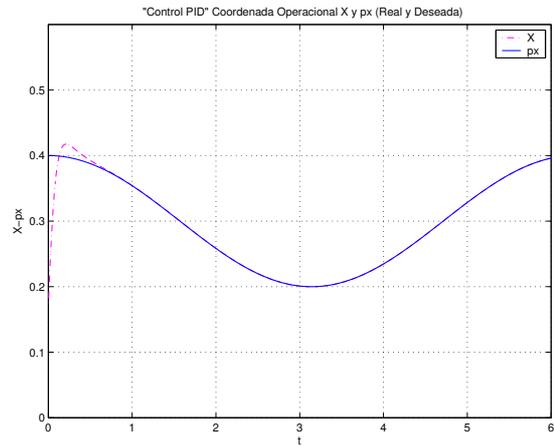


Figura 5.21: Control PID, Coordenada Operacional X y p_x (Real y Deseada).

Para el eje y, los valores obtenidos por el robot presentan una oscilación antes de alcanzar los valores deseados, una vez que estos son aproximadamente iguales se mantiene esta condición en el resto de la simulación (Ver Figura 5.22).

Debido a que el robot parte de una condición inicial, se puede apreciar que los valores obtenidos por este en la simulación alcanza los valores deseados que en toda la simulación son igual a cero (Figura 5.23).

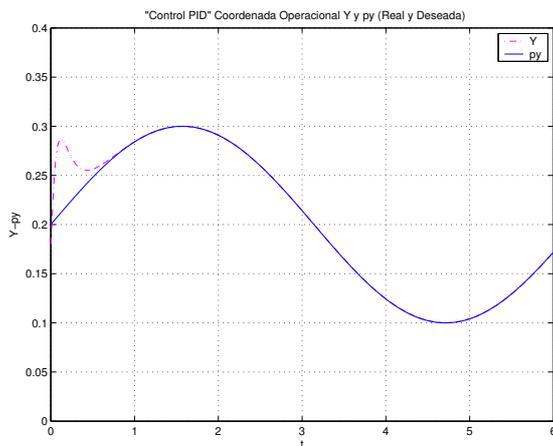


Figura 5.22: Control PID, Coordenada Operacional Y y p_y (Real y Deseada).

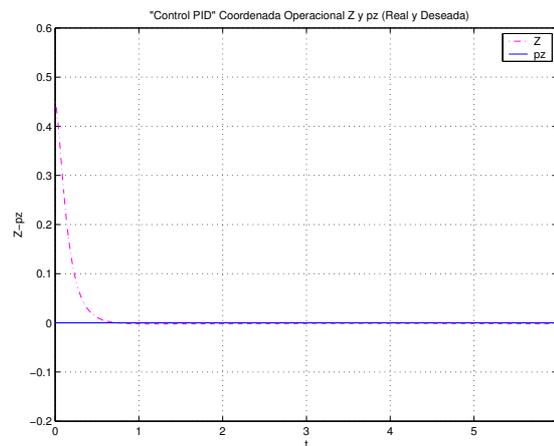


Figura 5.23: Control PID, Coordenada Operacional Z y p_z (Real y Deseada).

Las trayectorias deseada y obtenida por el robot se pueden apreciar en la figura 5.24 en donde se nota la semejanza entre ellas.

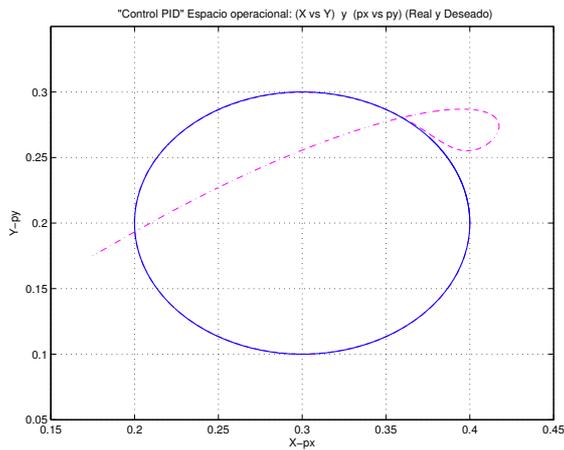


Figura 5.24: Control PID, Espacio operacional (X vs Y) y (px vs py)(Real y Deseado).

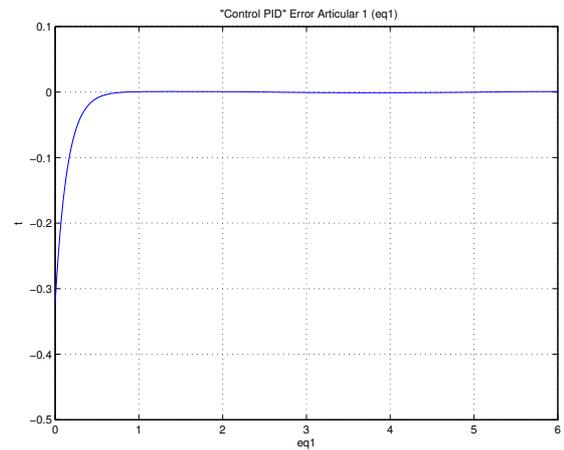


Figura 5.25: Control PID, Error Articular 1 (eq1)

Como se menciona anteriormente en la segunda articulación se aprecia un error (Figura 5.19). Dicho error se muestra de manera gráfica en la Figura 5.26 y dicho error es menor al que se observa en el control PD.

La gráfica para el error de articulación 3 muestra que en la mayor parte de la simulación dicho error es 0 (Figura 5.27).

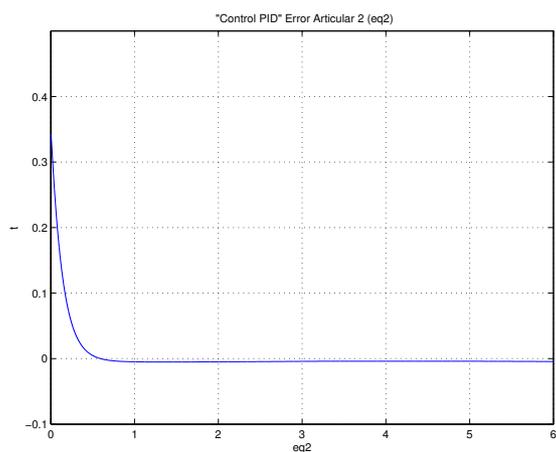


Figura 5.26: Control PID, Error Articular 2 (eq2)

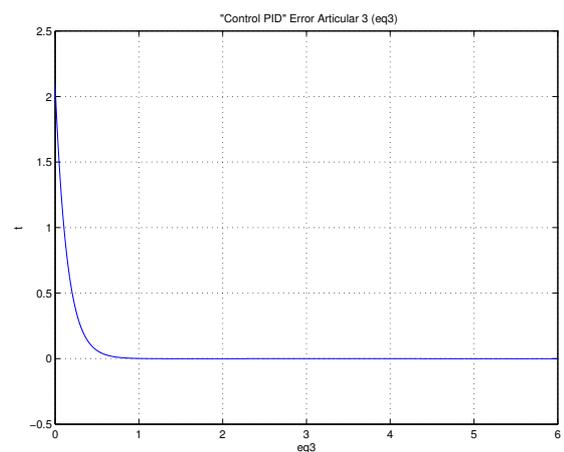


Figura 5.27: Control PID, Error Articular 3 (eq3)

Los errores operacionales en los tres ejes cartesianos son valores muy próximos a 0 (Figuras 5.28,5.29, 5.30).

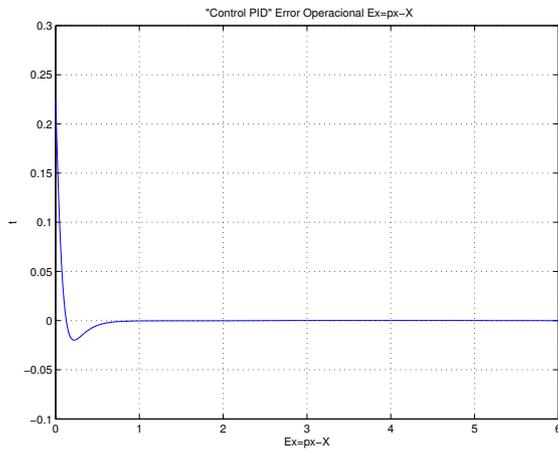


Figura 5.28: Control PID, Error Operacional ($E_x = p_x - X$)

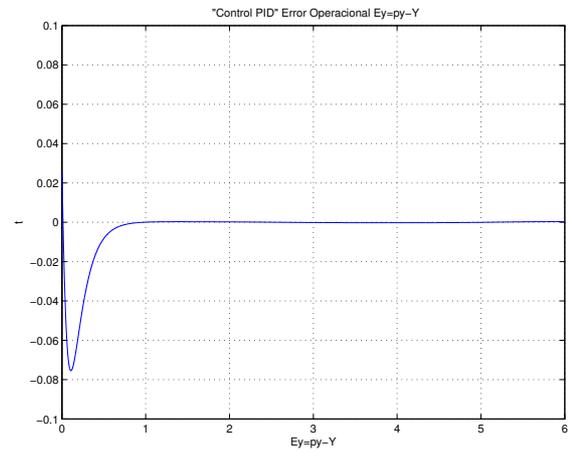


Figura 5.29: Control PID, Error Operacional ($E_y = p_y - Y$)

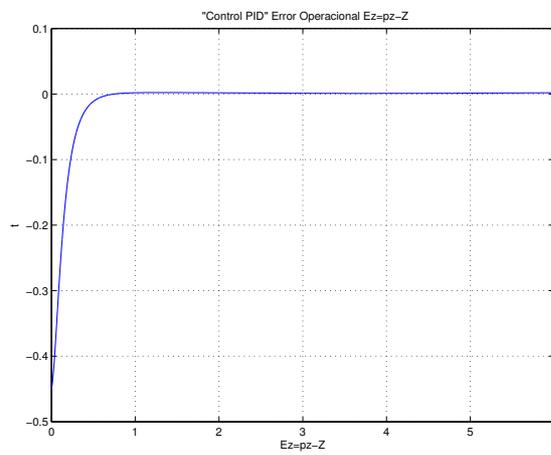


Figura 5.30: Control PID, Error Operacional ($E_z = p_z - Z$)

5.4.3. PIDNL

La simulación digital en MatLab, consiste en desarrollar el seguimiento de una trayectoria en el espacio operacional del robot virtual, la cinemática y dinámica está expuesta en el Capítulo 5 sección 2 y 3 respectivamente, las trayectorias deseadas de posición son 5.151 y 5.152:

$$x(t) = h + r \cos(wt)$$

$$y(t) = k + r \sin(wt)$$

Donde:

$$w = 0,4\pi$$

$$h = 0,3$$

$$k = 0,2$$

$$r = 0,1$$

Para dicha simulación se asume que el dispositivo (robot virtual) parte de un estado de reposo con una condición inicial de posición, que está definida en:

$$q_1 = 45^\circ$$

$$q_2 = -45^\circ$$

$$q_3 = -45^\circ$$

las ganancias del control son

$$k_p = [121, 0, 0; 0, 100, 0; 0, 0, 100]$$

$$k_d = [22, 0, 0; 0, 20, 0; 0, 0, 20]$$

$$k_i = k_p \times k_d / 10$$

En la Figura 5.31 se puede observar el desempeño de la coordenada generalizada q_1 aplicando un control PIDNL, además de poder realizar una comparación con respecto a los valores deseados de la misma (q_{d1}), se puede distinguir que existe un error ya que q_1 se encuentra desplazada ligeramente con respecto de q_{d1} , nótese que no existen oscilaciones al tratar de alcanzar el valor deseado por parte de q_1 .

En la Figura 5.32 se puede apreciar los valores obtenidos durante el desarrollo de la simulación, así como en la gráfica anterior se comparan los valores del robot virtual con los valores deseados para ésta coordenada generalizada q_2 , se puede notar que en el intervalo de tiempo de 0 a 0.5 existen oscilaciones de q_2 al tratar de lograr los valores deseados, en el intervalo de tiempo de 0.5 a 3 y de 4.5 a 7 se distingue un desplazamiento de q_2 con respecto a q_{d2} , lo que se interpreta como un error, en el intervalo de tiempo de 3 a 4.5 el error se reduce significativamente y se puede decir que en éste intervalo las gráficas son iguales.

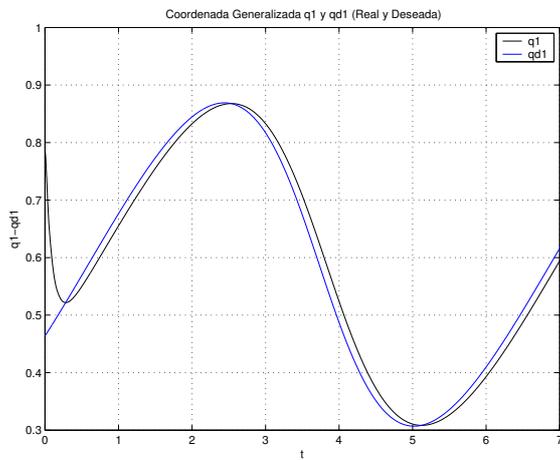


Figura 5.31: Control PIDNL, Coordenada Generalizada q1 y qd1 (Real y Deseada).

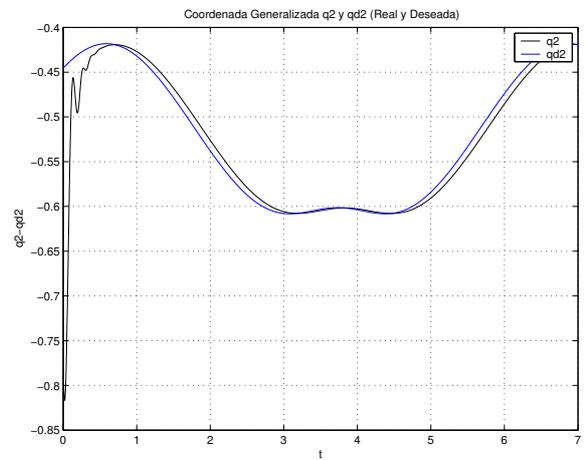


Figura 5.32: Control PIDNL, Coordenada Generalizada q2 y qd3 (Real y Deseada).

Como se ha mencionado en este trabajo de tesis, el robot virtual consta de 3 grados de libertad por lo cual se tiene otra coordenada generalizada q3, en la Figura 5.33 se muestra la gráfica de esta coordenada y los valores deseados para la misma al igual que en las coordenadas anteriores existe un valor entre los valores obtenidos y los deseados.

La trayectoria deseada es una circunferencia la cual se muestra en la Figura 5.34, además en esta figura se puede observar la trayectoria desarrollada por el robot virtual desde el punto de partida hasta el momento en el que se termina el trazo de la trayectoria.

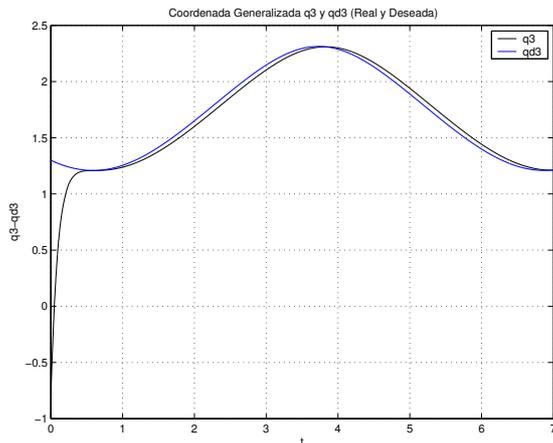


Figura 5.33: Control PIDNL, Coordenada Generalizada q3 y qd3 (Real y Deseada).

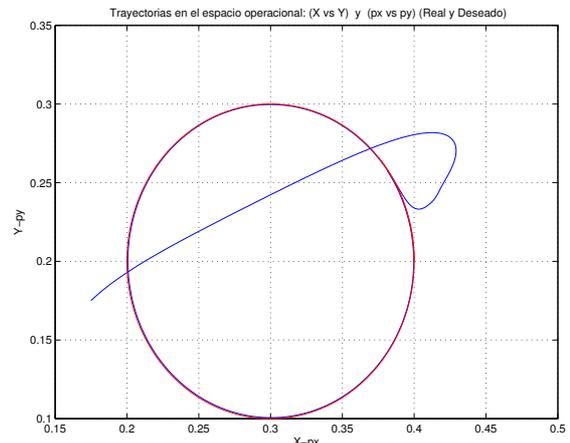


Figura 5.34: Control PIDNL, Trayectorias en el espacio operacional: (x,y) y (px,py), (Real y Deseada).

Como se ha mencionado anteriormente en las coordenadas generalizadas existen errores, que se pueden observar en la Figura 5.35, estos errores son muy pequeños y tienden a cero como se muestra.

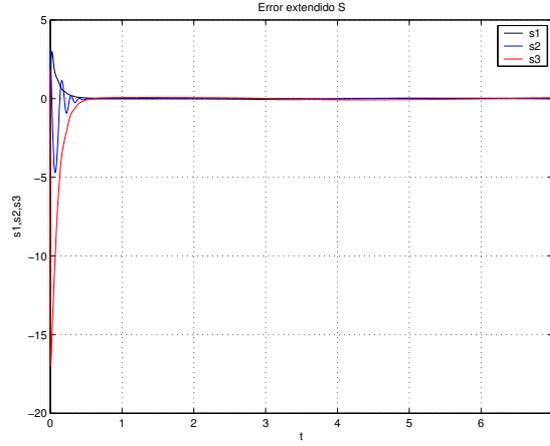


Figura 5.35: Control PIDNL, Error extendido S.

5.4.4. Control adaptable pasivo robusto

A continuación se efectúa la parametrización del modelo dinámico despreciando la fricción, aprovechando la propiedad $T = Y\theta$.

Donde:

Y : Representa a un regresor lineal que depende de posición y velocidad articular.

θ : Representa un vector de parámetros y que depende de las masas y longitudes de los eslabones. Haciendo $\theta = q$ (variable articular para evitar confusión con vector θ).

Para la ecuación 1:

$$\begin{aligned}
 T_1 = & \underbrace{((m_2 + m_3)L_2^2 C_2^2 + 2m_3L_2L_3 C_2C_{23} + m_3L_3^2 C_{23})}_{\theta_1} \ddot{q}_1 - \underbrace{(2(m_2 + m_3)L_2^2 S_2C_2 + 2m_3L_2L_3(S_{23}C_2 + S_2C_{23}))}_{\theta_1} \ddot{q}_2 \\
 & + \underbrace{2m_3L_3^2 S_{23}C_{23}}_{\theta_3} \dot{q}_1 \dot{q}_2 - \underbrace{(2m_3L_2L_3 S_{23}C_2 + 2m_3L_3^2 S_{23}C_{23})}_{\theta_2} \dot{q}_1 \dot{q}_3
 \end{aligned} \tag{5.153}$$

Para la ecuación 2:

$$\begin{aligned}
 T_2 = & \underbrace{((m_2 + m_3)L_2^2 + m_3L_3^2 + 2m_3L_2L_3 C_3)}_{\theta_1} \ddot{q}_2 + \underbrace{((m_3L_3^2 + m_3L_2L_3 C_{23}))}_{\theta_3} \ddot{q}_3 - \underbrace{(2m_3L_2L_3 S_3)}_{\theta_2} \dot{q}_2 \dot{q}_3 \\
 & + \underbrace{(m_2 + m_3)L_2^2 S_2C_2 + m_3L_2L_3(C_2S_{23} + C_{23}S_2)}_{\theta_1} \ddot{q}_1 - \underbrace{m_3L_3^2 S_{23}C_{23}}_{\theta_3} \dot{q}_1^2 - \underbrace{(m_3L_2L_3 S_3)}_{\theta_2} \dot{q}_3^2 \\
 & - \underbrace{(m_2LCG_2 + m_3L_2)}_{\theta_4} gC_2 - \underbrace{(m_3LCG_3)}_{\theta_5} gC_{23}
 \end{aligned} \tag{5.154}$$

Para la ecuación 3:

$$T_3 = \underbrace{(m_3 L_3^2)}_{\theta_3} + \underbrace{m_3 L_2 L_3 C_3}_{\theta_2} \ddot{q}_2 + \underbrace{(m_3 L_3^2)}_{\theta_3} \ddot{q}_3 + \underbrace{(m_3 L_2 L_3 S_3 C_2)}_{\theta_2} + \underbrace{m_3 L_3^2 S_{23} C_{23}}_{\theta_3} \dot{q}_1^2 + \underbrace{(m_3 L_2 L_3 S_3)}_{\theta_2} \dot{q}_2^2 - \underbrace{(m_3 L C G_3)}_{\theta_5} g C_{23} \quad (5.155)$$

Expresando el modelo dinámico en términos de los parámetros θ_1 , θ_2 , θ_3 , θ_4 y θ_5 , donde su equivalencia en función de masas y longitudes del robot se especifican posteriormente.

Ecuación 1:

$$T_1 = (\theta_1 C_2^2 + \theta_2 C_2 C_{23} + \theta_3 C_{23}^2) \ddot{q}_1 - (2\theta_1 S_2 C_2 + 2\theta_2 (S_{23} C_2 + S_2 C_{23}) + 2\theta_3 S_{23} C_{23}) \dot{q}_1 \dot{q}_2 - (2\theta_2 S_{23} C_2 + 2\theta_3 S_{23} C_{23}) \dot{q}_1 \dot{q}_3$$

Desarrollando la ecuación

$$T_1 = \theta_1 C_2^2 \ddot{q}_1 + \theta_2 C_2 C_{23} \ddot{q}_1 + \theta_3 C_{23}^2 \ddot{q}_1 - 2\theta_1 S_2 C_2 \dot{q}_1 \dot{q}_2 - 2\theta_2 \dot{q}_1 \dot{q}_2 (S_{23} C_2 + S_2 C_{23}) + 2\theta_3 S_{23} C_{23} \dot{q}_1 \dot{q}_2 - 2\theta_2 S_{23} C_2 \dot{q}_1 \dot{q}_3 - 2\theta_3 S_{23} C_{23} \dot{q}_1 \dot{q}_3$$

Factorizando

$$T_1 = \theta_1 (C_2^2 \ddot{q}_1 - 2S_2 C_2 \dot{q}_1 \dot{q}_2) + \theta_2 (2C_2 C_{23} \ddot{q}_1 - 2(S_{23} C_2 + S_2 C_{23}) \dot{q}_1 \dot{q}_2 - 2S_{23} C_2 \dot{q}_1 \dot{q}_3) + \theta_3 (C_{23}^2 \ddot{q}_1 - 2S_{23} C_{23} \dot{q}_1 \dot{q}_2 - 2S_{23} C_{23} \dot{q}_1 \dot{q}_3) \quad (5.156)$$

Ecuación 2:

$$T_2 = (\theta_1 + \theta_3 + 2\theta_2 C_3) \ddot{q}_2 + (\theta_3 + \theta_2 C_3) \ddot{q}_3 - (2\theta_2 S_3) \dot{q}_2 \dot{q}_3 + (\theta_1 S_2 C_2 + \theta_2 (C_2 S_{23} + C_{23} S_2) + \theta_3 S_{23} C_{23}) \dot{q}_1^2 - (\theta_2 S_3) \dot{q}_3^2 - \theta_4 g C_2 - \theta_5 g C_{23}$$

Desarrollando:

$$T_2 = \theta_1 \ddot{q}_2 + \theta_3 \ddot{q}_2 + 2\theta_2 C_3 \ddot{q}_2 + \theta_3 \ddot{q}_3 + \theta_2 C_3 \ddot{q}_3 - 2\theta_2 S_3 \dot{q}_2 \dot{q}_3 + \theta_1 S_2 C_2 \dot{q}_1^2 + \theta_2 (C_2 S_{23} + C_{23} S_2) \dot{q}_1^2 + \theta_3 S_{23} C_{23} \dot{q}_1^2 - \theta_2 S_3 \dot{q}_3^2 - \theta_4 g C_2 - \theta_5 g C_{23}$$

Factorizando:

$$T_2 = \theta_1(\ddot{q}_2 S_2 C_2 \dot{q}_1^2) + \theta_2(2C_3 \ddot{q}_2 + C_3 \ddot{q}_3 - 2S_3 \dot{q}_2 \dot{q}_3 + (S_{23} C_2 + C_{23} S_2) \dot{q}_1^2 - S_3 \dot{q}_3^2) + \theta_3(\ddot{q}_2 + \ddot{q}_3 + S_{23} C_{23} \dot{q}_1^2) + \theta_4(-gC_2) - \theta_5(-gC_{23}) \quad (5.157)$$

Ecuación 3:

$$T_3 = (\theta_3 + \theta_2 C_3) \ddot{q}_2 + (\theta_3) \ddot{q}_3 + (\theta_2 S_{23} C_2 + \theta_3 S_{23} C_{23}) \dot{q}_1^2 + (\theta_2 S_3) \dot{q}_3^2 - (\theta_5) g C_{23}$$

Desarrollando

$$T_3 = \theta_3 \ddot{q}_2 + \theta_2 C_3 \ddot{q}_2 + \theta_3 \ddot{q}_3 + \theta_2 S_{23} C_2 \dot{q}_1^2 + \theta_3 S_{23} C_{23} \dot{q}_1^2 + \theta_2 S_3 \dot{q}_3^2 - \theta_5 g C_{23}$$

Factorizando

$$T_3 = \theta_2(C_3 \ddot{q}_2 + S_{23} C_2 \dot{q}_1^2 + S_3 \dot{q}_3^2) + \theta_3(\ddot{q}_2 + \ddot{q}_3 + S_{23} C_{23} \dot{q}_1^2) + \theta_5(-gC_{23}) \quad (5.158)$$

La representación matricial, equivalente del modelo dinámico es:

$$\begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix}^{3 \times 1} = \begin{bmatrix} Y_1(q, \dot{q}, \ddot{q}) \\ Y_2(q, \dot{q}, \ddot{q}) \\ Y_3(q, \dot{q}, \ddot{q}) \end{bmatrix}^{3 \times 5} * \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \end{bmatrix}^{5 \times 1} \quad (5.159)$$

Donde:

$$\theta_1 = (m_2 + m_3)L_2^2$$

$$\theta_2 = m_3 L_2 L_3$$

$$\theta_3 = m_3 L_3^2$$

$$\theta_4 = m_2 L C G_2 + m_3 L_2$$

$$\theta_5 = m_3 L C G_3$$

La representación matricial es la siguiente:

$$T = \begin{bmatrix} Y_{11} & Y_{12} & Y_{13} & Y_{14} & Y_{15} \\ Y_{21} & Y_{22} & Y_{23} & Y_{24} & Y_{25} \\ Y_{31} & Y_{32} & Y_{33} & Y_{34} & Y_{35} \end{bmatrix} \times \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \end{bmatrix} \quad (5.160)$$

$$T = Y(q, \dot{q}, \ddot{q})\theta \quad (5.161)$$

Donde:

$Y(q, \dot{q}, \ddot{q})$: regresor

θ : vector de Parámetros.

También donde:

$$Y_{11} = C_2^2 \ddot{q}_1 - 2S_2 C_2 \dot{q}_1 \dot{q}_2 \quad (5.162)$$

$$Y_{12} = 2C_2 C_{23} \ddot{q}_1 - 2(S_{23} C_2 + S_2 C_{23}) \dot{q}_1 \dot{q}_2 - 2S_2 C_2 \dot{q}_1 \dot{q}_3 \quad (5.163)$$

$$Y_{13} = 2C_{23}^2 \ddot{q}_1 - 2S_{23} C_{23} \dot{q}_1 \dot{q}_2 - 2S_{23} C_{23} \dot{q}_1 \dot{q}_3 \quad (5.164)$$

$$Y_{14} = 0 \quad (5.165)$$

$$Y_{15} = 0 \quad (5.166)$$

$$Y_{21} = \ddot{q}_2 + S_2 C_2 \dot{q}_1^2 \quad (5.167)$$

$$Y_{22} = 2C_3 \ddot{q}_2 + C_3 \ddot{q}_3 - 2S_3 \dot{q}_2 \dot{q}_3 + (S_{23} C_2 + S_2 C_{23}) \dot{q}_1^2 - S_3 \dot{q}_3^2 \quad (5.168)$$

$$Y_{23} = \ddot{q}_2 + \ddot{q}_3 + S_{23} C_{23} \dot{q}_1^2 \quad (5.169)$$

$$Y_{24} = -gC_2 \quad (5.170)$$

$$Y_{25} = -gC_{23} \quad (5.171)$$

$$Y_{13} = 0 \quad (5.172)$$

$$Y_{32} = C_3\ddot{q}_2 + S_{23}C_2\dot{q}_1^2 + S_3\dot{q}_2^2 \quad (5.173)$$

$$Y_{33} = \ddot{q}_2 + \ddot{q}_3 + S_{23}C_{23}\dot{q}_1^2 \quad (5.174)$$

$$Y_{34} = 0 \quad (5.175)$$

$$Y_{35} = -gC_{23} \quad (5.176)$$

Parametrización lineal del modelo dinámico

Tomando en cuenta la fricción tenemos:

$$H(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + F(q) = T$$

$$Y(q, \dot{q}, \ddot{q})\hat{\theta} = T$$

$$H(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + F(q) = Y\hat{\theta} \quad (5.177)$$

Entonces tenemos el modelo dinámico con fricción a continuación:

Ecuación 1:

$$\begin{aligned} T_1 = & \underbrace{((m_2 + m_3)L_2^2 C_2^2)}_{\theta_1} + 2 \underbrace{m_3 L_2 L_3 C_2 C_{23}}_{\theta_2} + \underbrace{m_3 L_3^2 C_{23}^2}_{\theta_3} \ddot{\theta}_1 - (2 \underbrace{(m_2 + m_3)L_2^2 S_2 C_2}_{\theta_1} \\ & + 2 \underbrace{m_3 L_2 L_3 (S_{23} C_2 + S_2 C_{23})}_{\theta_2} + 2 \underbrace{m_3 L_3^2 S_{23} C_{23}}_{\theta_3}) \dot{\theta}_1 \dot{\theta}_2 - \underbrace{(2m_3 L_2 L_3 S_{23} C_2}_{\theta_2} \\ & + 2 \underbrace{m_3 L_3^2 S_{23} C_{23}}_{\theta_3}) \dot{\theta}_1 \dot{\theta}_3 + \underbrace{b_1}_{\theta_4} \dot{\theta}_1 + \tanh(\underbrace{\beta, \theta_1}_{\theta_5}) \end{aligned} \quad (5.178)$$

Ecuación 2:

$$\begin{aligned}
T_2 = & \underbrace{(m_2 + m_3)L_2^2}_{\theta_1} + \underbrace{m_3L_3^2}_{\theta_3} + 2 \underbrace{m_3L_2L_3}_{\theta_2} C_3 \ddot{\theta}_2 + \underbrace{(m_3L_3^2)}_{\theta_3} + \underbrace{m_3L_2L_3}_{\theta_2} C_3 \ddot{\theta}_3 \\
& - \underbrace{(2m_3L_2L_3S_3)}_{\theta_2} \dot{\theta}_2 \dot{\theta}_3 + \underbrace{((m_2 + m_3)L_2^2)}_{\theta_1} S_2 C_2 + \underbrace{m_3L_2L_3}_{\theta_2} (C_2 S_{23} + C_{23} S_2) \\
& + \underbrace{m_3L_3^2 C_{23}}_{\theta_3} S_{23} \dot{\theta}_1^2 - \underbrace{(m_3L_2L_3S_3)}_{\theta_2} \dot{\theta}_3^2 - \underbrace{(m_2LCG_2 + m_3L_2)}_{\theta_6} g C_2 \\
& - \underbrace{m_3gLCG_3C_{23}}_{\theta_7} + \underbrace{b_2\dot{\theta}_2}_{\theta_8} + \tanh(\beta, \underbrace{\dot{\theta}_2}_{\theta_9})
\end{aligned} \tag{5.179}$$

Ecuación 3:

$$\begin{aligned}
T_3 = & \underbrace{(m_3L_3^2)}_{\theta_3} + \underbrace{m_3L_2L_3}_{\theta_2} C_3 \ddot{\theta}_2 + \underbrace{(m_3L_3^2)}_{\theta_3} \ddot{\theta}_3 + \underbrace{(m_3L_2L_3)}_{\theta_2} C_2 S_{23} + m_3L_3^2 C_{23} S_{23} \dot{\theta}_1^2 \\
& + \underbrace{(m_3L_2L_3S_3)}_{\theta_2} \dot{\theta}_2^2 - \underbrace{m_3LCG_3C_{23}}_{\theta_7} + \underbrace{b_3\dot{\theta}_3}_{\theta_{10}} + \tanh(\beta, \dot{\theta}_3)
\end{aligned} \tag{5.180}$$

Expresando el modelo dinámico en términos de los parámetros θ :

Ecuación 1:

$$\begin{aligned}
T_1 = & (\theta_1 C_2^2 + 2\theta_2 C_2 C_{23} + \theta_3 C_{23}) \ddot{q}_1 - (2\theta_1 S_2 C_2 + 2\theta_2 (S_{23} C_2 + S_2 C_{23}) + 2\theta_3 S_{23} C_{23}) \dot{q}_1 \dot{q}_2 \\
& - (2\theta_2 S_{23} C_2 + 2\theta_3 S_{23} C_{23}) \dot{q}_1 \dot{q}_3 + (\theta_4) \dot{q}_1 + K_1 \tanh(\theta_5 \dot{q}_1)
\end{aligned}$$

Desarrollando

$$\begin{aligned}
T_1 = & \theta_1 C_2^2 \ddot{q}_1 + 2\theta_2 C_2 C_{23} \ddot{q}_1 + \theta_3 C_{23} \ddot{q}_1 - 2\theta_1 S_2 C_2 \dot{q}_1 \dot{q}_2 - 2\theta_2 \dot{q}_1 \dot{q}_2 (S_{23} C_2 + S_2 C_{23}) + 2\theta_3 S_{23} C_{23} \dot{q}_1 \dot{q}_2 \\
& - (2\theta_2 S_{23} C_2 \dot{q}_1 \dot{q}_3 - 2\theta_3 S_{23} C_{23} \dot{q}_1 \dot{q}_3 + \theta_4 \dot{q}_1 + K_1 \tanh(\theta_5 \dot{q}_1))
\end{aligned}$$

Factorizando θ :

$$\begin{aligned}
T_1 = & \theta_1 (C_2^2 \ddot{q}_1 - 2S_2 C_2 \dot{q}_1 \dot{q}_2) + \theta_2 (2C_2 C_{23} \ddot{q}_1 - 2\dot{q}_1 \dot{q}_2 (S_{23} C_2 + S_2 C_{23}) - 2S_{23} C_2 \dot{q}_1 \dot{q}_2) + \theta_3 (C_{23}^2 \ddot{q}_1 \\
& + 2S_{23} C_{23} \dot{q}_1 \dot{q}_2 - 2S_{23} C_{23} \dot{q}_1 \dot{q}_3 + \theta_4 \dot{q}_1 + K_1 \tanh(\theta_5 \dot{q}_1))
\end{aligned} \tag{5.181}$$

Ecuación 2:

$$T_2 = (\theta_1 + \theta_3 + 2\theta_2 C_3)\ddot{q}_2 + (\theta_3 + \theta_2 C_3)\ddot{q}_3 - (2\theta_2 S_3)\dot{q}_1\dot{q}_2 + (\theta_1 S_2 C_2 + \theta_2(S_{23}C_2 + S_2 C_{23}) + \theta_3 C_{23} S_{23})\dot{q}_1^2 - (\theta_2)\dot{q}_3^2 - (\theta_6)gC_2 - (\theta_7)gC_{23} + \theta_8\dot{q}_2 + K_2 \tanh(\theta_9\dot{q}_2)$$

Desarrollando:

$$T_2 = \theta_1\ddot{q}_2 + \theta_3\ddot{q}_2 + 2\theta_2 C_3\ddot{q}_2 + \theta_3\ddot{q}_3 + \theta_2 C_3\ddot{q}_3 - 2\theta_2 S_3\dot{q}_1\dot{q}_2 + \theta_1 S_2 C_2\dot{q}_1^2 + \theta_2(S_{23}C_2 + S_2 C_{23})\dot{q}_1^2 + \theta_3 C_{23} S_{23}\dot{q}_1^2 - \theta_2\dot{q}_3^2 - \theta_6 g C_2 - C_{23}\theta_7 g + \theta_8\dot{q}_2 + K_2 \tanh(\theta_9\dot{q}_2)$$

Factorizando:

$$T_2 = \theta_1(\ddot{q}_2 + S_2 C_2\dot{q}_1^2) + \theta_2(2C_3\ddot{q}_2 + C_3\ddot{q}_3 - 2S_3\dot{q}_1\dot{q}_2 + \dot{q}_1^2(S_{23}C_2 + S_2 C_{23}) - S_3\dot{q}_3^2 + \theta_3(\ddot{q}_2 + \ddot{q}_3 + C_{23}S_{23}\dot{q}_1^2) + \theta_6(-gC_2 - \theta_7(-gC_{23}) + \theta_8(\dot{q}_2) + K_2 \tanh(\theta_9\dot{q}_2) \quad (5.182)$$

Ecuación 3:

$$T_3 = (\theta_3 + \theta_2 C_3)\ddot{q}_2 + (\theta_3)\ddot{q}_3 + (\theta_2 C_2 S_{23} + \theta_3 S_{23} C_{23})\dot{q}_1^2 + (\theta_2 S_3)\dot{q}_2^2 - (\theta_7 C_{23} + (\theta_{10}\dot{q}_3 + K_3 \tanh(\theta_{11}\dot{q}_3)$$

Desarrollando:

$$T_3 = \theta_3\ddot{q}_2 + \theta_2 C_3\ddot{q}_2 + \theta_3\ddot{q}_3 + \theta_2 C_2 S_{23}\dot{q}_1^2 + \theta_3 S_{23} C_{23}\dot{q}_1^2 + \theta_2 S_3\dot{q}_2^2 - \theta_7 C_{23} + \theta_{10}\dot{q}_3 + K_3 \tanh(\theta_{11}\dot{q}_3)$$

Factorizando:

$$T_3 = \theta_2(C_3\ddot{q}_2 + S_{23}C_2\dot{q}_1^2 + S_3\dot{q}_2^2) + \theta_3(\ddot{q}_2 + \ddot{q}_3 + C_{23}S_{23}\dot{q}_1^2) - \theta_7(C_{23}) + \theta_{10}(\dot{q}_3) + K_3 \tanh(\theta_{11}\dot{q}_3) \quad (5.183)$$

Representación matricial de acuerdo a :

$$\begin{bmatrix} T_1 \\ T_1 \\ T_1 \end{bmatrix}^{3 \times 1} = \begin{bmatrix} Y_1(q, \dot{q}, \ddot{q}) \\ Y_2(q, \dot{q}, \ddot{q}) \\ Y_3(q, \dot{q}, \ddot{q}) \end{bmatrix}^{3 \times 11} * \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \\ \theta_6 \\ \theta_7 \\ \theta_8 \\ \theta_9 \\ \theta_{10} \\ \theta_{11} \end{bmatrix}^{11 \times 1} \quad (5.184)$$

Entonces:

$$T = \begin{bmatrix} Y_{11} & Y_{12} & Y_{13} & Y_{14} & Y_{15} & Y_{16} & Y_{17} & Y_{18} & Y_{19} & Y_{110} & Y_{111} \\ Y_{21} & Y_{22} & Y_{23} & Y_{24} & Y_{25} & Y_{26} & Y_{27} & Y_{28} & Y_{29} & Y_{210} & Y_{211} \\ Y_{31} & Y_{32} & Y_{33} & Y_{34} & Y_{35} & Y_{36} & Y_{37} & Y_{38} & Y_{39} & Y_{310} & Y_{311} \end{bmatrix}$$

Donde:

$$Y_{11} = C_2^2 \ddot{q}_1 - 2S_2 C_2 \dot{q}_1 \dot{q}_2 \quad (5.185)$$

$$Y_{12} = 2C_2 C_{23} \ddot{q}_1 - 2\dot{q}_1 \dot{q}_2 (S_{23} C_2 + S_2 C_{23}) - 2S_{23} C_{23} \dot{q}_1 \dot{q}_3 \quad (5.186)$$

$$Y_{13} = C_{23}^2 \ddot{q}_1 + 2S_{23} C_{23} \dot{q}_1 \dot{q}_2 - 2S_{23} C_{23} \dot{q}_1 \dot{q}_3 \quad (5.187)$$

$$Y_{14} = \dot{q}_1 \quad (5.188)$$

$$Y_{15} = \tanh(\beta, \dot{q}_1) \quad (5.189)$$

$$Y_{16} = 0 \quad (5.190)$$

$$Y_{17} = 0 \quad (5.191)$$

$$Y_{18} = 0 \quad (5.192)$$

$$Y_{19} = 0 \quad (5.193)$$

$$Y_{110} = 0 \quad (5.194)$$

$$Y_{111} = 0 \quad (5.195)$$

$$Y_{21} = \ddot{q}_2 + S_2 C_2 \dot{q}_1^2 \quad (5.196)$$

$$Y_{22} = 2C_3\ddot{q}_2 + C_3\ddot{q}_3 - 2S_3\dot{q}_1\dot{q}_2\dot{q}_1^2(S_{23}C_2 + S_2C_{23}) - S_3\dot{q}_3^2 \quad (5.197)$$

$$Y_{23} = \ddot{q}_2 + \ddot{q}_3 + C_{23}S_{23}\dot{q}_1^2 \quad (5.198)$$

$$Y_{24} = 0 \quad (5.199)$$

$$Y_{25} = 0 \quad (5.200)$$

$$Y_{26} = -gC_2 \quad (5.201)$$

$$Y_{27} = -gC_{23} \quad (5.202)$$

$$Y_{28} = \dot{q}_2 \quad (5.203)$$

$$Y_{29} = \tanh(\beta, \dot{q}_2) \quad (5.204)$$

$$Y_{210} = 0 \quad (5.205)$$

$$Y_{211} = 0 \quad (5.206)$$

$$Y_{31} = 0 \quad (5.207)$$

$$Y_{32} = C_3\ddot{q}_2 + S_{23}C_2\dot{q}_1^2 + S_3\dot{q}_1^2 \quad (5.208)$$

$$Y_{33} = \ddot{q}_2 + \ddot{q}_3 + C_{23}S_{23}\dot{q}_1^2 \quad (5.209)$$

$$Y_{34} = 0 \quad (5.210)$$

$$Y_{35} = 0 \quad (5.211)$$

$$Y_{36} = 0 \quad (5.212)$$

$$Y_{37} = C_{23} \quad (5.213)$$

$$Y_{38} = 0 \quad (5.214)$$

$$Y_{39} = 0 \quad (5.215)$$

$$Y_{310} = \dot{q}_3 \quad (5.216)$$

$$Y_{311} = \tanh(\beta, \dot{q}_3) \quad (5.217)$$

En la Figura 5.36 se pueden apreciar los valores que se obtuvieron durante la simulación, de la coordenada generalizada q1, al mismo tiempo se muestra una comparación gráfica con respecto a los valores que se desea obtener, a diferencia del control PIDNL se nota como las gráficas son iguales a excepción del inicio de la simulación debido a la condición inicial del robot, el error tiende a cero lo cual no ocurre en el control PIDNL.

En la Figura 5.37 se puede observar la representación gráfica de los valores de la coordenada generalizada q2 y de los valores deseados qd2, se puede apreciar que en la mayor parte del tiempo de simulación el error es cero, hecho que no ocurre con el control PIDNL ya que en este sólo ocurre en un intervalo de la simulación.

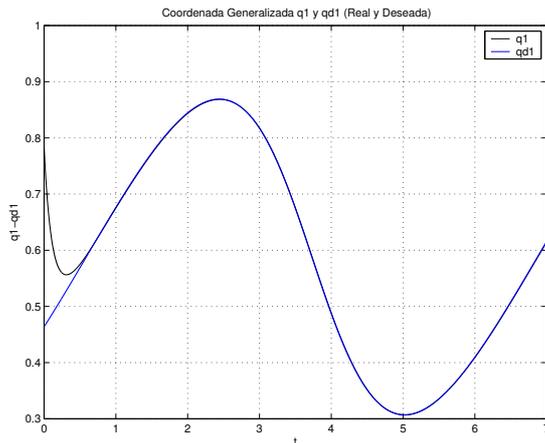


Figura 5.36: Control Adaptable, Coordenada Generalizada q1 y qd1 (Real y Deseada).

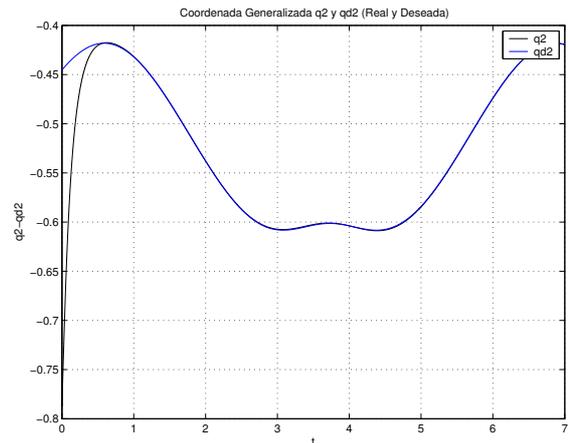


Figura 5.37: Control Adaptable, Coordenada Generalizada q2 y qd2 (Real y Deseada).

El comportamiento de la tercera coordenada generalizada q3, se puede observa en la Figura 5.38, es muy similar a las anteriores coordenadas, en la forma en que los valores del robot virtual alcanzan a los valores deseados, provocando que el error sea nulo en la mayor parte de la simulación.

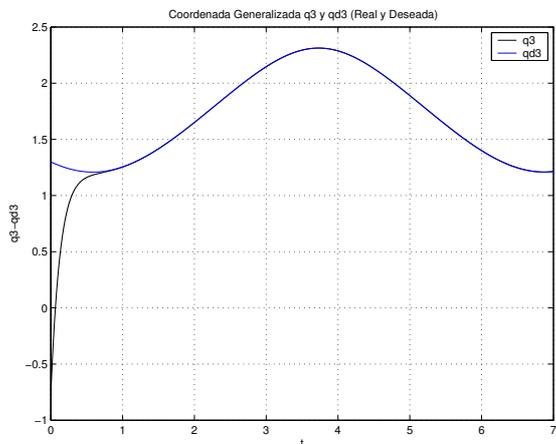


Figura 5.38: Control Adaptable, Coordenada Generalizada q_3 y q_{d3} (Real y Deseada).

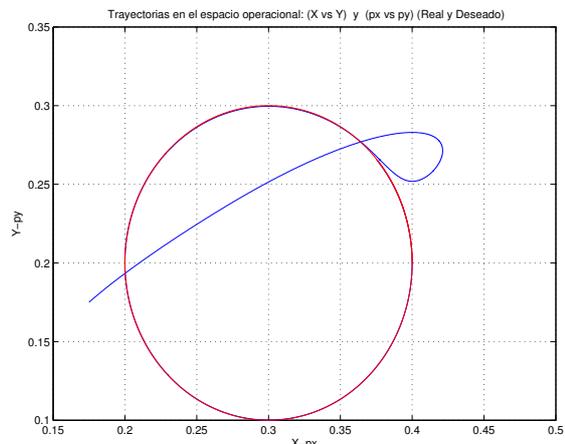


Figura 5.39: Control Adaptable, Trayectoria en el espacio operacional: (x,y) y (p_x,p_y) , (Real y Deseada).

La trayectoria realizada por el robot virtual y la trayectoria deseada se muestran en la Figura 5.39, se puede apreciar que ambas trayectorias son iguales en la mayor parte de la simulación, claro no al iniciar debido a las condiciones iniciales del robot virtual, este logra cumplir la trayectoria (circunferencia).

En este punto se puede apreciar que el control adaptable elimina el error extendido al seguir la trayectoria deseada como se puede observar en la Figura 5.40.

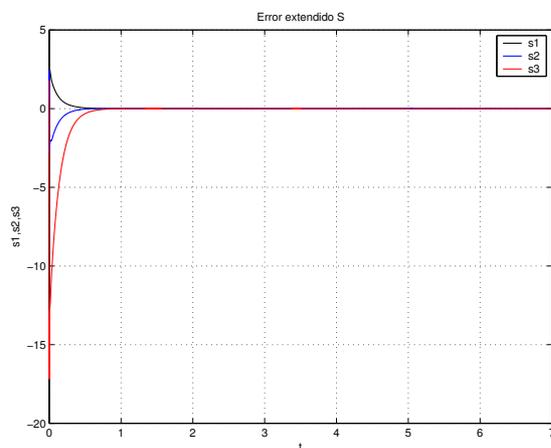


Figura 5.40: Control Adaptable, Error extendido S.

5.5. Seguimiento exacto: Convergencia en Tiempo Finito.

Un problema típico en el control de movimiento restringido de robots manipuladores, extensivo al guiado háptico, es la conmutación de movimiento libre a movimiento restringido. Para ello es necesario detectar el momento en el que ocurre la colisión o contacto entre el efector final, y el objeto que define la superficie geométrica de

restricción (superficie virtual de contacto para el caso de estudio). Para ellos es necesario el conocimiento exacto de los estados en algún instante de tiempo, por lo tanto la estabilización asintótica complica considerablemente la transición por lo complicado que es definir con precisión el instante en el que ocurre la colisión. En estas condiciones la convergencia en tiempo finito, permite el conocimiento exacto de los estados en algún instante de tiempo. Para lograr la convergencia en tiempo finito, se requiere de un generador tiempo base TBG en el contexto de movimiento libre.

Función implementada en TBG

$$f(t) = 1/2 - 1/2 \cos(wt) \quad (5.218)$$

donde:

$$w = 2\pi$$

$$f = 2\pi/t$$

$$t = \text{periodo} = 2tb$$

$$w = 2\pi/2tb$$

$$w = \pi/tb \text{ rad/seg}$$

$$f(t) = 1/2 - 1/2 \cos(\pi/tbt)$$

$$f'(t) = \pi/2tb \text{sen}(\pi/tbt)$$

$$f''(t = 1/2tb) = 0$$

$$f''(t) = \pi^2/2tb^2 \cos(\pi/tbt)$$

$$f''(1/2tb) = \pi^2/2tb^2 \cos(\pi/2) = 0$$

5.5.1. PIDNL con TBG

Para la implementación del control PINL con generador de tiempo bas la referencia nominal q_r en lazo cerrado, para que el robot tenga un comportamiento estable y convergencia en tiempo finito, se propone la siguiente ecuación

$$\dot{q}_r = \dot{q}_d - \alpha(t) \Delta q + S_d - K_i \sigma \quad (5.219)$$

$$\dot{\sigma} = \text{sgn}(S_q) \quad (5.220)$$

donde,

$$S_q = S - S_d \quad (5.221)$$

$$S_d = S(t_0) e^{-kt} \quad (5.222)$$

donde $S = \dot{\Delta}q + \alpha(t) \Delta q$, $K_i = K_i^T \in \mathfrak{R}^{n \times n}$ es una matriz diagonal positiva definida, $k > 0$, $\text{sgn}(\ast)$ denota la función signo de (\ast) Notese que $\ddot{q}_r = \ddot{q}_d - \alpha(t) \dot{\Delta}q - \dot{\alpha}(t) \Delta q + \dot{S}_d - K_i \text{sgn}(S_q)$ es discontinua, pero es utilizada solo para la prueba de estabilidad, y que $S_q(t_o) = 0$ para cualquier condición inicial. La coordenada del error extendido S_r ,

$$S_r = S_q + K_i \sigma \quad (5.223)$$

El control PID no lineal propuesto es el siguiente,

$$\tau = -K_d S_r \quad (5.224)$$

en términos de las anteriores definiciones,

$$\tau = -K_p \Delta q - K_v \dot{\Delta}q + K_d S_d - K_I I \quad (5.225)$$

$$I = \sigma = \int_{t_0}^t \text{sgn}(S_q(\varsigma)) d\varsigma \quad (5.226)$$

donde $K_d = K_d^T \in \mathfrak{R}^{n \times n}$ es una matriz diagonal positiva definida, $K_p(t) = K_d \alpha(t)$, $K_v = K_d$ y $K_I = K_d K_i$ son matrices diagonales que representan las ganancias de control, y $\alpha(t)$ es definida como $\sigma(t)$ en la ecuación (??) y con un generador de tiempo base $\xi(t)$ tal que se obtengan las trayectorias presentadas en la Figura ??.

La simulación digital en MatLab, consiste en desarrollar el seguimiento de una trayectoria en el espacio operacional del robot virtual con convergencia en tiempo finito, la cinemática y dinámica esta expuesta en el capítulo 5 sección 2 y 3 respectivamente, las trayectorias deseadas de posición son 5.151 y 5.152:

$$x(t) = h + r \cos(wt)$$

$$y(t) = k + r \text{sen}(wt)$$

Donde:

$$w = 0,4\pi$$

$$h = 0,3$$

$$k = 0,2$$

$$r = 0,1$$

Para dicha simulación se asume que el dispositivo (robot virtual) parte de un estado de reposo con una condición inicial de posición, que esta definida en:

$$q_1 = 45^\circ$$

$$q_2 = -45^\circ$$

$$q_3 = -45^\circ$$

las ganancias del control son:

$$k_p = [0,001, 0, 0; 0, 0,001, 0; 0, 0, 0,001]$$

$$k_d = [24, 0, 0; 0, 122, 0; 0, 0, 4, 5]$$

$$k_i = k_p \times k_d / 10$$

el tiempo de convergencia es:

$$t_b = 0,4 \text{ seg}$$

con:

$$\alpha_0 = 1,001$$

$$\delta = 0,001$$

En la Figura 5.41 se puede apreciar que con la implementación de TBG en un control PIDNL la coordenada generalizada q_1 se comporta de manera diferente al control PINDL (figura 1), el tiempo de convergencia empleado en la simulación fue de 0.4 seg. Tiempo en el cual q_1 alcanza los valores deseados provocando un mejor desempeño del robot ya que el error se reduce de manera significativa.

En la Figura 5.42 se puede apreciar que con la implementación de TBG en un control PIDNL la coordenada generalizada q_2 se comporta de manera diferente al control PINDL, el tiempo de convergencia empleado en la simulación fue de 0.4 seg. Tiempo en el cual q_2 alcanza los valores deseados provocando un mejor desempeño del robot ya que el error se reduce de manera significativa, además de eliminar las oscilaciones que presenta con el control PIDNL.

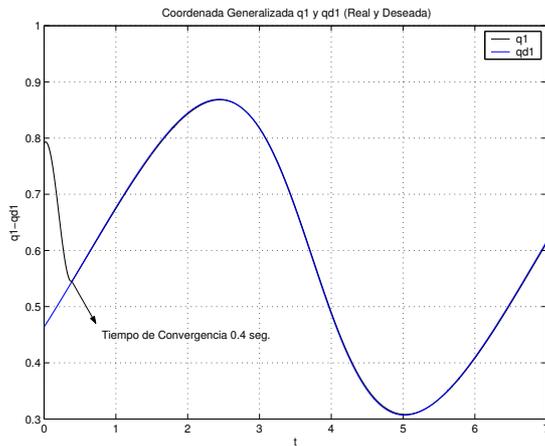


Figura 5.41: Control PIDNL con TBG, Coordenada Generalizada q_1 y q_{d1} (Real y Deseada).

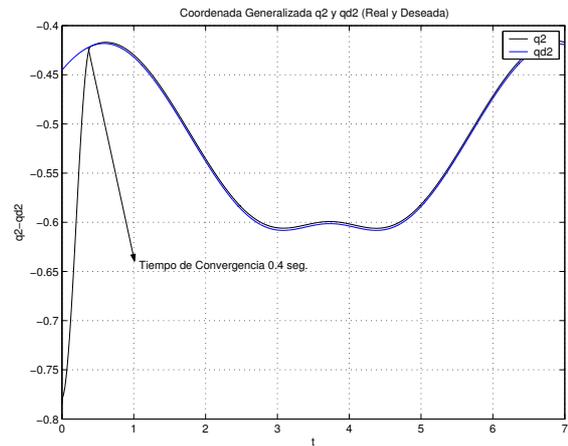


Figura 5.42: Control PIDNL con TBG, Coordenada Generalizada q_2 y q_{d2} (Real y Deseada).

En la Figura 5.43 se puede apreciar que con la implementación de TBG en un control PIDNL la coordenada generalizada q_3 se comporta de manera diferente al control PINDL, el tiempo de convergencia empleado en la simulación fue de 0.4 seg. Tiempo en el cual q_3 alcanza los valores deseados provocando un mejor desempeño del robot ya que el error se reduce de manera significativa.

La trayectoria realizada por el robot virtual y la trayectoria deseada, además de apreciar que ambas trayectorias son iguales en la mayor parte de la simulación, claro no al iniciar debido a las condiciones iniciales del robot virtual, este logra cumplir la trayectoria (circunferencia), se puede notar que la trayectoria del robot al tratar de alcanzar la trayectoria deseada es diferente a la Figura 5.44.

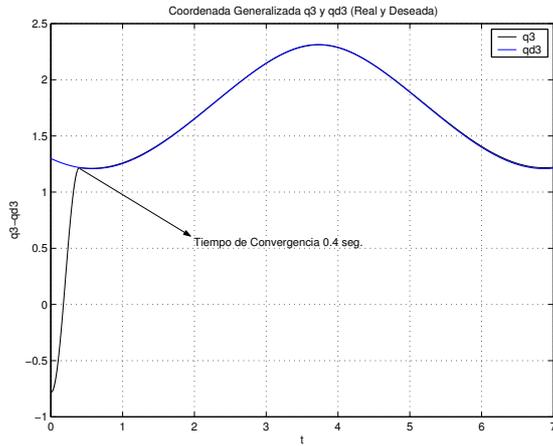


Figura 5.43: Control PIDNL con TBG, Coordenada Generalizada q_3 y q_{d3} (Real y Deseada).

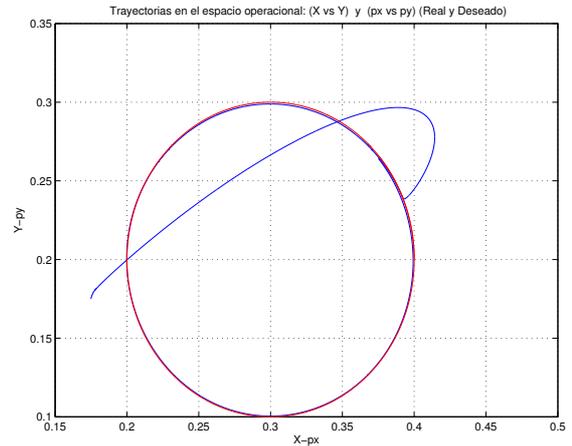


Figura 5.44: Control PIDNL con TBG, Trayectorias en el espacio operacional: (x,y) y (p_x,p_y) , (Real y Deseada).

Se puede afirmar que con la implementación del TBG en el control PIDNL mejora el desempeño del robot virtual, ya que el error extendido al seguir la trayectoria deseada tiende a cero o es cero en la mayor parte de la simulación, como se puede observar en la Figura 5.45.

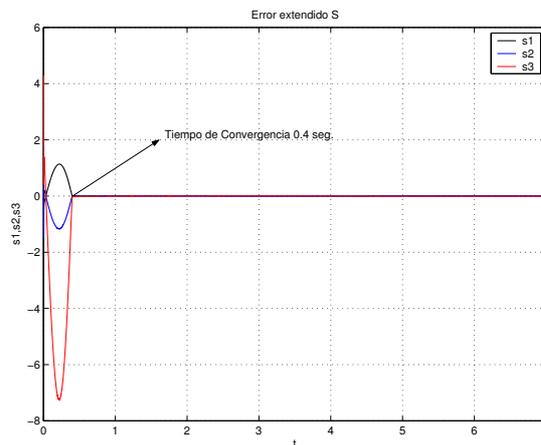


Figura 5.45: Control PIDNL con TBG, Error extendido S .

5.5.2. Control adaptable pasivo robusto con TBG

La simulación digital en MatLab, consiste en desarrollar el seguimiento de una trayectoria en el espacio operacional del robot virtual con convergencia en tiempo finito, la cinemática y dinámica esta expuesta en el capítulo

5 sección 2 y 3 respectivamente, las trayectorias deseadas de posición son 5.151 y 5.152:

$$x(t) = h + r \cos(wt)$$

$$y(t) = k + r \operatorname{sen}(wt)$$

Donde:

$$w = 0,4\pi$$

$$h = 0,3$$

$$k = 0,2$$

$$r = 0,1$$

Para dicha simulación se asume que el dispositivo (robot virtual) parte de un estado de reposo con una condición inicial de posición, que esta definida en:

$$q_1 = 45^\circ$$

$$q_2 = -45^\circ$$

$$q_3 = -45^\circ$$

las ganancias del control son:

$$\text{alfa} = [8, 0, 0; 0, 0, 8, 8, 0; 0, 0, 7, 5]$$

$$\text{gamma} = 1,5$$

$$\text{kd} = 20$$

el tiempo de convergencia es:

$$t_b = 0,4 \text{seg}$$

con:

$$\alpha_0 = 1,001$$

$$\delta = 0,001$$

En la Figura 5.46 se pueden ver los valores que se obtuvieron durante la simulación, de la coordenada generalizada q_1 , al mismo tiempo se muestra una comparación grafica con respecto a los valores que se desea obtener, además de la implementación del TBG al control Adaptable, a diferencia del control adaptable sin TBG el único cambio es la convergencia en un x tiempo, el cual en la simulación fue de 0.4 seg. , el error como puede apreciarse es cero.

En la Figura 5.47 se puede observar la representación gráfica de los valores de la coordenada generalizada q_2 y de los valores deseados qd_2 , así como la implementación del TBG al control Adaptable, a diferencia del control adaptable sin TBG cambio es la convergencia en un x tiempo, el cual en la simulación fue de 0.4 seg., se puede apreciar también que en la mayor parte del tiempo de simulación el error es cero.

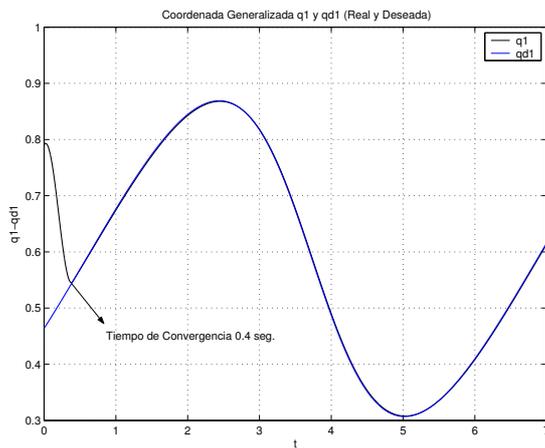


Figura 5.46: Control adaptable con TBG, Coordenada Generalizada q_1 y qd_1 (Real y Deseada).

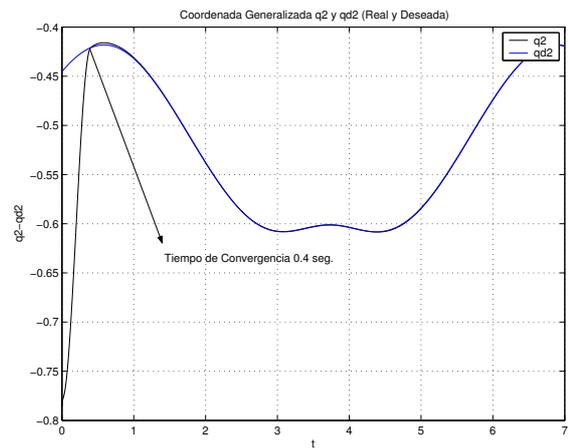


Figura 5.47: Control adaptable con TBG, Coordenada Generalizada q_2 y qd_2 (Real y Deseada).

La implementación del TBG al control Adaptable y comportamiento de la tercera coordenada generalizada q_3 , se puede observa en la Figura 5.48, es muy similar a las anteriores coordenadas, en la forma en que los valores del robot virtual alcanzan a los valores deseados, el cambio observado es la convergencia en un tiempo x , el cual en la simulación fue de 0.4 seg., ya que el error es nulo en la mayor parte de la simulación.

La trayectoria realizada por el robot virtual y la trayectoria deseada se muestran en la Figura 5.49, se puede apreciar que ambas trayectorias son iguales en la mayor parte de la simulación, si se observa la Figura 5.50 se puede notar que la trayectoria del robot al tratar de alcanzar la trayectoria deseada es diferente a la 5.49.

En la Figura 5.50 se puede apreciar que el control adaptable con TBG elimina el error extendido al seguir la trayectoria deseada, además de lograrlo en un tiempo x , el cual en la simulación fue de 0.4 seg.

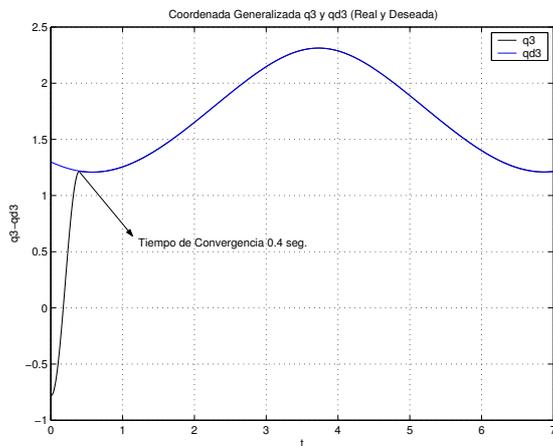


Figura 5.48: Control adaptable con TBG, Coordenada Generalizada q_3 y q_{d3} (Real y Deseada).

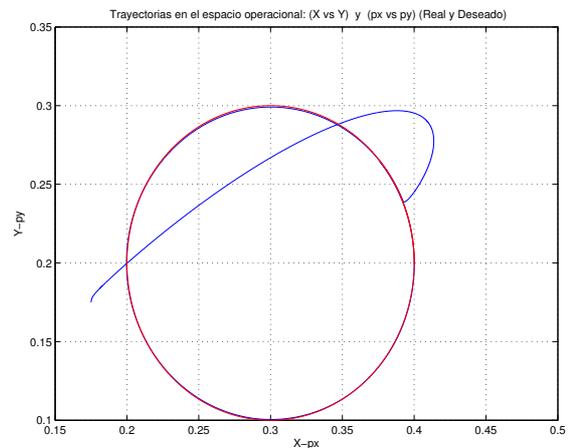


Figura 5.49: Control adaptable con TBG, Trayectorias en el espacio operacional: (x,y) y (p_x,p_y) , (Real y Deseada).

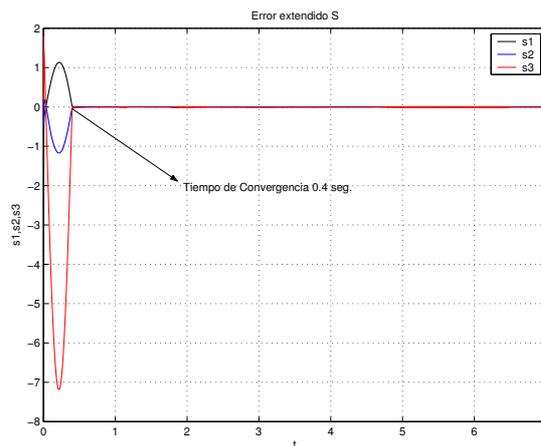


Figura 5.50: Control adaptable con TBG, Error extendido S .

5.6. Conclusiones

En este capítulo se puede observar la obtención matemática de cada uno de los modelos del robot, se presenta todo el desarrollo matemático de estas ecuación a ecuación, esto ayudó a la comprensión de estos modelos y para su codificación y adaptación a la interfaz de usuario propuesta.

Así mismo se presentan las propiedades del modelo dinámico del robot, en el apéndice C se presentan las gráficas de dichas propiedades al realizar la tarea de simulación en donde se comprueba la validez del modelo obtenido.

Se presentan también los controles PD, PID, PIDNL y el control Adaptable pasivo Robusto en éste ultimo se presenta además la parametrizaciones lineales del modelo dinámico despreciando la fricción existente en el sistema y tomando en cuenta la fricción del sistema, para las simulaciones se empleo la última opción; además de la integración de un generador de tiempo base que nos ayuda a que el robot converja en la posición deseada en un determinado tiempo denominado tiempo de convergencia, se a los controles PIDNL y Adaptable.

Para la determinación de qué control es mejor con respecto a los mencionados, se realizó una comparación entre los controles: PD y PID, PIDNL y Adaptable; en donde se puede decidir que el mejor es el control Adaptable, PIDNL, PUID y PD en ese orden, esta conclusión fue en relación al error obtenido en las simulaciones que se hicieron en MatLab, cabe hacer mención que los cuatro controles terminaron la trayectoria. La implementación del generador de tiempo base mejoró el comportamiento del robot en ambos controles.

Capítulo 6

Conclusiones y perspectivas.

6.1. Conclusiones generales

Este trabajo profesional resolvió el problema de visualización virtual y control de un robot manipulador del tipo antropomórfico, aplicado al seguimiento perfecto de trayectoria, sin embargo ésta metodología puede ser extendida a diferentes configuraciones de robot, así como diferentes aplicaciones mecatrónicas con partes móviles. Para lograr estos resultados, fué necesario hacer una revisión de los modelos matemáticos de robots manipuladores, las estrategias de control de movimiento clásicas y modernas, las técnicas de planificación de trayectorias y las herramientas para virtualización con comportamiento complejo, sin embargo se realizó una revisión de literatura para establecer dentro de la metodología propuesta y describir profesionalmente los resultados, situación que permitió el análisis y diseño del sistema.

Además de realizar una modelación analítica de la geometría de un robot, la descripción del desplazamiento espacial del mismo, es decir la cinemática del movimiento, de igual forma se abordó las formulaciones de movimiento del brazo indicando las leyes físicas que intervienen en el proceso denominando a esto como dinámica.

En este documento se presentó los modelos cinemáticos y dinámicos del robot definido como caso de estudio. La cinemática del robot fué descrita con la metodología Denavit - Hartenberg, y la dinámica fué sintetizada con la formulación Euler - Lagrange, para verificar que estos modelos sean correctos se analizaron sus correspondientes propiedades.

Se puede definir el control como “un algoritmo que permite la regulación de la energía aplicada a los motores que generan el movimiento del robot, de manera que alcancen los parámetros deseados con el fin de disminuir el error”.

Estos algoritmos pueden controlar la posición, posición y velocidad, también existen aquellos que únicamente toman en cuenta la cinemática, en el presente trabajo los controles toman en cuenta la cinemática y dinámica del robot en los algoritmos propuestos.

Para las estrategias de control de regulación y seguimiento de trayectorias se estudiaron, a partir de la simulación digital en Matlab, los controladores PD, PID y un control por modos deslizantes de segundo orden o PID no lineal. Es importante mencionar que para la solución numérica de la acción integral se implementó el método de Runge - Kutta de orden superior. Las trayectorias fueron descritas por las ecuaciones paramétricas de una circunferencia y para lograr la convergencia en tiempo finito es empleado un generador de tiempo base.

Un control adicional implementado fué el control adaptable pasivo robusto diseñado a partir de la parametrización lineal del robot, y partiendo del control adaptable tipo Slotine - Li. El ambiente virtual fué desarrollado en Visual C++ con OpenGL y una interfaz de usuario con Delphi 7.0 logrando resultados significativos.

¿Qué control es mejor? con respecto a los mencionados, se realizó una comparación entre los controles: PD y PID, PIDNL y Adaptable; en donde se puede decidir que el mejor es el control Adaptable, PIDNL, PUID y

PD en ese orden, esta conclusión fue en relación al error obtenido en las simulaciones realizadas en MatLab, cabe hacer mención que los cuatro controles terminaron la trayectoria. La implementación del generador de tiempo base mejoró el comportamiento del robot en ambos controles (PIDNL y Adaptable).

Se consideran 11 apéndices de códigos, propiedades dinámicas y manual de usuario que permiten una mejor interpretación de la tesis.

6.2. Perspectivas y trabajo futuro

Este trabajo deja bases técnicas y tecnológicas para lograr el siguiente trabajo futuro:

1. Estudios de control moderno e inteligentes de robots en ésta plataforma.
2. Establecer tareas de posicionamiento en el espacio.
3. Establecer otros tipos de tareas industriales en el visualizador virtual.
4. Aplicar la metodología para robots cooperativos.

Apéndice A

Abreviaturas

AFNOR Asociación Francesa de Normalización.

AFRI Asociación Francesa de Robótica Industrial.

AGV Vehículos de Guiado Automático.

AMF American Machine y Foundry Company.

API Application Programming Interface.

AV Vehículos Autónomos.

CA Corriente Alterna.

CD Corriente Directa.

$c_i \cos(i)$

$c_{ij} \cos(i + j)$

CMOS Semiconductor Complementario de Óxido de Metal.

DEA Digital Electric Automation.

DOF Degrees Of Freedom. Grados de Libertad.

E-L Euler - Lagrange.

EID Entorno Integrado de desarrollo.

F Fuerza.

HW Hardware.

IBM International Business Machines.

IEP Interfase Electrónica de Potencia.

J Matriz Jacobiana.

JIRA Asociación Japonesa de Robots Industriales.

IEP Interfaz Electrónica de Potencia.

IFR Federación Internacional de Robótica.

ISO Organización Internacional de Estándares.

K Energía cinética.

kg Kilogramo.

LCG Longitud de Centro de Gravedad.

MCDP Modelo cinemático directo de posición.

MCIP Modelo cinemático inverso de posición.

MCDV Modelo cinemático directo de velocidad.

MCIV Modelo cinemático inverso de velocidad.

MCDA Modelo cinemático directo de aceleración.

MCIA Modelo cinemático inverso de aceleración.

MIT Instituto Tecnológico de Massachussets.

MTH Matriz de transformación homogénea.

ME Matriz elemental.

N Newton.

N-E Newton - Euler.

NASA National Aeronautics and Space Administration.

P Energía potencial.

PC Computadora Personal.

PD Proporcional más Derivativo.

PDH Parámetros Denavit-Hartenberg.

PI Proporcional Integral.

PID Proporcional Integral Derivativo.

PUMA Manipulador Universal Programable para Tareas de Ensamblaje.

P-R Pasivo Robusto.

R Robot.

RAD Rapid Application Development.

RIA Asociación de Robots Industriales.

RV Realidad virtual.

rad radian.

SCARA Selective Compliance Assembly Robot Arm.

s_i $sen(i)$

s_{ij} $sen(i + j)$

seg Segundos.

SW Software.

T Periodo.

T3 The Tomorrow Tol.

V Velocidad de desplazamiento.

VCL Visual Components Library.

w Velocidad angular.

Apéndice B

Glosario

Acción de control

Ver Variable de control[41].

Adaptación

Tipo evolutivo de cambio (usualmente involucra una modificación progresiva de cierta estructura o estructuras) que un sistema realiza para hacer frente con los cambios del entorno, mientras mantiene los atributos esenciales de la estructura y constantes del sistema[41].

Aleatorio

Algo que carece de todo patrón regular debido a su inmunidad a cualquier ley que la forzaría a una secuencia aunque sea levemente repetitiva que podría ser considerada como patrón. La existencia de aleatoriedad es solamente teórica, dado que no hay forma de probar que algo sea infinitamente irregular e impredecible. Las secuencias de números aleatorios (“números locos”) generadas computacionalmente se consideran, así, pseudo-aleatorias, ya que algún programa, aunque sea muy intrincado, es la causa de cada elemento de ellas. La aleatoriedad en pequeña escala, considerada como pseudoaleatoriedad, sirve como herramienta útil en simulaciones de Monte Carlo y en teoría de los juegos. Un proceso aleatorio puro es imposible de ser controlado con instrumento alguno. Difiere de un proceso caótico, que es controlable[41].

Articulación prismática (Lineal)

Las articulaciones prismáticas implicar un movimiento deslizante o de traslación de las uniones de conexión[5].

Articulación de revolución

Tipo de articulación giratoria en la que la unión de entrada es paralela al eje de rotación y la de salida es perpendicular a dicho eje. Esencialmente, la unión de salida gira alrededor de la de entrada como si estuviera en órbita[5].

Articulación de rotación

Suministra un grado de libertad consistente en una rotación alrededor del eje de la articulación. Está articulación es, con diferencia, la más empleada.

Autómata

Palabra que procede del griego (automatos = actuar por sí mismo). Es un mecanismo artificial que imita com-

portamientos de la vida real[41].

Brazo(Robot)

Una de las partes del manipulador. Soportado en la base de éste, sostiene y maneja la muñeca[41].

Cadena cinemática

Es un ensamble de eslabones y juntas interconectados de modo que proporcionen un movimiento de salida controlado en respuesta a un movimiento de entrada proporcionado.

Centro de gravedad

Punto imaginario en el cual puede considerarse que se concentra y actúa el peso total de un cuerpo. Punto en el que quedaría en equilibrio un cuerpo colgado de un cable[41].

Centro de masa

Es el punto donde puede considerarse que está concentrada toda la masa de un cuerpo para estudiar determinados aspectos de su movimiento. El centro de masas de una esfera de densidad uniforme está situado en el centro de la esfera. El centro de masas de una varilla cilíndrica de densidad uniforme está situado a la mitad de su eje. En algunos objetos, el centro de masas puede estar fuera del objeto.

Cinemática del brazo del robot

Trata con el estudio analítico de la geometría del movimiento de un brazo de robot con respecto a un sistema de coordenadas de referencia fijo sin considerar las fuerzas o movimientos que originan el movimiento[5].

Codificador óptico

En este se tiene una barra o disco translúcido con incrustaciones de material opaco o viceversa. De esta forma y usando un emisor y un detector de luz infrarroja, se generan pulsos a la salida de un fototransistor que ha sentido el movimiento de la barra o disco. Cabe mencionar que muchas veces el material del disco es opaco, y que simplemente se le maquinan hoyos simétricamente espaciados para lograr el mismo efecto.

Existen 2 tipos de codificadores ópticos: los incrementales y los absolutos.

Los codificadores ópticos incrementales son aquellos que producen pulsos igualmente espaciados y que pueden ser acumulados por un contador (para incrementar o decrementar). El conteo es la indicación del desplazamiento. El punto de origen puede fijarse o programarse en el dispositivo de lectura.

Los codificadores absolutos son aquellos que generan un número codificado digitalmente, y es indicativo de la posición. Esto se hace mediante un conjunto de cabezas lectoras y una disposición de segmentos en multicanal. Los códigos digitales típicos son: binario natural, gray y BCD (decimal codificado en binario).

Control

Selección de las entradas de un sistema de manera que los estados o salidas cambien de acuerdo a una manera deseada. Ver SISTEMA DE CONTROL. Ver PRINCIPIO DE VARIEDAD NECESARIA[41].

Dos tipos de control pueden ser usados en diferentes partes del robot:

Lazo cerrado: se monitorea continuamente la posición del brazo del robot mediante un sensor de posición, y se modifica la energía que se manda al actuador de tal forma que el movimiento del brazo se obedece al camino deseado, tanto en dirección como en velocidad. Éste control se puede usar cuando la tarea que se ha de llevar a cabo está dirigida mediante un camino definido por la misma pieza, tal como sería soldar, revestir y ensamblar.

En un sistema de lazo abierto: el controlador no conoce la posición de la herramienta mientras el brazo se mueve de un punto a otro. Éste tipo de control es muy usado cuando el movimiento que debe seguir el brazo se encuentra determinado previamente, al ser grabado con anterioridad y reproducido sin cambio alguno, lo cual es útil cuando todas las piezas a ser tratadas son exactamente iguales.

Control en lazo abierto

Control en el cual la salida no afecta la acción de control. Es decir, no se mide ni se realimenta la salida para compararla con la entrada. Ejemplo: control de tránsito, lavadora (el remojo, lavado y enjuague operan con base en el tiempo). La precisión del control depende de la calibración y no funciona correctamente en presencia de perturbaciones[41].

Control en lazo cerrado

Ver CONTROL REALIMENTADO[41].

Control realimentado

Operación que, en presencia de perturbaciones (impredecibles), tiende a reducir la diferencia entre la salida de un sistema y alguna entrada de referencia y lo continúa haciendo con base en esta diferencia. Ejemplos: control de velocidad, control de un robot, control de temperatura, sistema empresarial[41].

Coordenada operacional

Posición en el espacio definido por los valores x, y, z referidos al origen.

Dinámica del brazo del robot

Trata con la formulación matemática de las ecuaciones del movimiento del brazo. Las ecuaciones dinámicas de movimiento de un manipulador con un conjunto de ecuaciones matemáticas que describen la conducta dinámica del manipulador[5].

Dinámico

Atributo que acentúa el movimiento, el cambio y el proceso, en oposición a “estático”[41].

Efecto de Coriolis

Tendencia de una masa a aumentar o disminuir su velocidad angular cuando su radio de rotación decrece o crece, respectivamente[41].

Eje (Robot)

Cada una de las líneas según las cuales se puede mover el robot o una parte de él (algún elemento de su estructura). Pueden ser ejes o líneas de desplazamiento longitudinal sobre sí mismo (articulación prismática) o ejes de giro (rotación). Cada eje define un “grado de libertad” del robot[41].

Energía cinética

Es la capacidad de un cuerpo para efectuar un trabajo en virtud de su movimiento[10].

Energía potencial

Energía almacenada que posee un sistema como resultado de las posiciones relativas de sus componentes. Por ejemplo, si se mantiene una pelota a una cierta distancia del suelo, el sistema formado por la pelota y la Tierra tiene una determinada energía potencial; si se eleva más la pelota, la energía potencial del sistema aumenta [10].

Entorno

Estado de la naturaleza o estado del mundo, concepto útil en modelado. Conjunto de todos los factores externos o fuerzas que están más allá de la influencia de la persona que toma una decisión, pero que no obstante afectan las consecuencias de sus acciones[41].

Equilibrio

Estado de un sistema cuya configuración o propiedades macroscópicas no cambian a lo largo del tiempo. Por ejemplo, si se introduce una moneda caliente en un vaso de agua fría, el sistema formado por el agua y la moneda alcanzará el equilibrio térmico cuando ambos estén a la misma temperatura.

Eslabón

Cuerpo rígido que posee al menos dos nodos, que son los puntos de unión con otros eslabones. El número de nodos le da su nombre al eslabón : Binario = dos nodos, Terciario = tres nodos, etc..

Espacio de estado

Espacio n-dimensional cuyos ejes de coordenadas son las VARIABLES DE ESTADO[41].

Espacio singular

Se denominan configuraciones singulares de un robot a aquellas en el que el determinante de su matriz Jacobiana (Jacobiano) se anula. Por esta circunstancia, en las configuraciones singulares no existe jacobiana inversa. Al anularse el Jacobiano, un incremento infinitesimal de las coordenadas cartesianas supondría un incremento infinito de las coordenadas articulares, lo que en la práctica se traduce en que las inmediateces de las configuraciones singulares, el pretender que el extremo del robot se mueva a velocidad constante, obligaría a movimientos de las articulaciones a velocidades inabordables por sus actuadores.

Espacio de trabajo

Las dimensiones de los elementos del manipulador, junto a los grados de libertad, definen el espacio de trabajo del robot. Se subdivide en áreas diferenciadas entre sí, por la accesibilidad específica del elemento terminal (aprehensor o herramienta), es diferente a la que permite orientarlo verticalmente o con el determinado ángulo de inclinación. También queda restringido el espacio de trabajo por los límites de giro y desplazamiento que existen en las articulaciones[6]. El espacio de trabajo puede ser explicado, como un espacio de configuraciones y consiste en el conjunto de todas las posiciones y orientaciones que el robot puede tomar. Obtener el espacio de configuraciones es equivalente a que el robot recorra y se posicione en todos los puntos posibles del espacio de trabajo.

Estabilidad

Condición en la cual las variables críticas de un sistema dinámico se mantienen invariables o permanecen dentro de unos límites determinados[41].

Estado

Valor numérico de un conjunto de propiedades relevantes de un sistema.

Frecuencia

Número de veces que se repite un proceso periódico por unidad de tiempo[41].

Fricción

Se define a la fricción como una fuerza resistente que actúa sobre un cuerpo, que impide o retarda el deslizamiento de este respecto a otro o en la superficie que este en contacto. Esta fuerza es siempre tangencial a la superficie en los puntos de contacto con el cuerpo, y tiene un sentido tal que se opone al movimiento posible o existente del cuerpo respecto a esos puntos. Por otra parte estas fuerzas de fricción están limitadas en magnitud y no impedirán el movimiento si se aplican fuerzas lo suficientemente grandes.

Fricción estática

La fricción estática representa una fuerza que tiende a prevenir el movimiento desde el comienzo.

Fricción de Coulomb

Es una fuerza que tiene una amplitud constante con respecto al cambio de velocidad, pero el signo de la fuerza de fricción cambia al invertir la dirección de la velocidad[4].

Fricción viscosa

La fricción viscosa representa una fuerza que es una relación lineal entre la fuerza aplicada y la velocidad.

Fuerza centrífuga

Fuerza que tiende a alejar de un centro.

Fuerza centrípeta

Fuerza dirigida hacia un centro, que hace que un objeto se desplace en una trayectoria circular. Por ejemplo, supongamos que atamos una pelota a una cuerda y la hacemos girar en círculo a velocidad constante. La pelota se mueve en una trayectoria circular porque la cuerda ejerce sobre ella una fuerza centrípeta. Según la primera ley del movimiento de Newton, un objeto en movimiento se desplazará en línea recta si no está sometido a una fuerza. Si se cortara la cuerda de repente, la pelota dejaría de estar sometida a la fuerza centrípeta y seguiría avanzando en línea recta en dirección tangente a la trayectoria circular (si no tenemos en cuenta la fuerza de la gravedad). En general, la fuerza centrípeta que debe aplicarse a un objeto de masa m para que se mueva en una trayectoria circular de radio r con una velocidad constante v es

Cuando se aplica una fuerza centrípeta, la tercera ley de Newton implica que en algún lugar debe actuar una fuerza de reacción de igual magnitud y sentido opuesto. En el caso de la pelota que gira con una cuerda, la reacción es una fuerza dirigida hacia el exterior, o centrífuga, experimentada por la mano que sujeta la cuerda. En el caso del carrusel, el cuerpo de la persona presiona hacia fuera contra el asiento como reacción a la fuerza

centrípeta ejercida por el asiento.

Fuerza de Coriolis

Fuerza resultante de la rotación de la Tierra. En el hemisferio norte la fuerza de Coriolis desvía el movimiento del aire hacia la derecha. En el hemisferio sur la fuerza de Coriolis desvía el movimiento del aire hacia la izquierda. La fuerza actúa en un ángulo recto con respecto a la dirección del viento y es directamente proporcional a la velocidad de éste. Es decir, al aumentar la velocidad del viento, la fuerza de Coriolis se incrementa. En determinada latitud, al duplicarse la velocidad del viento se duplicará la fuerza de Coriolis. Esta puede fluctuar desde cero en el ecuador hasta un máximo en los polos. Influye en la dirección del viento en todo lugar, salvo en puntos situados sobre el Ecuador, aunque sus efectos son más pronunciados en latitudes medias y altas[41].

Grados de libertad

Para un robot de coordenadas cartesianas los tres grados de libertad son: movimiento vertical (movimiento eje z), movimiento hacia adentro y afuera (movimiento eje y) y movimiento derecha o izquierda (movimiento eje x). Estos se consiguen por los movimientos correspondientes de los 3 dispositivos de deslizamiento ortogonales del brazo del robot[5].

Gravedad

Propiedad de atracción mutua que poseen todos los objetos compuestos de materia.

Hamiltoniano

Es una reformulación de la mecánica clásica. La mecánica hamiltoniana puede ser formulada por sí misma, usando los espacios simplécticos, sin referir a cualesquiera conceptos anteriores de fuerza o de la mecánica lagrangiana.

Inercia

Propiedad de la materia que hace que ésta se resista a cualquier cambio en su movimiento, ya sea de dirección o de velocidad. Esta propiedad se describe con precisión en la primera ley del movimiento del científico británico Isaac Newton: un objeto en reposo tiende a permanecer en reposo, y un objeto en movimiento tiende a continuar moviéndose en línea recta, a no ser que actúe sobre ellos una fuerza externa.

Interfaz

Conexión mecánica o eléctrica que permite el intercambio de información entre dos dispositivos o sistemas. Habitualmente se refiere al “software” y “hardware” necesarios para unir dos elementos de proceso en un sistema o bien para describir los estándares recomendados para realizar dichas interconexiones. También se conoce por su denominación inglesa: “interface” [41].

Lagrangiano

Un lagrangiano es una función diseñada para resumir un sistema entero; el dominio apropiado del lagrangiano es un espacio de fase, y debe obedecer las ecuaciones de Euler-Lagrange. El concepto fue utilizado originalmente en una reformulación de la mecánica clásica conocida como la mecánica lagrangiana. En este contexto, el lagrangiano se toma comúnmente como la energía cinética de un sistema mecánico menos su energía potencial. El concepto también se ha probado útil en su extensión a la mecánica cuántica. El lagrangiano es también útil en que simplifica grandemente los cálculos con los sistemas dinámicos.

Lineal

Relación proporcional entre dos cantidades (como en $y = cx$). El término proviene de la línea recta con la cual tal relación puede ser visualizada. La noción de linealidad es generalizada para la relación entre más cantidades (expresada en forma de ecuaciones lineales) a un concepto de tiempo que evoluciona desde un pasado indefinido a un futuro indefinido sin caminos paralelos o lazos de realimentación[41].

Manipulador

En general, cualquier dispositivo mecánico capaz de reproducir los movimientos humanos para el manejo de objetos. En particular, suele referirse a los elementos mecánicos de un robot que producen su adecuado posicionamiento y operación[41].

Matriz de escalamiento

Es una matriz cuadrada que tiene todos sus elementos nulos excepto los de la diagonal principal que son iguales[43].

Matriz de rotación

Una matriz de rotación de 3×3 se puede definir como una matriz de transformación que opera sobre un vector de posición en un espacio euclídeo tridimensional y transforma sus coordenadas expresadas en un sistema de coordenadas rotado OUVW (sistema ligado al cuerpo) a un sistema de coordenadas de referencia OXYZ[9].

Matriz de transformación homogénea

Es una matriz 4×4 que transforma un vector de posición expresado en coordenadas homogéneas desde un sistema de coordenadas hasta otro sistema de coordenadas[9].

Mecanismo

Sistema de elementos dispuestos para transmitir movimiento en un modo predeterminado. Ejemplos: sacapuntas de manivela, obturador de cámara fotográfica, reloj analógico, silla plegadiza, lámpara ajustable de escritorio y sombrilla.

Modelo

Representación de una manera más sencilla de ciertos aspectos de un proceso o elemento, y el cual es utilizado con objetivos de análisis, control y predicción. Todo modelo se basa en una teoría, pero dicha teoría puede no estar indicada en una forma concisa. Tipos de modelos: a escala, descriptivos, gráficos, analógicos y matemáticos[41].

Modelo cinemático

conjunto de expresiones que determinan el movimiento sin considerar causas que lo producen.

Modelo dinámico

Conjunto de expresiones que determinan las causas del movimiento.

Momento de inercia

Resistencia que un cuerpo en rotación opone al cambio de su velocidad de giro. A veces se denomina inercia rotacional.

Órgano terminal

Este término se utiliza para describir la mano o herramienta que está unida a la muñeca. Representa el herramienta especial que permite al robot de uso general realizar una aplicación particular[5].

Perturbación

Variable no deseada aplicada a un SISTEMA y la cual tiende a afectar adversamente el valor de una variable controlada. Si la perturbación se genera dentro del sistema se denomina interna, en tanto que una perturbación externa se produce fuera del sistema y es una entrada[41].

Realidad virtual

Sistema que permite a uno o más usuarios ver, moverse y reaccionar en un mundo simulado por ordenador o computadora. Los distintos dispositivos de interfaz permiten al usuario ver, tocar y hasta manipular objetos virtuales. Los mundos virtuales y todo lo que contienen (incluyendo imágenes computerizadas de los participantes) se representan con modelos matemáticos y programas de computadora.

Robot

Es un sistema mecánico cuyo control debe estar encaminado a efectuar una tarea de utilidad[5].

Robot industrial

Es un manipulador multifuncional reprogramable diseñado para desplazar materiales, piezas, herramientas o dispositivos especiales mediante movimientos programados variables para la ejecución de una diversidad de tareas[5].

Robot manipulador

Los manipuladores son brazos articulados con un número de grados de libertad, que oscila entre dos y cinco, cuyos movimientos de tipo secuencial, se programan mecánicamente o a través de una memoria o de un controlador lógico programable. No permiten la combinación simultánea de movimientos ni el posicionamiento de su elemento terminal[9].

Robusto

En control, poco sensible a errores o incertidumbres de modelado. En la actividad computacional, tanto una máquina como un algoritmo se consideran robustos si son capaces de actuar y trascender a pesar de dificultades que accidentarían a máquinas o algoritmos más simples o menos flexibles. La tolerancia a las fallas o fallos se considera sinónimo de robustez[41].

Ruido

Entrada indeseada a un sistema cibernético. Típicamente es una PERTURBACIÓN aleatoria que no representa ninguna parte de un mensaje o señal deseada de una fuente específica[41].

Sensor

Es un transductor que se utiliza para medir una variable física de interés[5].

Sensores propioceptivos

Suministra información del estado interno del robot: posiciones, velocidades y aceleraciones[57].

Servomecanismo

Sistema automático de control de movimiento que usa la realimentación para controlar una salida deseada como posición, velocidad o aceleración[41].

Sistema de control

Sistema diseñado para lograr el comportamiento deseado de un sistema dinámico. Puede ser digital (microprocesado) o analógico (por ejemplo, circuitos eléctricos, sistemas hidráulicos, sistemas mecánicos, etc.), manual o automático[41].

Sistema Dinámico

Sistema con un número finito de grados de libertad y que puede ser modelado matemáticamente por medio de ecuaciones diferenciales que dependen del tiempo[41].

Sistema lineal

Un sistema es lineal si se cumple el principio de superposición. Es decir, la relación entre la entrada y salida del sistema es una línea recta[41].

Torque

Cuando se empuja un objeto, se está aplicando una fuerza. Cuando se gira algo, tal como una puerta, se está aplicando una fuerza rotacional. La fuerza rotacional se denomina torque[58].

Transmisión

Las transmisiones son los elementos encargados de transmitir el movimiento desde los actuadores hasta las articulaciones[6].

Transmisión mecánica

Incrementa la fuerza y el par de salida. Estas transmisiones pueden adoptar la forma de: engranes, tornillos sinfín, impulsores armónicos, bandas, cadenas, etc[44].

Transductor

Es un dispositivo que transforma un tipo de variable física[5].

Trayectoria de un robot

Para realizar un tarea determinada el robot debe moverse desde un punto inicial a la final. Este movimiento puede ser realizado según infinitas trayectorias espaciales[6].

Variable articular

Valor que puede adquirir el ángulo de la articulación.

Variable controlada

Variable dinámica que se regula. Esta variable es medida por los sensores[41].

Variable de control

Variable entregada por el controlador para ser aplica al actuador, después de ser transformada y amplificada. Esta variable depende de la variable error[41].

Variable de error

Desviación de la variable controlada del punto de referencia. El regulador digital responde en dependencia de esta variable[41].

Variable dinámica

Parámetro físico que cambia espontáneamente o por influencia externa. Algunas variables dinámicas pueden requerir regulación. Ejemplos: temperatura, presión, voltaje, etc.[41].

Variable manipulada

Variable dinámica que cambia como función de la variable de control y que modifica directamente la variable controlada. Por ejemplo, en el proceso de calentamiento de un líquido la variable de control es el voltaje, el cual se aplica a una resistencia, y la variable manipulada es el calor[41].

Variable medida

Variable dinámica dada por los sensores, proporcional a la variable controlada. Normalmente es un nivel de voltaje o corriente[41].

Apéndice C

Apéndice: Propiedades dinámicas

La dinámica de un robot se encarga de la relación entre las fuerzas que actúan sobre un cuerpo y el movimiento que en él se origina. Esta relación se obtiene mediante el modelo dinámico, que relaciona matemáticamente: la localización del robot definida por sus variables articulares o por las coordenadas de localización de su extremo, y sus derivadas: velocidad y aceleración., las fuerzas y pares aplicados en las articulaciones y los parámetros dimensionales del robot, como longitud, masas e inercias de sus elementos.

MatLab es un lenguaje de programación interpretado en el que las variables son matrices y, por tanto, las operaciones básicas aritméticas y lógicas son operaciones matriciales. Esto hace que MatLab sea una herramienta muy adecuada para cálculo matricial y, en concreto, para simulación de robots.

En este apéndice se presenta el código Matlab y las gráficas que representan las simulaciones digitales que comprueban las ecuaciones del modelo dinámico de un robot antropomórfico de tres grados de libertad, después de haber analizado el modelo dinámico de dicho robot de forma teórica.

Desarrollo

Para dicha simulación se crearon dos archivos en Matlab los cuales se denominaron Dinamica.m y Dinamica1.m.

En el primero se declara una función global llamada Dinamica en la cual se resolverá con el método de Runge Kutta, dentro de la función se definen las ecuaciones del modelo, las características del robot y del modelo son las masas, longitudes, longitudes al centro de masas, coeficientes de fricción seca, viscosa y gravitación. Así como también en este primer archivo se evalúan las propiedades de dicho modelo.

El segundo archivo es un script para ejecutar el modelo dinámico de un robot antropomórfico de tres grados de libertad, en el cual se especifican las condiciones que son los valores de q_1 , q_2 , q_3 , qp_1 , qp_2 , qp_3 , x , y , y z , para llamar el método Runge Kutta con la instrucción `ode45`, además en este script es en donde se realizan las gráficas de coordenadas operacionales y propiedades del Modelo.

Se dan las condiciones iniciales para que se pueda realizar la simulación, algunos de los valores son: $q_1 = 0,7854$, $q_2 = -0,7854$, $q_3 = -0,7854$, $qp_1 = 0$, $qp_2 = 0$, $qp_3 = 0$, $x = 0,175$, $y = 0,175$, $z = 0,4475$.

Las condiciones iniciales deberán ser validas de acuerdo con el modelo cinemática, es decir, la posición la velocidad y aceleración según dicho modelo, además la posición deberá pertenecer al espacio de trabajo.

Se propone un robot con las siguientes características, las masas son: $m_1 = 0,45$, $m_2 = 0,25$, y $m_3 = 0,15$, las longitudes son: $L_1 = 0,35$, $L_2 = 0,35$, y $L_3 = 0,2$, también es importante mencionar que se tomará en cuenta que el centro de masas de cada uno de los eslabones se localiza en la parte media de cada uno de estos.

Problemas

El principal problema al que se enfrenta al hacer la simulación fue la falta de comprensión de las fórmulas matemáticas, esto se resolvió revisando los apuntes del modelo cinemático y dinámico, no bastando la simple revisión también se fueron desarrollando paso a paso cada fórmula hasta llegar a comprenderlas bien.

Otro problema que se tuvo en la graficación de la propiedad de antisimetría, ya que al momento de graficar salía exactamente cero siendo que ésta propiedad sólo debe acercársele. Se hizo la corrección en ésta parte del código $if(abs(PROP) < 1e-6)$, se cambió el signo de $>$ (mayor que) por el de $<$ (menor que).

Resultados

A continuación se muestran las gráficas como resultado de la simulación. En la figura C.1 se muestran los resultados de la simulación, se puede observar el comportamiento de las coordenadas generalizadas y la velocidad angular en cada una de las articulaciones durante el desarrollo de la simulación. En donde las coordenadas generalizadas representan cada grado de libertad del sistema y la velocidad angular es la derivada de dichas coordenadas.

Explicación de resultados:

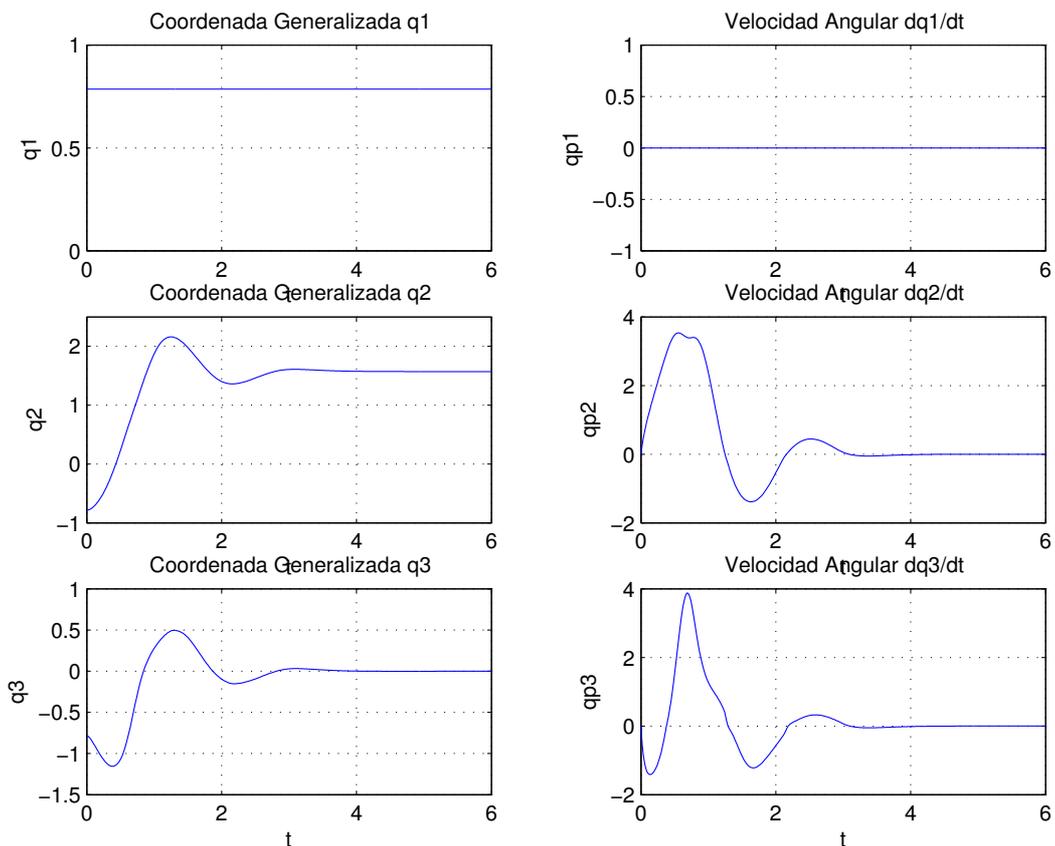


Figura C.1: Representación de las coordenadas generalizadas y las velocidades angulares

- La variable generalizada q_1 se mantiene estática debido a que no tiene un torque o fuerza que cambie su posición, por tal su velocidad es cero.
- Al contrario con las coordenadas q_2 y q_3 cuando:

- En q_2 observamos que a partir de un punto inicial de $45^\circ = 0,745$ radianes se posiciona oscilando en 1.54 radianes, para esa coordenada su velocidad qp_2 se incrementa pero casi de inmediato se estabiliza en cero, debido a que como el segundo eslabón dejó de moverse su coordenada articular también.
- En q_3 de igual forma partiendo del mismo punto inicial que q_2 , esta variable articular toma finalmente un valor de cero; para esa coordenada su velocidad qp_3 se incrementa pero se estabiliza en cero, debido a que como el tercer eslabón dejó de moverse su coordenada articular también.

Esto nos da a entender que en un momento en que el brazo se le quita el torque que lo mantenía en una posición, cae de tal forma que la gravedad actúa sobre él haciendo que tome la posición final, mostrada en las coordenadas generalizadas, en pocas palabras el brazo se venció.

En la Figura C.2 se muestran los resultados de la simulación, los cuales se pueden expresar en puntos que ocupa el efector final durante la tarea dentro del espacio de trabajo en cada uno de los ejes (x,y,z).

Explicación de resultados:

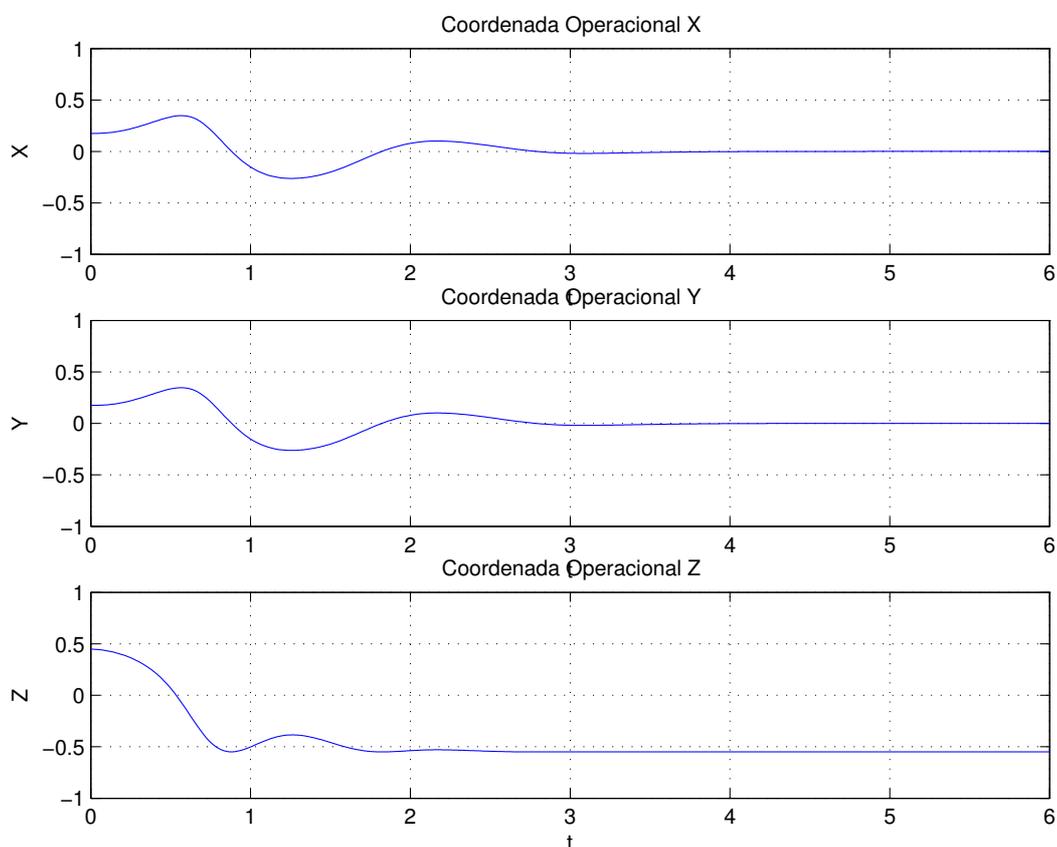


Figura C.2: Representación de la posición del órgano terminal.

- Las coordenadas operacionales X y Y tienen un momento de oscilación idénticas por un momento hasta que llegan a un punto de equilibrio en este caso a cero.
- Por su parte la coordenada operacional Z parte de un punto, comienza a oscilar hasta llegar a un punto de equilibrio, el punto final que tendrá el efector final será negativo porque el brazo se cae.

En la Figura C.3 se muestran los resultados de la evaluación de las propiedades de la matriz de inercia así como la propiedad de matriz antisimétrica.

Explicación de resultados:

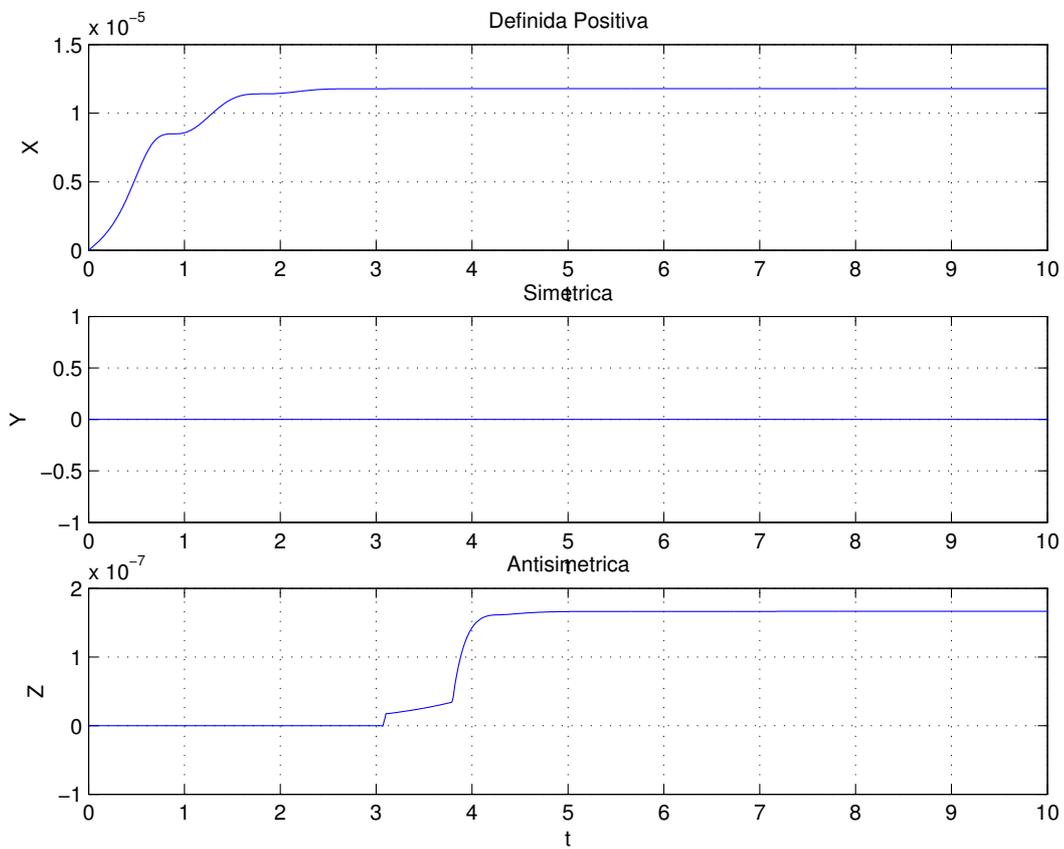


Figura C.3: Representación de las propiedades del modelo dinámico del robot antropomórfico

- Se puede observar que en la propiedad positiva los valores deben ser mayores de cero pero menores de 1 hasta llegar a un punto de equilibrio.
- En la propiedad simétrica se mantiene en cero, no hay oscilación,
- Y por último, la propiedad antisimétrica se mantiene constante durante un tiempo, después oscila un poco hasta quedar en un punto de equilibrio.

Apéndice D

Apéndice: Manual de usuario del sistema.

Para iniciar el sistema de “Seguimiento de Trayectoria” es necesario primero instalar el software necesario (Ver Figura D.1), como lo es el sistema mismo, la interfaz “PHANToM Virtual TRAZO” y otros archivos que son requeridos por esta última, para ello se proporciona un programa de instalación llamado “Seguimiento de trayectoria” que se encarga de crear un directorio llamado “Control” dentro del directorio raíz (Ver Figura D.2), en donde se copiarán todos los archivos necesarios, además de crear una opción dentro del menú de inicio (Ver Figura D.3), así mismo se podrá crear un icono en el escritorio si así lo desea (Ver Figura D.4), esto habilitando la opción cuando se lleve a cabo la instalación que es sencilla.



Figura D.1: Proceso de instalación Paso 1.



Figura D.2: Proceso de instalación Paso 2.

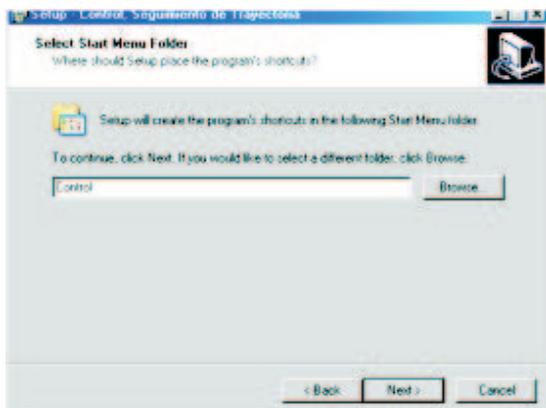


Figura D.3: Proceso de instalación Paso 3.

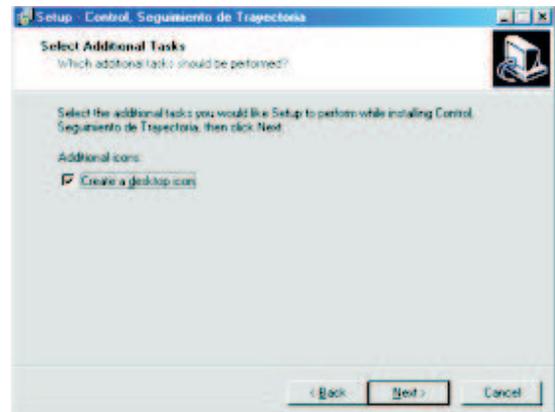


Figura D.4: Proceso de instalación Paso 4.

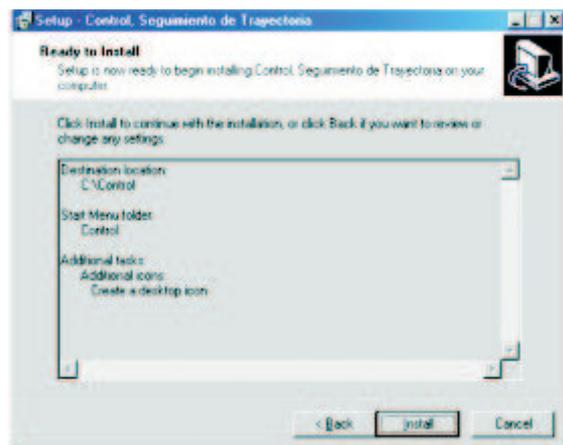


Figura D.5: Proceso de instalación Paso 5.

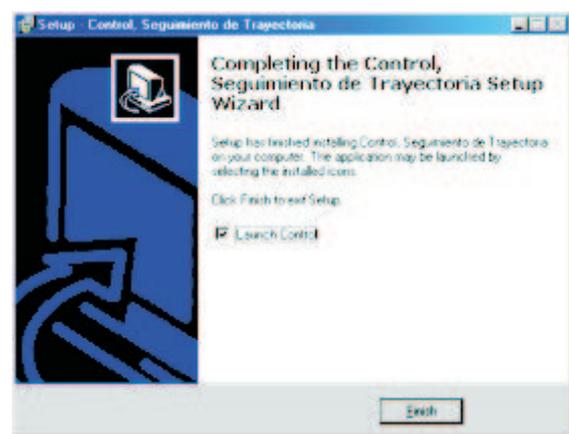


Figura D.6: Proceso de instalación Paso 6.

Para iniciar el programa una vez instalado deberá seleccionar el botón de inicio, de la barra de tareas, elegir programas y posteriormente elegir control e iniciar la aplicación de “Control” Figura D.7.

Una vez iniciada la aplicación aparecerá el menú principal en Figura D.8 donde el usuario podrá elegir las siguientes tareas:

- Trabajar con la interfaz “PHANToM Virtual TRAZO”.
- Trabajar con un control PIDNL.
- Trabajar con un control Adaptable (Demo MatLab).
- Salir de la aplicación

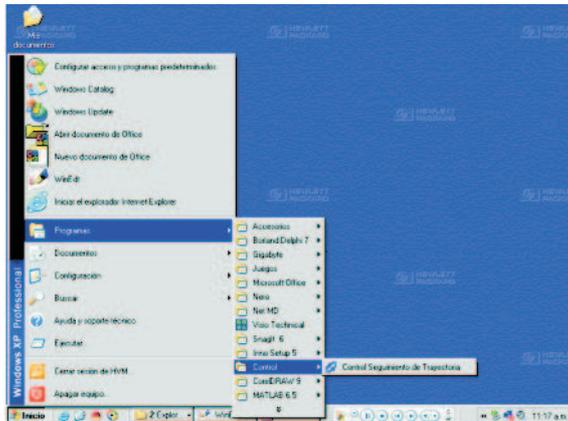


Figura D.7: Ubicación de la aplicación en el menú de inicio.



Figura D.8: Menú principal.

D.1. Robot Virtual

En la opción de robot virtual podrá trabajar con la “PHANToM Virtual TRAZO” (Ver Figura D.9), interactuar con el robot virtual esta interfaz solo se podrá manejar a través del teclado; nos permite una gran variedad de operaciones con el robot a continuación se presentan alguna de las operaciones básicas para hacer uso de ésta interfaz:

- Para ver el menú de opciones se utiliza la tecla M, de igual forma para desaparecer el menú.
- Mover las articulaciones del robot para dicha acción se utilizaran las teclas:
 - Q / W movimiento de la primer articulación.
 - A / S movimiento de la segunda articulación.
 - Z / K movimiento de la tercer articulación.
- También es posible mover el escenario con las siguientes teclas:
 - E / R con éstas teclas es posible acercar o alejar el escenario respectivamente.
 - D / F con éstas teclas es posible mover hacia arriba o hacia abajo el escenario respectivamente.
 - C / V con éstas teclas es posible mover hacia la izquierda o hacia la derecha el escenario respectivamente.
- Es posible incrementar o decrementar el espesor de cada uno de los elementos del robot mediante las teclas O / I respectivamente cabe hacer mención que la operación se realiza de forma simultanea en cada uno de los elementos del robot.
- Es posible rotar los ejes de referencia X, Y, esto mediante las teclas de desplazamiento como se indica:
 - Eje Y $\leftarrow\rightarrow$.
 - Eje X $\uparrow\downarrow$.
- Con la tecla F1 se pueden ver objetos sólidos u objetos trazados solo por líneas y vértices.
- Para restablecer el escenario después de una serie de movimientos, se utiliza la tecla F2.
- Para llevar al robot virtual a una posición inicial que esta definida en la interfaz se utiliza la tecla F3.
- Y tal vez la operación mas utilizadas para el seguimiento de trayectorias de esta interfaz serian por lo menos las dos primeras debido a que son la combinación para llevar acabo el seguimiento de trayectorias:
 - F5, carga los valores de la trayectoria.
 - F6, ejecuta latrayectoria a seguir por el brazo del robot.

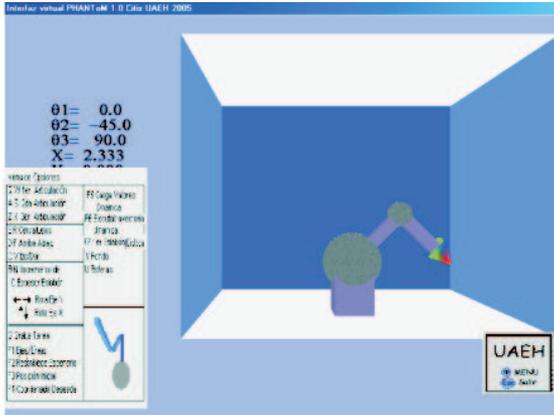


Figura D.9: Interfaz “PHANToM Virtual TRAZO” despliegue de menú de opciones.



Figura D.10: Coordenadas generalizadas y operacionales del robot virtual.

- P, al esta tecla la trayectoria seguida por el robot será marcada por una línea o un trazo y para quitar el ultimo trazo y volver a marcar se deberá teclear dos veces.
- Existen más operaciones, pero las anteriores son las necesarias para iniciar a manipular el robot.

En la interfaz “PHANToM Virtual TRAZO” es posible observar los valores que toman las coordenadas generalizadas y coordenadas operacionales en cualquier movimiento del Robot virtual (Ver Figura D.10).

D.2. Control PIDNL

En la opción de control PIDNL el usuario podrá experimentar con un control PIDNL ya sea con o sin convergencia en tiempo finito, además de observar el control PIDNL con TBG resuelto en MatLab (Ver Figura D.11), para llevar acabo el seguimiento de trayectoria de una circunferencia con centro en $C(0,3,0,2)$ y un radio de $r = 0,1$, partiendo siempre de una posición inicial que es $q_1 = 45^\circ, q_2 = -45^\circ, q_3 = -45^\circ$, el usuario podrá modificar las ganancias del robot par efectuar un mejor seguimiento al sintonizar cada vez mejor al robot.

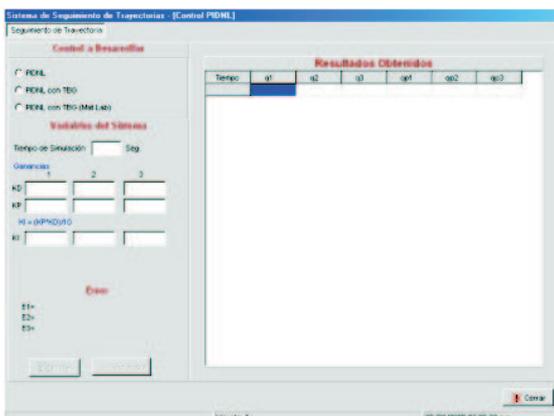


Figura D.11: Control PIDNL.

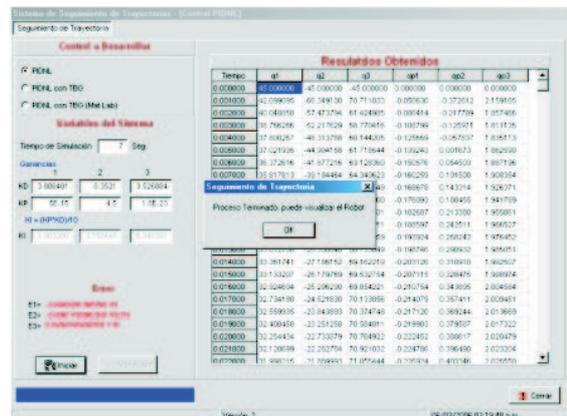


Figura D.12: Simulación Terminada Control PIDNL.

Al elegir la opción que desea ejecutar se habilitara el botón de “iniciar” en el caso de elegir PIDNL o PIDNL con TBG, o se habilitara el botón de ” ver robot” si se eligió el demo de MatLab, en el caso que ya se haya realizado alguna simulación al elegir una nueva opción los resultados y errores de esta son borrados para continuar con una nueva simulación y los valores de las ganancias y tiempos son restablecidos.

Al elegir PIDNL el usuario podrá iniciar la simulación con los parámetros que la aplicación tiene designados en:

- Tiempo de simulación
- Ganancias KD y KP

Éstos valores pueden ser cambiados por el usuario a su antojo, para experimentar que sucede en la simulación, las ganancias KI no podrán ser modificadas por el usuario debido a que dependen de las ganancias KD y KP; al iniciar la simulación aparecerá una barra en la esquina inferior derecha que nos indica progreso de la simulación y al terminar la simulación se despliega un mensaje que no indica que se ha terminado el proceso y es posible ahora ver el robot con el seguimiento de la trayectoria (Ver Figura D.12).

Al terminar la simulación no solo es posible ver el robot virtual, si no que también se muestran en pantalla en la parte izquierda los valores de los errores en cada una de las articulaciones, así como los valores obtenidos en la simulación de la tarea (Ver Figura D.14), de igual forma se habilitará un botón “Siguiente” para mostrar los valores en forma grafica; en el caso de que el usuario elija ver el robot virtual para ejecutar la trayectoria y ver el trazo de la misma, una vez que aparezca la interfaz “ PHANToM Virtual TRAZO” deberá pulsar la siguiente secuencia de teclas P, F5y F6 y el robot desarrollara la trayectoria (Ver Figura D.13).

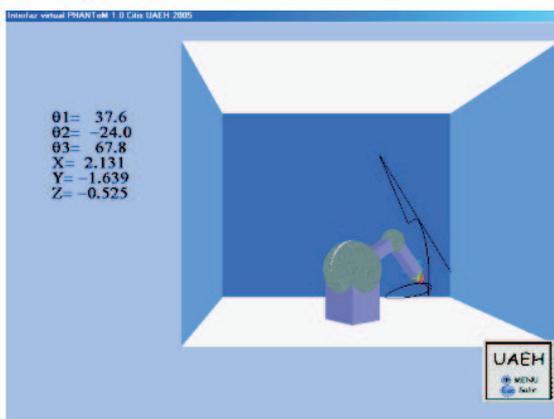


Figura D.13: Trayectoria realizada por el Robot virtual Control PIDNL.

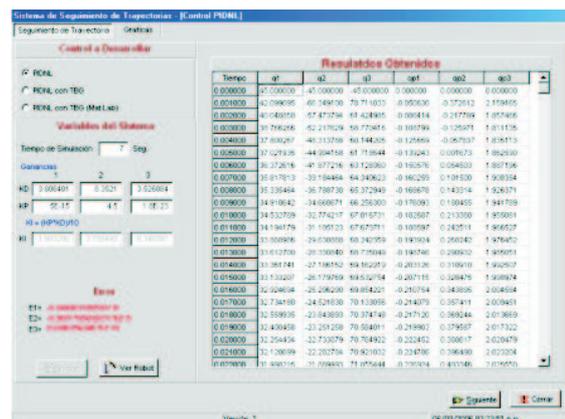


Figura D.14: Resultados de la simulación Control PIDNL.

En las graficas el usuario podrá observar los valores de las coordenadas generalizadas, así como los valores de los errores al realizar la simulación (Ver Figuras D.15, D.16, D.17 y D.18). Si se deseará llevar acabo otra simulación, sin que el sistema restablezca los valores solo basta con cambiar el tiempo de simulación o cualquier valor de las ganancias para que el botón de “iniciar” se habilite de nuevo.

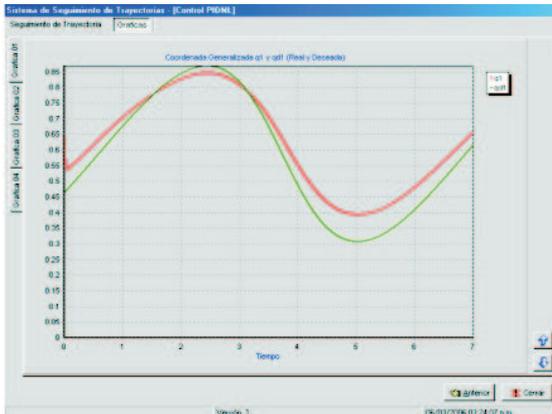


Figura D.15: Gráfica coordenada generalizada q_1 y q_{d1} Control PIDNL.

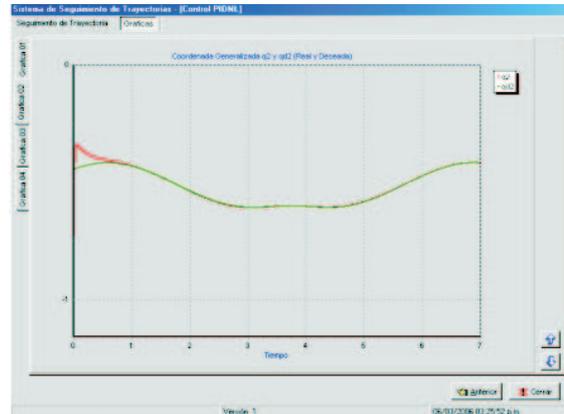


Figura D.16: Gráfica coordenada generalizada q_2 y q_{d2} Control PIDNL.

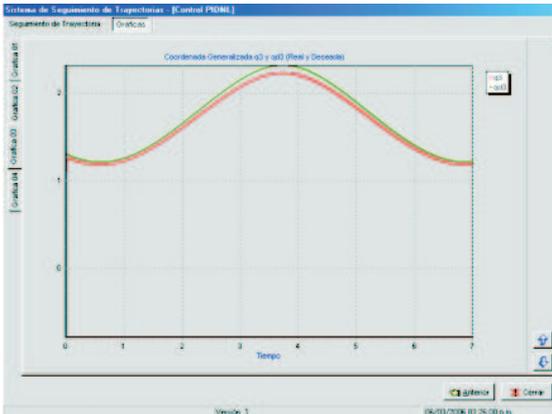


Figura D.17: Gráfica coordenada generalizada q_3 y q_{d3} Control PIDNL.



Figura D.18: Gráfica de errores Control PIDNL.

Al elegir PIDNL con TBG el usuario podrá iniciar la simulación con los parámetros que la aplicación tiene designados en:

- Tiempo de simulación
- Ganancias KD y KP
- Tiempo de Convergencia

Éstos valores pueden ser cambiados por el usuario a su antojo, para experimentar que sucede en la simulación, las ganancias KI no podrán ser modificadas por el usuario debido a que dependen de las ganancias KD y KP; al iniciar la simulación aparecerá una barra en la esquina inferior derecha que nos indica progreso de la simulación y al terminar la simulación se despliega un mensaje que no indica que se ha terminado el proceso y es posible ahora ver el robot con el seguimiento de la trayectoria(Ver Figura D.19).

Al terminar la simulación no solo es posible ver el robot virtual si no que también se muestran en pantalla en la parte izquierda los valores de los errores en cada una de las articulaciones, así como los valores obtenidos en la simulación de la tarea (Ver Figura D.21), de igual forma se habilitará un botón “Siguiente” para mostrar los valores en forma grafica;

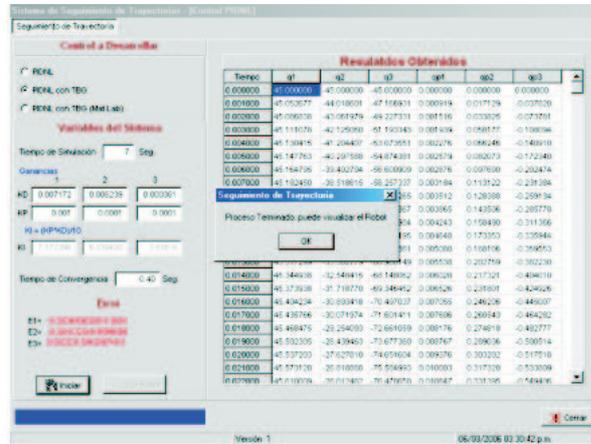


Figura D.19: Simulación Terminada Control PIDNL con TBG.

en el caso de que el usuario elija ver el robot virtual para ejecutar la trayectoria y ver el trazo de la misma, una vez que aparezca la interfaz “PHANToM Virtual TRAZO” deberá pulsar la siguiente secuencia de teclas P, F5 y F6 y el robot desarrollara la trayectoria (Ver Figura D.20).

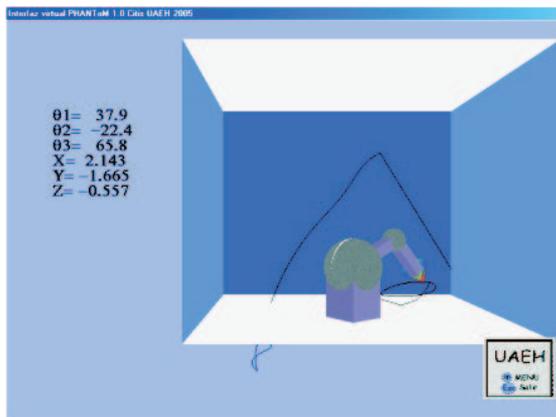


Figura D.20: Trayectoria realizada por el Robot virtual Control PIDNL con TBG.

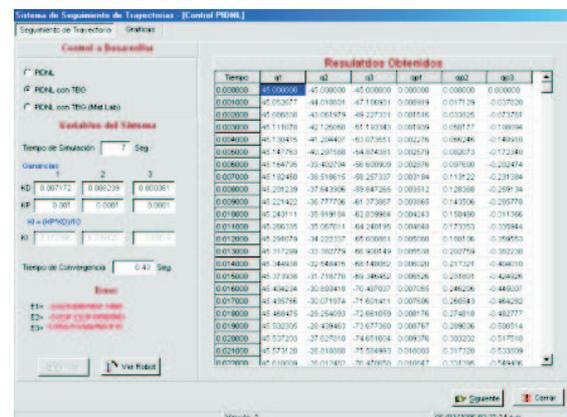


Figura D.21: Resultados de la simulación Control PIDNL con TBG.

En las graficas el usuario podrá observar los valores de las coordenadas generalizadas, así como los valores de los errores al realizar la simulación y observara la convergencia del robot (Ver Figuras D.22, D.23, D.24 y D.25). Si se deseará llevar acabo otra simulación, sin que el sistema restablezca los valores solo basta con cambiar el tiempo de simulación o tiempo de convergencia o cualquier valor de las ganancias para que el botón de “iniciar” se habilite de nuevo.

Al elegir PIDNL con TBG (Demo) el usuario podrá ver los resultados obtenido en MatLab en la parte derecha de la pantalla, en el mismo formato que en las simulaciones realizadas por esta aplicación, en la parte izquierda se muestran las condiciones de simulación (Ver Figura D.26), también podrá ver el código de MatLab (Ver Figura D.27), así como las graficas obtenidas(Ver Figura D.28). Y podrá de igual forma que las opciones anteriores ver el robot virtual y el seguimiento de la trayectoria pulsando el botón “Ver robot” y deberá pulsar la siguiente secuencia de teclas P, F5 y F6 y el robot desarrollara latrayectoria (Ver Figura D.29).

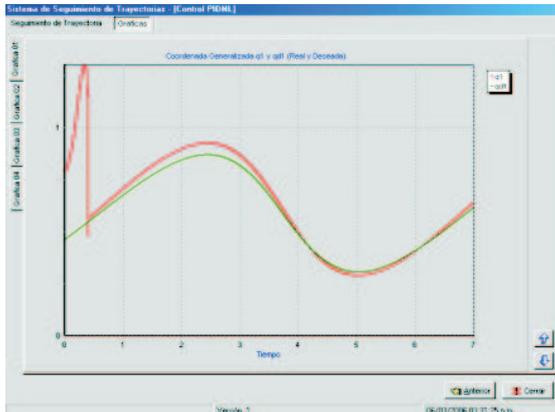


Figura D.22: Gráfica coordenada generalizada q_1 y q_{d1} Control PIDNL con TBG.

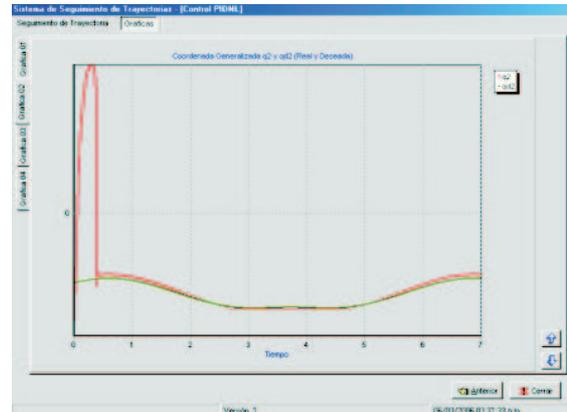


Figura D.23: Gráfica coordenada generalizada q_2 y q_{d2} Control PIDNL con TBG.



Figura D.24: Gráfica coordenada generalizada q_3 y q_{d3} Control PIDNL con TBG.



Figura D.25: Gráfica de errores Control PIDNL con TBG.

D.3. Control Adaptable

Al elegir Adaptable se muestra un Demo de un control adaptable con TBG en donde el usuario podrá ver los resultados obtenido en MatLab en la parte derecha de la pantalla, en el mismo formato que en las simulaciones realizadas por esta aplicación, en la parte izquierda se muestran las condiciones de simulación (Ver Figura D.30), también podrá ver el código de MatLab (Ver Figura D.31), así como las graficas obtenidas (Ver Figura D.32). Y podrá de igual forma que las opciones anteriores ver el robot virtual y el seguimiento de la trayectoria pulsando el botón “Ver robot” y deberá pulsar la siguiente secuencia de teclas P, F5 y F6 y el robot desarrollara la trayectoria (Ver Figura D.33).



Figura D.26: “Demo MatLab Control PIDNL con TBG”.

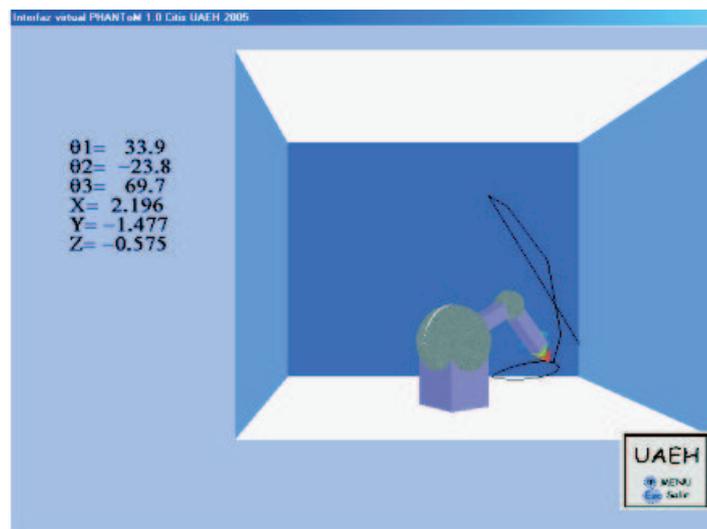


Figura D.27: Trayectoria realizada por el robot virtual Control PIDNL con TBG.

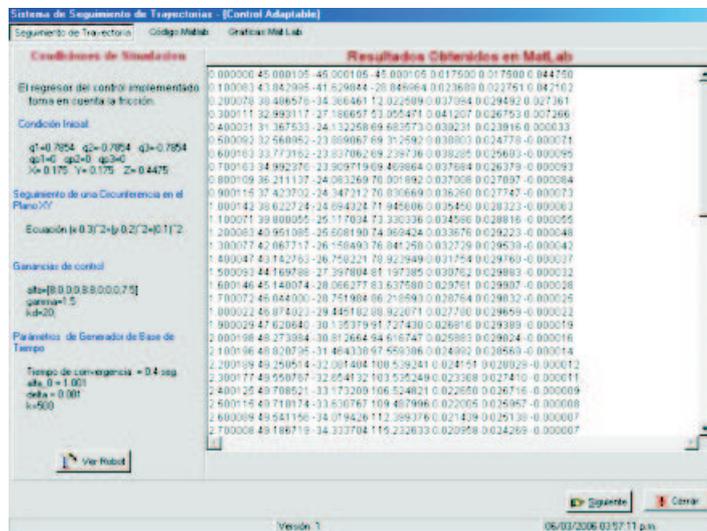


Figura D.30: “Demo MatLab Control Adaptable con TBG”.

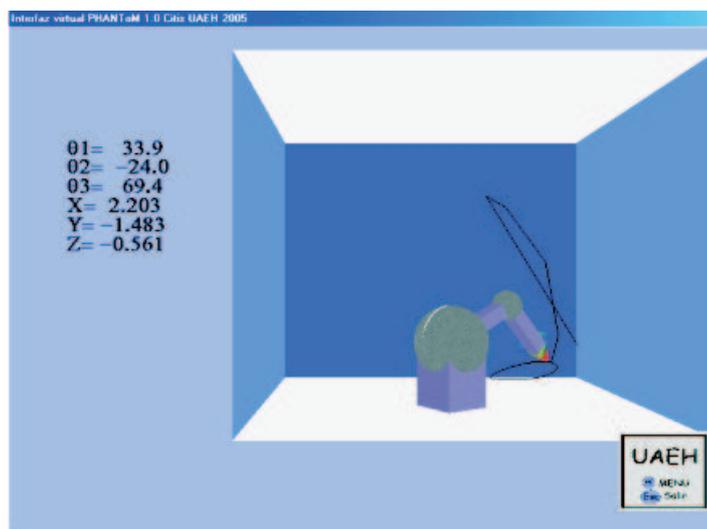


Figura D.31: Trayectoria realizada por el robot virtual Control Adaptable con TBG.

```

Sistema de Seguimiento de Trayectorias - [Control Adaptable]
Seguimiento de Trayectorias      Código Matlab      Graficos MatLab

Programa
% Programa que Efectúa un Control Adaptable Pasivo para Seguir
% en un robot antropomorfo de tres grados de libertad.
% Parte de la Condición Inicial: q1=0.7854, q2=-0.7854, q3=-0.7
% x=0.175, y=0.175 y z=0.4475
% Seguimiento de una Circunferencia en el Plano XY con Ecuador
% El rozador no toma en cuenta la fricción.
funcion di = adaptable(t,x)
%Parámetros supuestos del Robot Puenton y Ganados
global di
%////////////////////////////////////
m1=0.45; % masa de los eslabones
m2=0.25;
m3=0.15;
l1=0.75; % longitud de los eslabones
l2=0.35;
l3=0.2;
lcg1=0.5*l1; % longitud al centro de masa
lcg2=0.5*l2;
lcg3=0.5*l3;
q=0.61; % Coeficiente de gravitación

Script
% Script 1 para ejecutar el Programa que Efectúa un Control Adaptable
% Seguimiento de una Circunferencia en el Dispositivo Haptico Phantx
% de Posición 3 grados de Libertad.
% Parte de la Condición Inicial: q1=0.7854, q2=-0.7854, q3=-0.7854
% x=0.175, y=0.175 y z=0.4475
% Seguimiento de una Circunferencia en el Plano XY con Ecuador (x)
% Elabore: Omar Arturo Domínguez Ramírez.
clear
global di;
di=zeros(20,1);
options=odeset('MaxStep',0.1,'InitialStep',0.1);
[t,x]=ode45('adaptable',[0 7],[0.7854 0 -0.7854 0 0.175 0 0.175]);
% Referencia Desada en el Plano Operacional: Circunferencia C(1,1)
h=3;
k=2;
i=1;
w=1;
p=hw*cos(w*t);
pr=i+pr*cos(w*t);
z=0;

```

Figura D.32: Código MatLab Control Adaptable con TBG.

Nota: en caso de que se pulse el botón “Ver Robot” y se despliegue el mensaje “La interfaz esta en ejecución.. verifique” , se deberá pasar a la paliación pulsando las teclas *Alt + tab* hasta llegar a la aplicación “Interfaz virtual PAHNTom 1.0 CITIS UAEH 2005” y cerrarla y volver a la aplicación “Seguimiento de Trayectoria” o pulsar F2, F3 y P.



Figura D.33: Gráficas MatLab Control Adaptable con TBG.

Apéndice E

Artículo: Visualización virtual del Sistema de Seguimiento de trayectorias (Realidad virtual).

Visualización virtual del Sistema de Seguimiento de trayectorias (Realidad virtual).

González Rodríguez Araceli, Pineda Ortega Manuel y Soberanes Leal Dely Madai
Omar Arturo Domínguez Ramírez

Universidad Autónoma del Estado de Hidalgo
Instituto de Ciencias Básicas e Ingeniería
Carretera Tulacingo-Pachuca Km ,
Pachuca, Hidalgo, México.

Centro de Investigación en Tecnologías de Información y Sistemas
Ingeniería en Electrónica y Telecomunicaciones
ara.gonzalez@gmail.com, adenip264@gmail.com, msobranes@gmail.com

Resumen

Este trabajo surge de la necesidad de integrar una plataforma virtual dinámica para la evaluación de estrategias de control, para lo que es necesario conocer las herramientas de edición, así como los lenguajes existentes para desarrollar un ambiente virtual, de igual forma tener conocimiento de los modelos matemáticos cinemático y dinámico, dado que ayudan a representar a los robots para la realización de simulaciones matemáticas y de igual forma de las estrategias de control, lo cual se aplica para el seguimiento de trayectorias.

En este trabajo se muestra la implementación de los modelos matemáticos mencionados que se basan en la formulación de Euler-Lagrange para el modelo dinámico y la formulación de Denavit - Hartenberg para el modelo cinemático, los cuales en las simulaciones digitales se resolvieron en línea por la aplicación propuesta, esto mediante la programación del método de Runge - Kutta de orden superior. Lo anterior para la evaluación en línea de un control PID no lineal por modos deslizantes de segundo orden con y sin la implementación de un generador de tiempo base que nos ayuda a la convergencia de tiempo finito de robots manipuladores en aplicaciones que requieren de precisión en tiempo y espacio.

Palabras claves: Realidad virtual, herramientas de edición, lenguajes.

1. Introducción

El desarrollo de nuevas tecnologías es cada vez más acelerado. Algunas producen grandes impactos mientras otras no logran superar las expectativas iniciales más simples. En este contexto, ¿Por qué el desarrollo de la Realidad Virtual se presupone importante?. Hay varias razones de orden sociológico, económico y práctico que permiten suponer que ésta tecnología alcanzará una posición relevante: un aumento explosivo de la información y de la complejidad del mundo, que requiere de herramientas y personas con mayor destreza en el manejo de variables múltiples, y con una rápida capacidad de reacción. Pero al mismo tiempo, se requiere que el riesgo sea mínimo para el individuo expuesto.

Un elemento indispensable para los hombres es la comunicación. Dado el gran auge que han tenido las computadoras, aparece la necesidad de mejorar la comunicación entre ellas y sus usuarios, es decir, intentar una comunicación más natural con la máquina. Es en este escenario que surge el concepto de realidad virtual y los ambientes virtuales, inicialmente como una interfaz hombre-máquina más eficiente.

En el presente Artículo cuenta con cuatro secciones adicionales en la sección 2 se presentan algunos conceptos, características, así como una clasificación de la realidad virtual, en la sección 3 se exponen algunas de las herramientas de edición y lenguajes que nos ayudarán a la creación de un ambiente virtual, en la sección 4 se presenta un pequeño esbozo del ambiente virtual utilizado, en la sección 5 se presenta la plataforma propuesta por este equipo de trabajo para la simulación en línea del control PIDNL y por último se presenta la conclusión.

2. Conceptos preliminares de realidad virtual

Definir “Realidad virtual” (RV, o simplemente VR de su término inglés Virtual Reality) es difícil. Existen tantas definiciones como investigadores en un área, que dada su reciente y rápida evolución no han permitido tener una definición única. De este modo, no resulta extraño que la realidad virtual resulte ser altamente relativa para las diferentes personas y en las diferentes situaciones.

- Es una manera mediante la cual los humanos visualizan, manipulan e interactúan con computadoras y datos extremadamente complejos.
- Es un paso más allá de lo que sería la simulación por computadoras, tratándose más bien de una simulación interactiva, dinámica y en tiempo real de un sistema.
- Consiste en simulaciones tridimensionales interactivas que reproducen ambientes y situaciones reales.

Existen diversos términos para referirse al mismo concepto, tal es el caso de “realidad sintética”, “mundos virtuales o ficticios” y “ciberespacio”, palabras diferentes, mismos significados, idénticas indefiniciones.

Por regla general se reconoce que el término Realidad Virtual fue acuñado originalmente por Jaron Lanier, ex director de VPL Research, en 1989 (Myron Krueger fue el primero en atribuírselo), para distinguir entre simuladores y la experiencia de inmersión de realidad virtual, aún que otros autores difieren de esta posición planteando que ya en 1987 David Zelzer (Media Lab) utilizaba dicha expresión. Aunque el origen exacto del uso de la expresión “realidad virtual” es ambiguo, y más bien anecdótico, la cronología del desarrollo de la misma es bastante más clara.

En los inicios de los 90, los sistemas de realidad virtual salen de los ambientes de laboratorios en busca de aplicaciones comerciales, apareciendo las cabinas de entrenamiento para pilotos de guerra, los conocidos simuladores de vuelo, los recorridos virtuales, entre otras muchas aplicaciones.

En la actualidad, estamos aún en presencia del crecimiento y consolidación de las técnicas y recursos de la realidad virtual, el cual ha sido posible gracias al esfuerzo e interés combinado de científicos, militares y visionarios, y por que no decirlo, al dinero de las empresas que ven en ella una nueva y prometedora máquina de hacer dinero.

Para poder evaluar a un sistema realidad virtual como tal, es necesario que cumpla con ciertos elementos, entre los que se destacan[1]:

- **Simulación:** simulación para modelar un sistema, cabe decir que ésta tiene que ser lo suficientemente realista para convencer al usuario. Es decir una simulación del modelo o del mundo a experimentar, donde regirán una serie de reglas, no necesariamente iguales a las de la vida real.
- **Interacción:** interacción para tener control de la exploración del sistema creado; de no existir ésta interacción el sistema no dejaría de ser una película o recorrido fijado a priori. Para lograr la interacción existen diversas interfaces hombre-máquina, que van desde teclado y mouse hasta guantes o trajes sensoriales. La interactividad con el mundo virtual supone que el usuario pueda mover objetos (además de sí mismo) y modificarlos, y que tales acciones, produzcan cambios en ese mundo artificial.
- **Percepción (Inmersión):** la percepción viene a ser el factor más importante, algunos sistemas de realidad virtual se dirigirán principalmente a los sentidos (visual, auditivo, táctil) por medio de elementos externos (cascos de visualización, guantes de datos, cabinas, etc.); otros tratarán de llegar directamente al cerebro, evitando así las interfaces sensoriales externas; y otros, los más simples recurrirán a toda la fuerza de la imaginación del hombre para experimentar una realidad virtual relativa.
- **Generación de imágenes:** en un sistema de realidad virtual las imágenes mostradas al usuario no se encuentran almacenadas en ningún sitio, sino que son generadas dependiendo de la perspectiva que pretenda observarse.
- **Tridimensionalidad:** debe existir realmente una dimensión de profundidad, similar a la que ofrecen los simuladores de vuelo. Para conseguir este efecto de tridimensionalidad los objetos del mundo virtual deben tener asociada una tercera dimensión que marque su profundidad en la pantalla[1].

Los tres primeros elementos, son imprescindibles en cualquier sistema de realidad virtual, aunque algunos investigadores plantean que las tres bases de la realidad virtual son las “3i” (Interacción, Inmersión e Imaginación)[1].

2.1. Clasificación de la realidad virtual.

El hardware de realidad virtual es bastante variado. Así que en función de estos elementos involucrados, se pueden clasificar cuatro grandes grupos de sistemas que se proclaman como realidad virtual[1]:

- *Sistemas Desktop de Realidad Virtual:* o realidad virtual de escritorio, donde se engloban todas aquellas aplicaciones que muestran una imagen 2D ó 3D en un monitor en lugar de proyectarla en un “casco”. Los ejemplos típicos son los simuladores de vuelos y la mayoría de los juegos para PC’s.
- *Realidad Virtual en Segunda Persona:* son aquellos sistemas en los cuales el usuario sabe que está en el mundo virtual porque se ve a sí mismo dentro de la escena. Es decir es un integrante “visible” del mundo virtual, por medio de la proyección de su imagen en un fondo o ambiente, las cuales son visualizadas en la pantalla.
- *Sistemas de Telepresencia:* utilizan generalmente elementos como cámaras, micrófonos, dispositivos táctiles y de fuerza con elementos de retroalimentación, ligados a elementos de control remoto para permitir al usuario manipular robots u otros dispositivos ubicados en lugares remotos mientras se está experimentando en forma virtual.

- *Sistemas de Inmersión de Realidad Virtual*: Son aquellos que literalmente sumergen al usuario en el mundo virtual, mediante la utilización de sistemas visuales del tipo HMD, equipos sensores de posición y movimientos así como procesadores de sonido[1].

3. Herramientas de edición



Figura1. Portada de Autocad 2006[2].

1. Autocad.

AutoCAD de Autodesk es una herramienta dentro del campo denominado CAD (Computer Aided Design) o Diseño Asistido por Computadora. Para ser más exacto AutoCAD. es un programa de diseño en CAD analítico (frente a otros sistemas de CAD paramétrico). La versatilidad del sistema lo ha convertido en un estándar general[2].

Características:

- Dibujar de una manera ágil, rápida y sencilla, con acabado perfecto y sin las desventajas que encontramos si se ha de hacer a mano.
- Permite intercambiar información no solo por papel, sino mediante archivos, y esto representa una mejora en rapidez y efectividad a la hora de interpretar diseños, sobretodo en el campo de las tres dimensiones.
- Tiene herramientas para que el documento en papel sea perfecto, tanto en estética, como, lo más importante, en información, que ha de ser muy clara[2].



Figura2. Portada de 3D Studio Max 7.0 [3].

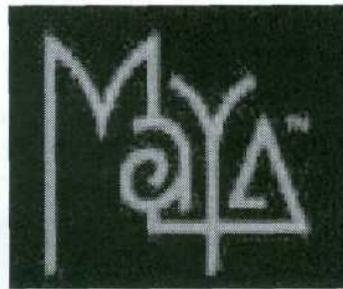


Figura3. Portada de Maya [4].

2. 3D Studio.

3D Studio Max es una aplicación basada en el entorno Windows (9x/NT) que permite crear tanto modelados como animaciones en tres dimensiones (3D) a partir de una serie de vistas o visores (planta y alzados). La utilización de 3D Studio Max permite al usuario la fácil visualización y representación de los modelos, así como su exportación y salvado en otros formatos distintos del que utiliza el propio programa.

Características:

- Es reescrito en un lenguaje de programación orientado a objetos.
- Funcionalidad reservada a costosas estaciones de trabajo gráficas.
- Integración de los entornos de modelado y animación.
- Manipulación en tiempo real con o sin presencia de tarjeta de aceleración de gráficos en 3D.

3. Maya.

Maya es un software de modelado 3D especialmente diseñado para la animación de personajes y la creación de efectos especiales[4].

Características:

Entre las principales características de animación de personajes en 3D implementadas en Maya se encuentran:

- Deformaciones que se pueden agrupar, reordenar y animar.
- Un conjunto de herramientas de IK (Inverse Kinematics) para tener el control preciso sobre el movimiento de los personajes.
- Control de la animación facial a partir de un conjunto de Blend Shapes.
- Herramientas de modelado de la piel.
- Control de las expresiones de los personajes.
- Soporte de captura de movimiento en vivo.

- Sincronización sonora integrada.
- Maya también incorpora un avanzado conjunto de herramientas para la creación de efectos visuales, como la caída de agua sobre un molino o la simulación de la colisión de dos galaxias.

3.1. Lenguajes

VRML



Figura4. Portada de VRML[5].

Es un acrónimo para Virtual Reality Modeling Language (Lenguaje para Modelado de Realidad Virtual). Técnicamente hablando, VRML no es un lenguaje para programar realidad virtual inmersiva ni tampoco un lenguaje de modelado. La realidad virtual inmersiva implica una experiencia tridimensional inmersiva y dispositivos externos como cascos o guantes digitales para lograr capturar otros sentidos diferentes al oído y a la vista. VRML no provee una inmersión sensorial total[5].

Características:

- Es un lenguaje estándar, y por consiguiente, universalmente utilizado en Internet como el lenguaje para simulaciones interactivas dentro de la Web.
- El VRML es un lenguaje para descripción de escenas no un lenguaje de programación.
- Provee un conjunto básico de primitivas para el modelaje geométrico tridimensional y tiene la capacidad de dar comportamiento a los objetos y asignar diferentes animaciones que pueden ser activadas por eventos generados por diferentes usuarios.
- Capaz de realizar rutas preestablecidas, como el movimiento del brazo de un robot, el giro de un ave en pleno vuelo, etc.
- Representación de los objetos en 3D.
- Recubriendo con Texturas un objeto.
- Manejo de cámaras.
- Manejo de luces y sombreados.
- Movimiento.
- Guión (Script)[6].

OpenGL

OpenGL es una API (Application Programming Interface) creada por Silicon Graphics Inc. para simplificar la programación de aplicaciones graficas en 2D ó 3D. OpenGL pone a nuestra disposición un pequeño conjunto de primitivas geométricas, a la par de estas primitivas, también brinda comandos para manipular la forma en que se verán los objetos. Una cosa importante de OpenGL es que es independiente del sistema operativo y del sistema de ventanas en el que se ejecute[7].

Características:

- Tecnología que facilitan la manipulación de los modelos 3D, e incluso puede ayudar a crearlos.
- No hay preocuparse por el hardware a la hora de programar, por lo tanto la programación no se hace tan complicada, de igual manera el código que se escribe con OpenGL será independiente de la plataforma y del sistema operativo.
- Sigue una serie de reglas para nombres de funciones y uso de variables.
- OpenGL permite al desarrollador escribir aplicaciones que se puedan desplegar en varias plataformas fácilmente.
- Primitivas geométricas: permiten construir descripciones matemáticas de objetos. Las actuales primitivas son: puntos, líneas, polígonos, imágenes y bitmaps.
- El doble buffering ayuda a eliminar el parpadeo de las animaciones. Cada fotograma consecutivo en una animación se construye en un buffer separado de memoria y mostrado sólo cuando está completo[7].

3.2. Visual C++

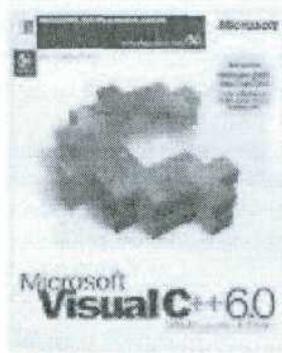


Figura5. Portada de Visual C++.

Como sabemos, Windows es el entorno más popular de interfaz gráfico de usuario (GUI). Desde este punto de vista, Windows es un entorno multitarea basado en ventanas, que representan programas, y que permite ejecución concurrente, permite la programación orientada a objetos (POO).

Características:

- Procesamiento de mensajes. El sistema operativo llama a main cuando el usuario ejecuta un programa.
- Programación basada en recursos. Para realizar una programación dirigida por datos en MS-DOS, es necesario codificar los datos como constantes de inicialización o proporcionar archivos de datos aparte para que los lea el programa.
- Gestión de memoria. Algunas de las funciones de gestión de memoria de Win16, como el GlobalAlloc, se trasladaron a Win32, pero se hizo para permitir a los desarrolladores modificar el código fuente con rapidez al pasar de una versión a otra.

3.3. Delphi 7.0



Figura6. Portada de Delphi 7.0.

¿Por qué Delphi?. La respuesta a esta pregunta puede no ser fácil. Se podría decir que la herramienta adecuada para un trabajo es aquella que permite la consecución correcta y a tiempo del mismo, además de estas condiciones mínimas, cada lenguaje se desempeña mejor en ciertas áreas o desarrollos; por ejemplo, Clipper fue concebido para manejar bases de datos con formato dBase, y es ese su fuerte.

Delphi “lenguaje de propósito general”, se puede definir como tal debido a que no tiene un lugar muy bien definido; Esto significa que se comporta bien ante tipos diferentes de problemas, y Borland ha puesto mucho ahínco en lograr que el rendimiento sea el mejor posible. Delphi se puede usar para casi cualquier tipo de programa, obteniéndose un rendimiento excelente con facilidad.

Características de Delphi:

- Posee un Entorno de Desarrollo Integrado para Windows (IDE, incluye un editor de texto ASCII, un visor de propiedades y eventos y un potente debugger.), con características de programación visual. Esto significa que la mayor parte del programa se hace gráficamente, lo hace ahorrar tiempo y esfuerzo.
- La programación se hace más intuitiva y sencilla con el uso de componentes.
- Genera ejecutables nativos, sin necesidad de librerías de run-time (salvo el caso de los programas que usan Bases de Datos).
- Se pueden crear componentes nuevos que se integran en el entorno de la misma manera que los nativos.

4. Construcción del Ambiente virtual

4.1. Animación al ambiente virtual

Una función importante para trabajar con el ambiente virtual cuando se emplea la función `glTranslatef(...)` y `glRotatef(...)`; es la función `glLoadIdentity()`, reestablece los valores de las coordenadas para el origen y también restablece los ángulos de giro de los ejes de referencia del píxel de trazo, (llevar al origen al centro del espacio virtual) y reestablecer ángulos de giro de los ejes de referencia

(llevarlos a cero).

La función `glTranslatef (...)` modifica las coordenadas (XYZ) origen del píxel de trazo. Por ejemplo:

```
glLoadIdentity (); //reestablece la coordenada del píxel de trazo
glTranslatef(3.0f, -2.0f, -5.0f); //traslación del origen del píxel
de trazo
```

El origen del píxel de trazo se desplazó 3 unidades a la derecha, 2 unidades hacia abajo, y 5 unidades hacia el fondo. Es importante explicar que los ejes están definidos en el espacio de trabajo virtual de la siguiente forma:

- El movimiento sobre la horizontal se hace sobre el eje X.
- El movimiento vertical se hace sobre el eje Y.
- EL movimiento hacia el fondo o acercar se hace sobre el eje Z

La función `glRotate3f(ANGULO, OPCIONX, OPCIONY, OPCIONZ)`; proporciona un giro en grados sexagesimales para los ejes de referencia del píxel de trazo.

El parámetro `ANGULO` puede ser positivo o negativo o mayor a 360, los parámetros `OPCIONX`, `OPCIONY` y `OPCIONZ` se les asignara el valor de 1.0f si se les desea aplicar el ángulo de rotación al eje correspondiente.

Las funciones de OpenGL dibujan y borran los objetos para producir una animación.

OpenGL emplea un doble buffer para presentar los objetos virtuales, esto quiere decir que un buffer contiene el escenario virtual que se visualiza, mientras que en el segundo buffer se esta dibujando la siguiente escena, y en su momento se intercambian los buffers

5. Plataforma Propuesta.

Considerando el propósito primordial de éste proyecto, que consiste en desarrollar una herramienta que permita la simulación, visualización del robot en el seguimiento de trayectorias (que en este caso de estudio es una circunferencia con un radio de 0.1 y con centro en (0.3,0.2)) así como la visualización de los resultados numéricos y su representación en forma gráfica, en ésta sección se presenta el análisis y diseño de ésta interfaz.

5.1. Análisis.

Para el desarrollo del sistema se consideraron los siguientes puntos:

- Visualización del Robot virtual.
- Simulación de un control PIDNL.
 - Control PIDNL.
 - Control PIDNL con convergencia en tiempo finito (TBG).

- Carga de Archivo de Matlab para la visualización de un PIDNL con convergencia en tiempo finito (TBG).
- Carga de Archivo de Matlab para la visualización de un control adaptable con convergencia en tiempo finito (TBG).

Nota: los puntos relacionados con control PIDNL así como adaptable la trayectoria que siguen es una circunferencia.

La visualización del robot se podrá realizar para cada uno de los puntos anteriores, la simulación se resuelve a través del método de solución de ecuaciones diferenciales de Runge Kutta, el método empleado es de orden superior debido a la exigencia de precisión en la tarea.

Para la visualización del robot virtual se retomó la aplicación “PHANToM Virtual TRAZO” [8], desarrollada bajo el lenguaje C++ con OpenGL, es por esto que la aplicación desarrollada se debió adaptarse al funcionamiento de la interfaz de visualización ya existente.

La carga de los archivos de Matlab se justifica debido al desarrollo de programas que resuelven el modelo matemático del robot a través de Matlab,.

A continuación se muestran los diagramas de flujo utilizados, en primer lugar se apoyó en un diagrama de flujo de información de sistema, el cual sirve para describir el flujo de información entre los distintos soportes físicos del sistema, refleja las operaciones normales para el desarrollo del proceso que realizan los componentes del sistema; de igual forma se muestran diagramas de flujo de detalle que abordan la solución del problema planteado.

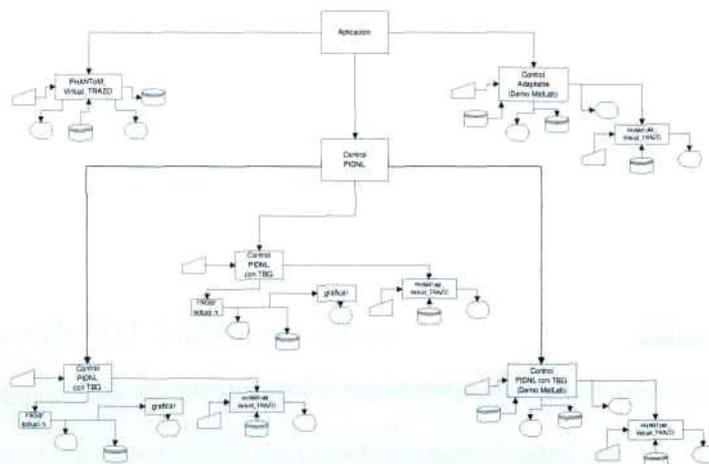


Figura7. Diagrama de flujo información del sistema “Seguimiento de trayectorias”.

El diagrama presentado en la Figura 8 se puede observar el procedimiento que se lanza o mejor dicho se inicia para dar paso al proceso de solución del problema de seguimiento de trayectoria con un control PIDNL ya sea con o sin TBG, en dicho procedimiento se inicializa al robot en una posición inicial, así como el primer punto a seguir que será cero para x, y y z, de acuerdo al tiempo especificado de simulación se calcula el número de iteraciones que se realizarán ya que el tiempo de simulación podrá ser cambiado por el usuario, en este procedimiento es donde se va generando el archivo de trayectoria que será leído por la interfaz "PHANToM Virtual TRAZO" de igual forma se generan las gráficas de las variables articulares y los errores de las mismas, para llevar acbo la solución se requiere de llamar a otros procedimientos y funciones.

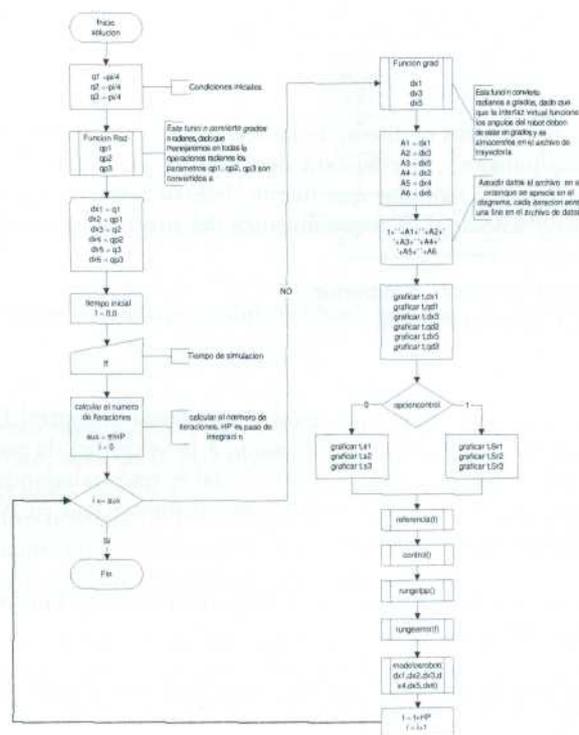


Figura8. Diagrama de flujo del procedimiento de solución.

Para llevar acabo el seguimiento se requiere saber el punto a seguir en cada una de la iteraciones, de lo que se encarga el procedimiento Referencia, el calculo de dichos puntos se hace por medio de la ecuación de la circunferencia, así también se hace uso de la primer y segunda derivada de la referencia, de igual forma es necesario calcular el modelo cinemático directo, cinemático inverso de velocidad y aceleración, la existencia de procedimiento de control, en el cual se se puede elegir un control PIDNL

sin TBG y otro con TBG, en este procedimiento se pone en práctica el Control PIDNL por modos deslizantes de segundo orden

El procedimiento rungetpp es el encargado de llevar a cabo el método de Runge Kutta de orden superior para la integración de q_{pp} y q_p para así obtener de cada integración anterior q_p y q .

Otro procedimiento utilizado es el de rungeerror que es de nuevo la interpretación del método de Runge Kutta de orden superior, pero esta vez es para la integración de una función denominada e_i la cual regresa un valor al procedimiento rungeerror.

Por último se muestra el procedimiento modelorobot el cual recibe parámetros que son las variables articulares, calculadas y así actualizar los modelos (dinámico y cinemático) del robot para las siguientes iteraciones.

5.2. Diseño.

En esta sección se presenta el diseño de la herramienta propuesta, "Modelado de Sistemas de control de un robot manipulador", el cual está desarrollado en Delphi 7 bajo la plataforma Windows XP; está formado por varios módulos que fueron desarrollados y probados independientemente antes de desarrollar un solo sistema. Los requerimientos del sistema desarrollado son:

- Sistema operativo win9x o superior.
- Procesador X86 o equivalente
- Teclado
- Mouse

Nota: El comportamiento de la aplicación al solucionar el control PINL en cualquiera de las dos opciones resueltas por dicha aplicación, en cuanto a la velocidad de respuesta dependerá de la velocidad del procesador del equipo de cómputo sobre el cual se esté trabajando; si se deseara que ésta aplicación se ejecutara sobre un SO UNIX se tendría que compilar con en Kilyx que es lenguaje Pascal para UNIX como Delphi es para Windows.

En ésta sección se muestra un diagrama de flujo de detalle del funcionamiento de la herramienta propuesta ver Figuras 9,10 y 11

En la aplicación se encuentran los siguientes módulos (ver Figura 12):

- Robot virtual.
- Control PIDNL.
- Control adaptable.

En el primer módulo se puede realizar la visualización del robot virtual así como interactuar con él Figura 13 .

En el segundo módulo se puede realizar la simulación del seguimiento de trayectoria ya mencionada con un control PIDNL con o sin generador de tiempo base (Figura 14, las dos opciones son resueltas en línea por mediante la aplicación del método de Runge Kutta antes mencionado, los valores que la aplicación despliega en referencia a las ganancias y tiempo de simulación y tiempo de convergencia en el caso del control PIDNL con generador de tiempo base, son valores de prueba mediante los cuales

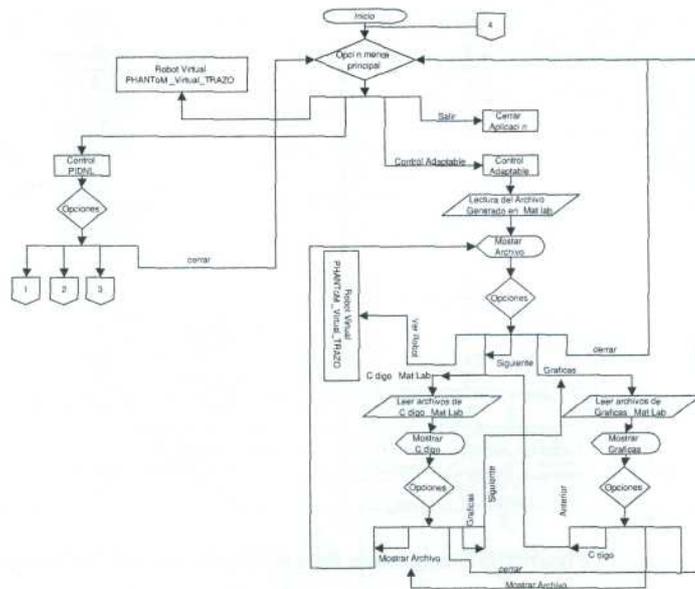


Figura9. Diagrama de flujo del funcionamiento del sistema.

este equipo obtuvo buenos resultados sin suponer que sean los mejores dado que esta aplicación es para experimentar con el robot, además la herramienta propuesta muestra las gráficas de las coordenadas generalizadas de las opciones resueltas en línea, este módulo presenta también un “demo” el cual realiza la carga de un archivo obtenido por Matlab, ya que también se trabajó con este software en donde se resolvió el mismo modelo matemático implementado en la aplicación; en este demo se muestra el código de MatLab Figura así como las gráficas obtenidas en MatLab.

El tercer módulo presenta un “demo” el cual realiza la carga de un archivo obtenido por Matlab, ya que también se trabajó con este software en donde se resolvió el mismo modelo matemático implementado en la aplicación; en cada una de las opciones se muestran los valores obtenidos numérica y gráficamente.

6. Conclusiones

Hoy en día la realidad virtual es un tema muy común o mejor dicho conocido, pero es necesario conocer todo lo que involucra la realidad virtual es por eso que se presenta una reseña, que incluye conceptos, clasificación y características generales de la realidad virtual. Existen diversas herramientas que se pueden usar para generar un ambiente virtual como los pueden ser los lenguajes y herramientas de edición, como AutoCad, 3D Studio, Maya, VRML, OpenGL, Visual C++, Delphi, entre otros.

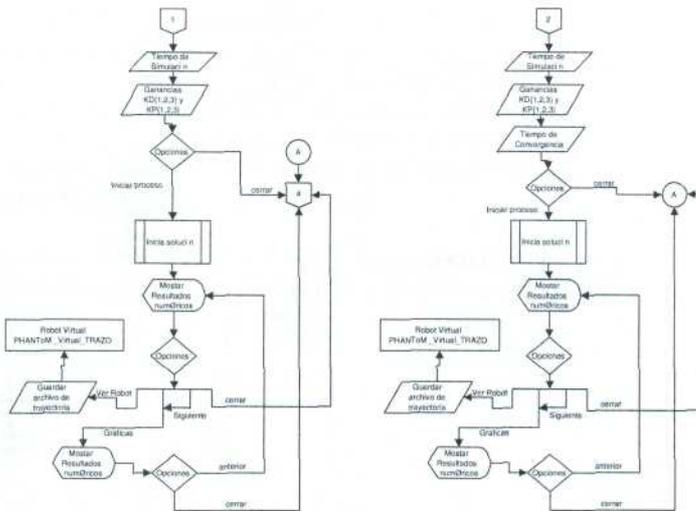


Figura10. Diagrama de flujo del funcionamiento del sistema.

Cabe señalar que al inicio de este proyecto la interfaz virtual se creó a partir de 3D Studio Max, se exportó a código Visual C++ con OpenGL y se comenzó a trabajar sobre este Visual C++, mientras tanto se liberó la aplicación “PHANToM Virtual TRAZO” desarrollada en el Centro de Investigación en Tecnologías de Información y Sistemas Ingeniería en Electrónica y Telecomunicaciones de esta Universidad, dicha aplicación está creada en lenguaje C++ con OpenGL y es una interfaz virtual que permite la interacción (movimientos y seguimiento de trayectorias) con un robot antropomórfico de tres grados de libertad, lo que se adapta a este trabajo de tesis, se retomo esta aplicación en este Capítulo se hace mención a la creación de esta aplicación y del comportamiento complejo de la misma.

Este equipo de trabajo desarrolló una interfaz de usuario que permite el seguimiento de trayectorias (en este caso una circunferencia), que inicialmente se trabajo en Visual C++ 6.0 y se termino en Delphi 7.0, debido a que este lenguaje (object pascal) y software es muy noble en aprendizaje, aplicación y colaboración con otras aplicaciones externas como “PHANToM Virtual TRAZO”, además de que puede trabajar con OpenGL de igual forma que Visual C++, si en un futuro se desea incorporar el robot a la aplicación propuesta.

Referencias

1. <http://espejos.unesco.org.uy/simplac2002/Ponencias/ambientes20digitales/AD003.doc>
2. <http://www.autodesk.es/adsk/servlet/index?siteID=455755id=5134721>
3. <http://www1.discreet.com/site/LatinAme.nsf/products/E38D3D66EB3A857A85256DD0004DD47C?opendocument>
4. <http://px379.narod.ru/infoftware3d.htm>
5. <http://agemenon.uniandes.edu.com/fde/isis380-2003-2/ejemplos/>

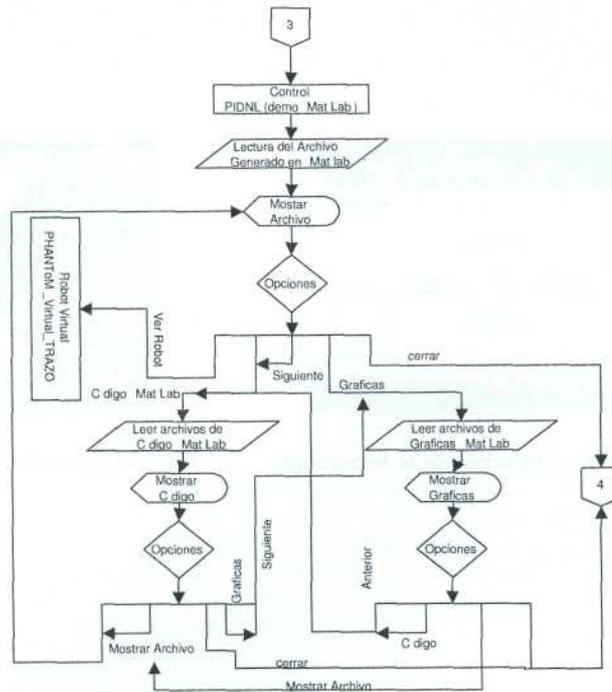


Figura11. Diagrama de flujo del funcionamiento del sistema.

6. <http://www.cosimir.com/VR/English/Technologie/Introduction.htm>

7. <http://dac.escet.urjc.es/formella/doc/ig03node31.html>

8. Herbert Lara Ordaz, "Análisis y diseño de una interfaz de visualización virtual del dispositivo háptico PHANTOM 1.0 basada en la dinámica de Euler-Lagrange", Universidad Autónoma del Estado de Hidalgo, Pachuca, Hidalgo, 2005.



Figura12. Menú principal de la herramienta propuesta.

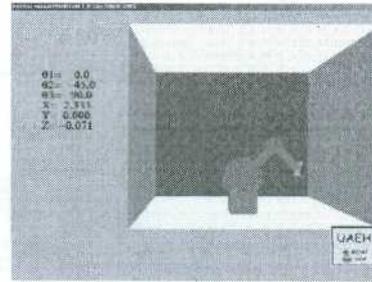


Figura13. Interfaz PHANToM Virtual TRAZO.

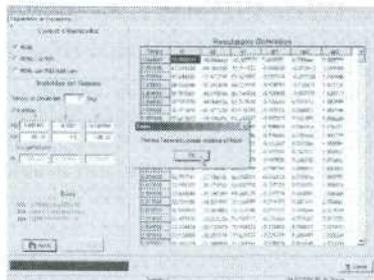


Figura14. Control PIDNL.

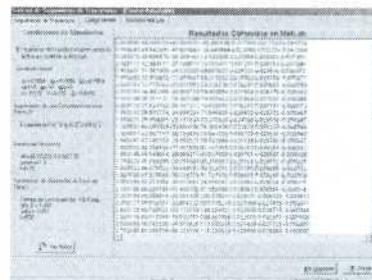


Figura15. Demo control adaptable con TBC realizado en MatLab.

Apéndice F

**Artículo: Control PID no Lineal con
Convergencia en Tiempo Finito.**

Control PID no Lineal con Convergencia en Tiempo Finito.

González Rodríguez Araceli, Pineda Ortega Manuel y Soberanes Leal Dely Madai
Omar Arturo Domínguez Ramírez

Universidad Autónoma del Estado de Hidalgo
Instituto de Ciencias Básicas e Ingeniería
Carretera Tulacingo-Pachuca Km ,
Pachuca, Hidalgo, México.
Centro de Investigación en Tecnologías de Información y Sistemas
Ingeniería en Electrónica y Telecomunicaciones
ara.gonzalez@gmail.com, adenip264@gmail.com, msobranes@gmail.com

Resumen

Este trabajo surge de la necesidad de integrar una plataforma virtual dinámica para la evaluación de estrategias de control, para lo que es necesario conocer los modelos matemáticos cinemático y dinámico, dado que ayudan a representar a los robots para la realización de simulaciones matemáticas, de igual forma se presentan estrategias de control clásicas y modernas, lo cual se aplica para el seguimiento de trayectorias. Los modelos matemáticos mencionados se basan en la formulación de Euler-Lagrange para el modelo dinámico y la formulación de Denavit - Hartenberg para el modelo cinemático, los cuales en las simulaciones digitales se resolvieron en MatLab con ayuda de la instrucción Ode45 (Runge - Kutta). Al control PID no lineal por modos deslizantes de segundo orden propuesto se le implementó un generador de tiempo base que ayuda a la convergencia de robots manipuladores en aplicaciones que requieren de precisión en tiempo y espacio. El control propuesto es aplicable a cualquier robot de 3 grados de libertad.

Palabras claves: robot, cinemática, dinámica, control.

1. Introducción

Robot es un sistema mecánico cuyo control debe estar encaminado a efectuar una tarea de utilidad[5]; Para un robot de coordenadas cartesianas los tres grados de libertad son: movimiento vertical (movimiento eje z), movimiento hacia adentro y afuera (movimiento eje y) y movimiento derecha o izquierda (movimiento eje x). Estos se consiguen por los movimientos correspondientes de los 3 dispositivos de deslizamiento ortogonales del brazo del robot[5], los grados de libertad de un robot se encuentran en relación con las articulaciones del mismo y éstas pueden ser de revolución o prismáticas[30]. Como todo sistema si consideramos a un robot como tal está constituido por diversos subsistemas como los son el electromecánico, comunicación, percepción, control y toma de decisión; existe una gran variedad de robots como lo pueden ser los manipuladores estos a su vez pueden ser de servicio o industriales, también existen los robots móviles autónomos, robots teleoperados, Androides, entre otros.

De acuerdo a la revisión del estado del arte, relativa a simuladores y visualizadores virtuales, no consideran las propiedades dinámicas inherentes al sistema físico que se desea representar virtualmente, existen trabajos como el proyecto alemán COSIMIR que permite la interacción de robots virtuales cooperativos

con propósitos educativos e industriales [1], sin embargo a pesar de que permite la programación real de trayectorias en robots industriales no emplea la dinámica del robot real, los movimientos virtuales están basados en las propiedades cinemáticas exclusivamente y los modelos son desarrollados mediante software para diseño CAD. Existe un laboratorio de robótica español denominado ROBOLAB [2], utilizado para la planificación de trayectorias para robots manipuladores, sin embargo la estrategia de planificación y ejecución virtual está basada en solo la cinemática del manipulador, teniendo como deficiencia la misma que el proyecto alemán COSIMIR. En México, los trabajos desarrollados en este ámbito son en el Tecnológico de Monterrey, ITESM, corresponde a un laboratorio virtual de robótica LVRM [3], cuyos propósitos son únicamente didácticos y permiten mediante una interfaz hombre máquina virtual planificar cinemáticamente los movimientos de robots manipuladores. Debido a las siguientes limitaciones: los modelos matemáticos están basados solo en aspectos cinemáticos, la cinemática es de posición, no consideran la cinemática diferencial (velocidad y aceleración), no consideran las fuerzas inerciales, gravitatorias, tribológicas, centrípetas y de Coriolis, como propiedades dinámicas inherentes a todos sistema robótico, evaluación de acciones de control lineal, cuando los robots son sistemas altamente no lineales, la planificación de trayectorias no considera el tiempo de convergencia.

Este trabajo propone un control PID no lineal con convergencia en tiempo finito en el seguimiento de trayectorias aplicable a un robot antropomórfico de 3 grados de libertad empleando metodologías de Denavit - Hartenberg para definir la cinemática directa e inversa, y la formulación de Euler - Lagrange para obtener el modelo dinámico del robot. El Presente artículo está ordenado en 7 secciones adicionales, la sección II presenta los modelos matemáticos del robot, cinemático y dinámico así como las propiedades de éste, en la sección III se expone el control PID no lineal propuesto así como los resultados obtenidos de la simulación de este control, en la sección IV se presenta la convergencia en tiempo finito, en la sección V se presentan los resultados de la integración del TBG al control PID no lineal, en la sección VI se muestran las conclusiones de éste trabajo y para finalizar las referencias utilizadas.

2. Modelos Matemáticos del robot

2.1. Cadena cinemática del robot

Es un ensamble de eslabones y juntas interconectados de modo que proporcionen un movimiento de salida controlado en respuesta a un movimiento de entrada proporcionado un ejemplo de una cadena cinemática se muestra en la figura 1.

2.2. Modelo Cinemático

La cinemática del brazo robot trata con el estudio analítico de la geometría del movimiento de un robot con respecto a un sistema de coordenadas de referencia fija como una función del tiempo, sin considerar las fuerzas/momentos que originan dicho movimiento. Así pues, trata con la descripción analítica del desplazamiento espacial del robot como función del tiempo, en particular las relaciones entre las variables espaciales de tipo articular, la posición y orientación del efector final del robot.

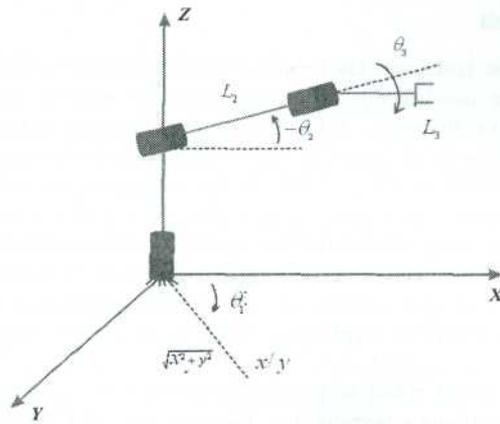


Figura1. Cadena cinemática de un robot de 3 grados de libertad.

Para el modelo cinemático directo de posición (MCDP):

$$x = f(q) \quad (1)$$

Modelo cinemático inverso de posición:

$$q = f^{-1}(x) \quad (2)$$

Modelo cinemático directo de velocidad:

$$\dot{x} = J\dot{q} \quad (3)$$

Modelo cinemático inverso de velocidad:

$$\dot{x} = \frac{d}{dt}f(q) = J\dot{q} \quad (4)$$

Modelo cinemático directo de aceleración:

$$\ddot{x} = J\ddot{q} + \dot{J}\dot{q} \quad (5)$$

Modelo cinemático inverso de aceleración:

$$\ddot{q} = J^{-1}[\ddot{x} + \dot{J}\dot{q}] \quad (6)$$

2.3. Modelo dinámico

La dinámica del robot trata con las formulaciones matemáticas de las ecuaciones de movimiento del brazo. Las ecuaciones de movimiento de un manipulador son un conjunto de ecuaciones matemáticas que describen su conducta dinámica. Tales ecuaciones son útiles para la simulación en computadora del movimiento del robot, el diseño de ecuaciones de control apropiadas para el robot y la evaluación del diseño y estructura del brazo. El objetivo del control de un manipulador basado en computadora es mantener la respuesta dinámica del mismo de acuerdo con algún rendimiento del sistema previamente especificado y objetivos deseados. En general, el rendimiento dinámico de un manipulador depende directamente de la eficacia de los algoritmos de control y de su modelo dinámico. El problema de control consiste en obtener modelos dinámicos del brazo del robot físico y a continuación especificar leyes o estrategias de control correspondientes para conseguir la respuesta y rendimiento del sistema deseado.

El modelo dinámico de un robot se puede obtener a partir de leyes físicas conocidas tales como las leyes de la mecánica newtoniana y lagrangiana. Este conduce al desarrollo de las ecuaciones de movimiento dinámico para las diversas articulaciones del manipulador en términos de los parámetros geométricos e inerciales de los elementos. Métodos convencionales como las formulaciones de Lagrange - Euler (L-E) y Newton - Euler (N-E) se pueden aplicar entonces sistemáticamente para desarrollar las ecuaciones de movimiento del robot.

2.4. Representación matricial del MD.

La representación matricial del modelo dinámico se basa en la siguiente expresión

$$T = H(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) \quad (7)$$

Donde (Ver Cuadro 1):

Función	Descripción
$H(q)\ddot{q}$	Matriz de inercia
$C(q, \dot{q})\dot{q}$	Matriz de coriolis y centrípetas
$G(q)$	Vector de fuerzas gravitatorias
$F(\dot{q})$	Vector de fuerzas de fricción seca y viscosa

Cuadro1. Descripción de las funciones de la representación matricial del modelo dinámico

Matriz de inercia

$$H(q) = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix} \quad (8)$$

Los elementos de la matriz de inercia son

$$\begin{aligned} H_{11} &= (m_2 + m_3) L_2^2 C_2^2 + 2m_3 L_2 L_3 C_2 C_{23} + m_3 L_3^2 C_{23}^2 \\ H_{12} &= 0 \\ H_{13} &= 0 \\ H_{21} &= 0 \\ H_{22} &= (m_2 + m_3) L_2^2 + m_3 L_2^2 + 2m_3 L_2 L_3 C_2 \\ H_{23} &= m_3 L_3^2 + m_3 L_2 L_3 C_3 \\ H_{31} &= 0 \\ H_{32} &= m_3 L_3^2 + m_3 L_2 L_3 C_3 \\ H_{33} &= m_3 L_3^2 \end{aligned} \quad (9)$$

La matriz de fuerzas de coriolis y centrípetas, así como sus elementos se definen a continuación

$$C(q, \dot{q}) = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \quad (10)$$

En donde:

$$\begin{aligned} C_{11} &= 0 \\ C_{12} &= -\{2(m_2 + m_3)L_2^2 S_2 C_2 + 2m_3 L_2 L_3 (S_{23} C_2 + S_2 C_{23}) + 2m_3 L_3^2 S_{23} C_{23}\} \dot{\theta}_1 \\ C_{13} &= \{2m_3 L_2 L_3 s_{23} c_2 + 2m_3 L_3^2 s_{23} c_{23}\} \dot{\theta}_1 \\ C_{21} &= \{(m_2 + m_3)L_2^2 S_2 C_2 + m_3 L_2 L_3 (S_{23} C_2 + S_2 C_{23}) + m_3 L_3^2 S_{23} C_{23}\} \\ C_{22} &= -\{2m_3 L_2 L_3 s_3\} \dot{\theta}_3 \\ C_{23} &= -\{m_3 L_2 L_3 S_3\} \dot{\theta}_3 \\ C_{31} &= \{m_3 L_2 L_3 S_{23} C_2 + m_3 L_3^2 S_{23} C_{23}\} \dot{\theta}_1 \\ C_{32} &= \{m_3 L_2 L_3 S_3\} \dot{\theta}_2 \\ C_{33} &= 0 \end{aligned} \quad (11)$$

El vector de fuerzas gravitatorias se define a continuación

$$G(q) = \begin{bmatrix} G_{11} \\ G_{21} \\ G_{31} \end{bmatrix} \quad (12)$$

y sus elementos son:

$$\begin{aligned} G_{11} &= 0 \\ G_{21} &= -m_2 L_{CG2} C_2 - m_3 \{L_2 C_2 - L_{CG3} C_{23}\} \\ G_{31} &= -m_3 L_{CG3} C_{23} \end{aligned} \quad (13)$$

El MD descrito en la ecuación 7 ha tomado en cuenta la fricción viscosa y seca, este vector fue definido como $F(\dot{q})$ el cual es la suma del vector de fricción seca y el vector de fricción viscosa que a continuación se describe:

$$F(\dot{q}) = Fv(\dot{q}) + Fs(\dot{q}) \quad (14)$$

en donde (ver Cuadro 2):

Vector	Descripción
$Fv(\dot{q})$	Fricción viscosa
$Fs(\dot{q})$	Fricción seca

Cuadro2. Descripción de los vectores de fricción.

$Fv(\dot{q})$ se describe a continuación

$$Fv(\dot{q}) = \begin{bmatrix} Fv_1(\dot{q}) \\ Fv_2(\dot{q}) \\ Fv_3(\dot{q}) \end{bmatrix} = \begin{bmatrix} b_1\dot{\theta}_1 \\ b_2\dot{\theta}_2 \\ b_2\dot{\theta}_2 \end{bmatrix} \quad (15)$$

en donde

b_i : es el coeficiente de fricción viscosa o rozamiento dinámico.

$\dot{\theta}_1$: es la velocidad angular.

$Fs(\dot{q})$ se describe a continuación

$$Fs(\dot{q}) = \begin{bmatrix} Fs_1(\dot{q}) \\ Fs_2(\dot{q}) \\ Fs_3(\dot{q}) \end{bmatrix} = \begin{bmatrix} K_1 \operatorname{sgn}(\dot{\theta}_1) \\ K_2 \operatorname{sgn}(\dot{\theta}_2) \\ K_2 \operatorname{sgn}(\dot{\theta}_3) \end{bmatrix} = \begin{bmatrix} K_1 \tanh(\beta_1 \dot{\theta}_1) \\ K_2 \tanh(\beta_2 \dot{\theta}_2) \\ K_2 \tanh(\beta_3 \dot{\theta}_3) \end{bmatrix} \quad (16)$$

en donde

K_i : es el coeficiente de fricción seca que depende del grado de lubricación de la articulación.

$\operatorname{sgn}(\cdot)$: es la función signo

$\dot{\theta}_i$: es la velocidad angular.

3. Control por modos deslizantes de segundo orden o PID no lineal.

En aplicaciones de control de robots, con tareas de regulación a una coordenada no singular del espacio de trabajo del manipulador, el control PD con compensación de gravedad y el control PID lineal tienen excelente desempeño. Sin embargo, estos controladores no tienen eficiente desempeño en tareas de seguimiento de trayectorias. Un control con eficiente desempeño en tareas de seguimiento, sin conocimiento de los parámetros dinámicos del robot, es el control PIDNL.

Si se sabe que S es el error y que éste contiene el error de posición (también llamado error extendido) y velocidad, entonces tenemos que la formulación es la siguiente:

$$S = e + \alpha \frac{de}{dt} \quad (17)$$

Y si entendemos que e está definida por:

$$e = \theta_{referencia} - \theta_{real} \quad (18)$$

Entonces derivando el error tenemos la siguiente ecuación:

$$\frac{de}{dt} = \frac{d}{dt}(\theta_{referencia} - \theta_{real}) \quad (19)$$

De esta forma el método es evaluado con el fin de determinar un momento que es próximo menor a los eslabones de referencia en comparación a los métodos anteriores.

Con esto nos puede llevar a un mayor control al momento de la ejecución. Sustituyendo en S los valores calculados se obtiene lo siguiente:

$$S = (\theta_{referencia} - \theta_{real}) + \alpha(\dot{\theta}_{referencia} - \dot{\theta}_{real}) \quad (20)$$

Por lo tanto la ecuación final es:

$$\tau = K_d * S + K_i \int \text{sgn}(s) dt \quad (21)$$

Donde:

$$\lim_{\beta \rightarrow \infty} \tanh(\beta S) = \text{Sgn}(S) \quad (22)$$

Por lo tanto:

$$\tau = K_d * S + K_i \int \tanh(\beta S) dt \quad (23)$$

Teniendo las siguientes condiciones:

$$S = e + \alpha \dot{e} \quad (24)$$

Donde la constante $\beta = 100$

Y las ganancias estarán definidas de la siguiente manera:

$$K_d = Wn^2 \quad (25)$$

$$\alpha = \frac{2}{Wn} \quad (26)$$

$$K_i \ll 2Wn^3 \quad (27)$$

Donde hay que mencionar que al referirse a Wn es la frecuencia natural que todo cuerpo posee y se tiene como constante, dentro del modelo es una constante que determina los valores iniciales a las constantes de proporción y derivación.

4. Seguimiento exacto: Convergencia en Tiempo Finito.

Un problema típico en el control de movimiento restringido de robots manipuladores, extensivo al guiado háptico, es la conmutación de movimiento libre a movimiento restringido. Para ello es necesario detectar el momento en el que ocurre la colisión o contacto entre el efector final, y el objeto que define la superficie geométrica de restricción (superficie virtual de contacto para el caso de estudio). Para ellos es necesario el conocimiento exacto de los estados en algún instante de tiempo, por lo tanto la estabilización asintótica complica considerablemente la transición por lo complicado que es definir con precisión el instante en el que ocurre la colisión. En estas condiciones la convergencia en tiempo finito, permite el conocimiento exacto de los estados en algún instante de tiempo. Para lograr la convergencia en tiempo finito, se requiere de un generador tiempo base TBG en el contexto de movimiento libre.

La función implementada para este caso de estudio se muestra en la ecuación 28:

$$f(t) = 1/2 - 1/2 \cos(wt) \quad (28)$$

donde:

$$w = 2\pi$$

$$f = 2\pi/t$$

$$t = \text{periodo} = 2tb$$

$$w = 2\pi/2tb$$

$$w = \pi/tb \text{ rad/seg}$$

$$\begin{aligned}
f(t) &= 1/2 - 1/2 \cos(\pi/tbt) \\
f'(t) &= \pi/2tb \operatorname{sen}(\pi/tbt) \\
f''(t = 1/2tb) &= 0 \\
f''(t) &= \pi^2/2tb^2 \cos(\pi/tbt) \\
f''(1/2tb) &= \pi^2/2tb^2 \cos(\pi/2) = 0
\end{aligned}$$

4.1. PIDNL con TBG

Para la implementación del control PINL con generador de tiempo bas la referencia nominal q_r en lazo cerrado, para que el robot tenga un comportamiento estable y convergencia en tiempo finito, se propone la siguiente ecuación

$$\dot{q}_r = \dot{q}_d - \alpha(t) \Delta q + S_d - K_i \sigma \quad (29)$$

$$\dot{\sigma} = \operatorname{sgn}(S_q) \quad (30)$$

donde,

$$S_q = S - S_d \quad (31)$$

$$S_d = S(t_0) e^{-kt} \quad (32)$$

donde $S = \dot{\Delta}q + \alpha(t) \Delta q$, $K_i = K_i^T \in \mathfrak{R}^{n \times n}$ es una matriz diagonal positiva definida, $k > 0$, $\operatorname{sgn}(\ast)$ denota la función signo de (\ast) . Notese que $\ddot{q}_r = \ddot{q}_d - \dot{\alpha}(t) \Delta q - \alpha(t) \dot{\Delta}q + \dot{S}_d - K_i \operatorname{sgn}(S_q)$ es discontinua, pero es utilizada solo para la prueba de estabilidad, y que $S_q(t_0) = 0$ para cualquier condición inicial. La coordenada del error extendido S_r ,

$$S_r = S_q + K_i \sigma \quad (33)$$

El control PID no lineal propuesto es el siguiente,

$$\tau = -K_d S_r \quad (34)$$

en términos de las anteriores definiciones,

$$\tau = -K_p \Delta q - K_v \dot{\Delta}q + K_d S_d - K_I I \quad (35)$$

$$I = \sigma = \int_{t_0}^t \operatorname{sgn}(S_q(\varsigma)) d\varsigma \quad (36)$$

donde $K_d = K_d^T \in \mathfrak{R}^{n \times n}$ es una matriz diagonal positiva definida, $K_p(t) = K_d \alpha(t)$, $K_v = K_d$ y $K_I = K_d K_i$ son matrices diagonales que representan las ganancias de control, y $\alpha(t)$ es definida como $\sigma(t)$ en la ecuación (37) y con un generador de tiempo base $\xi(t)$ tal que se obtengan las trayectorias presentadas en la Figura 2.

$$\sigma(t) = \sigma_0 \frac{\dot{\xi}}{1 - \xi + \delta} \quad (37)$$

donde $\sigma_0 = 1 + \varepsilon$, $0 < \varepsilon \ll 1$ y $0 < \delta \ll 1$. El generador de tiempo base $\xi(t) \in C^2$ se diseña de forma tal que tenga un desempeño suave de 0 a 1 en un tiempo finito arbitrario $t = t_b > 0$ y $\dot{\xi}(t)$ sea una campana tal que $\dot{\xi}(t_0) = \dot{\xi}(t_b) \equiv 0$ como se representa en la Figura 2.

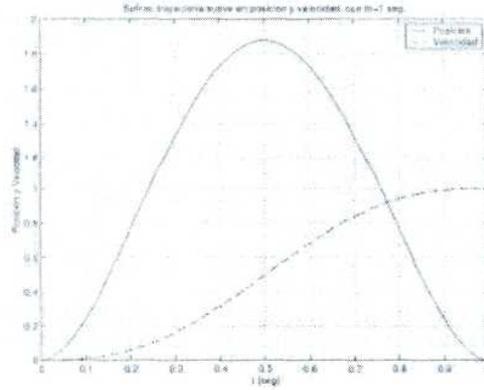


Figura2. Generador de base de tiempo $\xi(t)$ y derivada $\dot{\xi}(t)$ con $t_b = 1$ segundo.

La simulación digital en MatLab, consiste en desarrollar el seguimiento de una trayectoria en el espacio operacional del robot virtual con convergencia en tiempo finito, las trayectorias deseadas de posición son 38 y 39:

$$x(t) = h + r \cos(wt) \quad (38)$$

$$y(t) = k + r \sin(wt) \quad (39)$$

Para dicha simulación se asume que el dispositivo (robot virtual) parte de un estado de reposo con una condición inicial de posición, que esta definida en: $q_1 = 45^\circ$, $q_2 = -45^\circ$, $q_3 = -45^\circ$. Las ganancias utilizadas en el control son: $k_p = [0,001, 0, 0; 0, 0,001, 0; 0, 0, 0,001]$, $k_d = [24, 0, 0; 0, 122, 0; 0, 0, 4,5]$, $k_i = k_p \times k_d / 10$, el tiempo de convergencia empleado es: $t_b = 0,4 \text{ seg}$, con: $\alpha_0 = 1,001$, $\delta = 0,001$

En la Figura 3 se puede apreciar que con la implementación de TBG en un control PIDNL la coordenada generalizada q_1 se comporta de manera diferente al control PIDNL (figura 1), el tiempo de convergencia empleado en la simulación fue de 0.4 seg. Tiempo en el cual q_1 alcanza los valores deseados provocando un mejor desempeño del robot ya que el error se reduce de manera significativa.

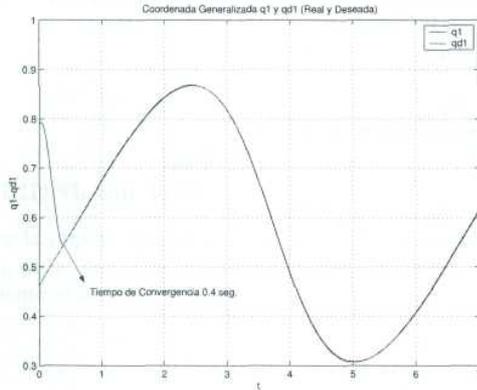


Figura3. Control PIDNL con TBG, Coordenada Generalizada q_1 y q_{d1} (Real y Deseada).

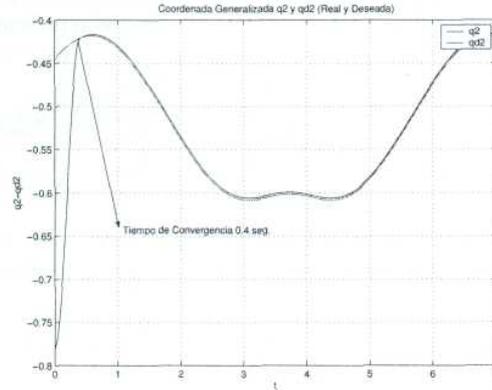


Figura4. Control PIDNL con TBG, Coordenada Generalizada q_2 y q_{d2} (Real y Deseada).

En la Figura 4 se puede apreciar que con la implementación de TBG en un control PIDNL la coordenada generalizada q_2 se comporta de manera diferente al control PIDNL, el tiempo de convergencia empleado en la simulación fue de 0.4 seg. Tiempo en el cual q_2 alcanza los valores deseados provocando un mejor desempeño del robot ya que el error se reduce de manera significativa, además de eliminar las oscilaciones que presenta con el control PIDNL.

En la Figura 5 se puede apreciar que con la implementación de TBG en un control PIDNL la coordenada generalizada q_3 se comporta de manera diferente al control PIDNL, el tiempo de convergencia empleado en la simulación fue de 0.4 seg. Tiempo en el cual q_3 alcanza los valores deseados provocando un mejor desempeño del robot ya que el error se reduce de manera significativa.

La trayectoria realizada por el robot virtual y la trayectoria deseada, además de apreciar que ambas trayectorias son iguales en la mayor parte de la simulación, claro no al iniciar debido a las condiciones iniciales del robot virtual, este logra cumplir la trayectoria (circunferencia), se puede notar que la trayectoria del robot al tratar de alcanzar la trayectoria deseada es diferente a la Figura 6.

Se puede afirmar que con la implementación del TBG en el control PIDNL mejora el desempeño del robot virtual, ya que el error extendido al seguir la trayectoria deseada tiende a cero o es cero en la mayor parte de la simulación, como se puede observar en la Figura 7.

5. Conclusión

El control PID no lineal a diferencia de otros controles clásicos, reduce el error estacionario que no es deseable en ninguna situación real, al implementarle el tiempo de convergencia finito ya que reduce el tiempo en que la trayectoria real alcanza a la deseada mejorando el desempeño del robot, esto es muy importante en la industria en tareas en las que el tiempo y la precisión son puntos importantes en el desarrollo de las mismas.

Referencias

1. <http://www.cosimir.com/VR/English/Technologie/Introduction.htm>
2. <http://disclab.ua.es/robofab/index.html>
3. <http://www-cia.mty.itesm.mx/gordillo/GV/personal.html>

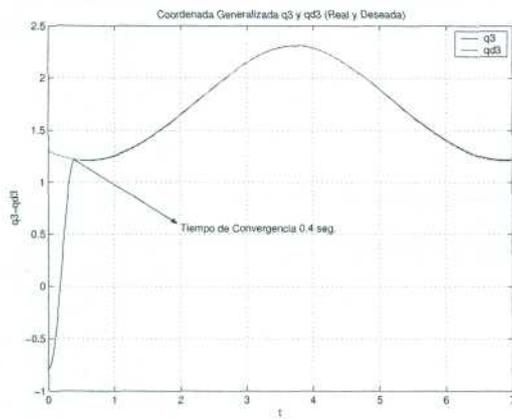


Figura5. Control PIDNL con TBG, Coordenada Generalizada q_3 y q_{d3} (Real y Deseada).

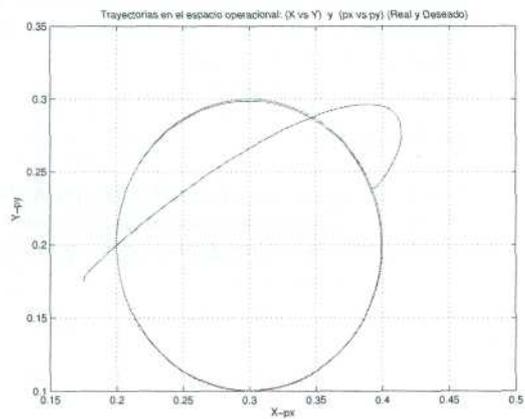


Figura6. Control PIDNL con TBG, Trayectorias en el espacio operacional: (x,y) y (p_x,p_y) , (Real y Deseada).

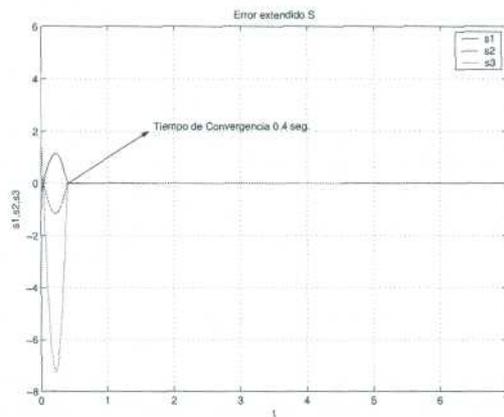


Figura7. Control PIDNL con TBG, Error extendido S.

Apéndice G

Artículo: Seguimiento Adaptativo de Trayectorias con Convergencia en Tiempo Finito de un Robot Antropomórfico Virtual.

Seguimiento Adaptativo de Trayectorias con Convergencia en Tiempo Finito de un Robot Antropomórfico Virtual.

González Rodríguez Araceli, Pineda Ortega Manuel y Soberanes Leal Dely Madai
Omar Arturo Domínguez Ramírez

Universidad Autónoma del Estado de Hidalgo
Instituto de Ciencias Básicas e Ingeniería
Carretera Tulancingo-Pachuca Km ,
Pachuca, Hidalgo, México.

Centro de Investigación en Tecnologías de Información y Sistemas
Ingeniería en Electrónica y Telecomunicaciones

ara.gonzalez@gmail.com, adenip264@gmail.com, msobranes@gmail.com

Resumen

Este trabajo surge de la necesidad de integrar una plataforma virtual dinámica para la evaluación de estrategias de control, para lo que es necesario conocer los modelos matemáticos cinemático y dinámico, dado que ayudan a representar a los robots para la realización de simulaciones matemáticas, de igual forma se presentan estrategias de control clásicas y modernas, lo cual se aplica para el seguimiento de trayectorias. Los modelos matemáticos mencionados se basan en la formulación de Euler-Lagrange para el modelo dinámico y la formulación de Denavit - Hartenberg para el modelo cinemático, los cuales en las simulaciones digitales se resolvieron en MatLab con ayuda de la instrucción Ode45 (Runge - Kutta). Al control adaptativo pasivo robusto propuesto se le implementó un generador de tiempo base que ayuda a la convergencia de robots manipuladores en aplicaciones que requieren de precisión en tiempo y espacio. El control propuesto es aplicable a cualquier robot de 3 grados de libertad.

Palabras claves: robot, cinemática, dinámica, control.

1. Introducción

Robot es un sistema mecánico cuyo control debe estar encaminado a efectuar una tarea de utilidad[5]; Para un robot de coordenadas cartesianas los tres grados de libertad son: movimiento vertical (movimiento eje z), movimiento hacia adentro y afuera (movimiento eje y) y movimiento derecha o izquierda (movimiento eje x). Estos se consiguen por los movimientos correspondientes de los 3 dispositivos de deslizamiento ortogonales del brazo del robot[5], los grados de libertad de un robot se encuentran en relación con las articulaciones del mismo y éstas pueden ser de revolución o prismáticas[30]. Como todo sistema si consideramos a un robot como tal está constituido por diversos subsistemas como los son el electromecánico, comunicación, percepción, control y toma de decisión; existe una gran variedad de robots como lo pueden ser los manipuladores estos a su vez pueden ser de servicio o industriales, también existen los robots móviles autónomos, robots teleoperados, Androides, entre otros.

De acuerdo a la revisión del estado del arte, relativa a simuladores y visualizadores virtuales, no consideran las propiedades dinámicas inherentes al sistema físico que se desea representar virtualmente, existen

trabajos como el proyecto alemán COSIMIR que permite la interacción de robots virtuales cooperativos con propósitos educativos e industriales [1], sin embargo a pesar de que permite la programación real de trayectorias en robots industriales no emplea la dinámica del robot real, los movimientos virtuales están basados en las propiedades cinemáticas exclusivamente y los modelos son desarrollados mediante software para diseño CAD. Existe un laboratorio de robótica español denominado ROBOLAB [2], utilizado para la planificación de trayectorias para robots manipuladores, sin embargo la estrategia de planificación y ejecución virtual está basada en solo la cinemática del manipulador, teniendo como deficiencia la misma que el proyecto alemán COSIMIR. En México, los trabajos desarrollados en este ámbito son en el Tecnológico de Monterrey, ITESM, corresponde a un laboratorio virtual de robótica LVRM [3], cuyos propósitos son únicamente didácticos y permiten mediante una interfaz hombre máquina virtual planificar cinemáticamente los movimientos de robots manipuladores. Debido a las siguientes limitaciones: los modelos matemáticos están basados solo en aspectos cinemáticos, la cinemática es de posición, no consideran la cinemática diferencial (velocidad y aceleración), no consideran las fuerzas inerciales, gravitatorias, tribológicas, centrípetas y de Coriolis, como propiedades dinámicas inherentes a todos sistema robótico, evaluación de acciones de control lineal, cuando los robots son sistemas altamente no lineales, la planificación de trayectorias no considera el tiempo de convergencia.

Este trabajo propone un control adaptable pasivo robusto con convergencia en tiempo finito en el seguimiento de trayectorias aplicable a un robot antropomórfico de 3 grados de libertad empleando metodologías de Denavit - Hartenberg para definir la cinemática directa e inversa, y la formulación de Euler - Lagrange para obtener el modelo dinámico del robot. El Presente artículo está ordenado en 7 secciones adicionales, la sección II presenta los modelos matemáticos del robot, cinemático y dinámico así como las propiedades de éste, en la sección III se expone el control adaptable propuesto y la parametrización lineal del modelo dinámico así como los resultados obtenidos de la simulación de este control, en la sección IV se presenta la convergencia en tiempo finito, en la sección V se presentan los resultados de la integración del TBG al control adaptable, en la sección VI se muestran las conclusiones de éste trabajo y para finalizar las referencias utilizadas.

2. Modelos Matemáticos del robot

2.1. Cadena cinemática del robot

Es un ensamble de eslabones y juntas interconectados de modo que proporcionen un movimiento de salida controlado en respuesta a un movimiento de entrada proporcionado un ejemplo de una cadena cinemática se muestra en la figura 1.

2.2. Modelo Cinemático

La cinemática del brazo robot trata con el estudio analítico de la geometría del movimiento de un robot con respecto a un sistema de coordenadas de referencia fija como una función del tiempo, sin considerar las fuerzas/momentos que originan dicho movimiento. Así pues, trata con la descripción analítica del desplazamiento espacial del robot como función del tiempo, en particular las relaciones entre las variables espaciales de tipo articular, la posición y orientación del efector final del robot.

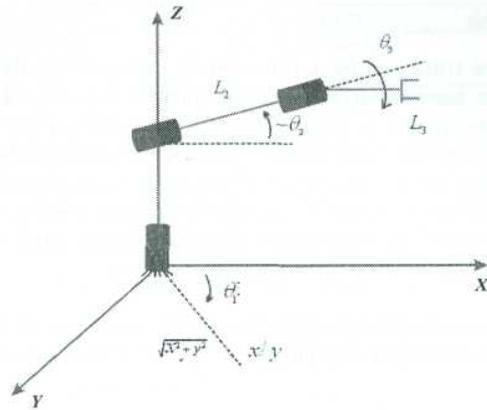


Figura1. Cadena cinemática de un robot de 3 grados de libertad.

Para el modelo cinemático directo de posición (MCDP):

$$x = f(q) \tag{1}$$

Modelo cinemático inverso de posición:

$$q = f^{-1}(x) \tag{2}$$

Modelo cinemático directo de velocidad:

$$\dot{x} = J\dot{q} \tag{3}$$

Modelo cinemático inverso de velocidad:

$$\dot{x} = \frac{d}{dt}f(q) = J\dot{q} \tag{4}$$

Modelo cinemático directo de aceleración:

$$\ddot{x} = J\ddot{q} + \dot{J}\dot{q} \tag{5}$$

Modelo cinemático inverso de aceleración:

$$\ddot{q} = J^{-1}[\ddot{x} + \dot{J}\dot{q}] \tag{6}$$

2.3. Modelo dinámico

La dinámica del robot trata con las formulaciones matemáticas de las ecuaciones de movimiento del brazo. Las ecuaciones de movimiento de un manipulador son un conjunto de ecuaciones matemáticas que describen su conducta dinámica. Tales ecuaciones son útiles para la simulación en computadora del movimiento del robot, el diseño de ecuaciones de control apropiadas para el robot y la evaluación del diseño y estructura del brazo. El objetivo del control de un manipulador basado en computadora es mantener la respuesta dinámica del mismo de acuerdo con algún rendimiento del sistema previamente especificado y objetivos deseados. En general, el rendimiento dinámico de un manipulador depende directamente de la eficacia de los algoritmos de control y de su modelo dinámico.

El problema de control consiste en obtener modelos dinámicos del brazo del robot físico y a continuación especificar leyes o estrategias de control correspondientes para conseguir la respuesta y rendimiento del sistema deseado.

El modelo dinámico de un robot se puede obtener a partir de leyes físicas conocidas tales como las leyes de la mecánica newtoniana y lagrangiana. Este conduce al desarrollo de las ecuaciones de movimiento dinámico para las diversas articulaciones del manipulador en términos de los parámetros geométricos e inerciales de los elementos. Métodos convencionales como las formulaciones de Lagrange - Euler (L-E) y Newton - Euler (N-E) se pueden aplicar entonces sistemáticamente para desarrollar las ecuaciones de movimiento del robot.

2.4. Representación matricial del MD.

La representación matricial del modelo dinámico se basa en la siguiente expresión

$$T = H(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) \quad (7)$$

Donde (Ver Cuadro 1):
Matriz de inercia

$$H(q) = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix} \quad (8)$$

Función	Descripción
$H(q, \dot{q})$	Matriz de inercia
$C(q, \dot{q})\dot{q}$	Matriz de coriolis y centrífugas
$G(q)$	Vector de fuerzas gravitatorias
$F(\dot{q})$	Vector de fuerzas de fricción seca y viscosa

Cuadro1. Descripción de las funciones de la representación matricial del modelo dinámico

Los elementos de la matriz de inercia son

$$\begin{aligned}
 H_{11} &= (m_2 + m_3) L_2^2 C_2^2 + 2m_3 L_2 L_3 C_2 C_{23} + m_3 L_3^2 C_{23}^2 \\
 H_{12} &= 0 \\
 H_{13} &= 0 \\
 H_{21} &= 0 \\
 H_{22} &= (m_2 + m_3) L_2^2 + m_3 L_2^2 + 2m_3 L_2 L_3 C_2 \\
 H_{23} &= m_3 L_3^2 + m_3 L_2 L_3 C_3 \\
 H_{31} &= 0 \\
 H_{32} &= m_3 L_3^2 + m_3 L_2 L_3 C_3 \\
 H_{33} &= m_3 L_3^2
 \end{aligned} \tag{9}$$

La matriz de fuerzas de coriolis y centrífugas, así como sus elementos se definen a continuación

$$C(q, \dot{q}) = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \tag{10}$$

En donde:

$$\begin{aligned}
C_{11} &= 0 \\
C_{12} &= -\{2(m_2 + m_3)L_2^2S_2C_2 + 2m_3L_2L_3(S_{23}C_2 + S_2C_{23}) + 2m_3L_3^2S_{23}C_{23}\} \dot{\theta}_1 \\
C_{13} &= \{2m_3L_2L_3s_{23}c_2 + 2m_3L_3^2s_{23}c_{23}\} \dot{\theta}_1 \\
C_{21} &= \{(m_2 + m_3)L_2^2S_2C_2 + m_3L_2L_3(S_{23}C_2 + S_2C_{23}) + m_3L_3^2S_{23}C_{23}\} \\
C_{22} &= -\{2m_3L_2L_3s_3\} \dot{\theta}_3 \\
C_{23} &= -\{m_3L_2L_3S_3\} \dot{\theta}_3 \\
C_{31} &= \{m_3L_2L_3S_{23}C_2 + m_3L_3^2S_{23}C_{23}\} \dot{\theta}_1 \\
C_{32} &= \{m_3L_2L_3S_3\} \dot{\theta}_2 \\
C_{33} &= 0
\end{aligned} \tag{11}$$

El vector de fuerzas gravitatorias se define a continuación

$$G(q) = \begin{bmatrix} G_{11} \\ G_{21} \\ G_{31} \end{bmatrix} \tag{12}$$

y sus elementos son:

$$\begin{aligned}
G_{11} &= 0 \\
G_{21} &= -m_2L_{CG2}C_2 - m_3\{L_2C_2 - L_{CG3}C_{23}\} \\
G_{31} &= -m_3L_{CG3}C_{23}
\end{aligned} \tag{13}$$

El MD descrito en la ecuación 7 ha tomado en cuenta la fricción viscosa y seca, este vector fue definido como $F(\dot{q})$ el cual es la suma del vector de fricción seca y el vector de fricción viscosa que a continuación se describe:

$$F(\dot{q}) = Fv(\dot{q}) + Fs(\dot{q}) \tag{14}$$

en donde (ver Cuadro 2):

Vector	Descripción
$Fv(\dot{q})$	Fricción viscosa
$Fs(\dot{q})$	Fricción seca

Cuadro2. Descripción de los vectores de fricción.

$Fv(\dot{q})$ se describe a continuación

$$Fv(\dot{q}) = \begin{bmatrix} Fv_1(\dot{q}) \\ Fv_2(\dot{q}) \\ Fv_3(\dot{q}) \end{bmatrix} = \begin{bmatrix} b_1\dot{\theta}_1 \\ b_2\dot{\theta}_2 \\ b_2\dot{\theta}_2 \end{bmatrix} \quad (15)$$

en donde

b_i : es el coeficiente de fricción viscosa o rozamiento dinámico.

$\dot{\theta}_i$: es la velocidad angular.

$Fs(\dot{q})$ se describe a continuación

$$Fs(\dot{q}) = \begin{bmatrix} Fs_1(\dot{q}) \\ Fs_2(\dot{q}) \\ Fs_3(\dot{q}) \end{bmatrix} = \begin{bmatrix} K_1 \operatorname{sgn}(\dot{\theta}_1) \\ K_2 \operatorname{sgn}(\dot{\theta}_2) \\ K_2 \operatorname{sgn}(\dot{\theta}_3) \end{bmatrix} = \begin{bmatrix} K_1 \tanh(\beta_1 \dot{\theta}_1) \\ K_2 \tanh(\beta_2 \dot{\theta}_2) \\ K_2 \tanh(\beta_3 \dot{\theta}_3) \end{bmatrix} \quad (16)$$

en donde

K_i : es el coeficiente de fricción seca que depende del grado de lubricación de la articulación.

$\operatorname{sgn}(\cdot)$: es la función signo

$\dot{\theta}_i$: es la velocidad angular.

3. Control Adaptable y parametrización del modelo dinamico

La técnica del Control Adaptable surge a partir de 1950 con el fin de dar solución a problemas de control en los que las características del sistema a controlar son variables o poco conocidas. Una de las definiciones que se han dado al control adaptable, según un foro realizado referente al tema, en el año de 1961: "Un sistema Adaptativo es aquel sistema físico que ha sido diseñado desde el punto de vista Adaptativo". En otras palabras es un sistema del cual se espera se "adapte" a los cambios en la dinámica del proceso a controlar, esperando con ello resultados óptimos.

El sistema de control adaptable debe proporcionar información continua sobre el estado actual del proceso. Es decir en todo momento realizar una IDENTIFICACIÓN. El sistema adaptable debe comparar el desempeño actual con el deseado y después de esto tomar la "mejor" decisión. En este caso los parámetros han experimentado un cambio, se han ADAPTADO.

Existe una diferencia muy clara entre el control clásico y el control adaptativo. El control clásico está diseñado y estructurado para sistemas lineales y con parámetros invariantes en el tiempo. En caso de que se presenten alteraciones en el funcionamiento del sistema, es posible que el control no responda de manera eficiente.

El control adaptativo presenta una alternativa diferente. Este responde a cambios inesperados en la dinámica del sistema y a perturbaciones. Es un control con parámetros variantes en el tiempo. Temas tales como sistemas lineales y no lineales, diseño de controladores, estabilidad, control estocástico, procesamiento de señales, estimación e identificación de sistemas, están íntimamente ligados con el control adaptativo. El control adaptable sirve como un regulador adaptativo diseñado para controlar sistemas en los cuales se tiene parámetros variantes con el tiempo. Esta variación de los parámetros está ligada con envejecimiento de los componentes del sistema, con alinealidades de los actuadores, así como a cambios súbitos en el proceso (perturbaciones).

Para el diseño del control, es necesario diseñar una referencia nominal q_r tal que en lazo cerrado, el robot tenga un comportamiento estable, se propone la siguiente:

$$\dot{q}_r = \dot{q}_d - \alpha \Delta q \quad (17)$$

sustituyendo Ecuación 17 en Ecuación 19 se obtiene el error extendido,

$$S = \dot{\Delta}q + \alpha \Delta q, \quad (18)$$

$$\tau = k_p e(t) + k_i \int e(t) dt + k_d \frac{de}{dt} \quad (19)$$

donde $\Delta q = q - q_d$, y q_d representa la trayectoria o coordenada de referencia.

El control adaptable τ propuesto es el siguiente

$$\tau = Y_r(q, \dot{q}, \dot{q}_r, \ddot{q}_r) \hat{\Theta} - K_d S \quad (20)$$

$$\dot{\hat{\Theta}} = -\Gamma Y_r(q, \dot{q}, \dot{q}_r, \ddot{q}_r)^T S \quad (21)$$

donde $K_d = K_d^T \in \mathbb{R}^{n \times n}$ y $\Gamma = \Gamma^T \in \mathbb{R}^{p \times p}$ son matrices diagonales definidas positivas, y $\hat{\Theta}$ representa el vector de parámetros estimados. La ecuación de error en lazo cerrado definida a partir de Ecuación 20 es

$$H(q) \dot{S} = -\{C(q, \dot{q}) + B\} S - Y_r(q, \dot{q}, \dot{q}_r, \ddot{q}_r) \Theta + Y_r(q, \dot{q}, \dot{q}_r, \ddot{q}_r) \hat{\Theta} - K_d S \quad (22)$$

simplificando,

$$H(q) \dot{S} = -\{C(q, \dot{q}) + B + K_d\} S + Y_r(q, \dot{q}, \dot{q}_r, \ddot{q}_r) \Delta\Theta \quad (23)$$

donde $\Delta\Theta = \hat{\Theta} - \Theta$.

3.1. Parametrización del Modelo Dinámico

Tomando en cuenta la fricción tenemos:

$$H(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + F(q) = T$$

$$Y(q, \dot{q}, \ddot{q})\hat{\theta} = T$$

$$H(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + F(q) = Y\hat{\theta} \quad (24)$$

Entonces tenemos el modelo dinámico con fricción a continuación:

Ecuación 1:

$$\begin{aligned}
 T_1 = & \underbrace{((m_2 + m_3)L_2^2 C_2^2}_{\theta_1} + 2 \underbrace{m_3 L_2 L_3}_{\theta_2} C_2 C_{23} + \underbrace{m_3 L_3^2 C_{23}^2}_{\theta_3}) \ddot{\theta}_1 - \underbrace{(2(m_2 + m_3)L_2^2 S_2 C_2}_{\theta_1} \\
 & + 2 \underbrace{m_3 L_2 L_3}_{\theta_2} (S_{23} C_2 + S_2 C_{23}) + 2 \underbrace{m_3 L_3^2 S_{23} C_{23}}_{\theta_3}) \dot{\theta}_1 \dot{\theta}_2 - \underbrace{(2m_3 L_2 L_3 S_{23} C_2}_{\theta_2} \\
 & + 2 \underbrace{m_3 L_3^2 S_{23} C_{23}}_{\theta_3}) \dot{\theta}_1 \dot{\theta}_3 + \underbrace{b_1 \dot{\theta}_1}_{\theta_4} + \tanh(\beta, \dot{\theta}_1)_{\theta_5}
 \end{aligned} \tag{25}$$

Ecuación 2:

$$\begin{aligned}
 T_2 = & \underbrace{((m_2 + m_3)L_2^2}_{\theta_1} + \underbrace{m_3 L_3^2}_{\theta_3} + 2 \underbrace{m_3 L_2 L_3}_{\theta_2} C_3) \ddot{\theta}_2 + \underbrace{(m_3 L_3^2}_{\theta_3} + \underbrace{m_3 L_2 L_3}_{\theta_2} C_3) \ddot{\theta}_3 \\
 & - \underbrace{(2m_3 L_2 L_3 S_3)}_{\theta_2} \dot{\theta}_2 \dot{\theta}_3 + \underbrace{((m_2 + m_3)L_2^2 S_2 C_2}_{\theta_1} + \underbrace{m_3 L_2 L_3 (C_2 S_{23} + C_{23} S_2)}_{\theta_2}) \\
 & + \underbrace{m_3 L_3^2 C_{23} S_{23}}_{\theta_3} \dot{\theta}_1^2 - \underbrace{(m_3 L_2 L_3 S_3)}_{\theta_2} \dot{\theta}_3^2 - \underbrace{(m_2 L C G_2 + m_3 L_2)}_{\theta_6} g C_2 \\
 & - \underbrace{m_3 g L C G_3 C_{23}}_{\theta_7} + \underbrace{b_2 \dot{\theta}_2}_{\theta_8} + \tanh(\beta, \dot{\theta}_2)_{\theta_9}
 \end{aligned} \tag{26}$$

Ecuación 3:

$$\begin{aligned}
 T_3 = & \underbrace{(m_3 L_3^2}_{\theta_3} + \underbrace{m_3 L_2 L_3}_{\theta_2} C_3) \ddot{\theta}_2 + \underbrace{(m_3 L_3^2}_{\theta_3} \ddot{\theta}_3 + \underbrace{(m_3 L_2 L_3}_{\theta_2} C_2 S_{23} + m_3 L_3^2 C_{23} S_{23})}_{\theta_1} \dot{\theta}_1^2 \\
 & + \underbrace{(m_3 L_2 L_3 S_3)}_{\theta_2} \dot{\theta}_2^2 - \underbrace{m_3 L C G_3 C_{23}}_{\theta_7} + \underbrace{b_3 \dot{\theta}_3}_{\theta_{10}} + \tanh(\beta, \dot{\theta}_3)
 \end{aligned} \tag{27}$$

Expresando el modelo dinámico en términos de los parámetros θ :

Ecuación 1:

$$T_1 = \theta_1(C_2^2\ddot{q}_1 - 2S_2C_2\dot{q}_1\dot{q}_2) + \theta_2(2C_2C_{23}\ddot{q}_1 - 2\dot{q}_1\dot{q}_2(S_{23}C_2 + S_2C_{23}) - 2S_{23}C_2\dot{q}_1\dot{q}_2) + \theta_3(C_{23}^2\ddot{q}_1 + 2S_{23}C_{23}\dot{q}_1\dot{q}_2 - 2S_{23}C_{23}\dot{q}_1\dot{q}_3 + \theta_4\dot{q}_1 + K_1 \tanh(\theta_5\dot{q}_1)) \quad (28)$$

Ecuación 2:

$$T_2 = \theta_1(\ddot{q}_2 + S_2C_2\dot{q}_1^2) + \theta_2(2C_3\ddot{q}_2 + C_3\ddot{q}_3 - 2S_3\dot{q}_1\dot{q}_2 + \dot{q}_1^2(S_{23}C_2 + S_2C_{23}) - S_3\dot{q}_3^2) + \theta_3(\ddot{q}_2 + \ddot{q}_3 + C_{23}S_{23}\dot{q}_1^2) + \theta_6(-gC_2 - \theta_7(-gC_{23}) + \theta_8(\dot{q}_2) + K_2 \tanh(\theta_9\dot{q}_2)) \quad (29)$$

Ecuación 3:

$$T_3 = \theta_2(C_3\ddot{q}_2 + S_{23}C_2\dot{q}_1^2 + S_3\dot{q}_1^2) + \theta_3(\ddot{q}_2 + \ddot{q}_3 + C_{23}S_{23}\dot{q}_1^2) - \theta_7(C_{23}) + \theta_{10}(\dot{q}_3) + K_3 \tanh(\theta_{11}\dot{q}_3) \quad (30)$$

Representación matricial de a cuerdo a :

$$\begin{bmatrix} T_1 \\ T_1 \\ T_1 \end{bmatrix}^{3 \times 1} = \begin{bmatrix} Y_1(q, \dot{q}, \ddot{q}) \\ Y_2(q, \dot{q}, \ddot{q}) \\ Y_3(q, \dot{q}, \ddot{q}) \end{bmatrix}^{3 \times 11} * \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \\ \theta_6 \\ \theta_7 \\ \theta_8 \\ \theta_9 \\ \theta_{10} \\ \theta_{11} \end{bmatrix}^{11 \times 1} \quad (31)$$

$$T = \begin{bmatrix} Y_{11} & Y_{12} & Y_{13} & Y_{14} & Y_{15} & Y_{16} & Y_{17} & Y_{18} & Y_{19} & Y_{110} & Y_{111} \\ Y_{21} & Y_{22} & Y_{23} & Y_{24} & Y_{25} & Y_{26} & Y_{27} & Y_{28} & Y_{29} & Y_{210} & Y_{211} \\ Y_{31} & Y_{32} & Y_{33} & Y_{34} & Y_{35} & Y_{36} & Y_{37} & Y_{38} & Y_{39} & Y_{310} & Y_{311} \end{bmatrix}$$

Donde:

$$Y_{11} = C_2^2 \ddot{q}_1 - 2S_2 C_2 \dot{q}_1 \dot{q}_2 \quad (32)$$

$$Y_{12} = 2C_2 C_{23} \ddot{q}_1 - 2\dot{q}_1 \dot{q}_2 (S_{23} C_2 + S_2 C_{23}) - 2S_{23} C_{23} \dot{q}_1 \dot{q}_3 \quad (33)$$

$$Y_{13} = C_{23}^2 \ddot{q}_1 + 2S_{23} C_{23} \dot{q}_1 \dot{q}_2 - 2S_{23} C_{23} \dot{q}_1 \dot{q}_3 \quad (34)$$

$$Y_{14} = \dot{q}_1 \quad (35)$$

$$Y_{15} = \tanh(\beta, \dot{q}_1) \quad (36)$$

$$Y_{16} = 0 \quad (37)$$

$$Y_{17} = 0 \quad (38)$$

$$Y_{18} = 0 \quad (39)$$

$$Y_{19} = 0 \quad (40)$$

$$Y_{110} = 0 \quad (41)$$

$$Y_{111} = 0 \quad (42)$$

$$Y_{21} = \ddot{q}_2 + S_2 C_2 \dot{q}_1^2 \quad (43)$$

$$Y_{22} = 2C_3\ddot{q}_2 + C_3\ddot{q}_3 - 2S_3\dot{q}_1\dot{q}_2\dot{q}_1^2(S_{23}C_2 + S_2C_{23}) - S_3\dot{q}_3^2 \quad (44)$$

$$Y_{23} = \ddot{q}_2 + \ddot{q}_3 + C_{23}S_{23}\dot{q}_1^2 \quad (45)$$

$$Y_{24} = 0 \quad (46)$$

$$Y_{25} = 0 \quad (47)$$

$$Y_{26} = -gC_2 \quad (48)$$

$$Y_{27} = -gC_{23} \quad (49)$$

$$Y_{28} = \dot{q}_2 \quad (50)$$

$$Y_{29} = \tanh(\beta, \dot{q}_2) \quad (51)$$

$$Y_{210} = 0 \quad (52)$$

$$Y_{211} = 0 \quad (53)$$

$$Y_{31} = 0 \quad (54)$$

$$Y_{32} = C_3\ddot{q}_2 + S_{23}C_2\dot{q}_1^2 + S_3\dot{q}_1^2 \quad (55)$$

$$Y_{33} = \ddot{q}_2 + \ddot{q}_3 + C_{23}S_{23}\dot{q}_1^2 \quad (56)$$

$$Y_{34} = 0 \quad (57)$$

$$Y_{35} = 0 \quad (58)$$

$$Y_{36} = 0 \quad (59)$$

$$Y_{37} = C_{23} \quad (60)$$

$$Y_{38} = 0 \quad (61)$$

$$Y_{39} = 0 \quad (62)$$

$$Y_{310} = \dot{q}_3 \quad (63)$$

$$Y_{311} = \tanh(\beta, \dot{q}_3) \quad (64)$$

4. Seguimiento exacto: Convergencia en Tiempo Finito.

Un problema típico en el control de movimiento restringido de robots manipuladores, extensivo al guiado háptico, es la conmutación de movimiento libre a movimiento restringido. Para ello es necesario detectar el momento en el que ocurre la colisión o contacto entre el efector final, y el objeto que define la superficie geométrica de restricción (superficie virtual de contacto para el caso de estudio). Para ellos es necesario el conocimiento exacto de los estados en algún instante de tiempo, por lo tanto la estabilización asintótica complica considerablemente la transición por lo complicado que es definir con precisión el instante en el que ocurre la colisión. En estas condiciones la convergencia en tiempo finito, permite el conocimiento exacto de los estados en algún instante de tiempo. Para lograr la convergencia en tiempo finito, se requiere de un generador tiempo base TBG en el contexto de movimiento libre.

La función implementada para este caso de estudio se muestra en la ecuación 65:

$$f(t) = 1/2 - 1/2 \cos (wt) \quad (65)$$

donde:

$$w = 2\pi$$

$$f = 2\pi/t$$

$$t = \text{periodo} = 2tb$$

$$w = 2\pi/2tb$$

$$w = \pi/tb \text{ rad/seg}$$

$$f(t) = 1/2 - 1/2 \cos (\pi/tbt)$$

$$f'(t) = \pi/2tb \text{sen} (\pi/tbt)$$

$$f''(t = 1/2tb) = 0$$

$$f''(t) = \pi^2/2tb^2 \cos (\pi/tbt)$$

$$f''(1/2tb) = \pi^2/2tb^2 \cos (\pi/2) = 0$$

5. Control adaptable con TBG.

La simulación digital en MatLab, consiste en desarrollar el seguimiento de una trayectoria en el espacio operacional del robot virtual con convergencia en tiempo finito, la cinemática y dinámica se expusó en anterioridad , las trayectorias deseadas de posición son las ecuaciones 66 y 67:

$$x(t) = h + r \cos (wt) \quad (66)$$

$$y(t) = k + r \text{sen} (wt) \quad (67)$$

Donde: $w = 0,4\pi$, $h = 0,3$, $k = 0,2$, $r = 0,1$

Para dicha simulación se asume que el dispositivo (robot virtual) parte de un estado de reposo con una condición inicial de posición, que esta definida en: $q_1 = 45^\circ$, $q_2 = -45^\circ$, $q_3 = -45^\circ$, las ganancias del control son: $\alpha = [8, 0, 0; 0, 0, 8, 8, 0; 0, 0, 7, 5]$, $\gamma = 1,5$, $k_d = 20$ y el tiempo de convergencia es: $t_b = 0,4\text{seg}$, con: $\alpha_0 = 1,001$, $\delta = 0,001$

En la Figura 2 se pueden ver los valores que se obtuvieron durante la simulación, de la coordenada generalizada q_1 , al mismo tiempo se muestra una comparación grafica con respecto a los valores que se desea obtener, además de la implementación del TBG al control Adaptable, a diferencia del control adaptable sin TBG el único cambio es la convergencia en un x tiempo, el cual en la simulación fue de 0.4 seg. , el error como puede apreciarse es cero.

En la Figura 3 se puede observar la representación gráfica de los valores de la coordenada generalizada q_2 y de los valores deseados q_d2 , así como la implementación del TBG al control Adaptable, a diferencia del control adaptable sin TBG cambio es la convergencia en un x tiempo, el cual en la simulación fue de 0.4 seg., se puede apreciar también que en la mayor parte del tiempo de simulación el error es cero.

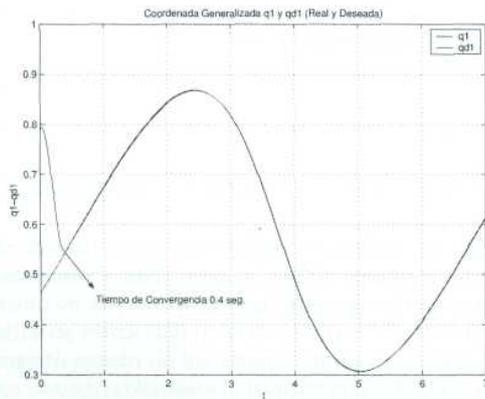


Figura2. Control adaptable con TBG, Coordenada Generalizada q_1 y q_d1 (Real y Deseada).

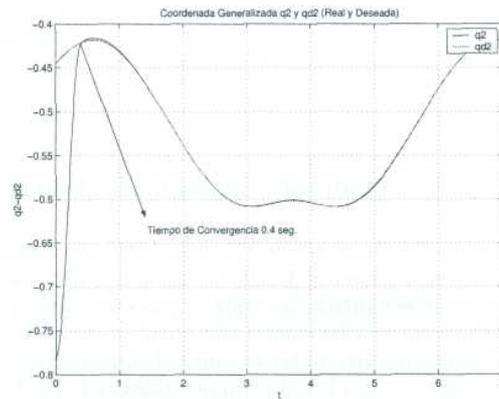


Figura3. Control adaptable con TBG, Coordenada Generalizada q_2 y q_d2 (Real y Deseada).

La implementación del TBG al control Adaptable y comportamiento de la tercera coordenada generalizada q_3 , se puede observar en la Figura 4, es muy similar a las anteriores coordenadas, en la forma en que los valores del robot virtual alcanzan a los valores deseados, el cambio observado es la convergencia en un tiempo x , el cual en la simulación fue de 0.4 seg., ya que el error es nulo en la mayor parte de la simulación.

La trayectoria realizada por el robot virtual y la trayectoria deseada se muestran en la Figura 5, se puede apreciar que ambas trayectorias son iguales en la mayor parte de la simulación, si se observa la Figura 6 se puede notar que la trayectoria del robot al tratar de alcanzar la trayectoria deseada es diferente a la 5.

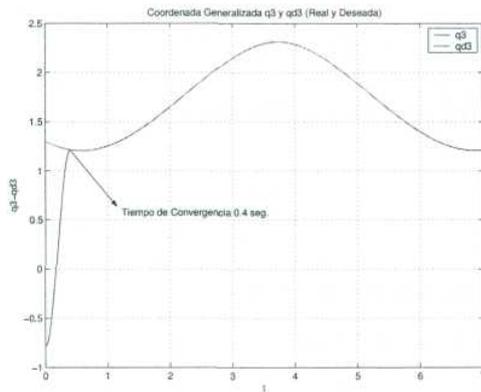


Figura4. Control adaptable con TBG, Coordenada Generalizada q_3 y q_{d3} (Real y Deseada).

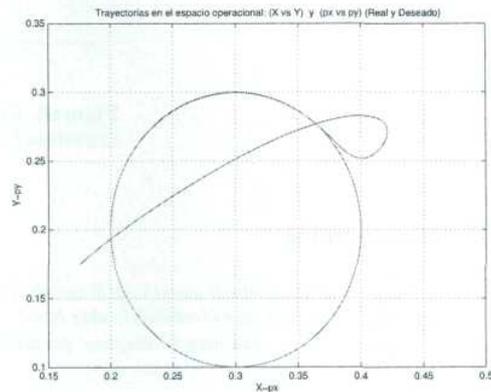


Figura5. Control adaptable con TBG, Trayectorias en el espacio operacional: (x,y) y (p_x,p_y) , (Real y Deseada).

En la Figura 6 se puede apreciar que el control adaptable con TBG elimina el error extendido al seguir la trayectoria deseada, además de lograrlo en un tiempo x , el cual en la simulación fue de 0.4 seg.

6. Conclusión

El control Adaptable pasivo Robusto a diferencia de otros controles como lo son los clásicos, elimina el error estacionario que no es deseable en ninguna situación real, además de tener la capacidad de adaptarse a eventualidades no previstas como lo son las alteraciones del sistema esto mediante la parametrización lineal del modelo dinámico que en este trabajo se realiza tomando en cuenta la fricción del sistema, la integración de un generador de tiempo base nos ayuda a que el robot converja en la posición deseada en un determinado tiempo denominado tiempo de convergencia, esto es muy importante en la industria en tareas en las que el tiempo y la precisión son puntos importantes en el desarrollo de las mismas.

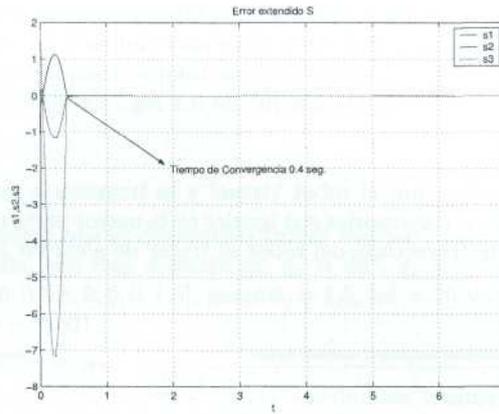


Figura6. Control adaptable con TBG, Error extendido S.

Referencias

1. <http://www.cosimir.com/VR/English/Technologie/Introduction.htm>
2. <http://disclab.ua.es/robofab/index.html>
3. <http://www-cia.mty.itesm.mx/gordillo/GV/personal.html>

Apéndice H

Apéndice: Contenido del CD.

El CD que se anexa a este trabajo contiene la siguiente información:

- Código en Matlab Control PD.
- Código en Matlab Control PID.
- Código en Matlab Control PIDNL.
- Código en Matlab Control Adaptable.
- Código en Matlab control PDNL con TBG.
- Código en Matlab Control Adaptable con TBG.
- Código en Delphi 7.0 de la Interfaz de Usuario.
- Instalador de la Aplicación de Control.

Bibliografía

- [1] Mark W. Spong, M. Vidyasagar. *“Robot Dynamics and Control”* John Wiley y Sons, U. S. A. 1989.
- [2] K. S. Fu, R. C. González, C. S. G. Lee. *“Robótica: Control, Visión e Inteligencia”*. Primera edición. McGraw Hill/Interamericana de España, S. A. 1988.
- [3] Katsuhiko Ogata. *“Ingeniería de Control Automático”*. Séptima edición. Prentice Hall Hispanoamericana, S. A. de C. V. 1996.
- [4] Benjamín C. Kuo. *“Sistemas de Control Automático”*. Séptima edición. Prentice Hall Hispanoamericana, S. A. de C. V. 1996.
- [5] Mikell P. Groover, Mitchell Weiss, Roger N. Nagel y Nicolas G. Odrey. *“Robótica Industrial: Tecnología, Programación y Aplicaciones”*. Primera edición. McGraw Hill/Interamericana de México, S. A. 1990.
- [6] Barrientos, A., PEÑIN,LF., BALAGUER C., ARACIL, R. *“Fundamentos de Robótica”*. Madrid: Mc-Graw Hill, 1997. 326p.
- [7] Cómo y cuándo aplicar un robot industrial Daniel Audí Piera Ed. Marcombo Boixareau
- [8] *“Fundamentos de Física”*. Frank J. Blatt. Ed. Prentice Hall Hispanoamericano, S.A.
- [9] ASIMOV, Isaac. *“Los robots”*: Ediciones Orbis S.A., 1986. 160p.
- [10] B. Gorla, M. Renaud. *“Modèles des Robots Manipulateurs, Application à Leur Commande”*, A and P, Francia 1984.
- [11] Schlüssel, K. (1985). *“Robotics and Artificial Intelligence Across the Atlantic and Pacific.”* IEEE Transactions on Industrial electronics. IE-30, 3, 244-251.
- [12] Brady, M.(1985). *“Artificial Intelligence and Robotics, Artificial Intelligence.”* 26, pp. 79-121.
- [13] McKerrow, P.J. (1986) *“Robotics, An Academic Dcipline?”*, Robotics, 2, 3, pp. 267-274.
- [14] John J. Craig, *“Introduction to Robotics Mechanics and Control”*, Addison Wesley, 1989, E.U.A.
- [15] Antti J. Koivo, *“Fundamentals for Control of Robotic Manipulators”*, John Wiley and Sons, 1989, E.U.A.
- [16] Ferraté, G., AMAT, J., AYSA, J., Y OTROS. *“Robótica Industrial.”* Barcelona : Marcombo, S.A., 1988. 382p.
- [17] L. Sciavicco and B. Siciliano, *“Modelling and Control of Robot Manipulators”*. Great Britain 2001. McGraw Hill.
- [18] *“Surgery: From Lister to Robotics”*, Lecture by Guillermo A Gomez, The University of Texas Medical Branch Mini-Medical School, Austin, Texas, April 6, 2004
- [19] Omar A. Domínguez Ramírez, J. Carrera, F. Garrido. *“Introducción a los Robots Manipuladores”*, CITIS, Universidad Autónoma de Hidalgo, Pachuca, Hgo. México 1999.

- [20] Pámanes García J. A. “*Modelado de Robots Manipuladores*”. 1994, México.
- [21] Timothy J. Maloney. “*Electrónica Industrial Moderna*”, Tercera Edición, Prentice may, 1997.
- [22] “*Robótica, una introducción D. McCloy*”, D. M. J. Harris Noriega editores.
- [23] Domínguez-Ramírez, O. A. and V. Parra-Vega, “*Interacción Háptica de Alto Desempeño con Objetos Virtuales Dinámicos Deformables*”, Avances en la Ciencia de la Computación en México, ISBN 970-36-0026-3, Centro de Investigación en Computación del Instituto Politécnico Nacional y ACM México. Research on Computing Science, vol 2, pp 294-310, 2003.
- [24] Omar A. Domínguez Ramírez, Herbert Lara Ordaz Ramón Soto de la Cruz, Virgilio López-Morales, “*Visualization and Control of Virtual Robot Manipulators*”, Citis, UAEH, CICINDI 2004, ISBN:970-36-0189-8.
- [25] Herbert Lara Ordaz, “*Análisis y diseño de una interfaz de visualización virtual del dispositivo háptico PHAN-ToM 1.0 basada en la dinámica de Euler-Lagrange*”, Universidad Autónoma del Estado de Hidalgo, Pachuca, Hidalgo, 2005.
- [26] Guía fácil: Robótica, J. Ma. Angulo Usategui José No Sánchez de León, Paraninfo
- [27] Domínguez-Ramírez Omar Arturo, Diseño e integración de una interfaz Háptica Activa, Centro de Investigaciones y Estudios Avanzados de IPN, División de mecatrónica, México D.F., 2004

Referencias electrónicas.

- [28] <http://www.nasa.gov>
- [29] <http://www.apuntes.robotica.com>
- [30] <http://www.itapizaco.edu.mx/paginas/robo/TTM1/>
- [31] <http://www.chi.itesm.mx/cim/robind/robotica.html>
- [32] <http://www.iso.org/iso/en/faqs/faq-standards.html>
- [33] www.cs.uiowa.edu/reu/summer95/Ovo-Dafe/ovo/ovo.html
- [34] <http://www.motoman.com>
- [35] www.superrobotica.com/S300205.htm
- [36] <http://www.vieartificielle.com/>
- [37] <http://www.stanford.edu/>
- [38] <http://www.robothalloffame.org/04inductees/shakey.html>
- [39] <http://www.roboticspot.com/spot/asifue/his2004b.html>
- [40] <http://www.cienciasmisticas.com.ar/tecnologia/automatizacion/robotica/index.html>
- [41] Intuitive Surgical Web site, Accessed April 20, 2004, <http://www.intuitivesurgical.com/products/index.html>
- [42] “*Coolest Inventions of 2003: Mine Sweeper*”, Time Online Magazine, November 17, 2003, Accessed April 20, 2004, <http://www.time.com/time/2003/inventions/involobster.html>
- [43] <http://disclab.ua.es/robofab/index.html>
- [44] <http://www.autodesk.es/adsk/servlet/index?siteID=455755id=5134721>
- [45] <http://www1.discreet.com/site/LatinAme.nsf/products/E38D3D66EB3A857A85256DD0004DD47C?opendocument>

- [46] <http://px379.narod.ru/infoftware3d.htm>
- [47] <http://agemenon.uniandes.edu.com/fde/isis380-2003-2/ejemplos/>
- [48] <http://dac.escet.urjc.es/formella/doc/ig03node31.html>
- [49] <http://www.cosimir.com/VR/English/Technologie/Introduction.htm>
- [50] <http://www-cia.mty.itesm.mx/gordillo/GV/personal.html>
- [51] <http://www.cyc.dfis.ull.es/simrob/pdf/proyecto.pdf>
- [52] <http://lim.ii.udc.es/docencia/phd-meccomp/>
- [53] <http://zip.rincondelvago.com/?00025343>
- [54] <http://espejos.unesco.org.uy/simplac2002/Ponencias/ambientes20digitales/AD003.doc>
- [55] <http://correo.puj.edu.co/bdelgado/Virtual.htm>
- [56] <http://www.access.control-systems.net/recursos/glosario/index.htm>
- [57] <http://www.fischertechnik.us/>
- [58] <http://personal5.iddeo.es/ztt/Tem/T6Matrices.htm>
- [59] <http://www.rec.ri.cmu.edu/education/webpage/torque.htm>
- [60] <http://www.used-industrial-robots.com/>
- [61] Kelly M. Rafael “Control de movimiento de Robots Manipuladores”, Notas de curso, Centro de sistemas de Manufactura, Instituto Tecnológico de Estudios Superiores de Monterrey, México.