



INSTITUTO TECNOLÓGICO DE SONORA
Educar para Trascender

“Diseño y desarrollo de un co-procesador para la operación de la convolución con Verilog en FPGA”

Tesis

que para el obtener el título de

“Ingeniero en Electrónica”

Presenta:

Roberto Emmanuel Valenzuela Armenta

Asesor:

Dr. Eduardo Romero Aguirre

No obstante, un matemático, que reforzara su profecía con fórmulas y terminologías matemáticas, podría no ser comprendido por nadie y, sin embargo, creído por todos.

Isaac Asimov, Preludio a la Fundación

AGRADECIMIENTOS

Agradezco principalmente a mi familia que por su esfuerzo estoy aquí terminando una etapa de mi vida y me dieron la oportunidad de estudiar esta carrera.

A los amigos que hice a lo largo de la carrera, que viví experiencias con ellos y nos ayudamos los unos a los otros a completar nuestros logros en los proyectos y trabajos.

RESUMEN

El siguiente proyecto se realiza con el fin de realizar un simulador utilizando herramientas de software libre para el curso tradicional de robótica a nivel licenciatura para ello se realiza una investigación de trabajos previos con el fin de conocer los distintos softwares que se han hecho anteriormente. Con la información recaudada se logra crear un diseño de un manipulador de 3 grados de libertad en V-Realm Builder y un panel de control el cual se piensa añadir al funcionamiento del manipulador para futuras interacciones.

LISTA DE FIGURAS

Figura 1.	Tipos de mecanismos	5
Figura 2.	Robot de 2GDL	6
Figura 3.	Codo hacia abajo y hacia arriba	7
Figura 4.	Diagrama de flujo	13
Figura 5.	Diagrama de flujo para la programación	18
Figura 6.	panel de control y robot en 3D.	21
Figura 7.	Logotipo del ITSON	24

LISTA DE TABLAS

Tabla I.	Parámetros del sistema Quanser.	25
----------	---	----

ÍNDICE

I. INTRODUCCIÓN	1
1.1. Antecedentes	1
1.2. Planteamiento del problema	2
1.3. Objetivos	2
1.4. Hipótesis	2
1.5. Justificación	3
1.6. Delimitaciones	3
1.7. Limitaciones	3
II. MARCO TEÓRICO	4
III. MÉTODO	12
3.1. Sujeto	12
3.2. Procedimiento	13
3.2.1. Identificación del problema e investigación	14
3.2.2. Determinación de los requerimientos y alcances del proyecto	14
3.2.3. Programación en MATLAB de algoritmos de convolución	14
3.2.4. Desarrollo de la arquitectura mediante herramientas de software	15
3.2.5. Implementación a nivel de hardware	15
3.2.6. Documentación	15
3.3. Materiales y Herramientas	16
IV. DESARROLLO	17
V. RESULTADOS	20

VI. CONCLUSIONES	22
BIBLIOGRAFÍA	23
A. Quanser IP02	24

I INTRODUCCIÓN

1.1. Antecedentes

ARREGLAR Un coprocesador es...

La convolución define en forma matemática como se combinan dos señales para formar una tercera señal. Es la técnica más importante en el procesamiento digital de señales, donde los sistemas se describen mediante una señal llamada respuesta al impulso. La convolución es importante debido a que relaciona las tres señales de interés: la señal de entrada, la señal de salida y la respuesta al impulso, y en un sistema lineal si se conoce su respuesta

al impulso, se conoce el comportamiento del mismo ante cualquier señal de entrada.

ARREGLAR Este tipo de cómputo es intensivo, es decir requiere que se realicen demasiadas operaciones matemáticas. En la actualidad, se utiliza la computación acelerada por medio del uso de GPU, que es el uso de GPU (graphics processing unit) junto a un CPU (central processing unit) para acelerar las aplicaciones de aprendizaje profundo [5].

1.2. Planteamiento del problema

ARREGLAR A medida de que la capacidad de los sistemas electrónicos de recopilar datos aumenta, la necesidad para procesar los mismos también aumenta. Los GPUs son circuitos electrónicos especializados que ayudan al cómputo intensivo, pero trabajan a nivel de software, lo cual no los hace la opción más rápida en cuestión de procesamiento de datos, además de consumir mucha potencia, lo cual resulta como una desventaja para sistemas que utilizan baterías. Existe la necesidad de encontrar alternativas a las tendencias actuales para el procesamiento de señales con mejores prestaciones de cómputo intensivo y con menor consumo de potencia.

1.3. Objetivos

ARREGLAR Desarrollar un coprocesador para la operación de la convolución. Describir una arquitectura compacta y bajo consumo de energía con el lenguaje de descripción de hardware Verilog e implementar en un FPGA, que pueda usarse como acelerador en hardware en sistemas que hagan uso intensivo de dicha operación.

1.4. Hipótesis

La arquitectura digital para la convolución implementada en el FPGA tendrá un desempeño mejor o comparable en términos de tiempos de procesamiento que un acelerador con GPU.

1.5. Justificación

A pesar de que en la actualidad existen opciones de dispositivos para acelerar el procesamiento de datos, como el GPU, este no está pensado para bajos consumos de potencia y al ser de propósito general, no está optimizado para sistemas que requieran movilidad. Por lo tanto, con una arquitectura en FPGA se logrará hacer un (co)procesador dedicado para esta función que disminuya los tiempos de procesamiento en el caso de procesar grandes volúmenes de datos. Si no se realiza esta arquitectura se estaría desaprovechando una oportunidad de analizar los datos con los que se cuentan hoy en día de manera más eficiente, además de que se estaría gastando mayor cantidad de potencia que se traduce en un gasto económico.

1.6. Delimitaciones

El diseño del procesador se limitará a implementar la arquitectura para un (co) procesador para la convolución. Se implementará en el FPGA y verificará a través de un entorno de simulación con la ayuda de Matlab.

1.7. Limitaciones

Las principales limitaciones del proyecto son: Tiempo disponible Dinero Recursos Licencias

II MARCO TEÓRICO

En la robótica un manipulador es una herramienta que ayuda a controlar objetos sin necesidad de hacer contacto directo [?]; consisten en una secuencia de cuerpos rígidos llamados eslabones, se conectan unos con otros mediante articulaciones formando una cadena cinemática. Se dice que una cadena cinemática es abierta si, numerando de manera secuencial los eslabones desde el primero, cada eslabón está conectado mediante articulaciones exclusivamente al eslabón anterior, y al siguiente, excepto el primero, que se suele fijar al suelo, y el último, uno de cuyos extremos queda libre.

Los movimientos que pueden realizar cada articulación de manera independiente se le denomina grado de libertad (GDL).

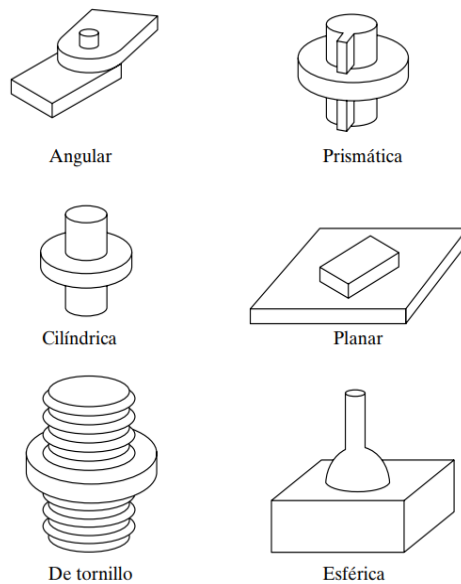


Figura 1: Tipos de mecanismos

- La articulación angular permite que dos eslabones giren, uno respecto del otro, alrededor del eje de articulación; tiene 1 GDL.
- La articulación prismática permite que dos eslabones arreglados en pares se deslicen, uno respecto al otro a lo largo de su eje; tiene 1 GDL.
- La articulación cilíndrica permite la rotación alrededor del eje de la articulación y el traslado independiente a lo largo de ella; tiene 2 GDL.
- La articulación plana permite dos traslados a lo largo de dos ejes independientes del plano de contacto y una rotación alrededor del eje perpendicular del plano.
- La articulación de tornillo permite que dos eslabones unidos giren alrededor del eje de la articulación y se trasladen, al mismo tiempo, a lo largo de él; consta de 1 GDL.
- La articulación esférica permite que uno de los eslabones pareados gire libremente en todas las orientaciones posibles respecto al otro alrededor del centro de una esfera; tiene 3 GDL.

Grado de libertad

Formalmente un grado de libertad de un sistema mecánico se define como el número de coordenadas mínimas necesarias para describir perfectamente su posición o configuración. Así, un cuerpo rígido que se mueve en el espacio cartesiano tridimensional tiene seis GDL, tres para posición y tres para orientación.

Cinemática de los robots

La cinemática es un derivado de la física que estudia el movimiento sin considerar las fuerzas o pares que lo causan, es decir, estudia las leyes de movimiento sin tener en cuenta aspectos como masas e inercias. Estudia la trayectoria que tiene el robot a lo largo del tiempo, considerando la posición, velocidad y en ocasiones la aceleración. En otras palabras se puede decir que describe de manera analítica el movimiento espacial del robot como una función del tiempo.

Cinemática Directa

Se le conoce como a los modelos matemáticos que permiten calcular la posición de los eslabones de un robot a partir de sus componentes fijos y configuraciones de las articulaciones. Se refiere al uso de ecuaciones para el cálculo de la posición del actuador final en un robot articulado a partir de los ángulos y/o desplazamientos de las articulaciones o de la posición y orientación de las bases de un robot móvil a partir de las velocidades de las ruedas.

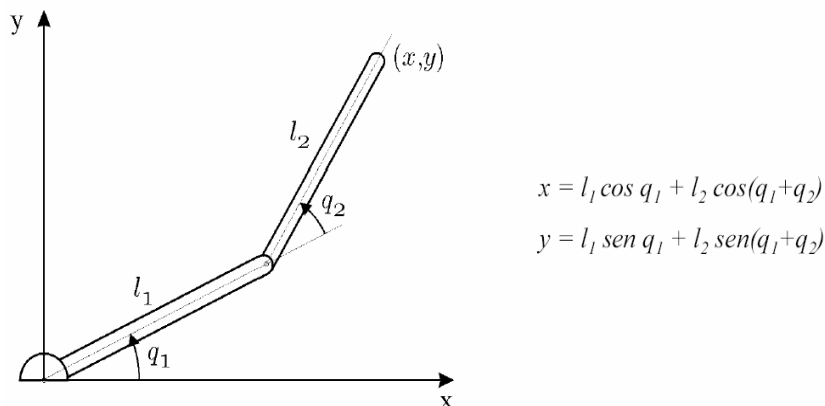


Figura 2: Robot de 2GDL

Cinemática Inversa

Es el proceso inverso por el cual se obtienen modelos matemáticos que permiten, a partir de una posición y/o desplazamientos de los actuadores. Por lo general podemos encontrar configuraciones que no son factibles, es decir, que no son alcanzables, soluciones múltiples, ya sea que el robot ejerza un movimiento con codo hacia arriba o con codo hacia abajo.

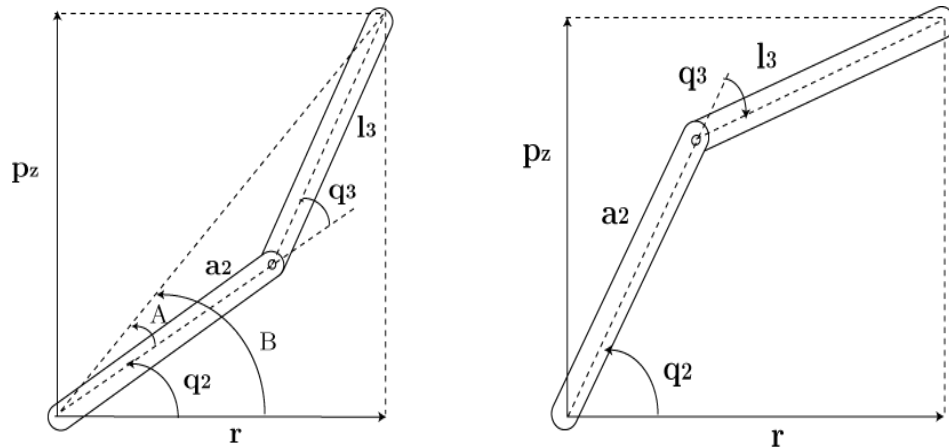


Figura 3: Codo hacia abajo y hacia arriba

Los problemas de la cinemática inversa suelen usar métodos aproximados para el cálculo de las variables de articulación, por ejemplo, a partir de la pseudo-inversa de la matriz Jacobiano. Para el caso de un robot articulado con 6GDL, la solución exacta pasa por desacoplar los tres primeros ejes de los tres últimos. En este sentido, la posición del efector final determina los valores de los tres primeros ejes, mientras que la orientación del efector final determina los valores de las tres últimas articulaciones.

Metodología de Denavit-Hartenberg

La metodología de Denavit-Hartenberg (D-H) permite establecer la ubicación de los sistemas de referencia de los eslabones en los sistemas robóticos articulados, ya sean prismáticos o de revolución, con cadenas cinemáticas abiertas. Jacques Denavit y Richard Hartenberg introdujeron esta convención en 1955 con el propósito de estandarizar la ubicación de los sistemas de referencia de los eslabones de un robot.

Parámetros D-H

Se trata de una metodología ampliamente utilizada en el ámbito académico y de investigación en robótica que permite definir las transformaciones relativas entre eslabones con tan solo cuatro parámetros siendo éste el número mínimo de parámetros para configuraciones genéricas.

La metodología de D-H define cuatro transformaciones que se aplican de forma consecutiva.

- d : Desplazamiento a lo largo de z anterior a la normal común.
- θ : Ángulo sobre la z anterior, de la x anterior a la nueva x .
- l : Longitud de la normal común. Suponiendo una articulación revoluta, este es el radio de z anterior.
- α : Ángulo sobre la normal común, del antiguo eje z al nuevo eje z .

El contenido anterior fue obtenido del libro de *Introducción a la robótica* [?] de Subir Kumar Saha y de *Fundamentos de la robótica* [?] de A. Barrientos, L. Peñin, C. Balaguer, R. Aracil.

SolidWorks

SolidWorks es un programa de diseño que consiste en la especificación de puntos, líneas, curvas o superficies a través de un conjunto de parámetros iniciales y de las relaciones existentes entre ellos. Para el diseño, el modelado paramétrico es un recurso muy importante ya que proporciona ventajas como reducir el tiempo y esfuerzo que se requieren al modificar y realizar variaciones en éste y elimina tareas repetitivas al realizar variantes en el diseño.

En el programa se crean tres tipos de archivos: piezas, ensambles y dibujo. Además estos archivos están relacionados entre sí, es decir, cuando se realiza la modificación del dibujo, ésta afecta automáticamente a todos los archivos donde se encuentra referenciada la pieza realizada. [?]

MatLab

MATLAB es una de las muchas sofisticadas herramientas de computación disponibles en el comercio para resolver problemas de matemáticas, tales como Maple, Mathematica y MathCad. Cada una permitirá efectuar cálculos matemáticos básicos, pero difieren en el modo como manejan los cálculos simbólicos y procesos matemáticos más complicados, como la manipulación de matrices. Por ejemplo, MATLAB es superior en los cálculos que involucren matrices, mientras que Maple lo supera en los cálculos simbólicos. El nombre mismo de MATLAB es una abreviatura de Matrix Laboratory, laboratorio matricial. En un nivel fundamental, se puede pensar que estos programas son sofisticadas calculadoras con base en una computadora. Son capaces de realizar las mismas funciones que una calculadora científica, y muchas más.

En muchas clases de ingeniería, la realización de cálculos con un programa de computación matemático como MATLAB sustituye la programación de computadoras más tradicional. MATLAB se han convertido en una herramienta estándar para ingenieros y científicos.

[?]

Simulink

Simulink® 3D Animation™ proporciona aplicaciones para vincular modelos Simulink y algoritmos MATLAB® a objetos gráficos 3D. Los objetos se pueden representar en los lenguajes de modelado 3D estándar X3D y VRML97. Puede animar un mundo tridimensional cambiando la posición, la rotación, la escala y otras propiedades del objeto durante el escritorio o la simulación en tiempo real. También puede detectar colisiones y otros eventos en el mundo virtual y alimentarlos de nuevo en sus algoritmos de MATLAB y Simulink. El video de las cámaras virtuales se puede transmitir a Simulink para su procesamiento.

Simulink 3D Animation incluye editores y visualizadores para renderizar e interactuar con escenas virtuales. Con 3D World Editor, puede importar formatos de archivos CAD y URDF, así como crear escenas detalladas ensambladas a partir de objetos 3D. El mundo 3D se puede ver de forma inmersiva con la visión estereoscópica. Puede incorporar múltiples vistas de escena en 3D dentro de las figuras de MATLAB e interactuar con el mundo virtual usando un joystick de retroalimentación forzada, un mouse espacial u otro dispositivo de hardware. [?]

Java 3D

Java 3D es una interfaz de programación de aplicaciones (API) desarrollada en Sun Microsystems para renderizar gráficos 3D interactivos utilizando el lenguaje de programación Java. Java 3D es una API de Java del lado del cliente. Otros ejemplos de API del lado del cliente de Sun incluyen Abstract Windows Toolkit (AWT) y Java Foundation Classes (JFC / Swing), que son ambas bibliotecas de clases Java para crear aplicaciones con una Interfaz gráfica de usuario (GUI). Las API de Java del lado del cliente contrastan con las API del lado del servidor de Sun tales como Enterprise Java-Beans (EJB) y los otros componentes de Java 2 Enterprise Edition (J2EE).

La principal ventaja de los desarrolladores de Java 3D para Java es que les permite programar en Java al 100 por ciento. En cualquier aplicación 3D grande, el código de representación solo constituirá una fracción de la aplicación total. Por lo tanto, es muy atractivo tener código de aplicación, persistencia y código de interfaz de usuario (UI) en un lenguaje fácilmente portátil, como Java. [?]

Megan Flanders [?] menciona que la realidad virtual está jugando un papel muy importante como herramienta educativa que hace mas tangible los conceptos para los estudiantes; concuerdo totalmente ya que una vez vistos los conceptos es mas sencillo aplicarlos en un programa y observar como se comportan de manera visual, además que se aprende a utilizar un software, se fortalece la teoría aplicándola de manera práctica en un programa.

En su artículo se observa que el menú es amigable con el usuario, fácil de interpretar y modificar las opciones para poder visualizar los resultados finales ya dentro del mismo simulador. Es la idea tomar en cuenta que como va orientado para alumnos, ellos tienen que comprender de la mejor manera los resultados arrojados por el mismo programa para que capten la teoría utilizando el software.

Se muestran resultados por parte de los alumnos ya que se realizaron encuestas los cuales mayormente el 55 % dicen que la herramienta sería una gran parte complementaria al curso, además el 61 % comenta que es fácil de usar el 61 % menciona que es mas fácil entender de manera interactiva con modelos en 3D que con dibujos en 2D, además se muestra que el 74 % comenta que ha incrementado ligeramente su conocimiento en estos temas.

Build-a-robot fue creado con MatLab y sus herramientas de VRLM para poder tener los modelos 3D. MatLab contiene muchas herramientas que facilitan a los ingenieros el poder simular, crear y desarrollar ideas, ya que cuenta con herramientas dentro del mismo programa que son complementarias y como corren dentro del mismo es mas sencillo tener acceso y combinarlas para poder crear un buen proyecto.

En el artículo de P. Abreu [?] habla sobre un laboratorio virtual a diferencia de lo que se quiere realizar este va enfocado a la industria, se tiene el mismo objetivo que es crear un simulador solo que este toma en cuenta además del robot maquinaria con la cual se desea trabajar, se simula el ambiente en el cual el robot estará operando y se puede llegar a una conclusión de si será optimo usar al robot para dicha etapa de producción o a lo que se requiera llevar a cabo. RobotStudio tiene un menú algo mas complicado en el aspecto que a comparación del de Megan y lo que se desea hacer este tiene un teach pendant y un distinto ambiente virtual, está lleno de símbolos, muestra la pantalla del teach pendants y los diferentes menús para poder utilizar el simulador, toman diferentes caminos, como se ha mencionado está hecho para un ambiente más robusto que enfoca otras características.

III MÉTODO

En este capítulo se abordará la ruta metodológica que se llevó a cabo para realizar el proyecto. Primero, se indicará el sujeto de estudio. Después, se mostrará un diagrama de flujo y la explicación de cada etapa del proyecto. Por último, se enlistan los materiales y herramientas utilizados en el trabajo.

3.1. Sujeto

- Investigadores e ingenieros del área de desarrollo de hardware.

3.2. Procedimiento

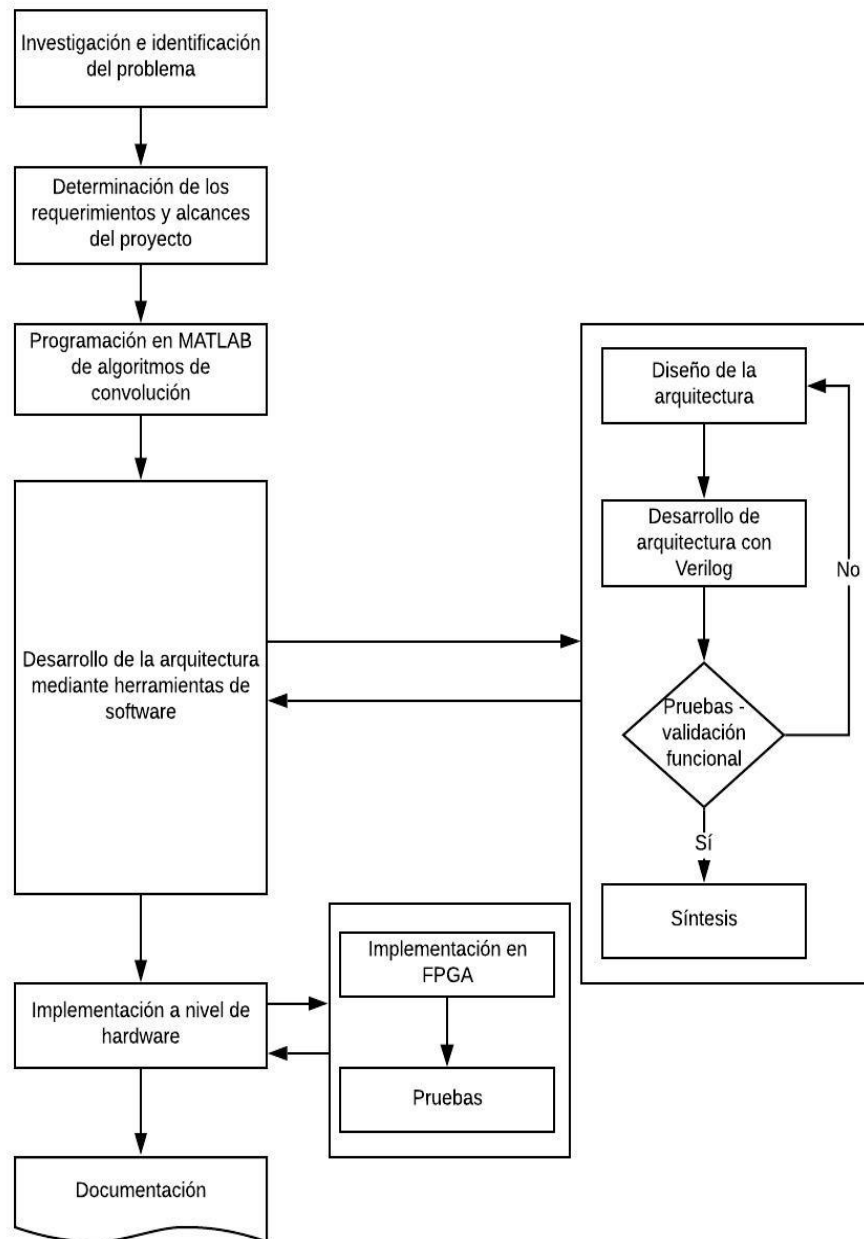


Figura 4: Diagrama de flujo

En la figura 4 se muestra la ruta metodológica a seguir para desarrollar el co-procesador para la convolución. A continuación se describe cada uno de los pasos.

3.2.1. Identificación del problema e investigación

Se determina la problemática, en base al análisis del entorno actual de la computación, al cual se le buscará solución con el proyecto que se presenta en este trabajo. El problema que se aborda es la necesidad de encontrar alternativas a las tendencias actuales para el procesamiento de señales con mejores prestaciones de cómputo intensivo y con menor consumo de potencia, específicamente para la operación de la convolución.

Además, se recopila información acerca de procesadores para la convolución en bases de datos de artículos científicos y en libros técnicos. Los cuales son trabajos similares al que se está proponiendo en este documento para poder comparar el rendimiento de la arquitectura propuesta en este trabajo y se analiza qué se puede mejorar con respecto a estos trabajos existentes en la actualidad. Esta etapa consiste en la investigación del estado del arte, es decir, cuál es la situación actual de esta tecnología.

3.2.2. Determinación de los requerimientos y alcances del proyecto

Se determina cuáles son los alcances y delimitaciones del proyecto, es decir hasta dónde se va a investigar y desarrollar. El diseño del procesador se limitará a diseñar la arquitectura para un coprocesador para la convolución. Se implementará en un FPGA y se verificará a través de un entorno de simulación con la ayuda de MATLAB. En este proyecto las limitaciones son el tiempo disponible, dinero, recursos y licencias.

3.2.3. Programación en MATLAB de algoritmos de convolución

Se desarrollan diferentes algoritmos existentes para resolver la convolución utilizando el lenguaje de programación MATLAB. La finalidad de esta etapa es comparar el rendimiento en tiempo de ejecución y uso de recursos de memoria de los diferentes algoritmos y analizar las ventajas y desventajas de cada uno de ellos.

3.2.4. Desarrollo de la arquitectura mediante herramientas de software

Se desarrolla la arquitectura utilizando metodologías de diseño y herramientas de software que facilitan el desarrollo de hardware.

Diseño de la arquitectura

Desarrollo de arquitectura con Verilog

Pruebas - validación funcional

Síntesis

3.2.5. Implementación a nivel de hardware

Implementación en FPGA

Una vez que se cuenta con la arquitectura sintetizada, se programa el archivo .bit en el FPGA utilizando el software Digilent Adept.

Pruebas

Se realizan pruebas al hardware mediante una interfaz de comunicación serial la cual conecta al FPGA con una PC. Se alimenta al FPGA con dos señales, las cuales se convolucionan en la arquitectura digital, y el FPGA da como resultado una tercera señal. Este resultado se compara con resultados obtenidos bajo un ambiente conocido en MATLAB y sirve para comprobar que el sistema digital está funcionando de manera correcta. Se realizan pruebas de velocidad de procesamiento y consumo de potencia.

3.2.6. Documentación

Se realiza un trabajo escrito en el cual se documentan los aspectos más relevantes del proyecto. Primero, se da una introducción al proyecto y se pone en contexto al lector, después se presenta el trabajo realizado así como se habla a detalle la forma en que se realizó cada etapa, por último, se presentan los resultados obtenidos así como un análisis de los mismos.

3.3. Materiales y Herramientas

- Laptop Dell Inspiron 13-5378
- MathWorks MATLAB R2015a
- Xilinx ISE 14.7
- Digilent Adept
- agregar los que faltan (FPGA, OS(?))

IV DESARROLLO

En este capítulo se describe el desarrollo del diseño para la implementación del laboratorio virtual. Se muestra el diagrama de flujo a seguir para llevar a cabo la programación y como es que va a funcionar. Finalmente se presenta la etapa que se esta siguiendo, con un robot de 3GDL el cual es realizar un simulador básico para poder comprender las características que se debe de tener.

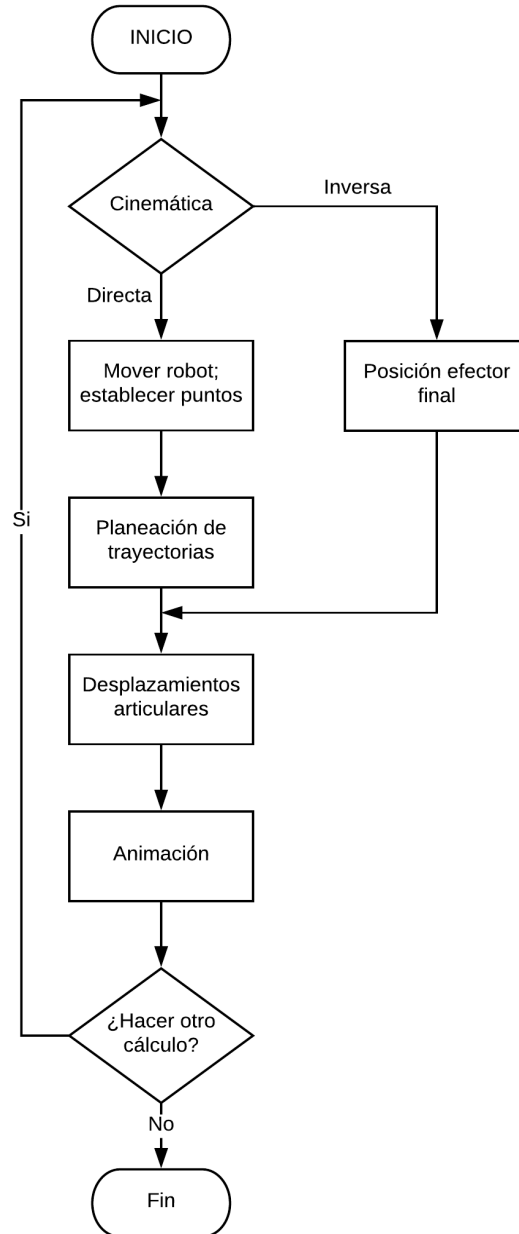


Figura 5: Diagrama de flujo para la programación

En la figura 5 se quiere iniciar con un menú el cual despliegue dos simples opciones, una la cual va a llevar al cálculo de la cinemática directa y otro de la cinemática inversa, según la opción que se seleccione en este caso si fue la directa se mostrará un menú el cual permita posicionar las articulaciones de manera libre, a continuación realizará una planeación de trayectorias para ejecutar los cálculos necesarios y que pueda hacer los desplazamientos

articulares mostrándolos mediante la animación; mostrará un menú el cual permita seleccionar si se desea finalizar con el programa o desea continuar con otro cálculo; se podrá continuar con la cinemática inversa el cual consta de posicionar el efecto final, el programa ejecutará los cálculos para poder hacer los desplazamientos articulares mostrándolos como se menciona anteriormente con su respectiva animación; de igual forma muestra el menú para continuar con otro calculo, de no ser así se finaliza el programa.

La etapa en la que se encuentra el proyecto es en base a un programa con el cual podemos crear un ambiente virtual que el mismo Matlab trae integrado es ir experimentando como posicionar piezas en un espacio, ensamblarlas, etc. Para que se logre crear un menú de igual forma con un comando el cual nos abrirá una ventana que se tiene que posicionar botones, sliders, etc; se necesita crear un link entre el panel creado y el robot en 3D es la etapa en la cual está el proyecto para poder finalmente interactuar con él.

Como se mencionó anteriormente dentro de Matlab se encuentra una herramienta llamada V-Realm Builder la cual permite crear desde figuras básicas o abrir diseños creados en CAD, en este trabajo se obtuvo un brazo robótico en SolidWorks, guardando la pieza en un formato de archivo .wrl se puede abrir en el programa dicho anteriormente, al abrirlos se deberán colocar en un punto del espacio mediante coordenadas hasta terminar formando la figura.

En la ventana de comandos de MatLab mandando llamar "GUIDE" abrirá otro menú el cual permite crear un panel con botones igual se puede personalizar con colores y las posiciones de los sliders, títulos, botones, etc.

Al finalizar el archivo .gui el cual se genera de la extensión "GUIDE" crea un código el cual se deberá modificar en cuanto a las características del archivo .WRL que tenemos del robot.

V RESULTADOS

En este capítulo se muestra hasta donde se logra llegar, cual es el estado actual del laboratorio virtual. Lo que se muestra a continuación son resultados parciales del proyecto.

Se logra llegar a lo que viene siendo un panel de control el cual controlará un manipulador de 3 GDL, también se observa el modelo en 3D hecho en V-Realm Builder diseñado con piezas sencillas del mismo programa el cual se menciona en el capítulo anterior que las piezas se deben posicionar en un espacio mediante coordenadas para poder alinear las figuras entre si y formar el brazo en este caso.

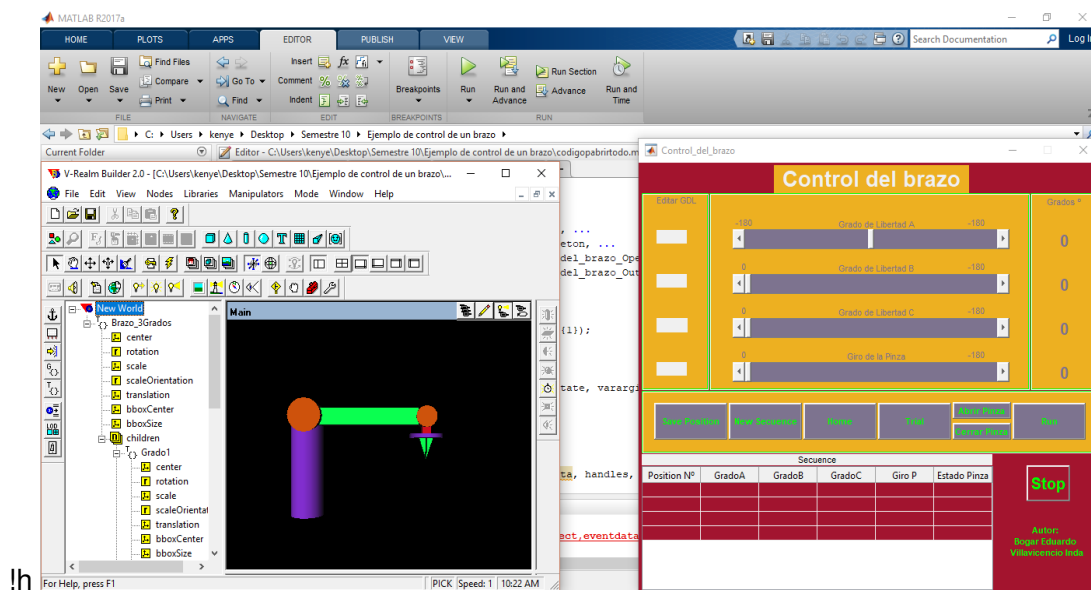


Figura 6: panel de control y robot en 3D.

Se muestra en la Figura 6 del lado izquierdo el manipulador en el programa V-Realm Builder, del lado derecho está el panel de control que muestra los botones y los sliders. El primer slider sirve para mover la parte morada del manipulador, tenemos que realiza giros de -180 a 180, la posición en la que se encuentra es la 0, el segundo slider mueve la parte verde de 0 a 180. el tercer slider mueve lo que viene siendo la posición de la muñeca de igual forma de 0 a 180, por último tenemos la opción del giro de la pinza que viene siendo la herramienta de 0 a 180.

Falta crear un enlace mediante código para poder hacer uso del panel con el robot y que se logren captar los movimientos en tiempo real. Se menciona en el capítulo anterior que el archivo GUIDE crea un scrip, dentro de éste es donde se modifican las funciones creadas para poder enlazar el panel y hacer utilidad del mismo para que el usuario interactúe finalmente con el manipulador y sea un paso mas a lo que se quiere llegar, que finalmente son cálculos de cinemáticas.

VI CONCLUSIONES

En este trabajo de tesis no se logra terminar el simulador, pero se aprovecha el potencial que Matlab tiene, ya que no se había manejado anteriormente el programa para crear diseños virtuales. Como no se había utilizado no se conocían las funciones del todo y poco a poco se fueron viendo hasta lograr lo que se ve en los resultados, de igual forma falta integrar la programación para enlazar el panel con el manipulador y poder finalmente tener un simulador, para así poder compararlo con lo que se quiere llegar, que es similar pero en Java3D. Teniendo los cálculos en Matlab será mas fácil hacer la programación, estudiando la API de Java3D para ver el código de importación y tener control sobre las piezas diseñadas anteriormente en SolidWorks.

Este tipo de herramientas crean un impacto positivo ya que en los trabajos vistos dentro de la investigación son favorables y se espera llegar o superar esos resultados.

Lo interesante de este proyecto es que una vez terminado se le puede dar seguimiento ya que es código abierto, en caso de querer añadir mas funciones será posible.

A Quanser IP02



INSTITUTO TECNOLÓGICO DE SONORA
Educar para Trascender

Figura 7: Logotipo del ITSON

Constante	Descripción	Valor	Unidades
M	Masa del carro	0.52	kg
B	Amortiguamiento viscoso	0.9	N·s/m
K	Rígidez del resorte	160	N/m
η_g	Eficiencia de engranes	1	
η_m	Eficiencia del motor	1	
K_g	Relación de transmisión	3.71	
K_t	Constante de par	0.00767	N·m/A
K_m	Constante de fuerza magnetomotriz	0.00767	V·s/rad
R_m	Resistencia de armadura	2.6	Ω
r_p	Radio pión	6.35×10^{-3}	m

Tabla I: Parámetros del sistema Quanser.