

96



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES.

CAMPUS ARAGÓN

290238

**DISEÑO DE UN CONTROL DIFUSO APLICADO AL MANEJO AUTOMÁTICO DE
ESTACIONAMIENTO DE UN CAMIÓN CON REMOLQUE IMPLEMENTADO CON
EL MICROCONTROLADOR 68HC11**

T E S I S
QUE PARA OBTENER EL TÍTULO DE
INGENIERO MECÁNICO ELÉCTRICO
P R E S E N T A:
VICTOR MANUEL AGUSTÍN TORRES OLGUÍN

ASESOR: M. EN I. DAVID GONZÁLEZ MAXINEZ

MÉXICO

2001.



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

	Contenido
Intruducción.....	3
1 Capítulo 1. Descripción global del problema	5
1.1. Origen de la lógica difusa y sus aplicaciones.....	6
1.2. Planteamiento del problema.....	10
1.3. Modelado.....	11
1.4. Planteamiento difuso.....	14
1.4.1. Fuzzyficación.....	18
1.4.2. Evaluación de reglas.....	19
1.4.2.1. Definición de reglas.....	23
1.4.3. Defuzzyficación.....	25
1.4.3.1. El método del centro de gravedad.....	25
1.4.3.2. Método de las cartas únicas.....	27
Capítulo 2. Simulación en software del problema propuesto	29
2.1. Descripción del software de diseño.....	30
2.2. Inferencia difusa.....	30
2.3. Programa de simulación del prototipo.....	36
Capítulo 3. Algoritmo de diseño con el microcontrolador	41
3.1. Características generales del microcontrolador 68HC11.....	42
3.2. Diagrama de flujo del problema.....	45
3.2.1. Lectura de datos de sensores y fuzzyficación.....	46
3.2.2. inferencia difusa.....	49
3.2.3. Defuzzyficación.....	51
3.2.3.1. Salida hacia el motor de pasos.....	52
Capítulo 4. Construcción del prototipo y programación del sistema	53
4.1. Descripción física del camión.....	54
4.2. Características de los circuitos.....	57
4.3. Los sensores.....	60
4.4. Circuito del Microcontrolador.....	61
4.4.1. Diagrama electrónico de la tarjeta MICRO 6811, conectores y puentes.....	62
4.5. Programa para el microcontrolador.....	64
Conclusión	77
Apéndice A: Programa de monitoreo (buffalo)	79
Apéndice B: Programa IASMI1	80
Apéndice C: Detalles de circuito del Microcontrolador	82
Bibliografía	85

La lógica difusa fue propuesta por el matemático, Lotfi Zadeh, a mediados de los años 60's. En ella no solo existe lo verdadero o lo falso, sino que se pueden presentar las dos condiciones al mismo tiempo, pero de forma gradual. Esto permite, una visión más clara y real del mundo. Es por ello que las aplicaciones de ésta son muy amplias.

Originalmente esta lógica fué pensada para el análisis en la psicología, la sociología y otras similares. Estas materias tienen en común, que suelen ser ambiguas si se comparan con las matemáticas, en el sentido, de que dependen de la apreciación y de la intuición del estudioso del tema para su evaluación. No es posible por ello establecer modelos matemáticos de los problemas que enfrentan estas ciencias, o los modelos pueden ser sumamente complicados.

Pero la utilidad práctica de esta teoría rebasó esta idea original, al demostrarse en 1973 que con la lógica difusa se pueden fabricar máquinas inteligentes, abriéndose entonces a las materias técnicas.

Una de las características importantes de esta lógica, es que permite el manejo de variables lingüísticas que suelen ser complicadas para la lógica booleana, como: muy, poco, alto, bajo, etc. permitiendo que la solución del problema se presente con tan solo proponer una serie de reglas que suelen ser deducidas a través de la experiencia de los operadores del sistema a controlar o a través de la observación del comportamiento del sistema.

Básicamente la lógica difusa está constituida por tres partes, y son:

- ⇒ La fuzzyficación, que transforma las entradas naturales en estradas fuzzy.
- ⇒ Evaluación de reglas, evalúa las reglas que pueden resolver el problema.
- ⇒ Y la defuzzyficación convierte las salidas fuzzy en salidas naturales aplicables al sistema diseñado.

En este trabajo se construye un sistema que pueda ser realmente aplicado a un camión con remolque, el cual tendrá la capacidad para estacionarse y ubicarlo en el punto donde pueda ser cargado o descargado con facilidad, sin necesidad de ser controlado por algún operador, independientemente del lugar donde se haya dejado. Para ello se cuenta con una tarjeta de desarrollo del microcontrolador 68HC11. El camión, para este trabajo, es un modelo a escala, ya que no se cuenta con uno real esto convierte a este trabajo en una tesis no teórica, sino práctica.

En el primer capítulo se esboza la historia y orígenes de la lógica difusa, además de estudiarse los conceptos y definiciones necesarias para lograr la construcción del sistema de control ya que debe de cumplir con las condiciones de trabajo de la lógica difusa.

Al mismo tiempo que se define el problema se va planteando una solución difusa teórica, además se dan las bases para la construcción de un modelo de computadora.

En el segundo capítulo se hizo un programa con el fin de probar la solución propuesta en el primer capítulo, la simulación permite hacer correcciones en las soluciones propuestas, con el fin de que en los siguientes capítulos pueda hacerse el salto al sistema real.

El capítulo tres estudia las características del microcontrolador, las formas de comunicación de este con la computadora, y se desarrolla el algoritmo de la inferencia difusa, los cálculos de mínimos y máximos y la defuzzyficación, y la salida del sistema al motor de control de la dirección.

En el capítulo cuarto, se explica la construcción del camión a escala y se analizan los circuitos del microcontrolador, los circuitos de salida, entrada y el programa para el microcontrolador 68HC11.

Se agregaron tres apéndices; el primero con las características del programa de control integrado al microcontrolador (buffalo), el segundo explica el programa de comunicación de la tarjeta con la pc y el tercero da detalles de la tarjeta de desarrollo para el microcontrolador 68hc11.

Capítulo 1. Descripción global del problema

1.1. Origen de la lógica difusa y sus aplicaciones

En este trabajo de investigación, se diseña el programa para el microcontrolador que manipula un camión con remolque dentro del estacionamiento de un muelle de carga a través del concepto de lógica difusa. Al utilizar este concepto se pretende que el modelado del problema sea lo más sencillo posible con respecto a formas tradicionales de control. Nuestro sistema de control deberá de ser capaz de estacionar al vehículo sin la necesidad de que algún operador humano intervenga en esta tarea. Este aspecto es posible realizarlo si se puede diseñar un sistema que refleje lo que la gente piensa, es decir un sistema que pueda modelar nuestra toma de decisiones, así como nuestro reconocimiento de vista y sonido, logrando con esto el diseño de máquinas más humanas. En nuestra investigación esto se puede realizar utilizando el concepto de lógica difusa.

La lógica difusa fue propuesta por el matemático de origen iraní, de la universidad de California en Berkeley, el profesor Lotfi Zadeh, a mediados de los años 60's; el profesor reconoce la infinita gama de valores que existen entre lo verdadero y lo falso a diferencia de la lógica Booleana, es decir en lugar de que un elemento sea 100% verdadero o falso, este puede ser parcialmente verdadero o parcialmente falso al mismo tiempo.

En 1973 Zadeh demuestra que con la lógica difusa, se pueden fabricar máquinas inteligentes y en 1974 E. H. Mandani profesor del colegio Queen Mary en la Universidad de Londres, aplica por primera vez, la teoría en un sistema de control de vapor. El profesor Zadeh ha podido demostrar junto con Bart Kosko que la lógica de Boole es un caso particular de la lógica difusa.

La lógica difusa es una extensión de multiniveles lógicos, en donde todo es una cuestión de grado. No solo contiene los valores clásicos (0 y 1), sino que hace uso de la teoría y probabilidad lógica sobre todo al momento de tomar decisiones. Estos fundamentos hacen posible establecer las siguientes características importantes que distingue a la lógica difusa de los sistemas de lógica tradicional, en la lógica difusa:

- El razonamiento exacto es visto como un límite del razonamiento aproximado.
- Completamente todo es visto como un grado de verdad.
- Los conocimientos son interpretados como una colección de límites elásticos en una colección de variables.
- En un sistema difuso, una proposición puede ser verdadera o falsa o puede tener un valor intermedio.
- Proporciona las herramientas necesarias para el tratamiento de la inexactitud, un concepto asociado al conocimiento humano.
- Permite manejar conceptos vagos como "alto", "sucio" o "edad".

La lógica difusa es muy poderosa y pueden aplicarse en:

-
- Sistemas donde una o más de las variables sean continuas.
 - Cuando se cuenta con la experiencia suficiente para poder controlar el sistema, aun cuando no se conozcan la interpretación matemática de éste, ni su variable de estado, ni su función de transferencia.
 - En algunos casos cuando el sistema no es lineal, pero es posible controlarlo mediante la experiencia humana.
 - En sistemas donde la vaguedad es común.
 - En sistemas cuando se conocen todas las posibles variables que pueden intervenir en la salida del sistema, ya que si alguna de estas se desconoce y no es tomada en cuenta, además de influir en gran parte en la salida, el control no sabrá que hacer.

Los ámbitos de aplicación del concepto son muy variados; podemos encontrar aspiradoras que ajustan su potencia de succión en función de la cantidad y consistencia del polvo que haya detectado mediante sensores; también hay lavadoras inteligentes que seleccionan el programa de lavado más adecuado, comprobando previamente la suciedad de la ropa, mediante un remojo previo y analizando el agua resultante; el metro de la ciudad de Sendai, en Japón, está dotado de un sistema informático difuso para evitar los arranques y frenadas bruscas, existe también un cambio automático que anula los típicos tirones al cambiar de marcha reduciendo el consumo de combustible, otra de las aplicaciones más importantes está en el reconocimiento de imágenes OCR (optical character recognizer, reconocedor óptico de caracteres). En la tabla 1 se muestran las industrias que tienen implementado el concepto de lógica difusa, así como la aplicación correspondiente.

ORGANIZACIÓN	APLICACIÓN O DESARROLLO
ISTUKAWANMA HARIMA HEAVY INDUSTRY	SOFTWARE PARA CONTROL INTELIGENTE
F. L. SMIDTH (DINAMARCA)	CONTROLADOR DIFUSO
OC. ENGINEERING	SOFTWARE PARA CONTROL DE PROCESOS
OLWI ELECTRONICS	CONTROLADOR DIFUSO (FOC-2001)
TOSHIBA - ELECTRONIC	CONTROLADOR DIFUSO (FZ 3000-6000)
TOSHIBA	FUZZY CONTROL ASSEMBLY SYSTEM (IFCS)
NISSHIN DENKI	CONTROLADOR DIFUSO
FUJI DENKI	SISTEMA DE CONTROL DIFUSO DE USO GENERAL
MICOM	ESTACIÓN DE DESARROLLO EN LÓGICA DIFUSA
HEIDEM SHA	CONTROLADOR DIFUSO DE USO GENERAL
OC ENGINEERING	SISTEMA DE CONTROL PARA CEMENTERA
KAWASAKI STEEL	CONTROL DE VELOCIDAD PARA LAMINADOS METÁLICOS
DAIDAN	CONTROL DE TEMPERATURA Y HUMEDAD
NISSIN DENKI	CONTROL DE SISTEMAS DE IONES
NKK	SISTEMA EXPERTO PARA EL CONTROL DE HORNOS
HITACHI M & TOKIO ELECTRIC POWER	CONTROL DE PUESTA EN MARCHA DE CENTRALES DE POTENCIA
MITSUBISHI HEAVY IND	SISTEMA DE CONTROL DE INCINERADORES
MITSUBISHI ELEC	AJUSTE DE PARÁMETROS DE PID
SHARP	SISTEMA DE CONTROL DE AIRE ACONDICIONADO
TATEISHI ELECTRIC	CONTROL DE AGARRE EN ROBOTS
TOSHIBA	AUTOMATIC ROBOT CONTROL
NAGATA STEEL	MONITOR PARA MÁQUINAS
NISSAN AUT	SISTEMA DE CONTROL DE CRUCERO
HITACHI MANUFAC	SISTEMA DE CONTROL DE ELEVADORES
HITACHI MANUFAC	SISTEMA DE OPERACIÓN PARA TRENES SUBURBANOS
HITACHI JAPAN HIGHWAY	SISTEMA DE CONTROL DEL FLUJO DE AIRE DE TÚNELES
FUJI HEAVY INDUSTRY	CONTROL AUTOMÁTICO DE VELOCIDAD PARA VEHÍCULOS
FIJI TECH	SISTEMA DE CONTROL DE ELEVADORES
FIJI ELECT	MOVIMIENTO BIDIMENSIONAL DE ROBOTS
MATSUSHITA ELEC	ROBOTS DE LIMPIEZA DOMÉSTICOS
MATSUSHITA HOME APPLIANCES	SISTEMA DE CONTROL DE MEZCLA PARA LAVADORAS
MATSUSHITA ELEC	SISTEMA DE CONTROL DE ELEVADORES
TATEISHI ELEC KIUSSHO TECH	COMPUTADOR DIFUSO ANALÓGICO
UNIR	
MICOM HOSEI UNIR	COMPUTADOR DIFUSO DIGITAL
AM TOGAI ENFRALOGIC INC.	COMPUTADOR DIFUSO DIGITAL

Tabla 1 Algunas de las aplicaciones de la lógica difusa

La estructura básica de un sistema difuso se muestra en el siguiente diagrama a bloques. Figura 1.

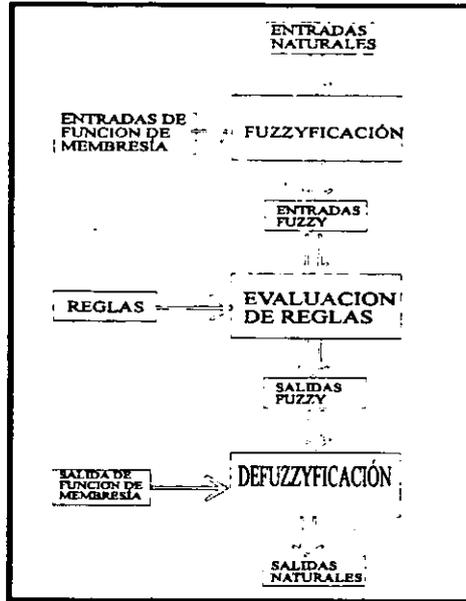


Figura 1. Diagrama a bloques de la lógica difusa

Donde las **Crisp input o entradas nítidas o entradas naturales**: son las entradas tomadas directamente del censor.

Fuzzyficación.- Es el método con el cual se transforman las entradas naturales en entradas fuzzy.

La **función de membresía**.- Define un conjunto fuzzy por mapear las entradas naturales desde su dominio a entradas fuzzy.

Entradas fuzzy o Grado del miembro.- Es el grado de pertenencia de la lectura tomada a un fuzzy set dado.

La **evaluación de reglas** también llamado **inferencia fuzzy**.- En este paso, la técnica llamada mínimo - máximo es generalmente usado para calcular un valor numérico representante de la verdad para una cierta acción consecuente, basado en un conjunto de reglas relacionadas a la consecuencia. el resultado de la evaluación de reglas es una salida fuzzy para cada tipo de acción consecuente.

Reglas.- son usualmente sentencias **SI... ENTONCES...** -que describen la acción que se toma en respuesta a varias entradas fuzzy, aun que se ven libres como lenguaje natural, ellas están confinadas a un conjunto predefinido de términos lingüísticos y una sintaxis estricta.

Defuzzyficación.- aquí se convierten las salidas fuzzy, obtenidas con la evaluación de reglas, en salidas naturales aplicables ya directamente al sistema.

Salidas naturales.- es el resultado de la defuzzyficación, que puede aplicarse al sistema.

La metodología para el diseño con lógica difusa tiene 5 pasos:

- * Análisis y división del sistema de control.
- * Definir las entradas y salidas del sistema de control.
- * Escribir las reglas.
- * Observar el comportamiento del modelo y verificar e iniciar el ajuste de reglas.
- * Optimizar el sistema.

1.2. Planteamiento del problema

El problema consiste, en estacionar un camión con remolque dentro de un estacionamiento, el cual es lo suficientemente grande para poder llevar a cabo las maniobras necesarias para que el vehículo se mueva hacia atrás permitiendo que las puertas de la caja se abran, una vez estacionado, y así poder cargarlo o descargarlo. En particular se desea que sea la zona central del lugar donde deberá arribar el camión, estando ubicado éste último en cualquier parte del inmueble, pero del lado contrario a la meta, el sistema a diseñar deberá entonces colocarlo correctamente sin necesidad de que algún operador intervenga en la tarea. Para que esto sea posible supóngase que el lugar está vacío y que el único movimiento dentro del área del sistema; sólo será el del trailer en estudio. En la figura 2 se muestra una posible ubicación del camión, la meta se puede ver del lado contrario al camión. El vehículo esta colocado al azar, dentro del estacionamiento, el camión y el remolque harán un ángulo cualquiera.

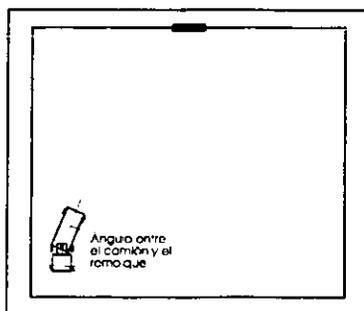


Figura 2 Posible ubicación del camión dentro del estacionamiento

1.3. Modelado

La descripción matemática del prototipo permitirá un mejor análisis del sistema, permitiendo la definición completa de las variables que actúan sobre él, y con ello el desarrollo del modelo en la computadora.

Para determinar la posición del prototipo, se considera que el estacionamiento es un cuadrante del plano cartesiano donde el origen de las coordenadas se encuentra en la esquina superior izquierda, como se ve en la figura 3, el ancho corresponde al eje de las X's que va de 0 a 100 unidades de longitud, lo mismo que el largo, el cual representa al eje de las Y's; siempre se considerarán positivos los valores en este plano.

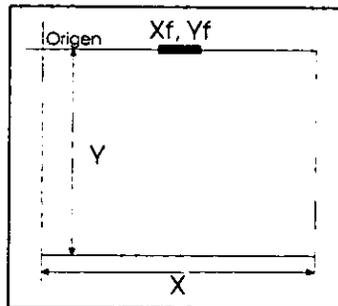


Figura 3. Variables de ubicación dentro del estacionamiento

Como se ve en la figura 3 el lugar donde deberá de llegar el prototipo es el punto $X_f = 50$ ul y $Y_f = 0$ ul, siendo este el caso ideal, considerando un buen resultado aquel con una desviación de hasta ± 5 ul de X_f . De las variables anteriores la que es de interés para este proyecto solo es la X, ya que la variable Y estará colocada siempre entre 85 UL y 100 ul, esto limita al sistema para que se coloque dentro de esta franja, ya que de otro modo el sistema no tendría el espacio suficiente para que se logre el objetivo.

Para simular el movimiento del prototipo, se ha considerado al camión y el remolque como un par de cajas (rectangulares), como se observa en la figura 4. Estas cajas se han definido a través de dos puntos, uno por cada una, y los cálculos de su movimiento y dirección se harán con respecto a estos puntos.

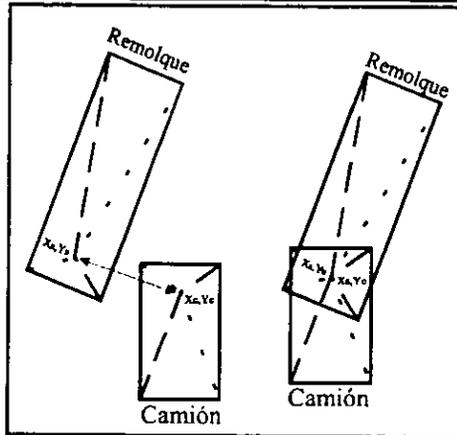


Figura 4 Especificación de los puntos de control del prototipo

El vehículo se desplaza a una velocidad constante y siempre hacia valores menores de Y , lo cual equivaldría, a que el camión se desplaza en reversa.

La variable de salida θ , es la dirección de las llantas. Se obtiene a través de la defuzzyficación.

El controlador a diseñar entonces deberá generar un ángulo θ de la dirección del camión al objetivo, el rango es de -30° a 30° , por definición el remolque tendría el mismo ángulo y rango; pero, con signo contrario al de la dirección (b). Para el modelo, estos últimos ángulos deben de sumarse al ángulo del camión (ϕ_c) y al ángulo del remolque (ϕ_R). Esto es para obtener el nuevo ángulo ϕ_c deberá de sumarse θ y para el nuevo ángulo ϕ_R se le sumará algebraicamente b, esto es b tendrá el signo contrario a θ .

ϕ_c ira de 0° a 360° y ϕ_R de -90° a 90° con respecto a la normal, al plano que corresponde al frente del camión. Estos ángulos se muestran en la figura 5.

Con estos ángulos se termina de definir la posición del prototipo; pero, también se ha considerado que si el camión se mueve sin que la dirección se mueva, el remolque se moverá, incrementando el ángulo, ϕ_R de forma constante y si esto no cambia podrá llegar a $\pm 90^\circ$ deteniendo el sistema para no causarse daño. Para la simulación por computadora no es posible que el sistema tenga un ángulo mayor a $\pm 90^\circ$, pero en el prototipo, este problema lo puede causar el sensado de las distancia o de los ángulos. Esto es si el sensado esta mal hecho o se desconecta o no hay lectura del microcontrolador y los sensores, si se puede presentar el choque del remolque y el camión.

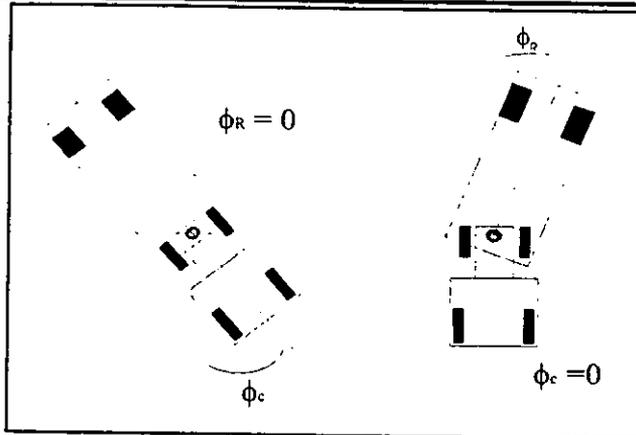


Figura 5. Ángulos que permiten determinar la ubicación del prototipo

Las fórmulas del modelo que definen al camión son:

θ = defuzzyficación

$$X_{cn+1} = X_{cn} - (K * \text{Cos}(\phi_C * 3.14159 / 180))$$

$$Y_{cn+1} = Y_{cn} - (K * \text{Sin}(\phi_C * 3.14159 / 180))$$

$$\phi_{cn+1} = \phi_{cn} \pm \theta$$

Las fórmulas que definen el remolque son:

$$\beta = -\theta$$

$$X_R = X_c; Y_R = Y_c$$

$$X_{Rn+1} = X_{Rn} - (K * \text{Cos}(\phi_R * 3.14159 / 180))$$

$$Y_{Rn+1} = Y_{Rn} - (K * \text{Sin}(\phi_R * 3.14159 / 180))$$

$$\phi_{Rn+1} = \phi_{Rn} \pm \beta$$

donde

(X_{cn+1}) nueva posición del camión en el eje X

(X_{cn}) posición actual del camión en el eje X

(Y_{cn+1}) nueva posición del camión en el eje Y

(Y_{cn}) posición actual del camión en el eje Y

(XR_{n+1}) nueva posición del remolque en el eje X

(XR_n) posición actual del remolque en el eje X

(YR_{n+1}) nueva posición del remolque en el eje Y

(YR_n) posición actual del remolque en el eje Y

(ϕ_R) ángulo del remolque

(ϕ_c) ángulo del camión con la horizontal.

(θ) ángulo de la dirección

(β) ángulo del remolque a actualizar a cada movimiento

K es constante, y permite definir si el sistema se mueve hacia atrás o hacia adelante.

El incremento de β que se presentaría si no existiera señal, sería constante y aproximadamente de 1° por cada avance.

1.4. Planteamiento difuso

La posibilidad de resolver este problema, a través de la descripción de los pasos, para hacer llegar el prototipo a la meta, permite la eliminación de los modelos matemáticos complejos. Del modelado se sabe, que las variables del problema son tres de entrada, X, ϕ_c y ϕ_R , y una variable de salida θ que es la dirección del volante. Es necesario definir el intervalo de valores con los cuales van a trabajar estas variables, la lógica difusa los conoce como Universo del discurso.

Universo del discurso o Universo de trabajo Intervalo de todos los valores posibles aplicables a una variable del sistema.

Los universos de cada una de las variables es el que sigue:

X su universo es de 0 ul a 100 ul.

La variable Y no se estudiará porque, se considera que el camión estará aproximadamente a la misma distancia siempre.

ϕ_c que es el ángulo que forma el camión con respecto a la horizontal que su universo va de 0° a 360° .

Y por último ϕ_R que es el ángulo del remolque y su universo va de -90° a 90° .

Una vez que se sabe el universo de las variables, se requiere que las variables se dividan en pequeños conjuntos llamados fuzzy set, normalmente estos conjuntos son propuestos en función de la experiencia del operador.

Fuzzy Set, Conjuntos Difusos o Funciones de Membresía Un conjunto con límites difusos. Un elemento en un conjunto fuzzy se manifiesta por el grado de membresía, en un intervalo de 0 (0 membresía) a 1 (membresía total) a el conjunto.

Estos conjuntos reciben nombres o etiquetas. No existen reglas rigurosas que determinen los límites estrictamente de los fuzzy set, basta con la intuición y buen criterio.

Label o Etiqueta: nombre descriptivo usado para identificar una función de membresía.

Es común utilizar cualquiera de las gráficas de la figura 6 como fuzzy set, ya que si bien es cierto, otras formas pueden ser más representativas de los fenómenos naturales, estas requieren ecuaciones de mayor complejidad.

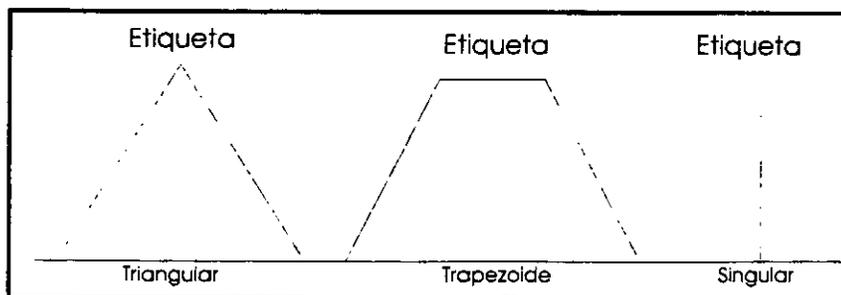


figura 6 Tipos de fuzzy set

La decisión de usar un tipo u otro, deberá ser justificada por la calidad de los resultados obtenidos. El número máximo de fuzzy set para una variable es de nueve, un mayor número complica el manejo del sistema.

Para la variable X se propone 5 grupos que son: Izquierda (LE) de 0 a 41.67 ul, centro izquierda (LC) de 41.67 a 47.5, centro (CE) de 47.5 a 52.5, centro derecha (RC) de 52.5 a 62.5 y derecha (RI) de 62.5 a 100.

Para este sistema se propone el triángulo como fuzzy set.

Los fuzzy set deben de empalmarse. Se recomienda que el grado de empalme no sea mayor al 50 %, el grado menor puede ser de 0% (singleton), este empalme se puede ajustar en función de los resultados que se obtengan en las pruebas de salida. Se ha decidido probar el sistema con un empalme de 50% para todas las variables que intervienen en este problema.

Los fuzzy set de las variables X de entrada, figura 7, tienen la siguiente forma:

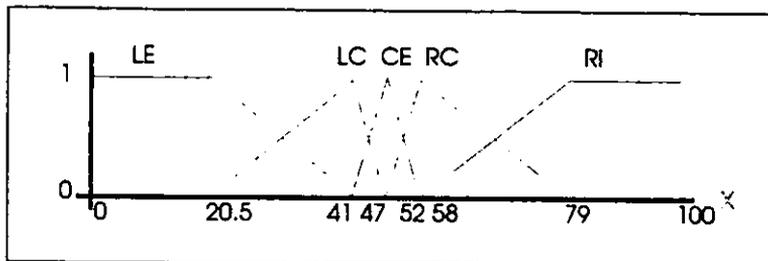


Figura 7 Fuzzy set para la variable X

Para la variable ϕ_c se proponen 9 fuzzy set que son: Bajo derecha (RB) que va de -90 a -10, (OD) de -10 a 10, superior derecha (SD) de 10 a 50, vertical derecha (VD) de 50 a 85, vertical (VE) de 85 a 95, vertical izquierda (VI) de 95 a 130, superior izquierda (SI) de 130 a 170, (OI) de 170 a 190 y bajo izquierda (BI) de 190 a 270. Ver figura 8.

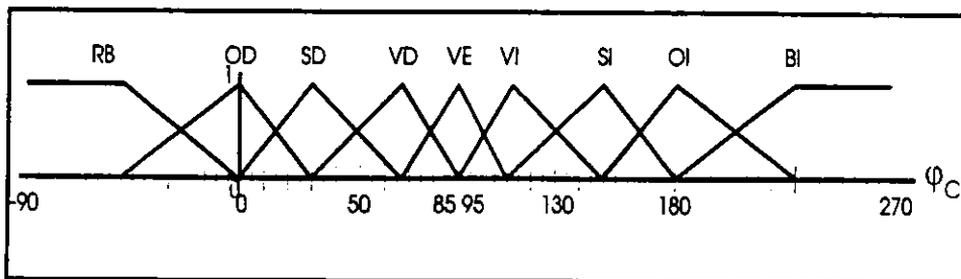


Figura 8. Variable ϕ_c

Variable ϕ_R . Características del desplazamiento del remolque, para la variable ϕ_c se propone 3 grupos: negativo (NE) de -90° a -10° , centro (ZR) de -10° a 10° y positivo (PO) de 10° a 90° . Figura 9.

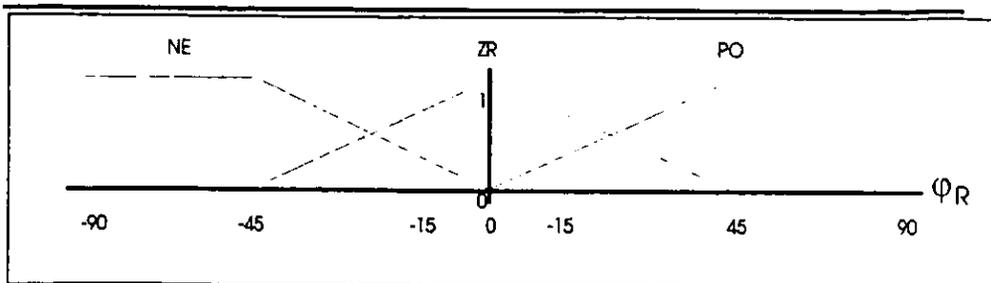


Figura 9. Variable ϕR de remolque.

Para la salida del sistema, también deben de proponerse el universo del discurso con sus fuzzy set. Sólo se analizará para θ por que β solo es un reflejo del primer ángulo, con signo contrario.

Para la variable de salida θ , se proponen 7 fuzzy set que son: negativo grande (NB) de -40° a -20° , negativo mediano (NM) de -20° a -10° , negativo pequeño (NS) de -10° a -5° , cero (ZE) de -5° a 5° , positivo pequeño (PS) de 5° a 10° , positivo mediano (PM) de 10° a 20° y positivo grande (PB) de 20° a 40° . Figura 10.

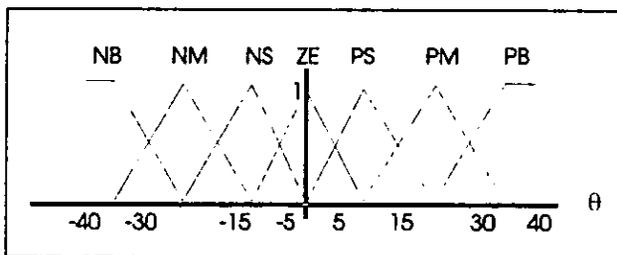


Figura 10. Fuzzy set de salida

El singular o singleton es recomendable sólo para definir las salidas del sistema, porque ésta forma permite una representación fácil para los algoritmos de defuzzyficación en las computadoras; una desventaja, es que no representa la salida tan fiel, como sería deseable.

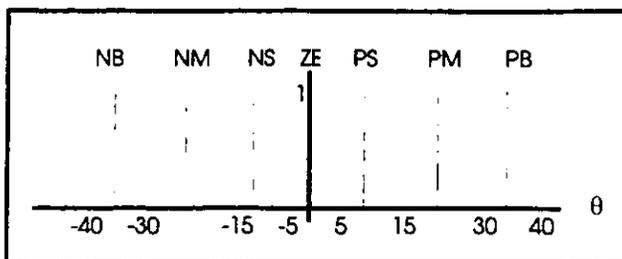


Figura 11. Fuzzy set de salida en forma de singleton

La definición de los fuzzy set es un paso necesario para trabajar con la lógica difusa, ya que es con ellos que se hace la fuzzyficación, que se explica en lo que sigue.

1.4.1. Fuzzyficación

La fuzzyficación, transforma las entradas naturales en entradas fuzzy. Al señalar el grado al que una variable es miembro de un conjunto. El Grado del miembro, de por ejemplo, el elemento X en el fuzzy set A está en el rango de 0 a 1, donde 0 indicando miembro nulo (o completamente falso) y 1 indicando miembro completo (o completamente verdadero), el grado de miembro será representado por la letra griega μ . que es llamado el "valor verdadero" y representa el grado al que una afirmación, es verdadera, la expresión entonces es:

$$\mu_A(X) \rightarrow [0,1].$$

Para obtener los valores de grados de membresía de forma gráfica; bastaría con trazar una línea, a partir del punto que representa la lectura tomada de la variable, esta línea se detendría al chocar con pendiente de la figura que representa el fuzzy set, la altura es entonces el grado de membresía para ese fuzzy set.

El método matemático con el que se calcula el grado de membresía es el siguiente:

$$\text{pendiente 1} = m1 = \frac{1-0}{\text{centro} - \text{punto 1}}; \text{pendiente 2} = m2 = \frac{1-0}{\text{centro} - \text{punto 2}}$$

Cuando la lectura está a la izquierda del triángulo:

$$\Delta 1 = \text{lectura} - \text{punto 1}; \Delta 2 = \text{punto 2} - \text{centro}$$

La lectura está a la derecha del triángulo:

$$\Delta 1 = \text{centro} - \text{punto 1}; \Delta 2 = \text{punto 2} - \text{lectura}$$

Cuando es un trapezoide solo se calcula la pendiente de interés.

grado de membresía = μ = valor mínimo de $(\Delta 1 * m1 * \Delta 2 * m2 * 1)$. Observar la figura 12.

Ejemplo:

Supóngase que se ha medido la distancia de la variable X, del prototipo al punto final.

La lectura es: X = 60 ul .

Al hacer la fuzzyficación de forma gráfica se obtiene que los fuzzy set tienen los siguientes valores:

$$LE = 0$$

$$LC = 0$$

$$CE = 0$$

$$RC = 0.76$$

$$\text{y } RI = 0.24$$

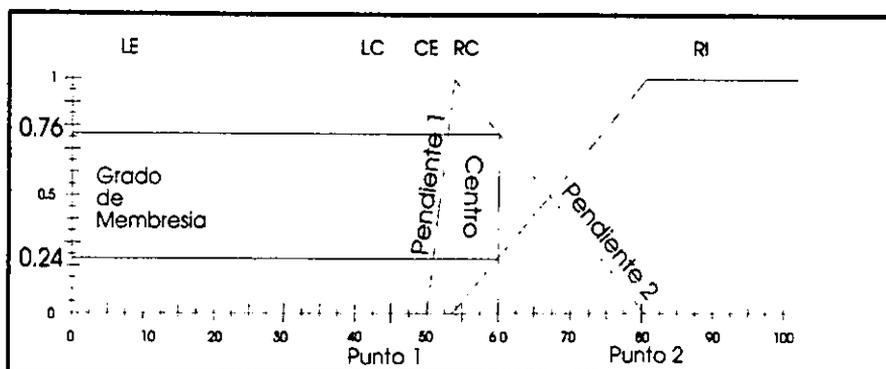


Figura 12. Ejemplo de fuzzyficación

Matemáticamente se obtienen los siguientes resultados:

Para el fuzzy set **RC**.

$$m1 = \frac{1-0}{54-50} = 0.25 ; \quad m2 = \frac{1-0}{54-80} = -0.03846 ; \quad \Delta 1 = 54 - 50 = 4 ; \quad \Delta 2 = 80 - 60 = 20 ; \quad \mu = (4 * 0.25 * 20 * 0.03846 * 1) = 0.7292.$$

Para el fuzzy set **RI**.

$$\text{pendiente 1} = m1 = \frac{1-0}{80-53} = 0.037 ; \quad \Delta 1 = 60 - 53 = 7 ; \quad \mu = (7 * 0.03703 * 1) = 0.2593.$$

1.4.2. Evaluación de reglas

Estas son propuestas por el experto que maneja el sistema, y se van corrigiendo en función de los resultados.

En este paso, se utilizan las variables lingüísticas, para determinar qué acción de control, debe ocurrir en respuesta a un conjunto dado de valores de entrada. Una variable lingüística, como su nombre lo indica, es una variable cuyos valores no son números, sino palabras o sentencias, en un lenguaje natural o sintético. En general los valores de una variable lingüística, pueden ser generados, de un término primario (por ejemplo "cerca"), su antónimo ("lejos"), una colección de modificaciones ("no", "muy", "más" o "menos", etc.) y los conectores "Y" y "O"; cada valor puede ser generado por un contexto libre gramático; cada valor de una variable lingüística, representa una posibilidad de distribución. Estas posibilidades de distribución, pueden ser calculadas de la probabilidad de distribución de los términos primarios y su antónimo a través del uso del atributo de técnicas gramaticales.

La evaluación de regla, también llamada inferencia fuzzy, aplica las reglas a las entradas fuzzy que fueron generados en el proceso de fuzzyficación, entonces, evalúa cada regla con las entradas que fueron generadas desde el proceso de fuzzyficación.

En este paso, la técnica llamada inferencia del mínimo - máximo, es usada generalmente para calcular un valor numérico, representando la verdad para una cierta acción consecuente, basada en un conjunto de reglas de acuerdo a la consecuencia; el resultado de la evaluación de reglas es una salida fuzzy para cada tipo de acción consecuente.

Aunque las reglas se ven libres, como lenguaje natural, ellas están confinadas a un conjunto predefinido de términos lingüísticos, y una sintaxis estricta como la que sigue:

SI ANTECEDENTE 1 Y ANTECEDENTE 2... ENTONCES CONSECUENCIA 1 Y CONSECUENCIA DOS..., que describen la acción que se toma en respuesta a varias entradas fuzzy.

Donde Y es uno de los operadores lógicos permitidos, y el antecedente tiene la forma de:

variable de entrada = etiqueta

Y la consecuencia está en la forma de:

variable de salida = etiqueta.

La forma de uso de **los operadores**, intersección (la base para el operador lógico Y) y Unión (La base para el operador lógico O), se utilizan en la lógica difusa de la siguiente manera:

para el operador Y, el mínimo valor de verdad del antecedente es seleccionado para determinar la fuerza de la regla.

Si X (valor de verdad = 0.3) y Y (valor de verdad = 0.7) entonces Z. La fuerza de la regla = 0.3

Con el operador O, el máximo valor de verdad del antecedente será seleccionado.

Si X (valor de verdad = 0.3) o Y (valor de verdad =0.7) entonces Z. La fuerza de la regla = 0.7

Sin embargo, es recomendable, que los operadores Y se usen preferentemente.

Otro operador importante es el no.

Si X (tiene un valor de verdad = 0.3) entonces no X tiene un valor de verdad de 0.7.

Los pasos en la evaluación de reglas son:

- Crear reglas que describan el comportamiento del sistema deseado, éstas sólo se modifican, si al hacerse la evaluación, no dan el resultado deseado.
- Para un valor de entrada natural particular, determinar el grado de verdad de cada antecedente por la fuzzyficación.
- Encontrar la fuerza de la regla entera, que es igual al mínimo del grado de antecedente de verdad.

La salida fuzzy es determinada por la máxima fuerza de regla de todas las reglas involucradas con la misma acción de salida.

Habrà una salida fuzzy por cada etiqueta de función de membresía de salida.

Si dos o más reglas tratan de afectar la misma salida, la regla que es más verdadera dominará.

- Derívese las salidas fuzzy que serán iguales a la máxima de las fuerzas de las reglas, para cada regla consecuente.

La lógica difusa evalúa las reglas que se proponen, tomando los grados de membresía obtenidos para cada variable, se realiza el producto lógico y la suma lógica.

El producto lógico de todos los grados de miembros es simplemente igual al valor mínimo de todos los grados de miembros involucrados, para el caso de la intersección (y).

La suma lógica toma los valores mayores de los fuzzy set de salida del sistema, la razón de esto es que en los resultados obtenidos con la operación anterior, se está afectando una misma regla dos o más veces y solo en estos casos se utiliza esta operación.

Para la investigación que se está haciendo se decidió tomar todos los fuzzy set de la variable X y se dibujó, sobre un cuadro que representa el estacionamiento, las divisiones de los fuzzy set, con el fin, de poder visualizar el problema, ver la figura 13. Recuérdese que esta división no es rígida, ya que un operador humano, solo puede utilizar su criterio y estas celdas variarán. Solo se utilizarán como referencia, para proponer las reglas que solucionen el sistema.

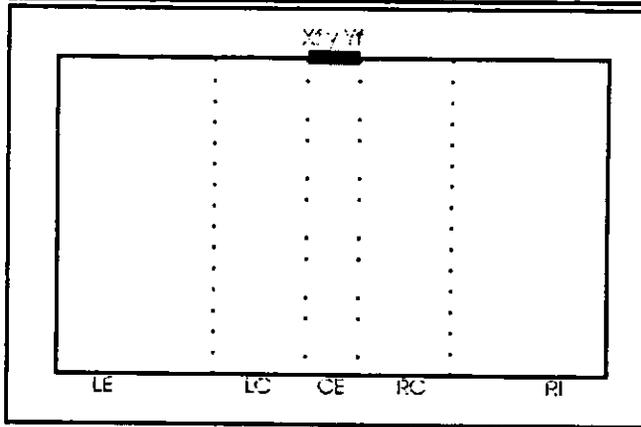


Figura 13. División del estacionamiento entre los distintos fuzzy set de la variable X

Con la variable de entrada ϕ_c se tomó un círculo y se hizo lo mismo que arriba. Figura 14.

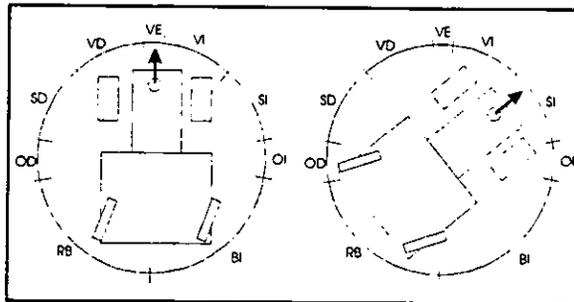


Figura 14. Gráfica para visualización de las reglas de la variable ϕ_c

Por último la variable ϕ_R se dividió sobre un semicírculo como se ve en la figura 15.

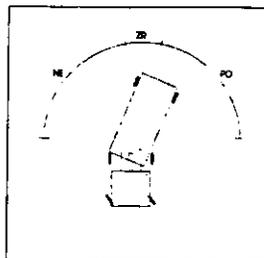


Figura 15. Variable ϕ_R

El siguiente paso fué colocar el dibujo del camión con remolque, dentro del dibujo del estacionamiento, en las distintas combinaciones posibles y por cada combinación se propuso el ángulo que se piensa más adecuado para girar la dirección, para que el prototipo se coloque en el lugar deseado. La cantidad de combinaciones que se analizaron para resolver este problema, son ciento treinta y cinco ($9 \cdot 5 \cdot 3$), que es también el número de reglas que se propusieron.

1.4.2.1. Definición de reglas

Ya que se necesitan 135 reglas, solo se definirán algunas de estas, en forma de texto y el resto se definirá como matrices:

Si el vehículo está a la izquierda y el ángulo del camión es bajo derecha y el ángulo del remolque es positivo **entonces** el ángulo de giro de la dirección es positivo grande.

Si el vehículo está a la izquierda y el ángulo del camión es lejos derecha y el ángulo del remolque es positivo **entonces** el ángulo de giro de la dirección es positivo pequeño.

-
-
-

Si el vehículo está a la derecha y el ángulo del camión es bajo izquierda y el ángulo del remolque es negativo **entonces** el ángulo de giro de la dirección es positivo grande.

Presentadas las reglas de forma matricial se tiene los siguiente:

po	rb	od	sd	vd	ve	vi	si	oi	bi
le	pb	ps	ps	pb	pm	pm	ps	ps	pm
lc	pb	pb	ps	pb	pm	pm	ps	ps	pm
ce	pb	pb	ps	nb	nb	nb	ns	ps	pm
rc	nm	ns	ns	nm	nm	nb	ns	nb	nb
ri	nm	ns	ns	nm	nm	nb	ns	ns	nb

Tabla 2 reglas

zr	rb	od	sd	vd	ve	vi	si	oi	bi
le	pb	pm	pm	pm	ps	pm	pm	ze	nm
lc	pb	pm	pm	pm	ps	ps	ze	nm	nm
ce	nb	pb	pb	nm	ze	nb	nb	pb	pb
rc	pm	pm	ze	ns	ns	nm	nm	nm	nb
ri	pm	ze	nm	nm	ns	nm	nm	nm	nb

Tabla 3 Reglas

ne	rb	od	sd	vd	ve	vi	si	oi	bi
le	nb	nb	ns	nb	nm	nb	ns	nm	nm
lc	nb	nb	ns	nb	nm	nb	ns	nm	nm
ce	nb	nb	ns	pm	pb	pb	ns	nm	nm
rc	pm	pm	ps	pb	pm	pb	ps	pb	pb
ri	pm	pm	ps	pb	pm	pb	ps	pb	pb

Tabla 4 Reglas

Ejemplo de evaluación de reglas:

Supóngase que se ha medido la distancia del prototipo con respecto a la variable "X" y es de 60 ul (unidades de longitud). El ángulo medido en ϕ_c es de 70° . Y el ángulo de ϕ_s es de 15° .

Al hacer la fuzzyficación de las entradas naturales se ha obtenido las siguientes entradas fuzzy. Para la variable "X".

$$LE = 0, LC = 0, CE = 0, RC = 0.76 \text{ y } RI = 0.24.$$

Para la variable " ϕ_c ".

$$RB = 0, OD = 0, SD = 0, VD = 0.97, VE = 0.09, VI = 0, SI = 0, OI = 0 \text{ y } BI = 0.$$

Para la variable " ϕ_s ".

$$NE = 0, ZR = 0.7 \text{ y } PO = 0.3.$$

Los fuzzy set que tienen aplicación en el proceso de evaluación de reglas solo son los valores mayores de 0 y entonces las reglas que son útiles son:

El producto lógico de todos los grados de miembro es simplemente igual al valor mínimo de todos los grados de miembro involucrados.

Si el vehículo está en el centro derecha 0.76 y el ángulo del camión es vertical derecha 0.97 y el ángulo del remolque es positivo 0.3 entonces el ángulo de giro de la dirección es negativo mediano. **nm = 0.3**.

Si el vehículo está en el centro derecha 0.76 y el ángulo del camión es vertical 0.09 y el ángulo del remolque es positivo 0.3 entonces el ángulo de giro de la dirección es negativo mediano. **nm = 0.09**.

Si el vehículo está a la derecha 0.24 y el ángulo del camión es vertical derecha 0.97 y el ángulo del remolque es positivo 0.3 entonces el ángulo de giro de la dirección es negativo mediano. **nm = 0.24**.

Si el vehículo está a la derecha 0.24 y el ángulo del camión es vertical 0.09 y el ángulo del remolque es positivo 0.3 entonces el ángulo de giro de la dirección es negativo mediano. $nm = 0.09$.

Si el vehículo está en el centro derecha 0.76 y el ángulo del camión es vertical derecha 0.97 y el ángulo del remolque es centro 0.7 entonces el ángulo de giro de la dirección es negativo pequeño. $ns = 0.7$.

Si el vehículo está en el centro derecha 0.76 y el ángulo del camión es vertical 0.09 y el ángulo del remolque es centro 0.7 entonces el ángulo de giro de la dirección es negativo pequeño. $ns = 0.09$.

Si el vehículo está a la derecha 0.24 y el ángulo del camión es vertical derecha 0.97 y el ángulo del remolque es centro 0.7 entonces el ángulo de giro de la dirección es negativo mediano. $nm = 0.24$.

Si el vehículo está a la derecha 0.24 y el ángulo del camión es vertical 0.09 y el ángulo del remolque es centro 0.7 entonces el ángulo de giro de la dirección es negativo pequeño. $ns = 0.09$.

Ya que se tienen los datos anteriores se hace la suma lógica, se toman los valores mayores de los fuzzy set de salida del sistema que no se repiten. Para este ejemplo los valores de salida serán: negativo mediano (nm) = 0.3 y negativo pequeño (ns) = 0.7 .

Pero los resultados obtenidos, son salidas fuzzy que no pueden utilizarse directamente en el sistema, debe transformarse estos en salidas naturales.

El proceso que permite la transformación de estos resultados, recibe el nombre de Defuzzyficación.

1.4.3. Defuzzyficación.

En ésta todas las salidas fuzzy, se combinan dentro de un comprensivo y específico resultado por variable de salida.

Existen dos métodos de defuzzyficación: el método del centro de gravedad y el método de la carta única.

1.4.3.1. El método del centro de gravedad

consiste en calcular el centro de masa del área del fuzzy set y consiste de varios pasos:

1. Se determina un punto central, sobre el eje x, para cada fuzzy set
2. Se tomó el grado de miembro obtenido con el producto lógico de cada fuzzy set.

3. Se calcula las áreas de las funciones de membresía.

4. La salida defuzzyficada se obtiene de un término promedio o término pesado de los puntos centrales del eje x y las áreas calculadas. Con el área calculadas sirviendo como peso.

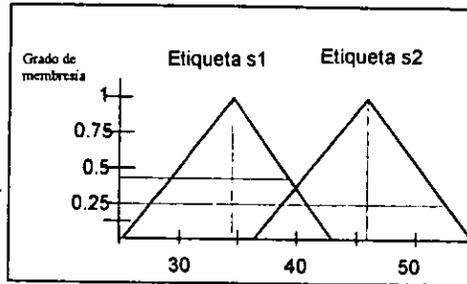


Figura 16. Método de centro de gravedad.

defuzzyficación medio

- Eje x, punto central
- grado de membresía
- Área sombreada

Mismos datos para largo. El área se calcula de la siguiente forma.

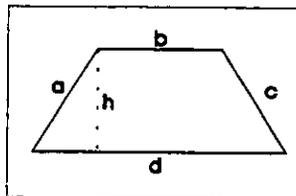


Figura 17. Área a calcular.

La fórmula para calcular el área es:

$$\text{Area} = A1 = \frac{(b + d)h}{2} \text{ con esta misma fórmula se calcula el área 2}$$

La salida natural se obtiene con la fórmula que sigue:

$$\text{salida natural} = \frac{(A1 * \text{punto_central1}) + (A2 * \text{punto_central2})}{A1 + A2}$$

1.4.3.2. Método de las cartas únicas

Este método es usado para simplificar el proceso de defuzzyficación. Una carta única en un fuzzy set singleton. Ya que la carta única intercepta al eje x en un solo punto, el cálculo del centro de gravedad, se reduce solo al cálculo del término promedio o término pesado de los puntos centrales del eje x y los grados de ajuste como pesos.

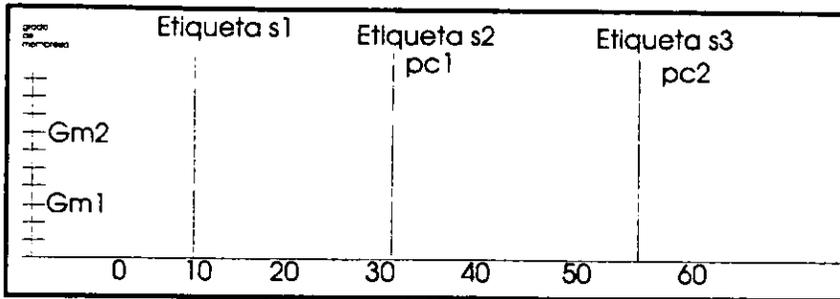


Figura 18 Método de las cartas únicas.

Defuzzyficación:

salida 1

- eje x punto central (pc1)
- grado de membresía (gm1)

salida 2

- eje x punto central (pc2)
- grado de membresía (gm2)

$$\text{salida natural del sistema} = \frac{(gm1 * pc1) + (gm2 * pc2)}{gm1 + gm2}$$

Ejemplo de defuzzyficación.

Con las salidas fuzzy obtenidas anteriormente se calcula la salida natural.

Gm1 = 0.3 (nm).

$$Gm2 = 0.7 \text{ (ns)}$$

$$Pc1 = 15$$

$$Pc2 = 7.5$$

$$\text{salida natural del sistema} \approx \frac{(0.3 * -15) + (0.7 * -7.5)}{0.3 + 0.7} = -9.75$$

El número que resulta es el ángulo que hace girar la dirección 9.75° a la izquierda.

Capítulo 2. Simulación en software del problema propuesto

2.1. Descripción del software de diseño

Existen en la actualidad muchos software¹ que permiten la simulación de la metodología difusa; pero no necesariamente permiten observar el modelo en funcionamiento, e incluso algunos no pueden manejar la cantidad de fuzzy set que se utilizan para analizar el problema de esta tesis. Por estas razones y la posibilidad de que el programa cumpla con todas las necesidades para el análisis, motivaron el diseño de un programa dedicado solo a este problema. El programa está hecho en basic, se ha seleccionado el Método de las cartas únicas para programar la defuzzyficación.

En él puede trabajarse con las gráficas de los fuzzy set o con el modelo desplazándose hacia la meta, bastaría con escribir las coordenadas para ubicar el modelo dentro del estacionamiento y el programa comenzará a calcular las entradas y salidas fuzzy, para obtener las nuevas coordenadas, terminará de calcular, el caso del modelo, al llegar a la meta o al chocar con alguna de las líneas que delimitan el estacionamiento o incluso al generarse más de 90° entre el camión y el remolque.

2.2. Inferencia difusa

En este punto, se analizan las reglas que fueron propuestas en el capítulo 1. Con el fin de que los resultados de la inferencia difusa puedan apreciarse se ha decidido sólo tomar las gráficas que representan al modelo moviéndose hacia la meta que se desea llegar.

Para cuantificarla se propone la ecuación siguiente:

$$\text{Error de entrada al muelle} = \sqrt{(90 - \phi_c)^2 + (X_F - X)^2 + (Y_F - Y)^2 + (0 - \phi_R)^2}$$

A continuación se muestran tres pruebas hechas con el programa diseñado y que se presenta al final de este capítulo.

La primera tiene las coordenadas:

X = 17, Y = 67, $\phi_C = 77^\circ$ y $\phi_R = 20^\circ$. Figura 19.

¹ Algunos de estos programas son: El FUDGE de Motorola 1994, El FULDEK de Bell Helicopter Textron 1993 entre otros.

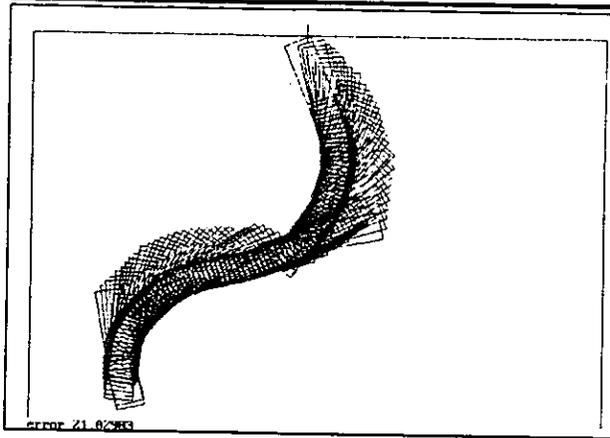


Figura 19. Primera prueba

La segunda prueba tiene las coordenadas son: $X = 50$, $Y = 67$, $\phi C = 85^\circ$ y $\phi R = 10^\circ$. Figura 20.

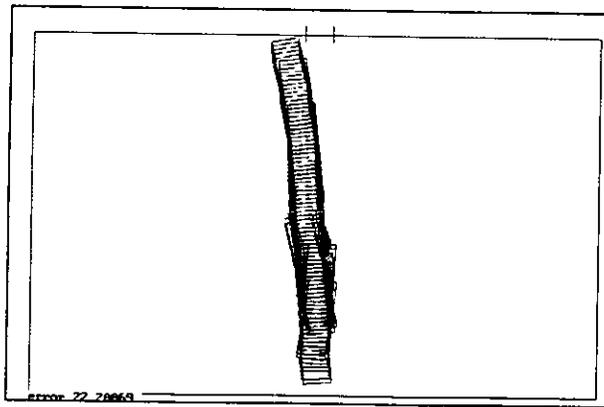


Figura 20. Segunda prueba

La tercera prueba tiene las coordenadas son: $X = 67$, $Y = 67$, $\phi C = 60^\circ$ y $\phi R = 15^\circ$.

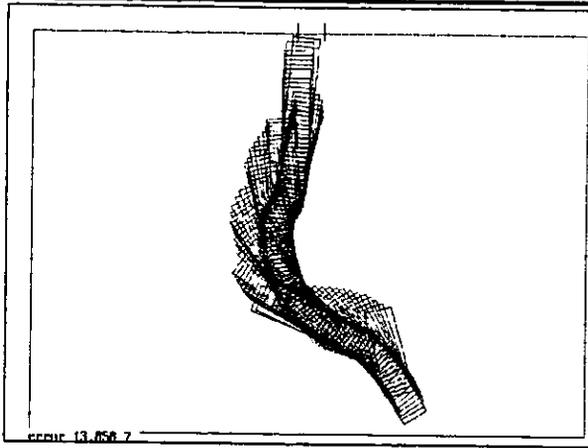


Figura 21. Tercera prueba

Se puede observar en las tres pruebas anteriores que el prototipo se mueve al final hacia afuera de la meta, se ha dicho arriba que puede deberse a que la forma seleccionada de los fuzzy set no sea la correcta o el empalme no sea el adecuado o incluso que las reglas esten mal planteadas.

Con el fin de mejorar los resultados se ha decidido modificar el empalme de los fuzzy set, cambiándolos de 50% a un 30%. Al hacer este movimiento la pendiente de los triángulos cambiará y es posible que el resultado sea el que se desea.

La primera prueba, las coordenadas son: $X = 17$, $Y = 67$, $\phi C = 77^\circ$ y $\phi R = 20^\circ$.

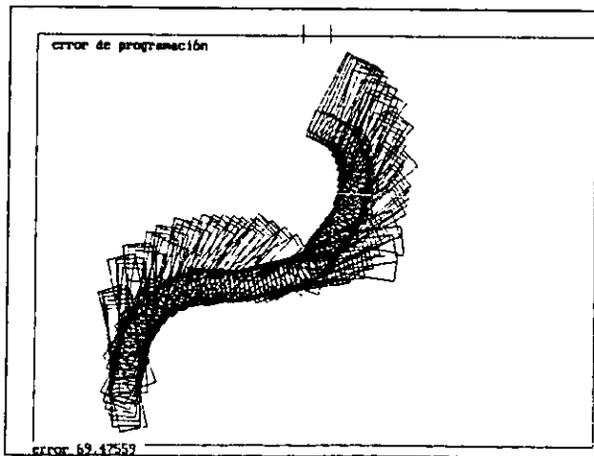


Figura 22. Primera prueba con empalme del 30% en los fuzzy set.

Segunda prueba, las coordenadas son: $X = 50$, $Y = 67$, $\phi C = 85^\circ$ y $\phi R = 10^\circ$. Figura 23.

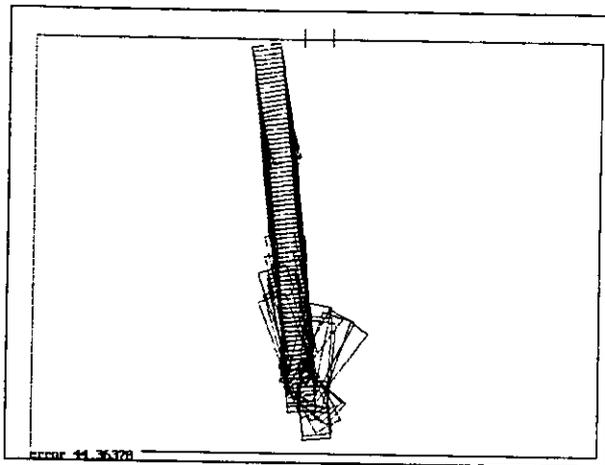


Figura 23. Segunda prueba con empalme de 30% en los fuzzy set.

Tercera prueba, las coordenadas son: $X = 67$, $Y = 67$, $\phi C = 60^\circ$ y $\phi R = 15^\circ$. Figura 24.

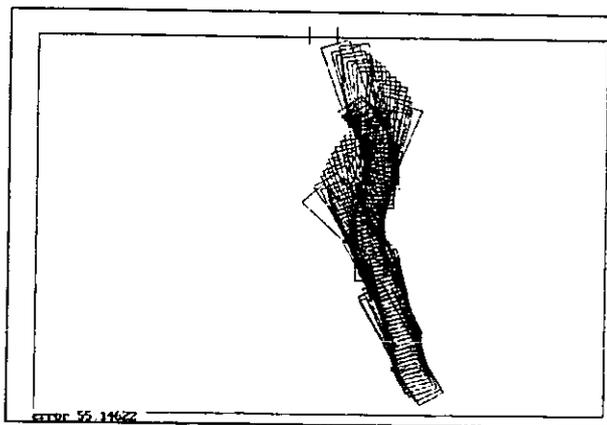


Figura 24. Tercera prueba con empalme de 30% en los fuzzy set.

Puede observarse que el error aumenta, se ha decidido no modificar el empalme, lo que se modificará entonces serán las reglas, en especial las que están relacionadas con el centro del estacionamiento. Y se proponen los siguientes cambios:

Se propone que las reglas asociadas con la variable "po" se conserven sin cambio. Tabla5.

po	rb	od	sd	vd	ve	vi	si	oi	bi
le	pb	ps	ps	pb	pm	pm	ps	ps	pm
lc	pb	pb	ps	pb	pm	pm	ps	ps	pm
ce	pb	pb	ps	nb	nb	nb	ns	ps	pm
rc	nm	ns	ns	nm	nm	nb	ns	nb	nb
ri	nm	ns	ns	nm	nm	nb	ns	ns	nb

Tabla 5 reglas

Para las reglas relacionadas con la variable "cr", se proponen algunos cambios. Tabla 6.

cr	rb	od	sd	vd	ve	vi	si	oi	bi
le	np	pb	pb	pb	ps	pm	pm	ze	nm
lc	pb	pb	pm	pb	ps	ze	ns	nm	nb
ce	pb	pb	pm	ps	ze	ns	nm	nb	nb
rc	pb	pm	ps	ze	ns	nb	nm	nb	nb
ri	pm	ze	nm	nm	nb	nb	nb	nb	pb

Tabla 6 Reglas

Las reglas que siguen "no" se modifican. Tabla 7.

ne	rb	od	sd	vd	ve	vi	si	oi	bi
le	nb	nb	ns	nb	nm	nb	ns	nm	nm
lc	nb	nb	ns	nb	nm	nb	ns	nm	nm
ce	nb	nb	ns	pm	pb	pb	ns	nm	nm
rc	pm	pm	ps	pb	pm	pb	ps	pb	pb
ri	pm	pm	ps	pb	pm	pb	ps	pb	pb

Tabla 7 Reglas

Ahora se presentan las pruebas realizadas con las reglas modificadas, con fuzzy set empalmados a un 50 %.

Primera prueba, las coordenadas son: $X = 17$, $Y = 67$, $\phi C = 77^\circ$ y $\phi R = 20^\circ$. Figura 25.

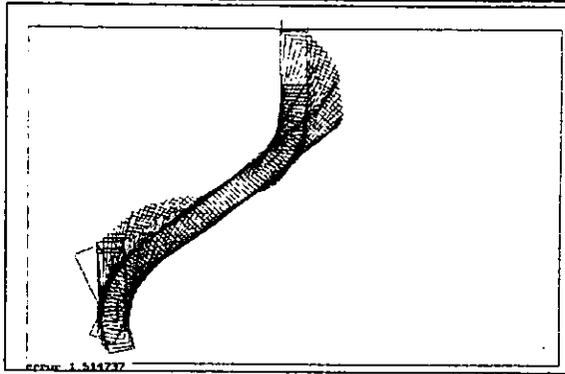


Figura 25. Primera prueba con reglas modificadas

Puede observarse que el error ha disminuido a un valor tolerable.

Segunda prueba, las coordenadas son: $X = 50$, $Y = 67$, $\phi_C = 85^\circ$ y $\phi_R = 10^\circ$. Figura 26.

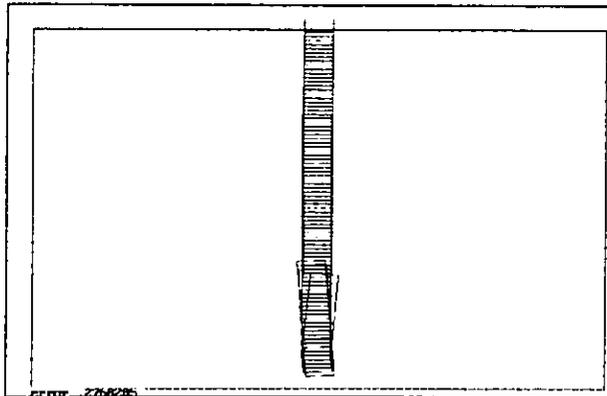


Figura 26. Segunda prueba con reglas ya modificadas.

Puede observarse que el modelo se mueve más limpio que en las pruebas anteriores.

Tercera prueba, las coordenadas son: $X = 67$, $Y = 67$, $\phi_C = 60^\circ$ y $\phi_R = 15^\circ$. Figura 27.

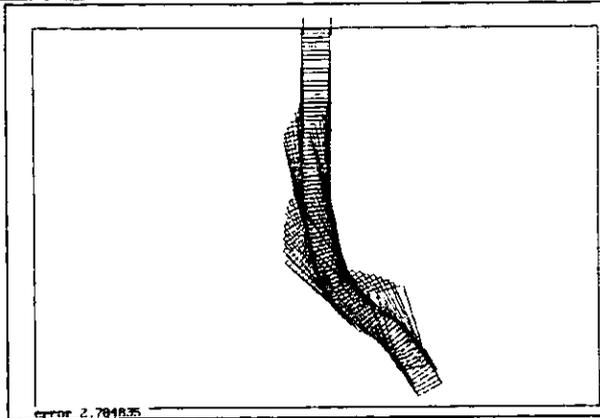


Figura 27. Tercera prueba con reglas modificadas

El error se reduce significativamente hasta un valor tolerable, de acuerdo a lo que se ha planteado en la definición del problema.

2.3. Programa de simulación del prototipo.

```

1 CLS: REM carro
i = 5: cl = 11: mo = SQR((15 ^ 2) + (80 ^ 2))
mm = SQR((15 ^ 2) + (40 ^ 2)): nn = ATN(15 / 40): pp = ATN(40 / 15)
no = ATN(15 / 80): p = ATN(80 / 15): jo = SQR((15 ^ 2) + (15 ^ 2))
REM " entrada de datos principales "
ent = 3: nent$(1) = "x": nent$(2) = "o": nent$(1) = "ko": xe = 50: xef = 550: ye = 210: yef = 10
REM " salidas de datos principales "
sal = 1: nsal$ = "phi": xs = 50: xsf = 550: ys = 450: ysf = 250
REM " entradas datos de fuzzy set "
vr(1) = 5: vr(2) = 9: vr(3) = 3: re = 50: re = re / 100
REM $ descripción de los grupos de trabajo $
g$(1, 1) = "le": g$(2, 1) = "lc": g$(3, 1) = "ce": g$(4, 1) = "rc": g$(5, 1) = "ri"
g$(1, 2) = "rb": g$(2, 2) = "od": g$(3, 2) = "sd": g$(4, 2) = "vd": g$(5, 2) = "ve": g$(6, 2) = "vi": g$(7, 2) = "si":
g$(8, 2) = "oi": g$(9, 2) = "bi"
g$(1, 3) = "po": g$(2, 3) = "cr": g$(3, 3) = "ne"
li(1, 1) = 0: li(2, 1) = 250: li(3, 1) = 285: li(4, 1) = 315: li(5, 1) = 375
ls(1, 1) = 250: ls(2, 1) = 285: ls(3, 1) = 315: ls(4, 1) = 375: ls(5, 1) = 600
li(1, 2) = -90: li(2, 2) = -10: li(3, 2) = 10: li(4, 2) = 50: li(5, 2) = 85: li(6, 2) = 95: li(7, 2) = 130: li(8, 2) = 170:
li(9, 2) = 190
ls(1, 2) = -10: ls(2, 2) = 10: ls(3, 2) = 50: ls(4, 2) = 85: ls(5, 2) = 95: ls(6, 2) = 130: ls(7, 2) = 170: ls(8, 2) = 190:
ls(9, 2) = 270
li(1, 3) = -90: li(2, 3) = -10: li(3, 3) = 10
ls(1, 3) = -10: ls(2, 3) = 10: ls(3, 3) = 90
REM " salidas de datos de fuzzy set "
fs(1) = 7: rs = 50: rs = rs / 100
gs$(1, 1) = "nb": gs$(2, 1) = "nm": gs$(3, 1) = "ns": gs$(4, 1) = "ze": gs$(5, 1) = "ps": gs$(6, 1) = "pm": gs$(7, 1)
= "pb"
lis(1, 1) = -30: lis(2, 1) = -20: lis(3, 1) = -10: lis(4, 1) = -5: lis(5, 1) = 5: lis(6, 1) = 10: lis(7, 1) = 20

```

```

lss(1, 1) = -20: lss(2, 1) = -10: lss(3, 1) = -5: lss(4, 1) = 5: lss(5, 1) = 10: lss(6, 1) = 20: lss(7, 1) = 30
f = 600: ii = -90: fs = 30: iis = -30
REM *          reglas          *
pr$(1, 1, 1) = "pb": pr$(1, 2, 1) = "ps": pr$(1, 3, 1) = "ps": pr$(1, 4, 1) = "pb": pr$(1, 5, 1) = "pm": pr$(1, 6, 1) =
"pm": pr$(1, 7, 1) = "ps": pr$(1, 8, 1) = "ps": pr$(1, 9, 1) = "pm"
pr$(2, 1, 1) = "pb": pr$(2, 2, 1) = "pb": pr$(2, 3, 1) = "ps": pr$(2, 4, 1) = "pb": pr$(2, 5, 1) = "pm": pr$(2, 6, 1) =
"pm": pr$(2, 7, 1) = "ps": pr$(2, 8, 1) = "ps": pr$(2, 9, 1) = "pm"
pr$(3, 1, 1) = "pb": pr$(3, 2, 1) = "pb": pr$(3, 3, 1) = "ps": pr$(3, 4, 1) = "nb": pr$(3, 5, 1) = "nb": pr$(3, 6, 1) =
"nb": pr$(3, 7, 1) = "ns": pr$(3, 8, 1) = "ps": pr$(3, 9, 1) = "pm"
pr$(4, 1, 1) = "nm": pr$(4, 2, 1) = "ns": pr$(4, 3, 1) = "ns": pr$(4, 4, 1) = "nm": pr$(4, 5, 1) = "nm": pr$(4, 6, 1) =
"nb": pr$(4, 7, 1) = "ns": pr$(4, 8, 1) = "ns": pr$(4, 9, 1) = "nb"
pr$(5, 1, 1) = "nm": pr$(5, 2, 1) = "ns": pr$(5, 3, 1) = "ns": pr$(5, 4, 1) = "nm": pr$(5, 5, 1) = "nm": pr$(5, 6, 1) =
"nb": pr$(5, 7, 1) = "ns": pr$(5, 8, 1) = "ns": pr$(5, 9, 1) = "nb"
REM
pr$(1, 1, 2) = "nb": pr$(1, 2, 2) = "pb": pr$(1, 3, 2) = "pb": pr$(1, 4, 2) = "pb": pr$(1, 5, 2) = "ps": pr$(1, 6, 2) =
"pm": pr$(1, 7, 2) = "pm": pr$(1, 8, 2) = "ze": pr$(1, 9, 2) = "nm"
pr$(2, 1, 2) = "pb": pr$(2, 2, 2) = "pb": pr$(2, 3, 2) = "pm": pr$(2, 4, 2) = "pb": pr$(2, 5, 2) = "ps": pr$(2, 6, 2) =
"ze": pr$(2, 7, 2) = "ns": pr$(2, 8, 2) = "nm": pr$(2, 9, 2) = "nb"
pr$(3, 1, 2) = "pb": pr$(3, 2, 2) = "pb": pr$(3, 3, 2) = "ps": pr$(3, 4, 2) = "ps": pr$(3, 5, 2) = "ze": pr$(3, 6, 2) =
"ns": pr$(3, 7, 2) = "nm": pr$(3, 8, 2) = "nb": pr$(3, 9, 2) = "nb"
pr$(4, 1, 2) = "pb": pr$(4, 2, 2) = "pm": pr$(4, 3, 2) = "ps": pr$(4, 4, 2) = "ze": pr$(4, 5, 2) = "ns": pr$(4, 6, 2) =
"nb": pr$(4, 7, 2) = "nm": pr$(4, 8, 2) = "nb": pr$(4, 9, 2) = "nb"
pr$(5, 1, 2) = "pm": pr$(5, 2, 2) = "ze": pr$(5, 3, 2) = "nm": pr$(5, 4, 2) = "nm": pr$(5, 5, 2) = "nb": pr$(5, 6, 2) =
"nb": pr$(5, 7, 2) = "nb": pr$(5, 8, 2) = "nb": pr$(5, 9, 2) = "pb"
REM
pr$(1, 1, 3) = "nb": pr$(1, 2, 3) = "nb": pr$(1, 3, 3) = "ns": pr$(1, 4, 3) = "nb": pr$(1, 5, 3) = "nm": pr$(1, 6, 3) =
"nb": pr$(1, 7, 3) = "ns": pr$(1, 8, 3) = "nm": pr$(1, 9, 3) = "nm"
pr$(2, 1, 3) = "nb": pr$(2, 2, 3) = "nb": pr$(2, 3, 3) = "ns": pr$(2, 4, 3) = "nb": pr$(2, 5, 3) = "nm": pr$(2, 6, 3) =
"nb": pr$(2, 7, 3) = "ns": pr$(2, 8, 3) = "nm": pr$(2, 9, 3) = "nm"
pr$(3, 1, 3) = "nb": pr$(3, 2, 3) = "nb": pr$(3, 3, 3) = "ns": pr$(3, 4, 3) = "pm": pr$(3, 5, 3) = "pb": pr$(3, 6, 3) =
"pb": pr$(3, 7, 3) = "ns": pr$(3, 8, 3) = "nm": pr$(3, 9, 3) = "nm"
pr$(4, 1, 3) = "pm": pr$(4, 2, 3) = "pm": pr$(4, 3, 3) = "ps": pr$(4, 4, 3) = "pb": pr$(4, 5, 3) = "pm": pr$(4, 6, 3) =
"pb": pr$(4, 7, 3) = "ps": pr$(4, 8, 3) = "pb": pr$(4, 9, 3) = "pb"
pr$(5, 1, 3) = "pm": pr$(5, 2, 3) = "pm": pr$(5, 3, 3) = "ps": pr$(5, 4, 3) = "pb": pr$(5, 5, 3) = "pm": pr$(5, 6, 3) =
"pb": pr$(5, 7, 3) = "ps": pr$(5, 8, 3) = "pb": pr$(5, 9, 3) = "pb"
CLS
INPUT "coordenadas en x hasta 600": ax
INPUT "coordenadas en y hasta 360": by
INPUT "grados O": o
10 INPUT "grados O, el rango de este ángulo es de -90 a 90": k
ko = k
IF k < -90 OR k > 90 THEN 10
IF k = 0 THEN k = o: GOTO 11
IF k > 0 THEN k = k + o: GOTO 11
IF k < 0 THEN k = k + o: GOTO 11
11 PRINT k, ko, o: INPUT rct
REM ***** cálculos *****
FOR m = 1 TO ent
FOR n = 1 TO vr(m)
lg(n) = ls(n, m) - li(n, m): mi(n) = lg(n) / 2: ct(n, m) = li(n, m) + mi(n): REM cálculos de triángulos primarios
s(n, m) = re * mi(n) / .5: REM calcula el porcentaje de empalme
NEXT n
NEXT m
FOR a = 1 TO sal

```

```

FOR b = 1 TO fs(a)
lgs(b) = lss(b, a) - lis(b, a): mis(b) = lgs(b) / 2: cts(b, a) = lis(b, a) + mis(b): REM cálculos de triángulos primarios
ss(b, a) = rs * mis(b) / .5: REM calcula el porcentaje de empalme
NEXT b
NEXT a
REM ===== cálculo de pendientes de salida =
FOR a = 1 TO sal
FOR b = 1 TO fs(a)
lis(b, a) = lis(b, a) - ss(b - 1, a): lss(b, a) = lss(b, a) + ss(b + 1, a): REM límites nuevos
ppis(b, a) = 1 / (cts(b, a) - lis(b, a)): REM pendiente del primer límite del triángulo
ppds(b, a) = 1 / (cts(b, a) - lss(b, a)): REM pendiente del segundo límite del triángulo
NEXT b
NEXT a
REM ===== cálculo de pendientes =
FOR m = 1 TO ent
FOR n = 1 TO vr(m)
li(n, m) = li(n, m) - s(n - 1, m): ls(n, m) = ls(n, m) + s(n + 1, m): REM límites nuevos
ppi(n, m) = 1 / (ct(n, m) - li(n, m)): REM pendiente del primer límite del triángulo
ppd(n, m) = 1 / (ct(n, m) - ls(n, m)): REM pendiente del segundo límite del triángulo
NEXT n
NEXT m
SCREEN 12
WINDOW (750, 650)-(-9, -90)
REM ***** marco *****
LINE (0, 12)-(0, 600): LINE (600, 12)-(600, 600)
LINE (0, 12)-(600, 12): LINE (0, 600)-(600, 600)
LINE (285, 0)-(285, 20): LINE (315, 0)-(315, 20)
LINE (195, 400)-(195, 20), 8, 3: LINE (285, 400)-(285, 20), 8, 3
LINE (315, 400)-(315, 20), 8, 3: LINE (405, 400)-(405, 20), 8, 3
LINE (0, 133)-(600, 133), 8, 3: LINE (0, 266)-(600, 266), 8, 3
507 REM /***** fuzzificación *****/
16 REM entrada de variable
FOR m = 1 TO ent
cnt(cnt) = 0
IF m = 1 THEN x = ax
IF m = 2 THEN x = o
IF m = 3 THEN x = ko
t(m) = x
VIEW PRINT 29 TO 30
PRINT "error": SQR((90 - o) ^ 2 + (300 - ax) ^ 2 + (105 - by) ^ 2 + (0 - ko) ^ 2)
VIEW PRINT
REM ***** cálculos de fuzzificación *****
FOR n = 1 TO vr(m)
IF x < ct(1, m) THEN x = ct(1, m): GOTO 50
IF x > ct(vr(m), m) THEN x = ct(vr(m), m): GOTO 60
50 IF x < ct(n, m) THEN d1(n, m) = x - li(n, m): d2(n, m) = ls(n, m) - ct(n, m)
60 IF x > ct(n, m) THEN d1(n, m) = ct(n, m) - li(n, m): d2(n, m) = ls(n, m) - x
IF x = ct(n, m) THEN d1(n, m) = ct(n, m) - li(n, m): d2(n, m) = ls(n, m) - ct(n, m)
gm(n, m) = d1(n, m) * ABS(ppi(n, m)) * d2(n, m) * ABS(ppd(n, m)) * 1
REM salidas de resultados de fuzzificación
IF gm(n, m) < 0 THEN 100
cnt(cnt) = cnt(cnt) + 1
cf(cnt(cnt), m) = gm(n, m): nn(cnt(cnt), m) = n
rcs(cnt(cnt), m) = gm(n, m): res$(cnt(cnt)) = g$(n, m)

```

```

100 NEXT n
IF m = 1 THEN cpv = cnt(ent)
IF cnt(ent) = 1 THEN res(2, m) = 0
IF m = 2 THEN cpb = cnt(ent)
IF cnt(ent) = 2 THEN res(3, m) = 0
NEXT m
IF db$ = "m" OR db$ = "M" THEN 120
120 REM ***** modelo *****
REM primer cuadro
ax = ax - (i * COS(o * 3.14159 / 180)); by = by - (i * SIN(o * 3.14159 / 180))
x1 = mm * COS((o * 3.14159 / 180) + nn); y1 = mm * SIN((o * 3.14159 / 180) + nn)
x2 = mm * COS(((o + 270) * 3.14159 / 180) + pp); y2 = mm * SIN(((o + 270) * 3.14159 / 180) + pp)
x3 = jo * COS((o + 225) * 3.14159 / 180); y3 = jo * SIN((o + 225) * 3.14159 / 180)
x4 = jo * COS((o + 135) * 3.14159 / 180); y4 = jo * SIN((o + 135) * 3.14159 / 180)
LINE (x2 + ax, y2 + by)-(x1 + ax, y1 + by); LINE (x3 + ax, y3 + by)-(x4 + ax, y4 + by)
LINE (x2 + ax, y2 + by)-(x3 + ax, y3 + by); LINE (x1 + ax, y1 + by)-(x4 + ax, y4 + by)
REM segundo cuadro
zx = ax; wy = by
zx = zx - (i * COS(k * 3.14159 / 180)); wy = wy - (i * SIN(k * 3.14159 / 180))
f1 = mo * COS(((k + 180) * 3.14159 / 180) + no); g1 = mo * SIN(((k + 180) * 3.14159 / 180) + no)
f2 = mo * COS(((k + 90) * 3.14159 / 180) + p); g2 = mo * SIN(((k + 90) * 3.14159 / 180) + p)
f3 = jo * COS((k + 45) * 3.14159 / 180); g3 = jo * SIN((k + 45) * 3.14159 / 180)
f4 = jo * COS((k + 315) * 3.14159 / 180); g4 = jo * SIN((k + 315) * 3.14159 / 180)
LINE (f2 + zx, g2 + wy)-(f1 + zx, g1 + wy)
LINE (f3 + zx, g3 + wy)-(f4 + zx, g4 + wy)
LINE (f2 + zx, g2 + wy)-(f3 + zx, g3 + wy)
LINE (f1 + zx, g1 + wy)-(f4 + zx, g4 + wy)
IF (g2 + wy) < 16 OR (g1 + wy) < 16 THEN END
130 REM ***** cálculo de mínimos *****
sum = 1
FOR fr = 1 TO cpv
FOR rr = 1 TO cnt(ent)
FOR er = 1 TO cpb
IF cf(er, 1) < cf(rr, 2) AND ef(fr, 1) < ef(fr, 3) THEN min(er, rr, fr) = cf(er, 1): si$(er, rr, fr) = pr$(nn(rr, 1), nn(er, 2), nn(fr, 3)): sum = sum + 1: GOTO 131
IF cf(rr, 2) < cf(er, 1) AND ef(rr, 2) < ef(fr, 3) THEN min(er, rr, fr) = cf(rr, 2): si$(er, rr, fr) = pr$(nn(rr, 1), nn(er, 2), nn(fr, 3)): sum = sum + 1: GOTO 131
IF cf(rr, 2) = cf(er, 1) AND ef(er, 1) < ef(fr, 3) THEN min(er, rr, fr) = cf(er, 1): si$(er, rr, fr) = pr$(nn(rr, 1), nn(er, 2), nn(fr, 3)): sum = sum + 1: GOTO 131
IF cf(er, 1) < ef(rr, 2) AND ef(er, 1) = ef(fr, 3) THEN min(er, rr, fr) = cf(er, 1): si$(er, rr, fr) = pr$(nn(rr, 1), nn(er, 2), nn(fr, 3)): sum = sum + 1: GOTO 131
IF ef(rr, 2) < ef(er, 1) AND ef(rr, 2) = ef(fr, 3) THEN min(er, rr, fr) = ef(rr, 2): si$(er, rr, fr) = pr$(nn(rr, 1), nn(er, 2), nn(fr, 3)): sum = sum + 1: GOTO 131
IF cf(rr, 2) = ef(er, 1) AND ef(er, 1) > ef(fr, 3) THEN min(er, rr, fr) = ef(fr, 3): si$(er, rr, fr) = pr$(nn(rr, 1), nn(er, 2), nn(fr, 3)): sum = sum + 1: GOTO 131
IF ef(er, 1) < cf(rr, 2) AND ef(rr, 2) > ef(fr, 3) THEN min(er, rr, fr) = ef(fr, 3): si$(er, rr, fr) = pr$(nn(rr, 1), nn(er, 2), nn(fr, 3)): sum = sum + 1: GOTO 131
IF cf(rr, 2) = ef(er, 1) AND ef(er, 1) = ef(fr, 3) THEN min(er, rr, fr) = cf(er, 1): si$(er, rr, fr) = pr$(nn(rr, 1), nn(er, 2), nn(fr, 3)): sum = sum + 1: GOTO 131
IF cf(rr, 2) < cf(er, 1) AND cf(rr, 2) > ef(fr, 3) THEN min(er, rr, fr) = cf(fr, 3): si$(er, rr, fr) = pr$(nn(rr, 1), nn(er, 2), nn(fr, 3)): sum = sum + 1: GOTO 131
131 NEXT er
NEXT rr
NEXT fr

```

```

REM ***** cálculo de máximos *****
zz = 0
FOR z = 1 TO sum
max = 0
FOR cr = 1 TO cpb
FOR fr = 1 TO cpv
FOR rr = 1 TO cnt(ent)
IF max < min(rr, cr, fr) THEN max = min(rr, cr, fr): cx = rr: cy = cr: cz = fr: zi$(z) = si$(rr, cr, fr)
NEXT rr
NEXT fr
NEXT cr
FOR fr = 1 TO cpv
FOR cr = 1 TO cpb
FOR rr = 1 TO cnt(ent)
IF si$(cx, cy, cz) = si$(rr, cr, fr) THEN min(rr, cr, fr) = 0
NEXT rr
NEXT cr
NEXT fr
IF max > 0 THEN zz = zz + 1: ga(zz) = max
NEXT z
FOR z = 1 TO zz + 2
FOR a = 1 TO sal
FOR b = 1 TO fs(a)
IF zi$(z) = gs$(b, a) THEN pc(z) = cts(b, a)
NEXT b
NEXT a
NEXT z
REM ***** defuzzificación *****
div = 0: db = 0
FOR df = 1 TO zz + 2
div = div + ga(df) * pc(df)
db = db + ga(df)
NEXT df
dfs = div / db
o = o + dfs: k = k - dfs
IF o > 270 THEN o = o - 360
IF o < -90 THEN o = 360 + o
REM condiciones de las variables
IF ko > 90 THEN SOUND 123, 22: LOCATE 2, 2: PRINT " error de programación": END
IF ko < -90 THEN SOUND 123, 22: LOCATE 2, 2: PRINT " error de programación": END
IF k > 270 AND o < 270 THEN k = k + 5: GOTO 20
IF o > k THEN k = k + 5
IF o < k THEN k = k - 5
IF o = k THEN k = k + 0
20 IF k > 360 THEN k = k - 360
IF k < 0 THEN k = k + 360
IF o > 360 THEN o = o - 360
IF o < 0 THEN o = o + 360
IF o = k THEN ko = 0: REM originales
IF o > k AND ax > 300 THEN ko = k - o
IF o < k AND ax > 300 THEN ko = (o - k) * -1
IF o > k AND ax < 300 THEN ko = o - k
IF o < k AND ax < 300 THEN ko = (k - o) * -1
GOTO 16

```

Capítulo 3. Algoritmo de diseño con el microcontrolador

3.1. Características generales del microcontrolador 68HC11.

El microcontrolador 68HC11 es uno de los más poderosos en 8 bits, está integrado, fabricado por motorola con tecnología de alta densidad metal Oxido Semiconductor complementaria, su cerebro es un microprocesador 6800 mejorado, conservando los códigos de operación de éste último, pero cuenta con más registros e instrucciones, puede trabajar a mayor frecuencia y el registro de código de condición se ha completado a 8 bits, puede estar en modo de espera (standby) en que consume muy poca energía.

Todas las versiones cuentan con memoria RAM, ROM y EEPROM, algunas versiones tiene eprom; los tamaños de las memorias dependen de la versión de que se trate.

El HC11 tiene 5 puertos paralelos, 2 puertos serie, uno síncrono y otro asíncrono, éste hecho para operar con la norma RS232 C.

Cuenta con un temporizador muy poderoso, que puede hacer operaciones de entrada y salida, acumular pulsos o tiempos, este solo bloque maneja 20 registros.

Tiene un convertidor analógico - digital, de 4 entradas que opera a buena velocidad y no requiere de ningún elemento de muestreo y retención.

Una característica importante es el gran número de interrupciones con que puede operar y que son 18 para todos los bloques, contando con una interrupción de tiempo real. Tiene 4 tipos de Reset siendo uno de ellos generado por fallas de programación al no detectar códigos de operación legales.

En la figura 28 se muestra un diagrama a cuadros del HC11.

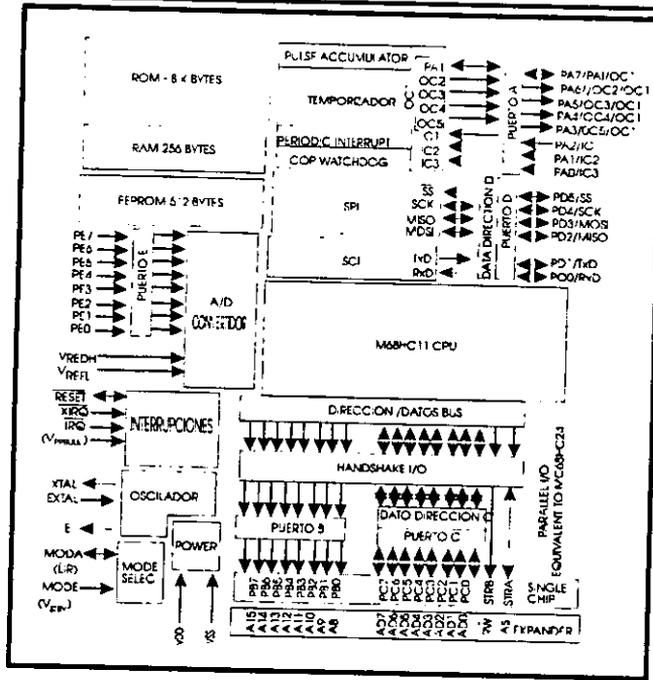


Figura 28. Diagrama a cuadros del microcontrolador.

Cabe destacar que el puerto C se pueden tener multiplexados los buses de datos y de direcciones. De entre las varias versiones de HC11 se muestra en la figura 29 las asignaciones de terminales para una el encapsulado de 52 patitas.

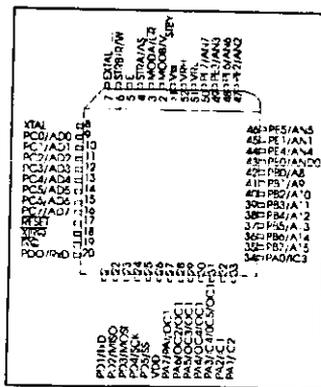


Figura 29. Asignación de terminales del MC68HC11E9/711E9

El voltaje de alimentación se aplica en las terminales VDD y VSS, debe ser de 5 V.

Las terminales VREFH y VREFL llevan el voltaje de alimentación del subsistema Conv. analógico a digital y el voltaje entre ellas debe ser entre 2.5 y 5 V. (precaución: en las tarjetas de evaluación VREFL está a tierra).

Xtal y Extal sirve para conectar el cristal que determina la frecuencia de operación del sistema, que es la frecuencia del cristal entre 4, se observa que XTAL va en la dirección de salida y EXTAL en la de entrada, esto es porque también puede aplicarse una señal de reloj directamente más que la de un cristal y esto tendrá que hacerse sólo por EXTAL.

E lleva la señal del reloj fase 2 a otros elementos conectados al HC11 que lo requieran (habilitación).

RESET, IRQ y EIRQ son terminales de interrupciones externas al HC11, las 3 actúan en bajo, por RESET puede haber una salida indicando que no se encuentran códigos legales, en IRQ también se puede aplicar un voltaje para programar memorias EEPROM, en XIRQ se puede aplicar el voltaje para programar memorias EPROM.

Las terminales MODA y MODB, en una de sus funciones operan juntas para seleccionar el modo de operación del sistema por circuitería, se tienen 4 posibilidades, para los modos llamados de un solo chip, bootstrap y de prueba. El sistema lee estas terminales cuando se aplica reset iniciando en el modo que indique, después de reset tienen otra función, LIR carga el registro de instrucciones que indica cuando se ejecuta una instrucción y que está pensada para correr programas por pasos y VSTBY que es para dar alimentación a la RAM y que ésta no pierda su información cuando no este presenta VDD.

Las líneas PA0 a PA7 son terminales de E/S del puerto paralelo A y también trabajan para el subsistema temporizador, PA0 a PA2 son sólo de entrada, PA3 a PA6 son sólo de salida y PA7 es bidireccional (PA3 es bidireccional en algunas versiones de HC11).

PB0 a PB7 son terminales de salida del puerto paralelo B; aunque en el modo expandido llevan la parte alta del bus de direcciones.

PC0 a PC7 son terminales del puerto paralelo C, bidireccionales, aunque en el modo expandido llevan multiplexados el bus de datos con la parte baja del BUS de direcciones.

STRA y STRE son terminales de control de los puertos paralelos B y C llamadas líneas Strobo, pero en el modo expandido son la línea R/W lectura/escritura y AS strobo de dirección que permiten mandar al exterior la orden de lectura/escritura y demultiplexar los buses de datos y direcciones.

PD0 a PD5 son terminales bidireccionales del puerto paralelo D, pero también llevan toda la información de datos y control de los puertos serie, SPI (interfaz periférica serie) y SCI (interfaz de comunicación serie), el sincrónico y el asincrónico.

PE0 a PE7 son terminales de entrada del puerto paralelo E y también llevan la entrada al convertidor analógico digital (las versiones de 48 terminales, dobles en línea, omiten las líneas de Pe4 a PE7).

3.2. Diagrama de flujo del problema

Para hacer el programas de simulación del modelo, se ha proporcionado a la computadora los límites de los fuzzy set, entonces la computadora calcula el centro de cada fuzzy set y los empalmes de cada uno de ellos, obteniendo estos datos el ordenador calcula las pendientes. En cambio para programar el microcontrolador se ha decidido, que lo datos de límites nuevos de los fuzzy set de las variables de entrada, los centros de los fuzzy set de la variable de salida y regla sean proporcionado al programa. La tarea del microcontrolador entonces es, leer los datos de los sensores para calcular la fuzzyficación de las entradas capturadas, hacer la inferencia fuzzy, calcular la defuzzyficación y controla los motores del prototipo.

Al analizar los datos que deben de proporcionarse al microcontrolador se observa que, para el caso de la variable X, que tiene 5 fuzzy set, el universo de trabajo va de 0 a 100 ul, al pasar estos números a base dos se puede observar que 100 necesita 7 bits (1100100). En cambio para la variable θ con 9 fuzzy set y un universo de trabajo de 0° a 360° , para representar el número mayor se necesitan 9 bits (101101000).

El microcontrolador tiene la facilidad de manejar 8 bits o 16 bits, pero el manejo de 16 bits implica el manejo de mucha más memoria y mayor tiempo en el proceso de datos. El sistema no tiene una gran precisión de trabajo por ello se ha decidido el manejo de solamente 8 Bits, el mayor número que se puede representar con 8 bits es 255; pero la mitad son números negativos solo quedan entonces 127, para todas las variables se respetaron los valores excepto para ϕc que se normalizó con respecto a 127. Los límites a programar en el microcontrolador son:

Para la variable X.

LE va de H00 a H29, LC va de H13 a H32, CE va de H29 a H39, RC va de H32 a H51 y RI va de H39 a H64.

Para la variable ϕc .

RB que va de HD9 a H00, OD va de HE8 a H08, SD va de H00 a H20, VD va de H0E a H2A, VE va de H20 a H35, VI va de H2A a H47, SI va de H35 a H55, OI va de H47 a H6c, B1 va de H55 a H7F.

Para la variable ϕR

Se sabe que su universo va de -90° a 90° , lo cual implica que se están manejando 180° , que son los que se han utilizado para calcular los límites de los fuzzy set evitando el manejo del complemento a dos. Los fuzzy set quedan como sigue:

PO va de H5A a H00, CR va de H2D a HD3 y NE va de H00 a HA6.

Para los fuzzy set de salida solo se necesitan los valores numéricos del centro porque se ha optado por hacer la defuzzyficación con el método de las cartas únicas.

Los datos anteriores se colocaron en el programa, para el microcontrolador, en las primeras localidades de memoria.

Para las salida los valores en hexadecimal son:

Para NB es HDD, NM es HE9, NS es HF6, ZE es igual a H00, PS es H0A, PM es H17 y PB es H23.

Las reglas ocupan 135 localidades de memoria, también manejan 8 bits. Estos datos se colocaron en las localidades de memoria arriba del fin del programa, porque se llaman las reglas al hacer la suma de los valores asignados a cada fuzzy set. Esto es, para la variable X se tiene que los fuzzy set se representan por unidades y son cinco números, para la variable ϕc los fuzzy set se representan por decenas hasta nueve y la última variable de entrada se representa por centenas solo tres datos. Por ejemplo si al hacer la inferencia difusa se obtiene LC y VD y CR la dirección donde está la regla es la 242. Este número es independiente del grado de membresía.

3.2.1. Lectura de datos de sensores y fuzzyficación

Los datos de los sensores serán generados por una computadora que actuará como sensor. Para lograr que el ordenador actúe como sensor se debe de transmitir las lecturas calculadas por el puerto serie del ordenador y se recibe igualmente por esta interfase, la comunicación debe de ser asincrónica, por que ésta permite mayor distancia a la transmisión.

En un puerto serie los datos se transmiten bit a bit. La Configuración de la mayoría de los puertos serie se basa, en el estándar RS-232 usando un conector de 25 patillas en cada extremo. Sin embargo, una gran cantidad de los puertos serie no emplea todas las señales especificadas en el estándar RS-232 se debe a que el puerto serie originalmente se diseñó para soportar un módem y cuando se usa con otros dispositivos, muchas de estas señales no sirven de nada.

Señal	Abreviatura	No. de terminal
Preparado para enviar	RTS	4
Línea de control	CTS	5
Datos preparados	DSR	6
Datos preparados	DTR	20
Transmisión de datos	Tx	2
Recepción de datos	Rx	3
Tierra	GRD	7

Tabla 8. salida de puerto serial

Formato de la comunicación serie en el mc68hc11.

El formato utilizado en el MC68HC11 es el de no regreso a cero (non return to zero) NZR, como el que se muestra en la figura 30. El formato debe cumplir con las siguientes características:

- a) La línea estará ociosa, teniendo un uno lógico antes de la transmisión ó recepción de datos.
- b) EL bit de comienzo será un cero lógico.
- c) Los datos son transmitidos y recibidos comenzando por el bit menos significativo.
- d) El bit de parada será un uno lógico e indicará el fin de transmisión de un carácter contenido en un paquete, el cual consistirá de: un bit de comienzo, un carácter de 8 ó nueve bits, y un bit de parada.
- e) Una terminación de transmisión ó recepción se llevará a cabo cuando la línea contenga un cero por lo menos por el tiempo completo de un paquete que contenga un carácter.

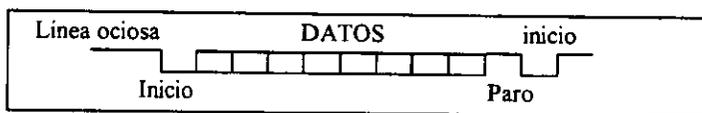


Figura 30. Formato NZR para comunicación serial.

El transmisor del SCI puede producir hileras de caracteres en uno lógico (estado ocioso o hileras de caracteres en cero lógico (estado de ruptura). En adición con la bandera de registro vacío TDRD (transmit data register empty), el SCI también proporciona una bandera de transmisión completa, que puede ser usada en aplicaciones con un módem. El corazón del sistema de transmisión es un registro de corrimiento serial. Usualmente este registro toma los datos del buffer de transmisión. Los datos entran en el Buffer cuando el usuario escribe en el registro (SCDR). Cuando entra en el registro de corrimiento procedente del buffer de transmisión, un cero es agregado en el primer bit del dato, para que opere como bit de comienzo y un bit de uno, es agregado en la última posición para que opere como bit de parada. En el caso previo al inicio a una comunicación serial el registro de corrimiento es llenado con unos para indicar línea ociosa. En caso de que se envíe un comando de ruptura el registro de corrimiento se llena con ceros. El SCI recibe datos del pin PxD, a través de un buffer, la recuperación de los datos se realiza mediante un registro de corrimiento de alta velocidad, el cual opera a 16 veces la velocidad del baud a que se este llevando la comunicación. El corazón del receptor de datos del SCI es un registro de corrimiento, este registro es habilitado por el bit RE que se localiza en el registro SCCR. EL bit M de registro SCCR1 determina la longitud de registro 10 o 11 bits de longitud (8 bits de datos o 9 bits de datos más bits de parada y comienzo). Después que se ha detectado un bit de parada, el dato recibido es transmitido del registro de corrimiento hacia el registro SCCDR, y la bandera de datos completos es colocada en uno (RDRF).

Como se presentó en el capítulo 1, la fuzzyficación, es la transformación de las entradas naturales en entradas fuzzy, proceso que debe hacerse para cada una de las variables. Teniendo la lectura de cada variable se fuzzyficará, dentro de los fuzzy set que se propusieron con anterioridad para la variable que le corresponda solamente. Al hacer la evaluación de todas las variables el menor

número de grados de membresía que se pueden obtener es de 3 y el mayor es de 9. La cantidad de grados de membresía obtenidos da una idea de cuantas reglas deberán de ser analizadas.

Se consideró que el primer paso para la fuzzyficación era determinar el lugar donde cae el valor de la lectura por que si éste está dentro de los extremos del universo de trabajo y el centro del primero y último fuzzy set éste toma el valor de uno como grado de membresía y los fuzzy set laterales no son de interés, solo en esa lectura. En cambio si la lectura cae exactamente en el centro de alguno de los fuzzy set, que no estén en las orillas, éste toma el valor de uno y los fuzzy set laterales toman el valor de cero esto se hace automáticamente y disminuye la cantidad de cálculos encargados al microcontrolador, pero cuando la lectura está fuera de las condiciones anteriores se hace necesario que el microcontrolador haga los cálculos de los grados de membresía. Y en este caso solo dos de los fuzzy set son afectados. El método de cálculo de grado de membresía se desarrolló en el capítulo 1 en el punto 1.4.1 fuzzyficación. El último de los casos posibles es que la lectura esté justo en el centro de alguno de los fuzzy set de las orillas esto provoca que el grado de membresía es uno y el fuzzy set lateral es cero ningún otro grado de membresía es de interés para esa lectura.

Como se explicó en el capítulo uno, se sabe que los grados de membresía están en el intervalo de uno a cero, esta particularidad es un obstáculo para hacer el programa ya que el microcontrolador no puede manejar números tan pequeños, si no se utiliza un algoritmo. Es posible utilizar la instrucción que permite la división de fracciones, pero el resultado sería de 16 bit no signados esto implica que se tienen 65535 números que representan fracciones y por ejemplo para representar el número 0.5 se tendría el número 32767. Si el programa solo se limitara a la fuzzyficación esto no sería problema pero al utilizar estos resultados en la evaluación de reglas éste análisis se complica, para evitar esto, se decidió que los resultados de la fuzzyficación se multiplicarán por 100, esto provoca que el máximo valor que se puede obtener es 100 y el menor es 0 y así para representar 0.5 se tendría como resultado 50. Por esta razón donde se debe de colocar un 1 se escribirá un 100 o en valor hexadecimal 64(h). Con este cambio las siguientes tareas ya no se ven afectadas.

El programa diseñado para el microcontrolador está diseñado para trabajar la fuzzyficación con los datos que están en la dirección 00 a la 0f esto implica que los datos de los límites de los fuzzy set de cada variable se muevan a este lugar, y los resultados se guardan a partir de la dirección 100 a la 10f para la primera variable, para la segunda variable se almacenan en la dirección 200 a la 20f y así sucesivamente hasta la última variable a analizar. El resultado de la fuzzyficación se va guardando junto con el fuzzy set afectado; por ejemplo para la variable X que tiene 5 fuzzy set, si la lectura es 50 se afecta al tercer fuzzy set (ce) que tendrá el grado de membresía como 100 o 64h y los fuzzy set laterales segundo (lc) y cuarto (rc) su grado de membresía es cero.

Como es posible que en algunos casos la lectura tomada provoque que solo uno o dos fuzzy set sean afectados, las localidades de memoria que no tengan nada se les carga con ff que para este programa querrán decir que no se tomen en cuenta estas localidades por que no son de interés. También es posible que por algún error, para alguna variable se le cargue con ff lo cual provoca que el sistema dée como resultado cero en la fuzzyficación

El diagrama de flujo en la figura 31 muestra los pasos para la fuzzyficación.

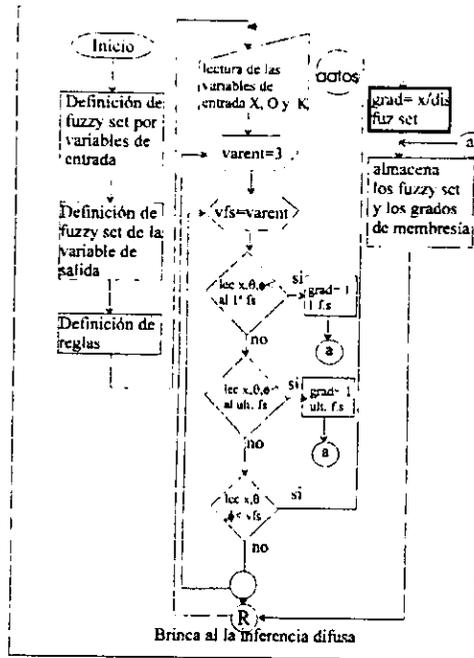


Figura 31 diagrama de fuzzyficación

3.2.2. inferencia difusa

Ya que se conocen las reglas se sabe que la máxima combinación posible es de 27. Como ya se dijo a cada variable se le asoció con números así para X se le asociaron las unidades y como solo son 5 fuzzy set el número más alto al que se le puede asociar una regla es 5. Para ϕc se tienen 9 fuzzy set y se le asociaron las decenas. Y por último para ϕR solo son 3 fuzzy set y se le asociaron las centenas. Así la regla obtenida por cada combinación posible da al sumarse la dirección en memoria donde se guarda la salida.

Por cada combinación solo uno de los grados de membresía es de interés para esta etapa, este es el número menor de todos los fuzzy set involucrados en esa regla. Para obtener éste se comparan los dos primeros grados de miembro de la regla estudiada se almacena el menor, éste se compara con el siguiente grado de membresía de la regla y se guarda el menor de estos dos junto con la regla involucrada. Esto se hace con todas las combinaciones existentes.

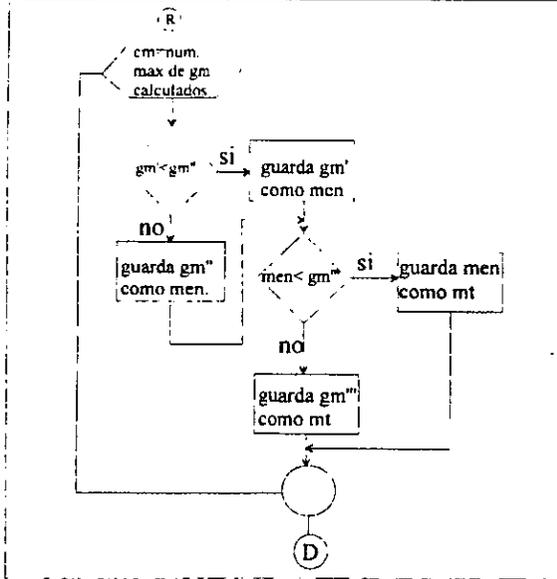


Figura 32. Diagrama de inferencia difusa cálculo de mínimos

Al finalizar lo anterior se puede observar que existen, en algunos casos, algunas salidas que se repiten con diferente valor o que algunas son zeros. Para determinar cuales son los valores de las salidas que serán defuzzyficadas se comparan los fuzzy set de salida repetida y se busca el mayor. Al hacer la comparación se eliminan los grados de membresía menores de las salidas que se repiten, guardando solo los otros grados de miembro. Con estos resultados se puede calcular las salidas naturales.

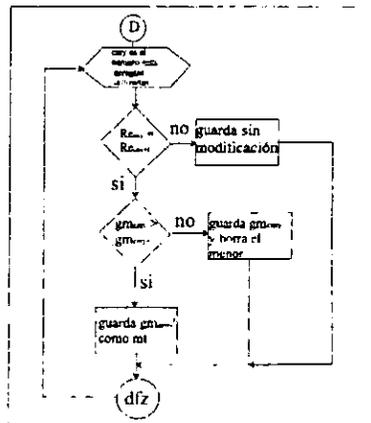


Figura 33 Diagrama de flujo de cálculo de máximos.

3.2.3. Defuzzyficación

Ya que se han obtenido los grados de membresía junto con las reglas que operan en la lectura que se esta estudiando, se debe de defuzzyficar para obtener las salidas naturales. Esta se calcula al multiplicar el grado de membresía por el valor numérico del centro del fuzzy set. Cada uno de los resultados que anteriormente se calcularon se suman, el resultado se divide entre la suma de todos los grados de membresía que fueron afectados por los resultados obtenidos. Este resultado representa el grado del ángulo que se mueve la dirección del camión. Normalmente este resultado tiene fracciones, para el mayor de paso que se tiene no se puede obtener este ángulo exactamente. El microcontrolador hará girar el motor en ángulo aproximado al calculado.

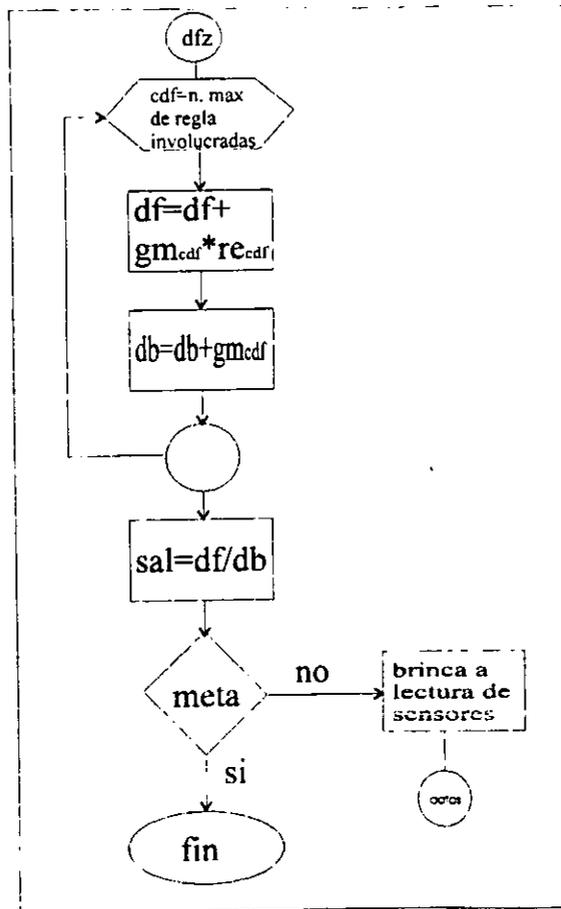


Figura 34. Diagrama de defuzzyficación

3.2.3.1. Salida hacia el motor de pasos

En esta parte se detalla la salida del sistema hacia el motor. Se sabe que existe una secuencia de salida para el giro del motor de pasos, el inicio de esta secuencia se coloca en la dirección \$40, la secuencia completa se encuentra en la dirección de memoria \$40 a \$46. La secuencia de giro contrario va en la dirección \$50 a \$56.

Entonces si se desea cambiar el sentido del giro basta con dar una u otra dirección. La secuencia permite giros de 11° mínimo y se muestra en la tabla 9:

Número	izquierda	Derecha
0	Ch	3h
1	8h	1h
2	9h	9h
3	1h	8h
4	2h	4h
5	4h	2h
6	3h	ch

Tabla 9. secuencia de giro del motor de paso

Para empezar la secuencia de giro se inicializa un contador desde cero, el acumulador b se carga con la dirección de giro de la forma \$aaaa, luego se incrementa x, el acumulador se envía a la dirección \$1004 que corresponde al puerto B del microcontrolador. Debido a la velocidad del microcontrolador se envía a una subrutina que produce un retardo de tiempo que permite que los relevadores funcionen correctamente. Luego el contador se almacena en el acumulador A donde se compara con el dato #b que es el número de giros que se necesita para alcanzar el grado calculado por la defuzzificación. Si el contador es igual al dato almacenado, se finaliza la secuencia de giro y entonces se apagan los relés, pero si aun no se alcanza el valor del dato se incrementa el contador y se continúa con la secuencia.

Capítulo 4. Construcción del prototipo y programación del sistema

4.1. Descripción física del camión

En este trabajo de investigación al tener los datos teóricos ya desarrollados y con el fin de implementar el controlador de forma real, se vuelve necesario la construcción de un camión, pero éste no requiere ser perfecto sino solo parecido a un camión real, se le ha dotado con lo mínimo para un funcionamiento adecuado. Los elementos son la estructura del camión, el motor de tracción, un motor de paso para controlar el giro de la dirección, los circuitos de salida para controlar los motores, los circuitos de los sensores y el microcontrolador. Como primer punto se describe la estructura del modelo construido.

El prototipo está hecho de un perfil de fierro y lámina, las ruedas son de plástico y tiene la forma que muestra la figura 35 pesa dos kg. y mide 28 cm de largo, 16.5 cm de ancho y 20 cm de alto.

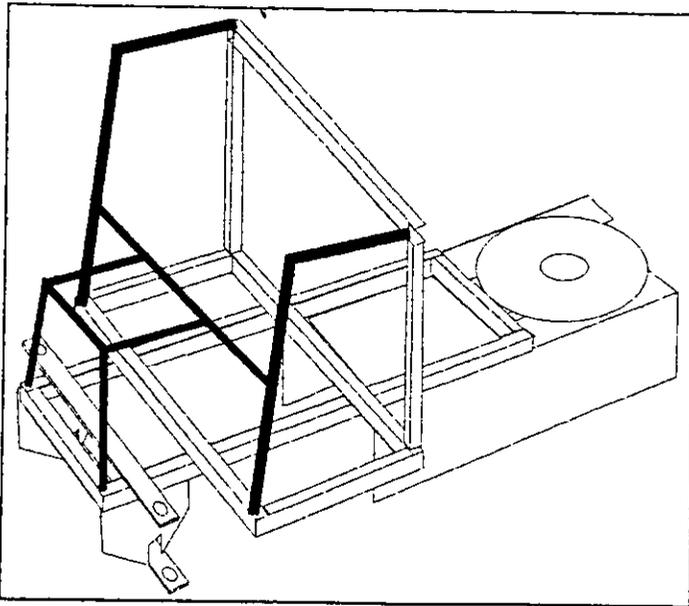


Figura 35. Modelo hecho en lámina

Los ejes de este prototipo están hechos de aluminio, excepto los ejes donde están empotrados los engranes de transmisión, los cuales son de fierro, empotrados en pequeños baleros.

Para las ruedas de la parte trasera, no se colocó un diferencial porque giran libres en cualquier dirección. En las orillas de los ejes se colocaron prisioneros que impiden la salida de las ruedas.

Atrás se colocó una base en la cual se monta el remolque. Tiene además una cabina que sirve para que se coloquen los circuitos de control y de potencia.

También es posible que si no se desea utilizar el remolque puede colocarse una batería para alimentar a los motores y a los circuitos, en la base destinada al remolque.

El diferencial está hecho de un cilindro de aluminio, dentro del cual hay dos baleros pequeños que solo giran en un sentido. El diferencial está conectado a un engrane, como se muestra en la figura 36.

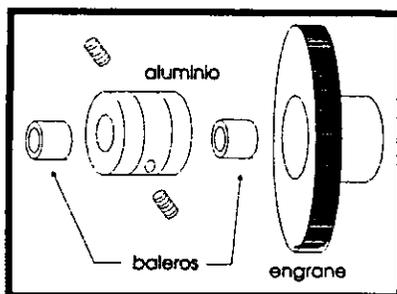


Figura 36. Diferencial

Cuando existe el movimiento hacia atrás o adelante los baleros giran en la misma dirección en la que está girando el diferencial cuando se hace girar el vehículo, uno de los baleros se detiene y el otro gira, el orden de los baleros se invierte al hacer girar el vehículo en dirección contraria.

La conexión del diferencial y el motor se muestra en la figura 37. El movimiento se transmite al engrane B que está libre, éste hace girar el eje con los engranes A y B, que no hace girar ninguna de las ruedas, el engrane A se conecta al engrane montado en el diferencial y éste hace girar las ruedas.

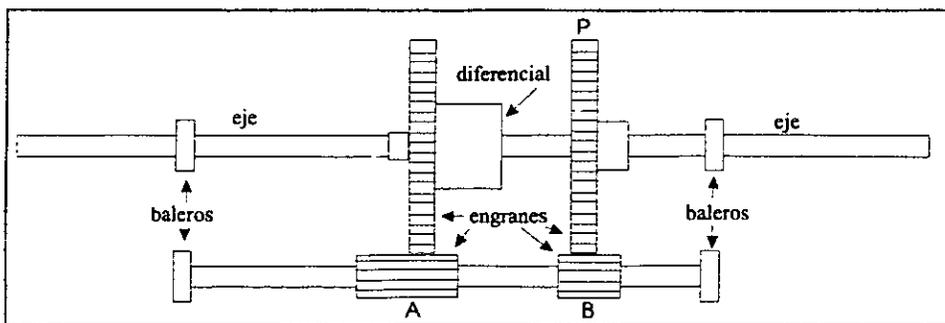


Figura 37. Montaje del diferencial y la tracción

El impulso lo da un motor de corriente directa, que trabaja con un voltaje de 12 V y 3.5 W. que además tiene un embrague controlado eléctricamente. El engrane del motor está conectado a un

engrane mayor a través de una banda, y éste último engrane esta acoplado a un engrane menor como se ve en la figura 38.

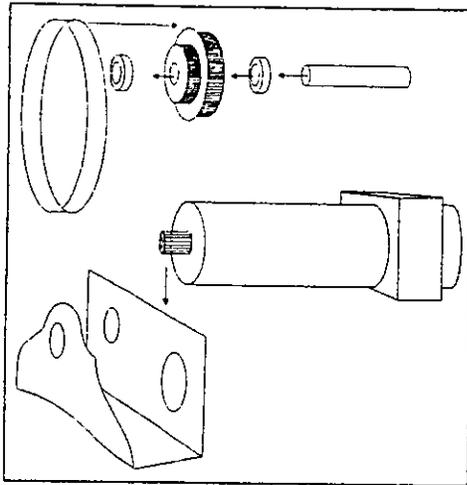


Figura 38. Motor del prototipo

La dirección está controlada por un motor más pequeño de pasos, el cual tiene un engrane chico que mueve a un engrane más grande, que hace girar una palanca y así mueve los ejes de las ruedas. El motor trabaja con 12 V y 4 W.

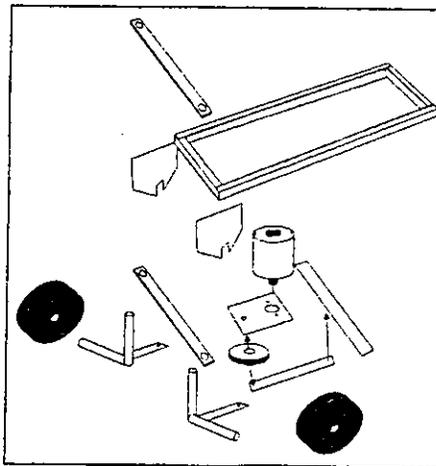


Figura 39. Motor para controlar la dirección

El prototipo completamente armado se muestra en la figura 40.

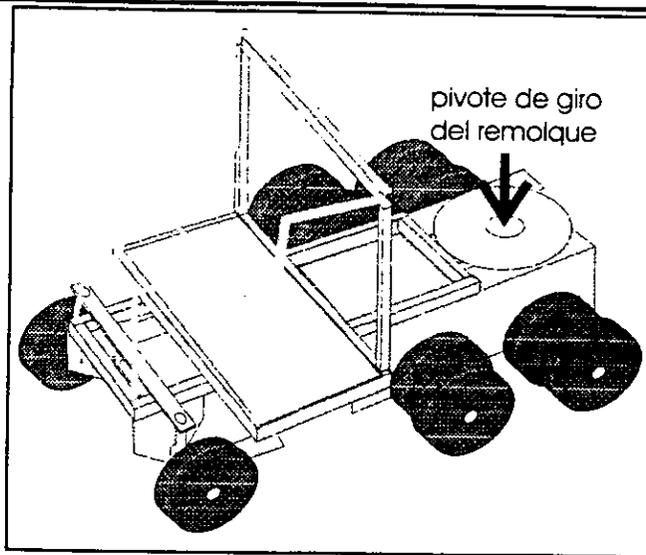


Figura 40. Prototipo completo sin circuitos

El remolque está hecho del mismo material, no tiene ninguna característica mecánica especial, pero la caja puede servir para colocar baterías, circuitos e incluso sensores necesarios al camión.

4.2. Características de los circuitos

Como se sabe el microcontrolador no puede ser conectado a los motores directamente, debido a que la corriente y el voltaje de salida son mucho menores a los que se necesitan para hacer funcionar cualquier motor, entonces se necesita un circuito capaz de aislar la circuitería de salida del microcontrolador para que no se dañe y que aporte la energía necesaria para el funcionamiento del motor. El circuito integrado buffer permite aislar el circuito de las salidas del microcontrolador. Además el circuito no consume directamente de los integrados de salida. El circuito "buffer" aporta capacidad para alimentar sin cargar las salidas.

El buffer 74LS245 es un transmisor/receptor de ocho vías (octal bus transmitter/receiver) para 8 líneas asíncronas, la comunicación es bidireccional y es controlada por una de las terminales o también puede aislarse a través de otra terminal. La figura 41 muestra las terminales del circuito.

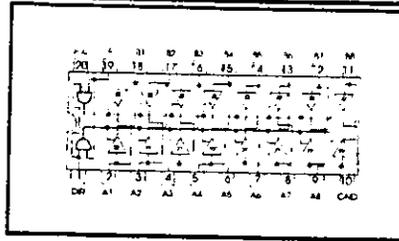


Figura 41 Integrado 74LS245

Entrada	DIR	Salida
E	DIR	
0	0	Datos del bus B al bus A
0	1	Datos del bus A al bus B
1	X	Alta impedancia

Tabla 10. tabla de verdad de 74LS245

Las terminales 2 a 9 se conectan a la salida del puerto B del microcontrolador, las terminales 1 y 20 se conectan a 5V, y las terminales 19 y 10 se conectan a tierra, las salidas están en las terminales 11 a 18.

Como complemento a este circuito se hace necesario un circuito que aporte la potencia necesaria para los motores que requieren de mayor potencia.

Un circuito controlador puede suministrar la potencia necesaria para controlar hasta 8 relés a partir de las señales del puerto de microcontrolador. Sustituye a los circuitos de control del relés hechos a base de transistores que obligan a utilizar más componentes. El circuito se muestra en la figura 42.

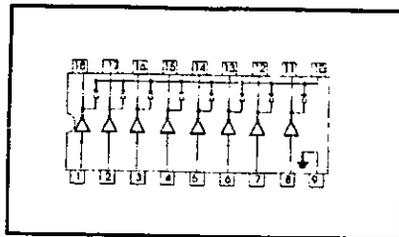


Figura 42. Integrado ULN 2803

Las terminales que se conectan para la entrada van de la 1 a 8, las salidas van de la terminal 11 a 18, la terminal 10 se conecta a la fuente de 5V y la terminal 9 se conecta a tierra.

El relé tiene la siguiente configuración figura 43.

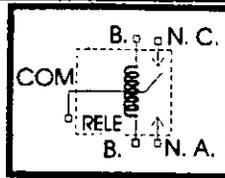


Figura 43. Diagrama de relé

N. A. - Normalmente abierto.

N. C.- Normalmente cerrado

B.- Bobina

COM.- Común

Voltaje.- 5 V c. d.

La resistencia de la bobina es de 70 ohms

La corriente nominal es de 72 mA.

Los valores en Com, N. A. y N. C. es de 2A a un voltaje de 125 V c. a. Los motores consume aproximadamente 300 mA cada uno.

El circuito completo, que controla el motor de tracción es el que se muestra en la figura 44. Al tener un 1 en la terminal 18 y un 0 en la terminal 17 el motor hace avanzar el camión hacia adelante, si se tiene el caso contrario 1 en terminal 17 y 0 en la terminal 18 el camión avanza hacia atrás, cuando en ambas terminales (17 y 18) se tienen unos o ceros el motor no gira.

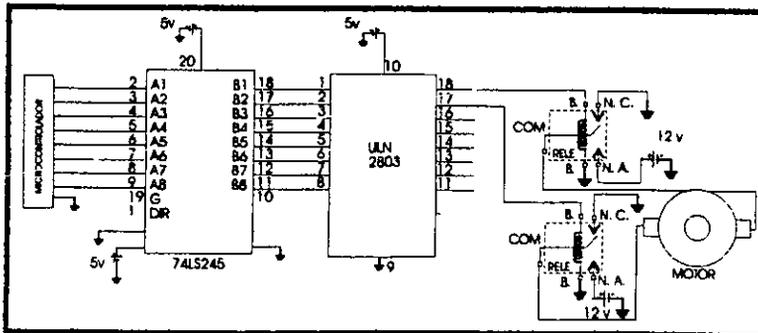


Figura 44. Circuito de control de motor de tracción

El motor de paso que se usa para el camión tiene la siguiente secuencia para giro completo:

izquierda	derecha
110000	001100
100000	000100
100100	100100
000100	100000
001000	010000
010000	001000
001100	110000

Tabla 11. secuencia de giro completo para el motor de pasos

Se recorrieron las salidas hacia la izquierda, los datos originales se ven en la tabla 9, para utilizar esos números para el funcionamiento del motor de tracción, la conexión al microcontrolador se ha hecho a través de relé. Se han tomado las salidas del circuito ULN 2803 de la 16 a la 13.

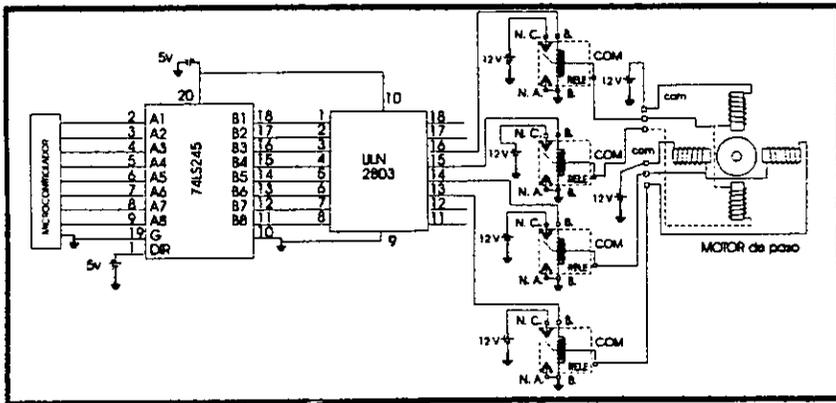


Figura 45. Circuito de control del motor de paso

4.3. Los sensores.

El sistema como se dijo en el tema 1 tiene 3 variables que son necesarias para garantizar que el sistema pueda generar las salidas fuzzy correctas, estas variables son:

La distancia en el eje de las "X".

El ángulo ϕ_c que va de 0° a 360° , y

El ángulo ϕ_R que se mide con respecto al pivote de giro del remolque.

Se hicieron varias pruebas con el fin de encontrar un sensor sencillo que pudiera ser capaz de medir alguno de los datos que se deben de sensar, y se llegó a la conclusión de que el desarrollo y perfeccionamiento de los sensores que se necesitan implica el desarrollo de una tesis completa con este fin, por ello se decidió usar una computadora que proporcionará al sistema los valores

El microcontrolador MC68HC11 es utilizado ampliamente en diversas aplicaciones como por ejemplo: En sistemas de inyección electrónica de automóviles, equipos de monitoreo atmosférico, alarmas, equipos biomédicos, control de motores eléctricos, conmutadores telefónicos, tacómetros, velocímetros etc.

La tarjeta MICRO-M-6811 ha sido diseñada para servir como un sistema de desarrollo para el microcontrolador MC68HC11.

Se puede transmitir programas mediante comunicación serial a la tarjeta MICRO-M-6811, y ejecutar el programa, depurarlo, o realizar alguna aplicación final de algún proyecto.

La tarjeta tiene 3 etapas principales

- 1.- Fuente de Alimentación, que está diseñada para trabajar de 8 a 20 volts de C.D.
- 2.- Comunicación Serial. La tarjeta cuenta con una interface de comunicación serial para poderse comunicar con una PC a través del puerto serial con tipo de conector DB9 o DB 25. Del protocolo de comunicación RS232, solo se toman 3 pines que son: el RX, RT y la tierra.
- 3.- Conector de entrada salida. Este conector permite al usuario conectar diversos dispositivos al microcontrolador, como por ejemplo teclado, pantalla etc.

4.4.1. Diagrama electrónico de la tarjeta MICRO 6811, conectores y puentes.

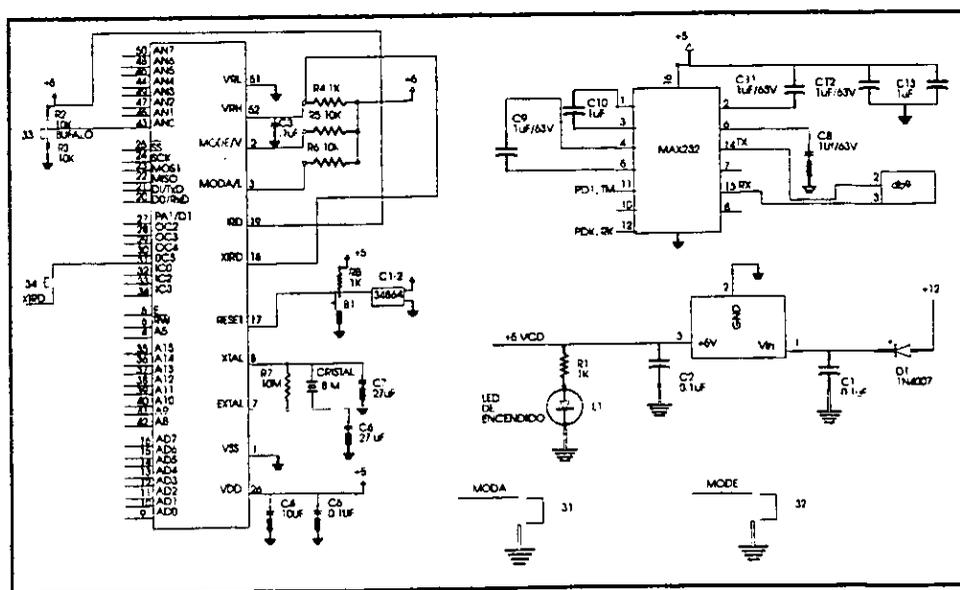


Figura 46. Diagrama electrónico de la tarjeta micro M 6811.

J1	J2	MODO DEL MICROCONTROLADOR
Instalado	Removido	Modo sencillo "BUFFALO"
Removido	Removido	Modo Expandido
Instalado	Instalado	Modo de arranque especial "PCBUG"
Removido	Instalado	Modo de prueba en fábrica.

Tabla 12. Puentes

Conector J3

EE  J3. Se ejecuta el programa localizado en la memoria EEPROM.

Conector J3

EE  J3. Se ejecuta el programa BUFFALO.

Nota.- Este conector no tiene efecto si se arranca en el modo de arranque especial.

Conector J4.



El conector J4 es usado para conectar la terminal XIRQ a la terminal PA3/OC5 lo cual es requerido por el programa BUFFALO. En aplicaciones que utilicen la terminal PA3/OC5, el conector debe ser removido. En este caso algunos comandos del programa BUFFALO serán inservibles.

Conector J5

El conector J5 de la tarjeta proporciona el modo de conexión para entradas y salidas a las terminales del microcontrolador, además de proporcionar VCC y tierra. En la tabla 13 se muestra la distribución de las terminales de salida y entrada de la tarjeta de desarrollo.

GND 60	OO	59 GND
VCC 58	OO	57 VCC
LIBRE 56	OO	55 LIBRE
LIBRE54	OO	53 LIBRE
VRH 52	OO	52 VRL
PE7 50	OO	49 PE 3
PE6 48	OO	47 PE2
PE5 46	OO	45 PE1
PE4 44	OO	43 PE0
PB0/A8 42	OO	41 PB1/A9
PB2/A10 40	OO	39 PB3/A11
PB4/A12 38	OO	37 PB5/A13
PB6/A14 36	OO	35 PB7/A15
PA0/IC3 34	OO	33 PA1/IC2
PA2/IC1 32	OO	31 PA3/OC5

PA4/OC4 30	OO	29 PA5/OC3
PA6/OC2 28	OO	27 PA7/OC1
VCC 26	OO	25 PD5/SS*
PD4/SCK 24	OO	23 PD3/MOSI
PD2/MISO 22	OO	21 PD1/TXD
PD0/RXD 20	OO	19 IRQ*
XIRQ*	OO	17 RESET*
PC7/AD7 16	OO	15 PC6/AD6
PC5/AD5 14	OO	13 PC4/AD4
PC3/AD3 12	OO	11 PC2/AD2
PC1/AD1 10	OO	9 PC0/AD0
XTAL 8	OO	7 EXTAL
STRB/R/W* 6	OO	5 E
STRA/AS 4	OO	3 MODA/LIR*
GND 2	OO	1 GND

Tabla 13. Distribución de las terminales en el conector J5

Conector J6

Conector de alimentación, se alimenta con una fuente de 8 a 20 volt de C.D. y da una salida de 5V necesaria para el funcionamiento del microcontrolador e incluso puede alimentar al circuito de control de salidas.



Vcc Tierra

Botón de reset. Al oprimir el botón de Reset se reiniciará al MC68HC11, dependiendo del modo en que se encuentren J1 y J2 tomará el modo de operación en función de lo explicado en la tabla 12.

4.5. Programa para el microcontrolador

```

org $0010 ; origen de datos
db $00,$13,$2a,$32,$3a,$51,$64,$ff,$ff,$ff,$ff,$ff,$00,$01,$07,$ff;dato de x
db $d6,$e8,$00,$0e,$20,$2a,$35,$47,$55,$6c,$7f,$ff,$00,$10,$0b,$ff;dato de o
db $a6,$d3,$00,$2d,$5a,$ff,$ff,$ff,$ff,$ff,$ff,$01,$00,$05,$ff;datos de k
db $0c,$08,$09,$01,$02,$04,$03,$00,$00,$00,$00,$00,$00,$00,$00,$00
db $03,$01,$09,$08,$04,$02,$0c,$00,$00,$00,$00,$00,$00,$00,$00,$00
cont ds $00
org $3000
ctl ds $01 ;aparta 1 byte para el contador de fuzzy set de la variables
dis ds $02 ;aparta 2 byte para guardar la distancia entre fuzzy set
dli ds $02 ;aparta 2 byte para guardar la distancia lado izquierdo
dld ds $02 ;aparta 2 byte para guardar la distancia lado derecho
lec ds $01 ;lectura
ctc ds $02 ;intercambio de resultados
v1 ds $02 ;aparta 2 byte, dirección de datos a comparar
v2 ds $02 ;aparta 2 byte, dirección a guardar
ctt ds $01 ;contador de variables

```


* tesis diseño de un controlador difuso aplicado al manejo automático de *
 * estacionamiento de un camión con remolque implementado con el *
 * microcontrolador 68hc11 *

```
org $2000 ;origen del programa
clr ct1 ;limpia contador
clr lec ;limpia lec
clr ctc ;limpia
```

```
ldaa $090 ;carga con dato de lectura de x
staa lec ;guarda en lec
ldd #$0010 ;carga en d, especifica la dir. a cambiar para comparar
std v1 ;guarda en v1
ldd #$0110 ;carga en d, especifica la dir. a cambiar para guardar
std v2 ;guarda en v2
jmp fuz ;brinca a fuz
```

seg ldaa \$091 ;guarda el acumulador la variable de entrada

```
staa lec ;guarda en lec
ldd #$0020 ;carga de con 0020 como dato
std v1 ;guarda en v1
ldd #$0120 ;carga 0120 dato
std v2 ;guarda en v1
jmp fuz ;brinca a fuz
```

ter ldaa \$092 ;carga lectura

```
staa lec ;guarda en lec
ldd #$0030 ;dato 0030
std v1 ;guarda en v1
ldd #$0130 ;dato 0130
std v2 ;guarda en v2
jmp fuz ;brinca a fuz
```

** carga los datos de cada variable para calcular la fuzzyficación*****

```
fuz ldx v1 ;carga x con las direcciones de la variable a analizar
ldy #$0000 ;carga y con la dirección 0000
ldv ldaa 0,x ;carga el acumulador a con los datos de las dir prop por x
staa 0,y ;descarga el acumulador en las dir prop por y
inx ;incrementa x
iny ;incrementa y
cpy #$000f ;compara el indice y con el incremento
beq carg ;brinca si es = a cero
bra ldv ;brinca a ldv lectura de datos de variable
carg clr ct1 ;inicia contador con 00
```

***** lectura igual a centros de fuzzy set *****

```
ldx #$0000 ;carga con 00 a x
com ldab lec ;cargar data a comparar de lectura de dato
cmpl $00,x ;compara b con el dato de la dirección 00+x
beq gmfs ;brinca si es menor
inx ;incrementa x
inc ct1 ;incrementa el contador en 1
ldaa ct1 ;carga el acumulador a con la variable ct1
cmpa $0e ;compara ct1 con el dato de la dir 07
beq sig ;si es = brinca a la otra zona de comparación
bra com ;brinca a com
```

***** calcula grado de membrestia *****

```
gmfs ldd #$0064 ;carga b con l dato
```

```

std $104 ;guarda grado de membresía
stx $106 ;guarda x
ldaa ct1 ;carga el contador en a
cmpa #000 ;compara si a y el contador son iguales a cero
beq lii ;si es igual a cero brinca a lii
inc ct1 ;incrementa el contador
ldaa ct1 ;carga el dato del contador en a
cmpa $0e ;compara con el dato de la dir 0e
beq lif ;si es igual brinca a lif
ldd #0000 ;carga a con 00
std $100 ;guarda en la dirección 102, grado de membresía
std $108 ;guarda en la dirección 114
dex ;decremento de x
stx $102 ;guarda en 102
inx ;incrementa
inx ;incrementa
stx $10a ;guarda x
jmp reg ;brinca a reglas
lii inx ;incrementa x
bra cmm ;brinca a cmm
lif dex ;decremento de x
bra cmm ;brinca a cmm
cmm ldd #$fff ;carga a con ff
std $100 ;y guarda en la dir 102
std $108 ;guarda
stx $106 ;guarda
dex ;decremento de x
stx $102 ;guarda x
inx ;incrementa x
inx ;incrementa x
stx $10a ;guarda x
jmp reg ;brinca a reglas
***** lectura diferentes a centros de fuzzy set *****
sig clr dis ;limpia dis
clr dli ;limpia dli
clr dld ;limpia dld
clr ct1 ;limpia ct1
ldx #0000 ;carga x con 000
cat ldab lec ;dato leído
cmpb $00,x ;compara b con el dato de la dirección 00+x
bmi cal ;brinca si b es menor
inx ;incrementa x
inc ct1 ;incrementa ct1
bra cat ;brinca a col
***** calcula grado de membresía *****
cal ldaa ct1 ;carga a con ct1
cmpa #01 ;compara
beq liid ;si es igual a cero brinca a liid
inc ct1 ;incrementa ct1
ldaa ct1 ;carga el acumulador a con ct1
cmpa $0e ;compara
beq liif ;si es igual brinca a liid
ldaa $00,x ;carga el acumulador a con el dato de la dir. 00+x
dex ;decremento de x

```

ldab \$00,x :carga el acumulador b con el dato de la dir. 00+x-1
 sba :resta para conocer la distancia
 staa \$200 :guarda en memoria distancia total
 ldab \$200 :carga el acumulador b con el dato de \$200
 ldaa #\$00 :carga el acumulador a con 00
 std dis :guarda el acumulador d en dis
 inx :incrementa x
 ldaa \$00,x :carga el acumulador a con la dir 00.x
 ldab lec :carga el acumulador con el dato leído
 sba :resta, distancia de lectura al limite derecho
 ldab #\$64 :carga b con 64 dato
 mul :multiplica a por b
 std did :el resultado se guarda en did
 stx \$106 :carga x con el dato de la dirección 106
 dex :decremento de x
 ldaa lec :carga acumulador a con lectura
 ldab \$00,x :carga acumulador b con limite izquierdo
 sba :resta, distancia
 ldab #\$64 :carga el acumulador b con 64 dato
 mul :multiplica
 std dli :guarda el resultado en dli
 stx \$102 :carga x con el dato de 102
 ldx dis :carga registro x con la distancia
 ldd dld :carga el doble acumulador con distancia derecha
 idiv :divide fracciones
 stx \$104 :resultado de división, grado de membresia
 ldx dis :carga registro x con distancia total
 ldd dli :carga el doble acumulador con la distancia izquierda
 idiv :división fracciones
 stx \$100 :resultado de división, grado de membresia
 ldd #\$ffff :carga el doble acumulado con datos de la dirección 0d
 std \$108 :último grado de membresia
 std \$10a :carga el doble acumulador con el dato de 10a
 jmp reg :brinca a reg
 liid ldd #\$ffff :carga el acumulador b con ff
 std \$100 :guarda el doble acumulador en 100 y 101
 dex :decremento de x
 stx \$102 :carga x con el dato de la dirección 102
 ldd #\$0000 :carga el doble acumulador con 000
 std \$108 :guarda en 108
 inx :incrementa x
 inx :incrementa x
 stx \$10a :y guarda en la dirección 10a
 ldd #\$0064 :carga el acumulador b con 01
 std \$104 :guarda el doble acumulador en 104 y 105
 dex :decremento de x
 stx \$106 :guarda x en 106 y 107
 jmp reg :brinca a reg
 liif ldd #\$0000 :carga el doble acumulador con 0000
 std \$100 :y guarda en 100
 dex :decremento de x
 dex :decremento de x
 stx \$102 :y guarda en 102 u 103
 ldd #\$ffff :carga el acumulador d con ffff

El desarrollo de esta tesis permitió construir un prototipo que puede demostrar una de las posibles aplicaciones prácticas de la teoría de la lógica difusa. Demostrando con ello que es posible la construcción de sistemas de control complejos, sin la necesidad de utilizar los medios de control tradicional, basando el diseño tan solo en el sentido común del operador de un camión. Si se observa en el capítulo uno, no existen ecuaciones complejas que modelen el desplazamiento del camión; de hecho es posible reducir este capítulo a un resumen de la experiencia de algún chofer de camión y con ello comenzar a programar el microcontrolador.

Para desarrollar este proyecto, ya que se definieron los conceptos de la lógica difusa, se diseñó un programa que simulará el funcionamiento del sistema, éste presentó resultados muy cercanos a los deseados; entonces, se hicieron algunas modificaciones a algunas reglas que ya se habían planteado, ganando con ello un mejor resultado. En resumen, los resultados obtenidos en la simulación del sistema, trabaja de acuerdo con lo deseado, incluso pueden agregarse otras condiciones, más variables que definan con mayor claridad la ubicación del camión, y éste trabajará bien; pero hay que recordar que es la misma computadora la que está proporcionando los datos de posición del sistema y los está tomando de forma directa de los resultados de los cálculos realizados obtenidos para la graficación del modelo, entonces se pueden esperar resultados excelentes.

Una vez que los datos de la simulación fueron los deseados se pasó a la construcción del camión; pero éste presentó algunas dificultades, ya que el prototipo no está montado sobre un carro a escala comercial, sino que se hizo un modelo, la razón por lo cual se hizo esto fué que en los camiones a escala, que se estudiaron, no se podía colocar los circuitos, y por otro lado otros modelos eran muy caros, fué cuando se decidió construir un camión a escala en función de las necesidades de espacio para los circuitos (microcontrolador y circuito de salida), los detalles de la construcción del vehículo se pueden ver en el capítulo cuatro, en particular uno de los problemas que presentó mayor complejidad fué el poder hacer funcionar el camión, para que éste funcionara se desarrolló un juego de engranes que pudiera transmitir el movimiento del motor a las ruedas, este diseño no parece ser muy adecuado, aunque puede mover un peso muerto de 9 Kg., gracias al uso de baleros, el motor que se utilizó para mover el camión es uno que se consiguió en un puesto que vende motores de máquinas deshuesadas lavadoras, fuentes de computadoras y otras que no conozco, por ello afirmo que este motor no está diseñado para dar el impulso a un carro a escala, de hecho los motores para carros a escala son más grandes, trabajan a 5V y consumen mayor amperaje, es por esto que se decidió utilizar el motor de 12V.

El siguiente problema que se presentó fué que la fuerza mecánica que proporciona el motor no se trasmite en el momento de dar vuelta hacia cualquier lado, esto hace necesario, el uso de un diferencial, el tamaño de éste es muy pequeño como para hacer uno, entonces se hizo necesario adaptar uno de un carro a escala de control remoto, este acoplamiento no es perfecto y produce cierta dificultad en el movimiento del camión, a esto hay que agregar que el sistema de dirección tampoco es adecuado, ya que la transmisión del giro tampoco se pudo lograr que fuera bueno ya que se acoplaron engranes que aunque están bien hechos estos no tienen el sostén adecuado y provoca que se desajusten. Todos estos son problemas mecánicos.

Otro de los limitantes, al tratar de pasar este modelo a un sistema real, fué la necesidad de desarrollar sensores que permitan cuantificar la ubicación del sistema en movimiento esto implicaría el desarrollo de toda una tesis completa que se encargue del desarrollo de dichos sensores.

Las características que tendría este sistema son las siguientes:

- ◆ Debe de comunicarse con el microcontrolador en tiempo real.
- ◆ Debe de ser capaz de identificar el ángulo del carro y de la caja.
- ◆ Debe de ser capaz de medir la distancia del vehículo y los límites del estacionamiento.
- ◆ Debe de poder reconocer la meta hacia donde se dirige.

En cuanto al funcionamiento del microcontrolador, fué necesario que éste no tuviera precargado el programa buffalo, ya que el programa diseñado para esta tesis es más grande que la capacidad de memoria del microcontrolador de uso normal.

Al evaluar el funcionamiento del microcontrolador, éste presenta un buen funcionamiento, lo más complejo es el tomar a la computadora como un sensor, ya que con los únicos datos que cuenta es con el ángulo de giro de la dirección que mide el potenciómetro y una velocidad aproximada, el ángulo del remolque, dado que no se conoce con certeza los demás datos, no permite el control del camión de una buena forma. El sistema funciona mejor sin el remolque.

Lo que en particular he aprendido es que no siempre los métodos tradicionales de solución de problemas son los más adecuados para el manejo de un problema; de hecho gran parte de la física moderna poco a poco se ha dado cuenta que la naturaleza se comporta de forma azarosa, sobre todo en la física de partículas. Una de las cosas sorprendentes con el uso de esta lógica es que abre el camino para la comprensión de fenómeno de mayor complejidad, como es posible determinar una aproximación de resultados en sistemas que no son estables.

Concluyendo, en este trabajo se construyó el camión y la caja de carga; pero debido a que fue poco fácil el controlar el remolque por la falta de sensores adecuados se terminó solo el camión. En él se colocaron los circuitos de salida a los motores y un sensor; los dos motores y dos baterías de 12V 1.2 A que le permiten cierta autonomía que no es completa, debido a que el circuito necesita permanecer conectado al puerto serial de una computadora. En las pruebas se tiene que bajar la velocidad del camión y se tiene que conocer las dimensiones del lugar donde se hace la prueba. Los resultados con el camión se acercan al resultado esperado sin ser buenos, una de las posibles razones de esto es que la dirección no se pudo alinear adecuadamente y esto produce errores en el sistema. Sin embargo son resultados que permiten que valga la pena, que el camión sea perfeccionado, en la parte mecánica, área que se deja para otra tesis de esa materia.

Apéndice A: Programa de monitoreo (buffalo)

El programa con el que se puede monitorear el microcontrolador es el llamado BUFFALO (bit user friendly aid to logical operations), que permite la comunicación del sistema con otros elementos por la interfaz de comunicación serie (SCI) y en particular con el puerto serie de una computadora PC bajo la norma RS-232 c.

El programa monitor consiste de 5 partes que son:

Inicialización, interpretación de comandos, rutinas de E/S, subrutinas utilizables y tablas de comandos. En este apéndice sólo se presenta la información necesaria para la inicialización y operación de la tarjeta.

Este programa emplea las localidades RAM entre \$0047 y \$00FF, que debe evitarse su uso.

Comandos del monitor.

ASM [<dirección>]	Ensamblador/desensamblador.
BF <dir.1> <dir.2> <dato>	Llena bloque de memoria con datos.
BR [-] [<dirección>]	Fija puntos de ruptura del programa.
Bulk	Borra en masa la EEPROM.
Bulkall	Borra la EEPROM + reg. Config.
Call [<dirección>]	Ejecuta subrutina.
G [<dirección>]	Ejecuta programa.
HELP	Desplega comandos del monitor
LOAD <T>	Carga en el microcontrolador por el SCI programas en formato S (S19)
MD [<dir.1> [<dir.2>]]	Despliega memoria en terminal.
MM [<dirección>]	Modifica memoria.
Move <dir.1> <dir.2> [<destino>]	Mueve memoria a nueva localidad.
P	Continúa de punto de ruptura
RM [p, x, y, a, b, c, s]	Modifica registros.
Stopat <dirección>	Parar en dirección.
T [<n>]	Corre por paso (\$1-\$FF).
Verify <T>	Compara memoria con programa cargado en PC

El paréntesis [] indica opcional.

ASM [<dirección>]; dirección es la de inicio de la operación de ensamblador, si no se da dirección, por omisión toma la de inicio de la ram. Permite trabajar con un ensamblador/editor interactivo, donde cada línea fuente es convertida a código de máquina y es almacenada en memoria, reescribiendo los datos que hubiera. Los códigos de operación ilegales son desplegados como ILLOP.

Las reglas de sintaxis más importantes son:

1. Todos los valores numéricos son considerados en hexadecimal.
2. Los operados deben ser separados por uno o más espacios.
3. Cualquier carácter después de un código válido y operando asociados es considerado comentario.

ESTA TESIS NO SALE
DE LA BIBLIOTECA

Apéndice B: Programa IASM11

Este programa de ejecución en computadora se corre tecleando IASN11.

Permite editar un programa en ensamblador, hace la comunicación por el puerto serie de la pc con un microcontrolador HC11, enlazándose con el programa BUFALO, y enviando programas que fueron grabados en formato S.19 del disco al sistema mínimo.

Operación

Al entrar muestra un programa en eNsamblador de nombre NONAME.

Se puede seleccionar un bloque (que podría ser el programa NONAME) con (control) KB al inicio y (control) KK al final, ese bloque puede:

Borrarse con (control) KY.

Imprimir con (control) KP.

Copiarse con (control) KC.

Cambiar nombre con (control) KN.

Si se borró el programa y se quiere EDITAR uno nuevo debe hacerse con el formato.

ETIQUETAS ENSAMBLADOR COMENTARIOS

Etiquetas, debe comenzar con letra, por lo que la primera columna no puede estar en blanco, hay que empezar en la primera columna, no se permiten espacios en las etiquetas.

Se debe de dejar un espacio para la primera columna de ensamblador y escribe una instrucción en lenguaje ensamblador, dejando un espacio entre el código de operación y los operandos, (en modo inmediato empieza los operadores con #).

Con (CONTROL) QT se fijan los espacios de tabulación (tecla TAB) fija tal como lo fijó en la instrucción anterior.

Al inicio de los programas hay que colocar la instrucción ORG especificando la localidad de memoria donde inicia el programa. Después de las instrucciones en ensamblador, si se pone “;” todo lo que sigue está tomado como comentario. Para ensamblar un programa hay que oprimir F10 (menú) y con “A” entrar a las formas de ensamblado posibles. El objeto se selecciona S.19 oprimiendo (RETORNO) y seleccionando las que se desee (listado para impresión), se sale con la tecla de ESCAPE.

Al oprimir F4 se ensambla en forma seleccionada y en caso de haber errores lo indicará, y no se ensamblará habiendo errores.

Con F10 y luego “F” puede revisar el directorio y cargar el programa deseado.

Para establecer la comunicación entre la tarjeta y la PC debe teclearse F10 y luego “C” de comunicación y seleccionar el puerto adecuado, oprimir (RETORNO) para pasar a comm 1 a

comm 2. Después hat que oprimir F7 que abrirá la ventana de comunicación, con reset en la tarjeta, deberá aparecer el letrero de buffalo, indicando que la comunicación está lista.

Si se desea llevar un programa del disco a la tarjeta, debe de teclearse LOAD T (RETORNO) y se estará dentro de un programa (T); se oprime ahora F6 y aparecerá un letrero pidiendo el nombre del programa, se teclea y oprime (retorno), se transmitirá el programa, al terminar (aparecerá el promp) se debe oprimir reset en la tarjeta para poder salir del programa T y poder trabajar con el programa cargado.

Apéndice C: Detalles de circuito del Microcontrolador

Circuito del microcontrolador, las conexiones hacia los otros circuitos se hacen a través de un cable plano, ver tabla 14.

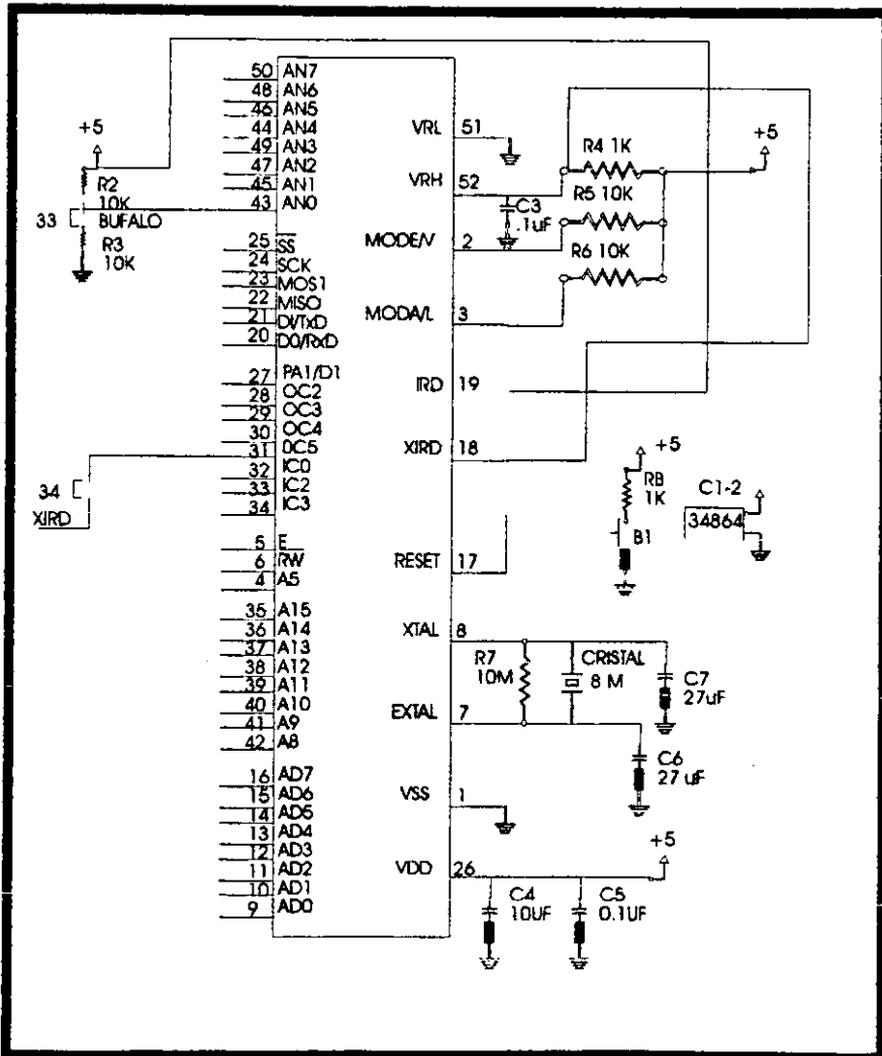


Figura 47. Conexiones del Circuito del microcontrolador

La tarjeta con la que se esta trabajando tiene concentrados los puertos de salida y entrada en el conector J5 y se especifica en la siguiente tabla.

GND 60	OO	59 GND
VCC 58	OO	57 VCC
LIBRE 56	OO	55 LIBRE
LIBRE54	OO	53 LIBRE
VRH 52	OO	52 VRL
PE7 50	OO	49 PE 3
PE6 48	OO	47 PE2
PE5 46	OO	45 PE1
PE4 44	OO	43 PE0
PB0/A8 42	OO	41 PB1/A9
PB2/A10 40	OO	39 PB3/A11
PB4/A12 38	OO	37 PB5/A13
PB6/A14 36	OO	35 PB7/A15
PA0/IC3 34	OO	33 PA1/IC2
PA2/IC1 32	OO	31 PA3/OC5
PA4/OC4 30	OO	29 PA5/OC3
PA6/OC2 28	OO	27 PA7/OC1
VCC 26	OO	25 PD5/SS*
PD4/SCK 24	OO	23 PD3/MOSI
PD2/MISO 22	OO	21 PD1/TXD
PD0/RXD 20	OO	19 IRQ*
XIRQ*	OO	17 RESET*
PC7/AD7 16	OO	15 PC6/AD6
PC5/AD5 14	OO	13 PC4/AD4
PC3/AD3 12	OO	11 PC2/AD2
PC1/AD1 10	OO	9 PC0/AD0
XTAL 8	OO	7 EXTAL
STRB/R/W* 6	OO	5 E
STRA/AS 4	OO	3 MODA/LIR*
GND 2	OO	1 GND

Tabla 14. Distribución de las terminales en el conector J5

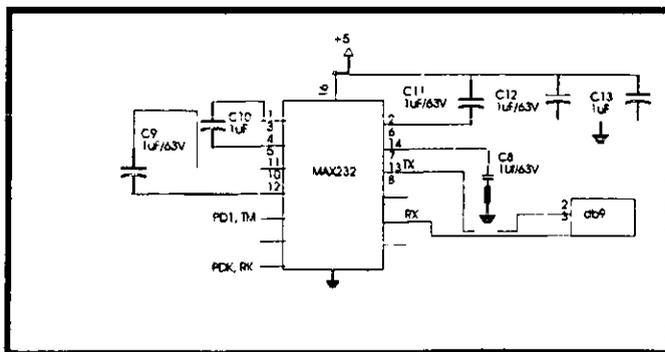


Figura 48. Circuito de comunicación de interfase RS232

La conexión a la computadora se hace a través de un db9 macho y un db9 hembra y los cables conectados son; 2 con 2,3 con 3 y 5 con 5. Del protocolo de conexión RS232 solo se tomaron los pines necesarios para la comunicación, si se desea saber un poco más de este tipo de conexión ver detalles en página 46.

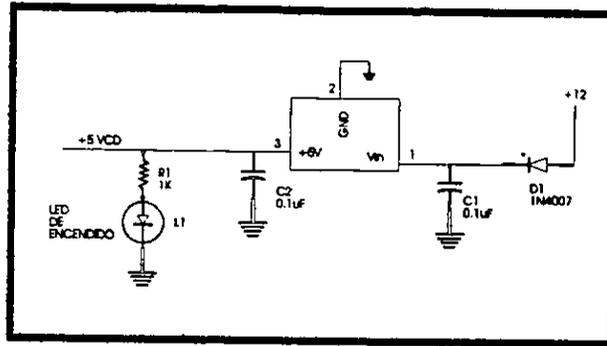


Figura 49. Fuente de voltaje de alimentación para el microcontrolador.

Esta fuente no permite que el circuito sea conectado mal. Ya que el diodo no permite el flujo de energía por el lado equivocado. Tiene un led para indicar que el circuito esta funcionando. Dado que éste está constituido por un regulador, se puede alimentar con un voltaje que puede ir de 8V a 18V y éste dará a la salida un voltaje continuo de 5 V y 1Amp. La energía, la proporcionan dos baterías, conectadas en paralelo, marca Yuasa de 12 V a 1.2 A.

- Jorge S, Gamboa Cabrera. **Electronica VI El microcontrolador M68HC11**. E. S. I. M. E.
- Motorola **Fast and LS TTL data**. Fifth edition. Impreso en EE.UU. 1992
- Roger L. Tokheim. **Fundamentos de los Microprocesadores**. McGraw Hill. segunda edición, Impreso en México.
- David Gonzalez Maxinez Andre Ferreyra Ramirez Angel R Zapata Ferrer Adriano de Luca Pennichea, **Fuzzy logic**, UNAM. Apuntes UNAM 1993
- Bart kosko_ **Fuzzy Thinking the new science of fuzzy logic**., Hyperio new york 1993
- Motorola **Linear/Interface Ics device data vol. II**. Fifth edition. Impreso en EE.UU. 1993
- Motorola **M68HC11 Reference Manual**, impreso en EE. UU.
- Ing. Porfirio Mora Villavicencio Samuel bolaños. **Microcontrolador MC68HC11 Introducción al uso del software para su programación**. E. S.I. M. E. Zacatenco Julio de 1995.
- Alfonso Magaña Ramírez y José Luis Ortega Vallejo **Sistemas Neuro fuzzy**. Tesis E. N. E: P. Aragon 1996, UNAM.