EE4800/EE5880
Real Time Embedded Systems

Lesson 21-24

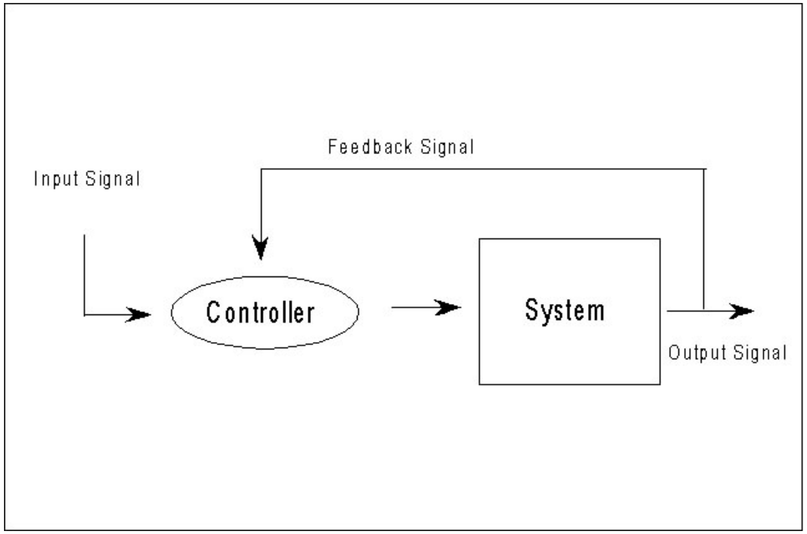
Fuzzy Logic

Overview

- What is a control system?
- Traditional vs Fuzzy Logic Control Systems
- Fuzzy Logic Controller Overview
- 68HC12 Fuzzy Logic Operations
- Membership Function Design Process
- Fuzzy Rule Implementation
- 68HC12 Fuzzification Process
- Rule Evaluation
- 68HC12 Defuzzification Process
- Mobile Robot Example

What is a control system?

- Any system, whose outputs are controlled by some inputs to the system, is called a control system
- Typical control system, Figure 4.1, pg 120



What is a control system?

- Uses model along with feedback provided by system output to generate a new system input
- Controller requires accurate mathematical model of system
  - Requires dynamic equations
  - Euler Lagrange Equation - Ref. Section 4.2.2.
  - Not fun stuff!!!

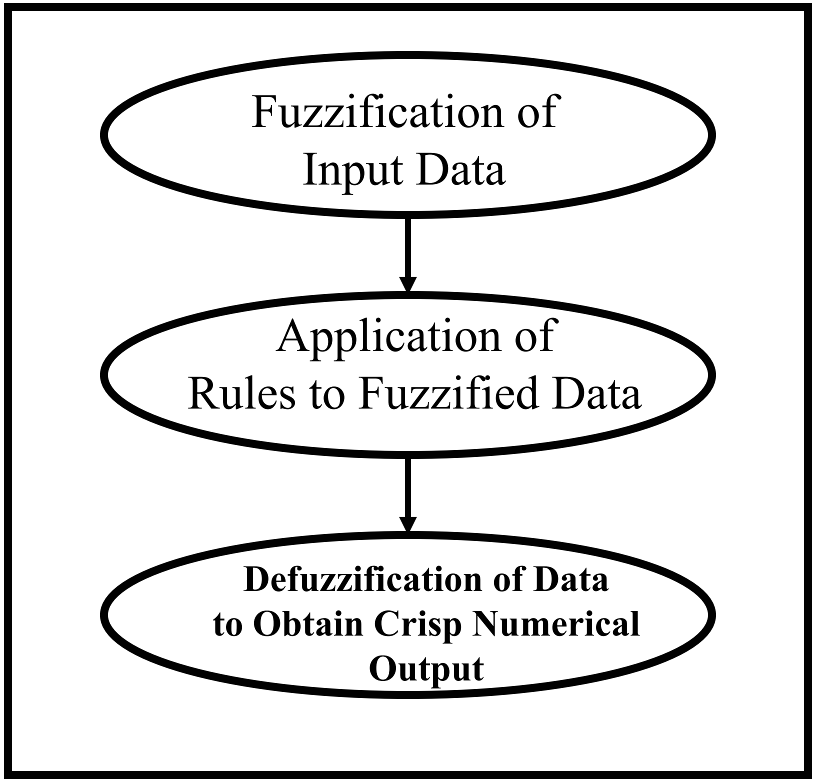
Traditional vs Fuzzy Logic Control Systems

- Traditional control method requires accurate, often complex system of equations
- Fuzzy logic proponents claim fuzzy systems are much closer to human control process
- Neural networks, fuzzy logic, and evolutionary systems share common thread
  - Do not rely on mathematical systems descriptions
- Interestingly, biological control systems do not rely on complex mathematical models
- Traditional camp treats these new approaches with skepticism for lack of mathematical rigor
- Fuzzy logic controllers in heavy use

Fuzzy Logic Controller Overview

- Fuzzy Logic Control Systems rely on physical rules rather than mathematical equations
- Simple form of an expert system
- Expert system contains rules that map input commands to output system responses
- Also an evidence accumulating system
  - Uses confidence factor to assign a system output according to the confidence measure computed using a well-defined process

Fuzzy Logic Control


Fuzzification of Input Data
Application of Rules to Fuzzified Data
Defuzzification of Data to Obtain Crisp Numerical Output

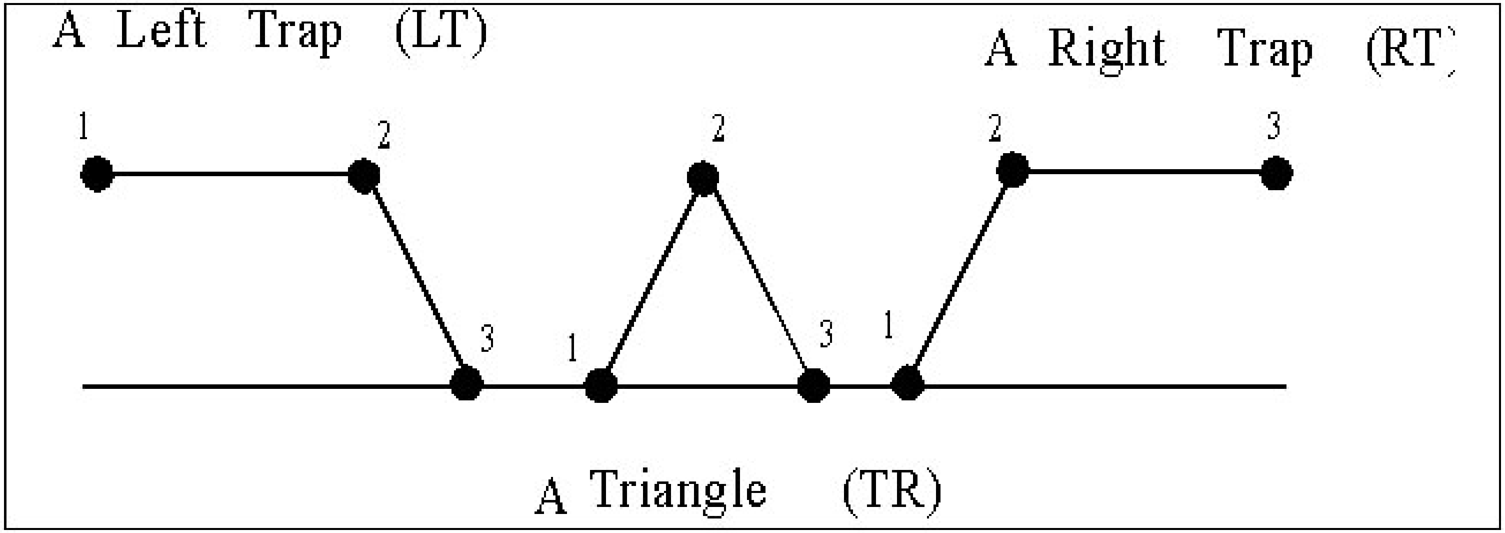
- Inputs should be sufficient to describe current status of system
- Need appropriate relationships between inputs and outputs
- Represent output control values in suitable form for system control

Fuzzy Logic Control

- Input values are first fuzzified by assigning an appropriate value (degree of membership) to each fuzzy membership function
- Fuzzy membership function is defined as an instance of a linguistic variable describing the fuzzy controller input
- All fuzzy membership functions have one of the following shapes: a left trap (LT), a trapezoid (TR) and a right trap (RT)

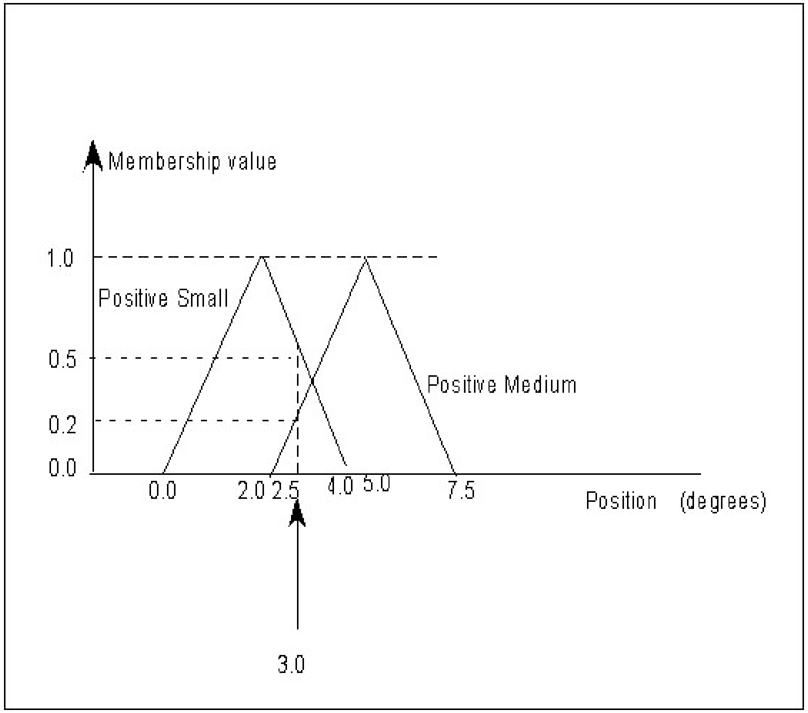
Fuzzy Logic Control

- HC12 implements all of its input membership functions using trapezoids
- Fig 4.5, page 128



Fuzzy Logic Control

- Given an input datum, the controller must first assign a degree of membership, a value ranging from zero to one, to each fuzzy membership function.
- Fig. 4.6, pg 129



Fuzzy Logic Control

- Apply expert rules to input membership functions
- Rules should cover all possible combinations of the input variables
- Rules map combinations of inputs to output force fuzzy membership functions
- Rule format:

  If (antecedent 1) AND (antecedent 2),

  Then (desired response)

Fuzzy Logic Control

- In applying rules, the **minimum** degree of membership among the input functions is assigned to the output membership function
  - – Known as minimum rule
- Forms confidence factor
- Minimum of the two is chosen since it guarantees that both antecedents are true

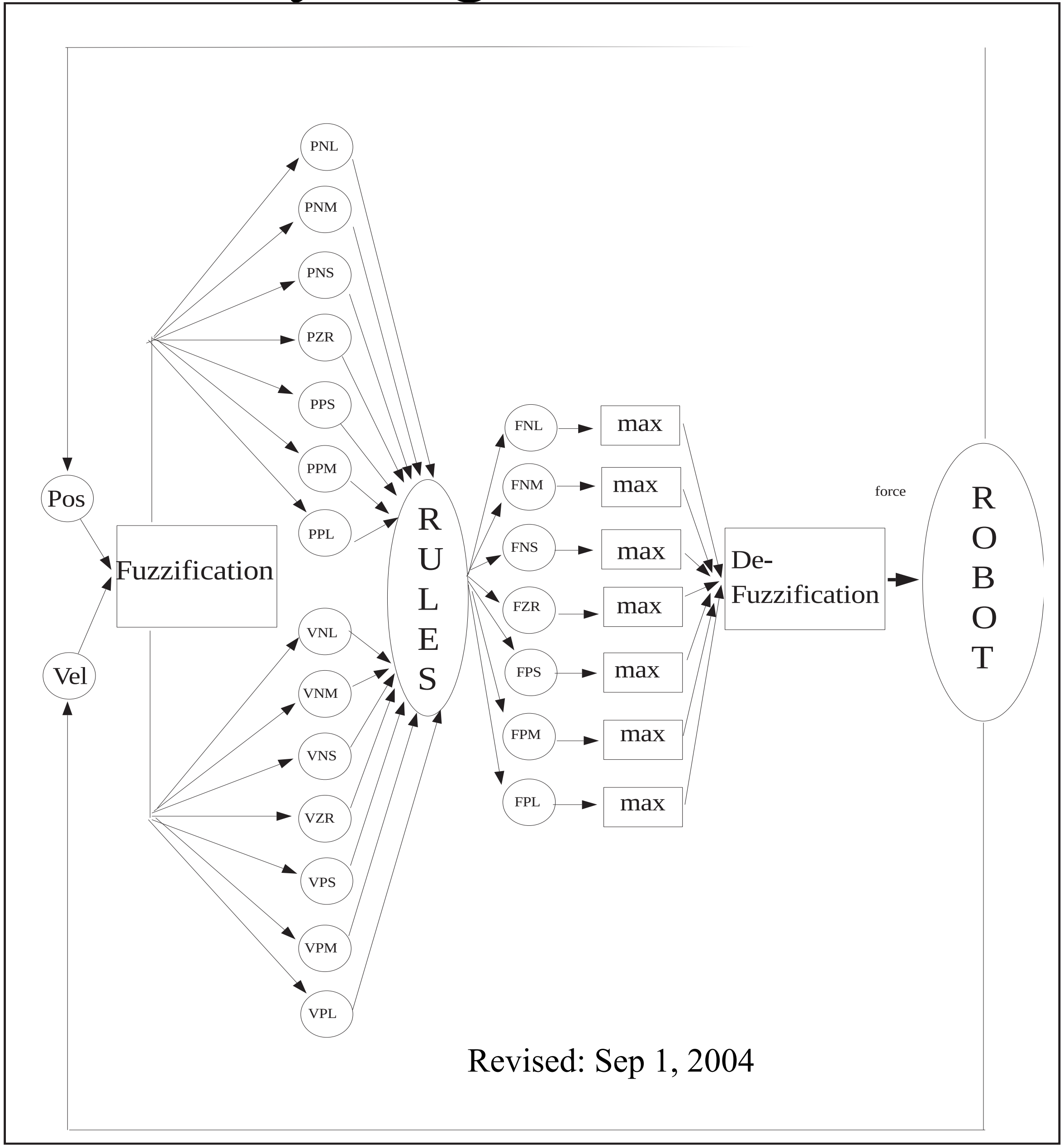
Fuzzy Logic Control

- Controller now takes the maximum of the confidence factors as the degree of the membership for the particular fuzzy function.
- Relies on the most dominant rule to assign the output membership function
- Need to convert the output membership function values to a form suitable to control system
  - Control value must be a numerical one
  - Need to reverse process of fuzzification - defuzzification

Fuzzy Logic Control

- Defuzzification – most common method is finding center of gravity or centroid of output membership functions
  - Combines all regions of the output fuzzy functions with OR operator
  - Computes center point
  - Generates crisp numerical value
- Gives equal weights to each output membership function

Fuzzy Logic Control

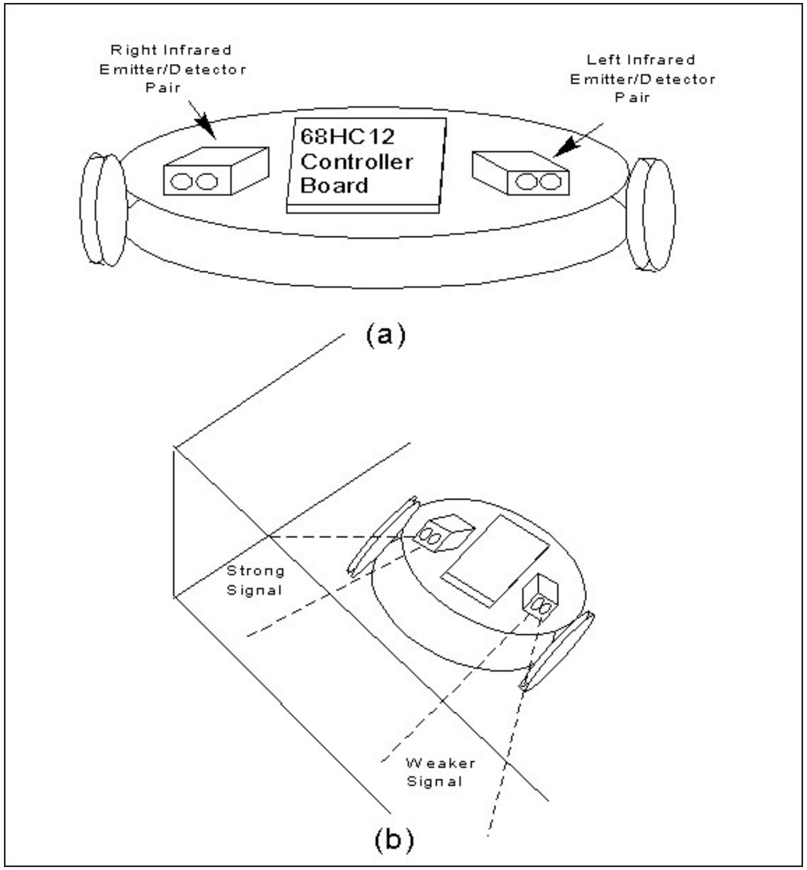


68HC12 Fuzzy Logic Operations

- Four 68HC12 instructions:
  - MEM: grade of membership
  - REV: MIN-MAX rule evaluation
  - REVW: MIN-MAX rule evaluation with optional rule weighting
  - WAV: performs weighted average calculations
- Supplementary instructions:
  - ETBL, TBL, EDIV, EDIVS, EMACS, EMAX, EMAXM, EMIND, EMINM, EMUL, EMULS

68HC12 Fuzzy Logic Operations

- Example: Design a fuzzy logic based robot controller - Fig 4.15, pg 149


Right Infrared Emitter/Detector Pair
Left Infrared Emitter/Detector Pair
68HC12 Controller Board
(a)
Strong Signal
Weaker Signal
(b)

68HC12 Fuzzy Logic Operations

1 . Determine linguistic variables describing all inputs to the controller

left sensor input: very weak, weak, medium, strong, very strong

right sensor input: very weak, weak, medium, strong, very strong

2 . Each linguistic variable must be defined with a set of input membership functions. For the HC12 a trapezoidal function must be used. Fig 4.16, pg 150

68HC12 Fuzzy Logic Operations

3 . Input membership function design process.

- Each membership function must be trapezoid: LT, T, RT
- Specify uniquely with farthest left point, farthest right point, left slope, and right slope
- Full membership corresponds to $FF - for each sensor

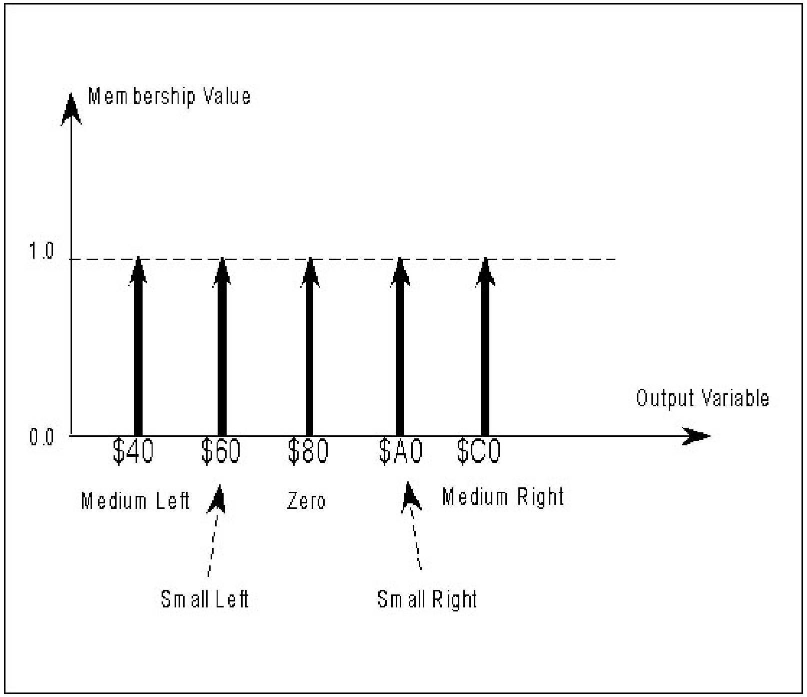
very weak	*$00, $50, $00, $10*
weak	*$40, $70, $10, $10*
medium	*$60, $A0, $10, $10*
strong	*$90, $C0, $10, $10*
very strong	*$B0, $FF, $10, $00*

68HC12 Fuzzy Logic Operations

4 . Output membership function design process.

- Use singleton functions for the fuzzy output membership functions (Singleton: each function is defined by a single value).
  - *Medium Left ($40)*
  - *Small Left ($60)*
  - *Zero ($80)*
  - *Small Right ($A0)*
  - *Medium Right ($C0)*
  - **Assumption:** $80 keeps robot on straight path.

Figure 4.17, pg 151



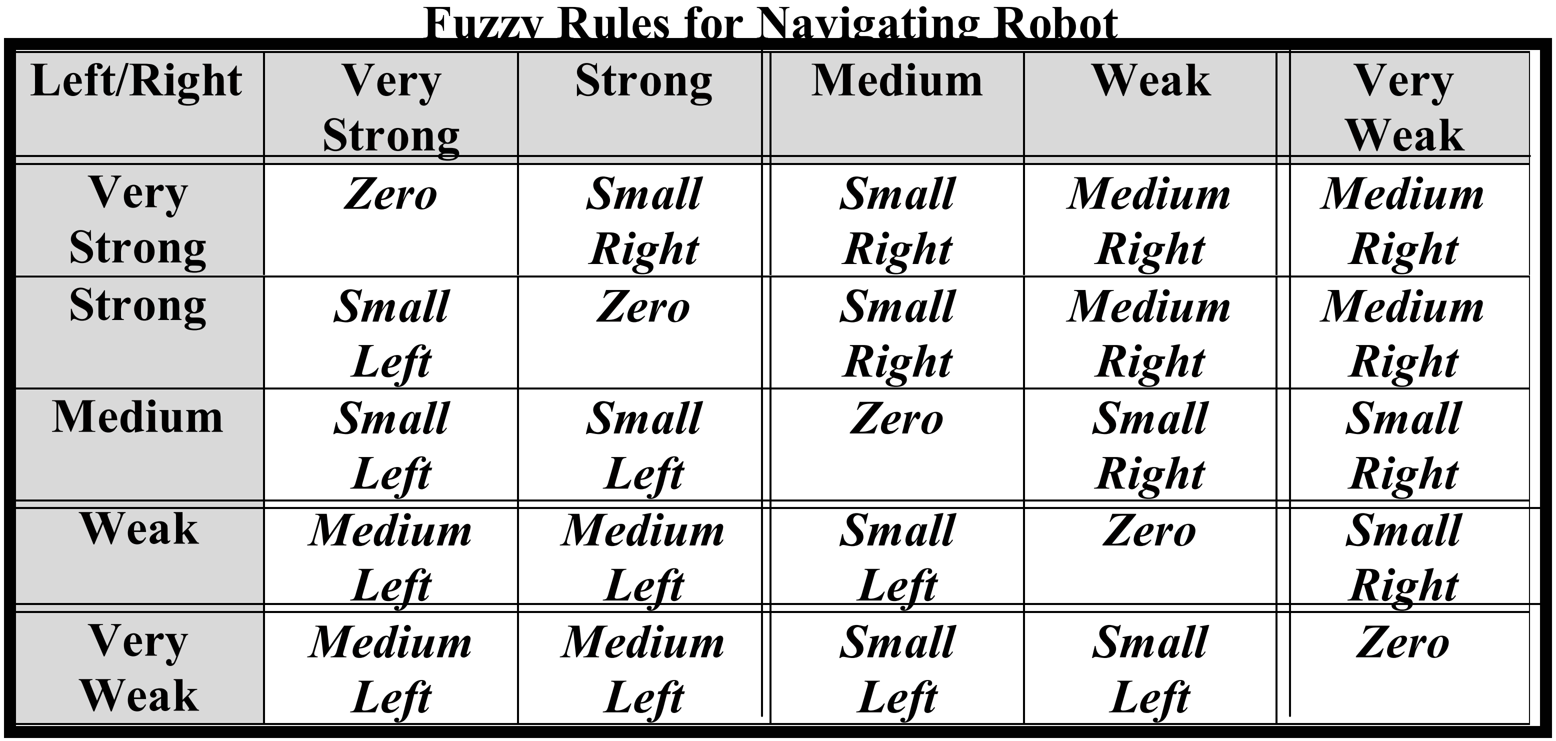
68HC12 Fuzzy Logic Operations

5. Determine fuzzy rules that map input to output membership functions. Rules are intuitive based on considering all possible input sensor value scenarios.

   Rule format:

   IF (*antecedent 1*) AND (*antecedent 2*),
   THEN (*desired output*)

68HC12 Fuzzy Logic Operations



Fuzzy Rules for Navigating Robot

Left/Right	**Very Strong**	**Strong**	**Medium**	**Weak**	**Very Weak**
Very Strong	***Zero***	***Small Right***	***Small Right***	***Medium Right***	***Medium Right***
Strong	***Small Left***	***Zero***	***Small Right***	***Medium Right***	***Medium Right***
Medium	***Small Left***	***Small Left***	***Zero***	***Small Right***	***Small Right***
Weak	***Medium Left***	***Medium Left***	***Small Left***	***Zero***	***Small Right***
Very Weak	***Medium Left***	***Medium Left***	***Small Left***	***Small Left***	***Zero***

68HC12 Fuzzy Logic Operations

6. Fuzzy Rule Implementation - encode rules for all input combinations using the format:

RULE 1:

antecedent one, antecedent two, $FE, desired output, $FE

Note: $FE separator indicates convenient break points from fuzzy logic processing should an interrupt occur.

68HC12 Fuzzy Logic Operations

7. Deffuzify: Combine truth values of all output fuzzy membership functions into a single numerical value.
   - Average the output function values - find the centroid of arbitrary output membership functions for singleton output membership functions.

Output = Σ (Si Fi) / Σ (Fi) for $i = i ..n$

where: Si is the Singleton value

Fi is the output membership value for fuzzy logic output function i

Mobile Robot Example Code

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;Fuzzy Logic Controller for A Mobile Robot
;Description: This program takes in two Infrared
;sensor values and computes a di-
;rection control signal to avoid
;wall collisions.  The two sensor
;values are read from memory
;locations $6000 and $6001 and
;the control output value is
;written to memory location $6002
;
; Authors: Daniel Pack and Steve Barrett
; Date: 8-21-2000
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

Mobile Robot Example Code

```
; Data Section
O_R_VS          EQU   $00   ;Offset values input and output mem fns
O_R_ST          EQU   $01   ; Right Sensor Strong
O_R_ME          EQU   $02   ; Right Sensor Medium
O_R_WE          EQU   $03   ; Right Sensor Weak
O_R_VW          EQU   $04   ; Right Sensor Very Weak
O_L_VS          EQU   $05   ; Left Sensor Very Strong
O_L_ST          EQU   $06   ; Left Sensor Strong
O_L_ME          EQU   $07   ; Left Sensor Medium
O_L_WE          EQU   $08   ; Left Sensor Weak
O_L_VW          EQU   $09   ; Left Sensor Very Weak
O_ML            EQU   $0A   ; Medium Left
O_SL            EQU   $0B   ; Small Left
O_ZR            EQU   $0C   ; Zero
O_SR            EQU   $0D   ; Small Right
O_MR            EQU   $0E   ; Medium Right
MARKER          EQU   $FE   ; rule separator
ENDR            EQU   $FF   ; end of rule marker
```

Mobile Robot Example Code

```
Line 33                   ORG   $6000
Line 34  RSENSOR          RMB   $01    ; Allocating
Line 35  LSENSOR          RMB   $01    ; memory locations
Line 36  CONTROLS         RMB   $01    ; for input/output variables

Line 37  ; Fuzzy Input Membership Function Definitions for Right Sensor
Line 38  R_Very_Strong    FCB   $B0, $FF, $10, $00
Line 39  R_Strong         FCB   $90, $C0, $10, $10
Line 40  R_Medium         FCB   $60, $A0, $10, $10
Line 41  R_Weak           FCB   $40, $70, $10, $10
Line 42  R_Very_Weak      FCB   $00, $50, $00, $10

Line 43  ; Fuzzy Input Membership Function Definitions for Left Sensor
Line 44  L_Very_Strong    FCB   $B0,$FF,$10,$00
Line 45  L_Strong         FCB   $90,$C0,$10,$10
Line 46  L_Medium         FCB   $60,$A0,$10,$10
Line 47  L_Weak           FCB   $40,$70,$10,$10
Line 48  L_Very_Weak      FCB   $00,$50,$00,$10
```

Mobile Robot Example Code

```
;Fuzzy Output Membership Function Definitions
Medium_Left      FCB   $40
Small_Left       FCB   $60
Zero             FCB   $80
Small_Right      FCB   $A0
Medium_Right     FCB   $C0

; Locations for fuzzy membership values
R_VS             RMB   $01
R_ST             RMB   $01
R_ME             RMB   $01
R_WE             RMB   $01
R_VW             RMB   $01
L_VS             RMB   $01
L_ST             RMB   $01
L_ME             RMB   $01
L_WE             RMB   $01
L_VW             RMB   $01
```

Mobile Robot Example Code

```
; Output Fuzzy Logic Membership Values - initialize them to zero
ML                FCB   $00
SL                FCB   $00
ZR                FCB   $00
SR                FCB   $00
MR                FCB   $00
```

Mobile Robot Example Code

```
; Rule Definitions
Rule_Start          FCB       O_R_VS,O_L_VS,MARKER,O_ZR,MARKER
                    FCB       O_R_VS,O_L_ST,MARKER,O_SL,MARKER
                    FCB       O_R_VS,O_L_MD,MARKER,O_SL,MARKER
                    FCB       O_R_VS,O_L_WE,MARKER,O_ML,MARKER
                    FCB       O_R_VS,O_L_VW,MARKER,O_ML,MARKER
                    FCB       O_R_ST,O_L_VS,MARKER,O_SR,MARKER
                    FCB       O_R_ST,O_L_ST,MARKER,O_ZR,MARKER
                    FCB       O_R_ST,O_L_MD,MARKER,O_SL,MARKER
                    FCB       O_R_ST,O_L_WE,MARKER,O_ML,MARKER
                    FCB       O_R_ST,O_L_VW,MARKER,O_ML,MARKER
                    FCB       O_R_MD,O_L_VS,MARKER,O_SR,MARKER
                    FCB       O_R_MD,O_L_ST,MARKER,O_SR,MARKER
                    FCB       O_R_MD,O_L_MD,MARKER,O_ZR,MARKER
                    FCB       O_R_MD,O_L_WE,MARKER,O_SL,MARKER
                    FCB       O_R_MD,O_L_VW,MARKER,O_SL,MARKER
                    FCB       O_R_WE,O_L_VS,MARKER,O_MR,MARKER
                    FCB       O_R_WE,O_L_ST,MARKER,O_MR,MARKER
                    FCB       O_R_WE,O_L_MD,MARKER,O_SR,MARKER
                    FCB       O_R_WE,O_L_WE,MARKER,O_ZR,MARKER
                    FCB       O_R_WE,O_L_VW,MARKER,O_SL,MARKER
                    FCB       O_R_VW,O_L_VS,MARKER,O_MR,MARKER
                    FCB       O_R_VW,O_L_ST,MARKER,O_MR,MARKER
                    FCB       O_R_VW,O_L_MD,MARKER,O_SR,MARKER
                    FCB       O_R_VW,O_L_WE,MARKER,O_SR,MARKER
                    FCB       O_R_VW,O_L_WE,MARKER,O_ZR,ENDR
```

Mobile Robot Example Code

```
; Main Program
         ORG     $4000

; Fuzzification-----------------------------------------------------
; Fuzzification Procedure
; MEM function determines grade of membership.
;
; Registers must be set up as follows:
;  A: must hold the current crisp value of system input variable
;     X: must point to 4-byte data structure describing the trapezoidal
;        membership function for a label of the system input
;     Y: must point to the fuzzy input (RAM location) where the resulting
;        grade of membership is to be stored
;-------------------------------------------------------------------

; Perform fuzzification for right sensor
         LDX   #R_Very_Strong      ; Start of Input Mem func
         LDY   #R_VS               ; Start of Fuzzy Mem values
         LDAA  RSENSOR             ; Right Sensor Value
         LDAB  #5                  ; Number of iterations
         Loopr MEM                 ; Assign mem value
         DBNE  B,Loopr             ; Do all five iterations
```

Mobile Robot Example Code

```
         ; Perform fuzzification for left sensor
Line 107         LDAA  LSENSOR          ; Left Sensor Value
Line 108         LDAB  #5               ; Number of iterations
Line 109         Loopl MEM              ; Assign mem value
Line 110         DBNE  B,Loopl          ; Do all five iterations
```

Mobile Robot Example Code

```
; Fuzzy Logic Rule Evaluation----------------------------------------
; REV function performs an unweighted evaluation of a list of rules,
; using fuzzy input values to produce fuzzy outputs.
;
;Before executing REV, perform the following set up operations.
; X: must point to the first 8-bit element in the rule list.
;     Y: must point to the base address for fuzzy inputs and outputs
;     A: must contain the value of $FF, and the CCR V bit must = 0
;         (LDAA #$FF places the correct value in A and clears V)
;     Clear fuzzy outputs to zeros
;--------------------------------------------------------------------

        LDY    #R_VS                ; Process rules
        LDX    #Rule_Start          ; Point X to the start addr
        LDAA   #$FF                 ; Initialize min and V bit
        REV                         ; Evaluate rules
```

Mobile Robot Example Code

```
; Defuzzification Process-------------------------------------
; WAV function performs weighted average calculations on
; values stored in memory.
;       X: used to reference one source operand list
;       Y: used to reference second source operand list
;       B: used a s counter to control number of elements to
;          be included in the weighted average
;-------------------------------------------------------------
        LDX   #Medium_Left        ; Start of output mem func
        LDY   #ML                 ; Start of mem values
        LDAB  #$05                ; Five elements sum
        WAV                       ; Computing a crisp value
        EDIV                      ; Extended divide - 32 bit/16 bit
        TFR   Y,D                 ; Store answer to D
        STAB  CONTROLS            ; Save the answer
        END
```