

# Soft Computing Control of a Mobile Robot

**Moufid Harb, PhD, PEng**

Research Scientist , [Larus Technologies](#)  
Research Collaborator, [University of Ottawa](#)  
[moufid.harb@larus.com](mailto:moufid.harb@larus.com)

CEG4910&4911 Computer Engineering Design Project - II (Winter 2009)

# The Project Outline

Problem Definition

Simulation Environment and Mathematical Model

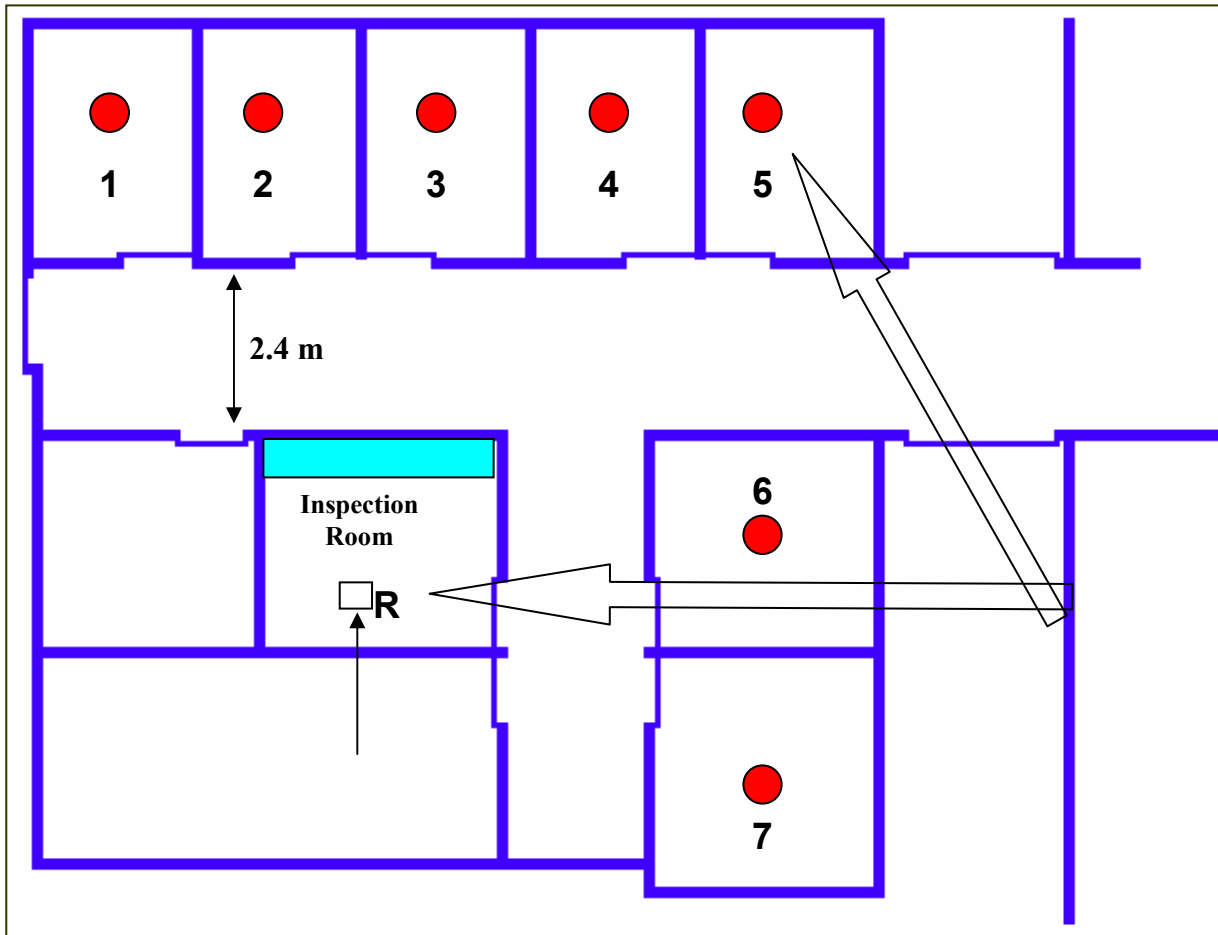
Ranging System

Neural Navigation and Environmental Recognition

Path Planning and Global Neural Navigation

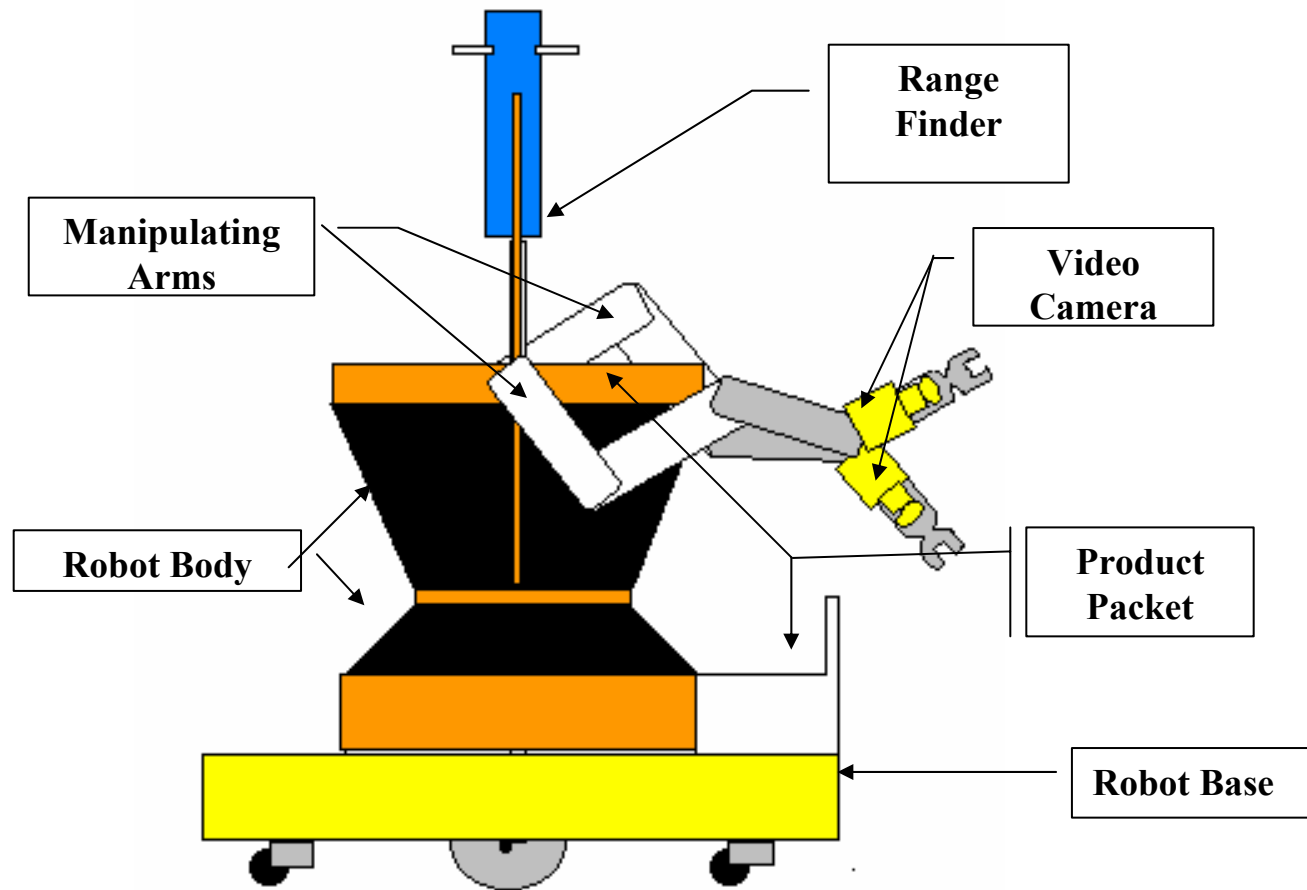
Incorporated Fuzzy-Neural Speed Control

# Problem Definition



- Floor plan of a medicine factory that manufactures tablets to treat heart diseases
- Because of:
  - The required very clean environment
  - The daily exposure to the factory air might cause a serious health problem to human workers

# Mobile Robot



# Computer Simulation Environment

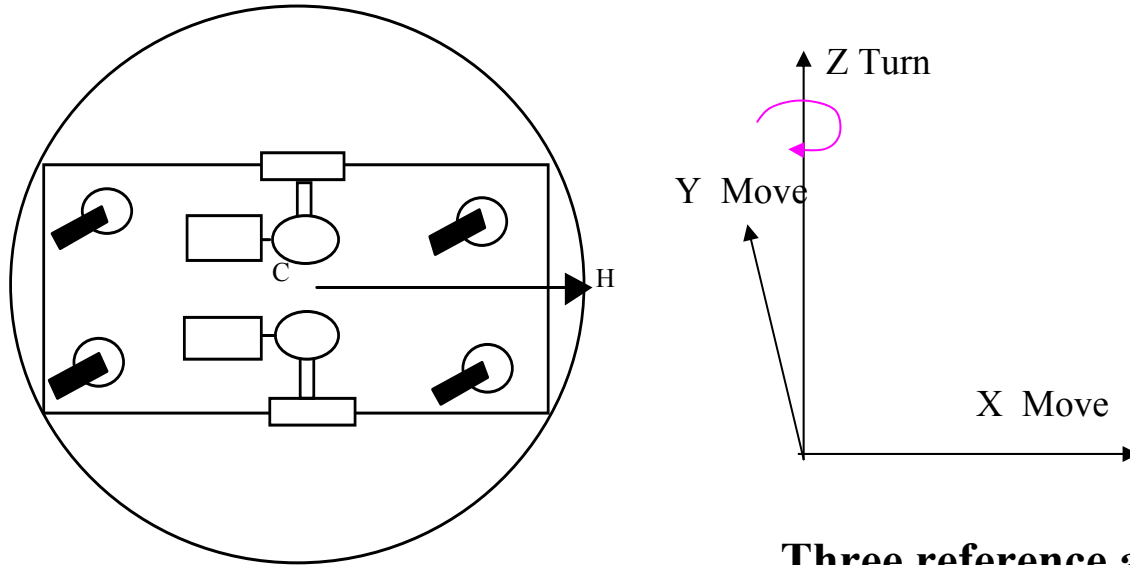
That saves:

Time, Efforts, and Money

Therefore:

Mathematical Model

# Mobile robot of differential drive arrangement vehicle



**Three reference axes**

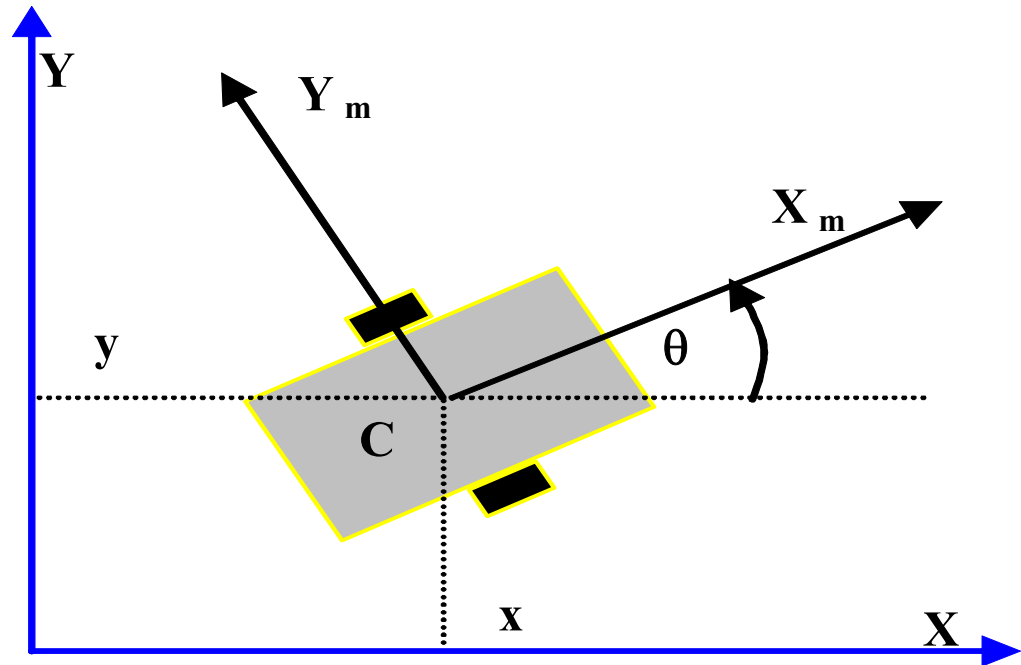
- Steering is affected by the difference in velocity of the right and left motorized wheels
- Forward orientation **angle**  $\theta$ , and Cartesian coordinates  $X_C$  and  $Y_C$  determine the vector that represents the location of the robot.
- The maximum dimension of the rectangular robot is 0.75 meter, which equals the diameter of the circle that surrounds the robot.

# Robot and Cartesian Coordinates

Dynamic Model Equation of the MR's Centre Position

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} -\sin\theta & 0 \\ \cos\theta & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \eta_1 \\ \eta_2 \end{pmatrix}$$

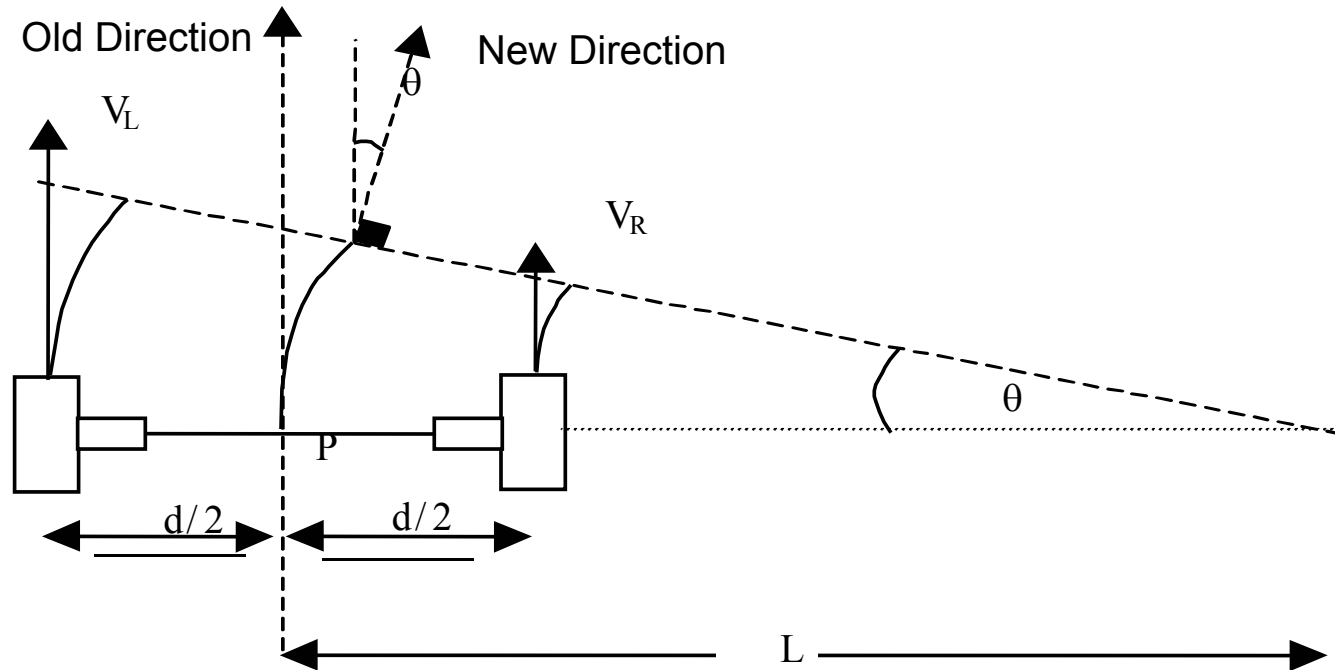
$\eta_1$  Velocity element along  $X_m$   
 $\eta_2$  Angular velocity



Robot Posture

$$\zeta = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$

# Kinematics Model: Differential Drive Arrangement



$$V_R = V - \frac{d \cdot \theta}{2 \cdot T_S}$$

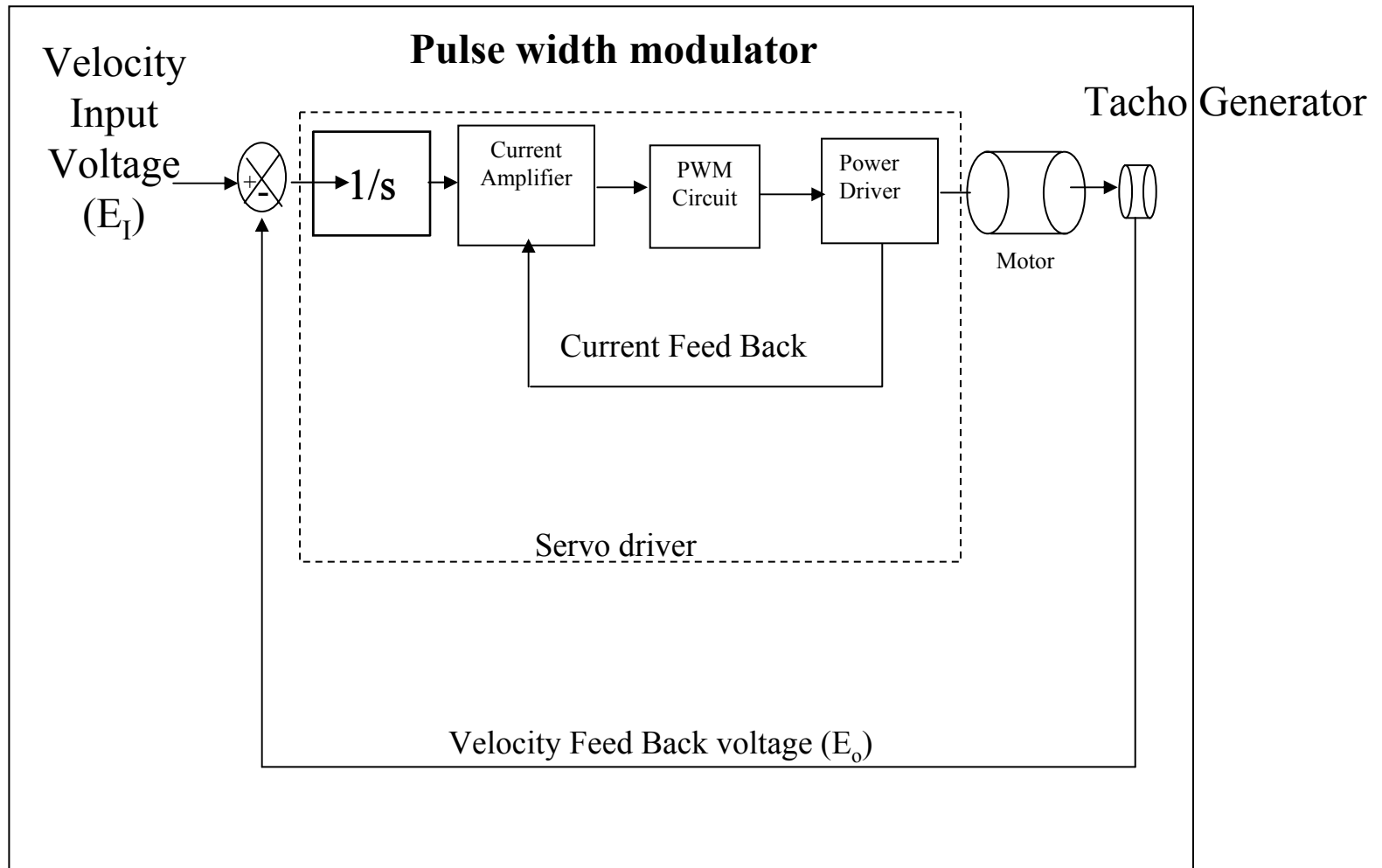
$$V_L = V + \frac{d \cdot \theta}{2 \cdot T_S}$$

$T_S$ : Sampling Time  
Between 2  
Measurements

$$V = \frac{V_L + V_R}{2}$$

$$\theta = (V_L - V_R) \frac{T_S}{d}$$

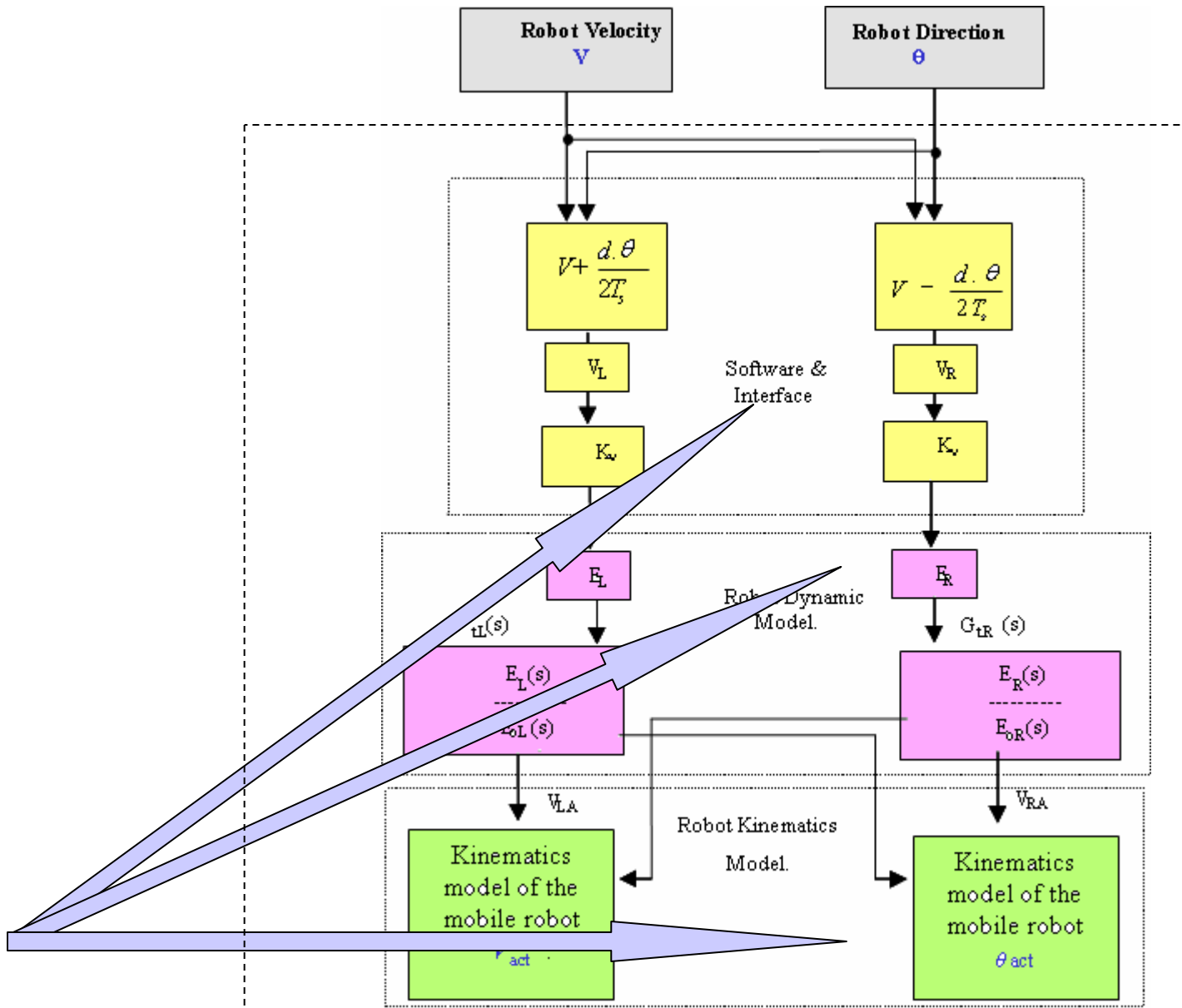
# Differential Drive Arrangements



# Speed and Direction Control

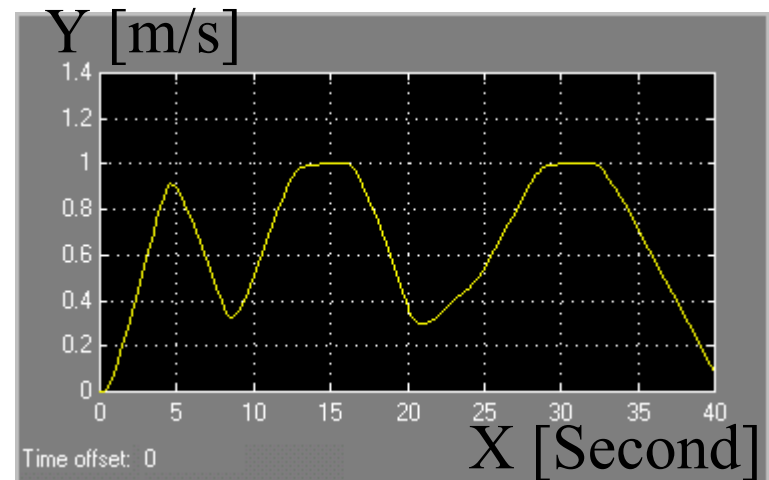
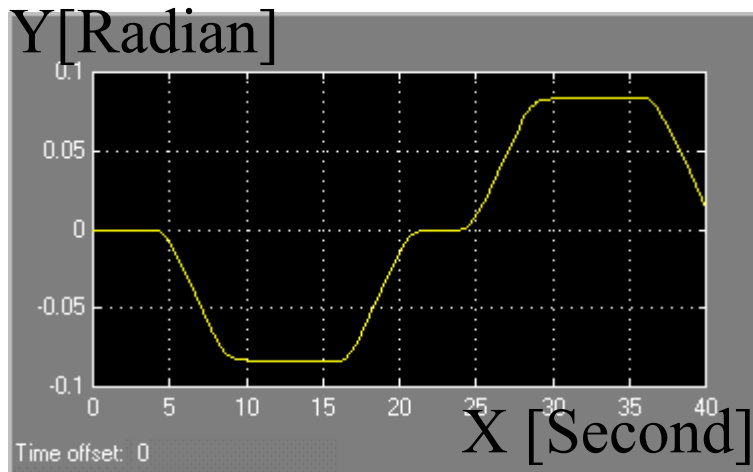
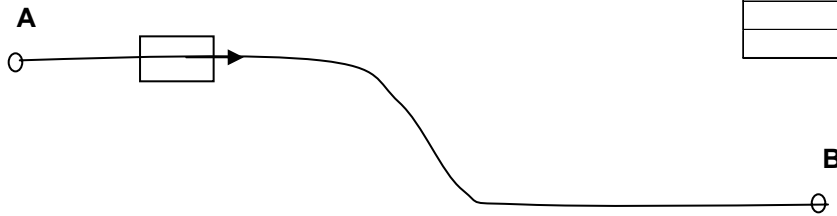
## Considering The Mathematical Model

## For Computer Simulation



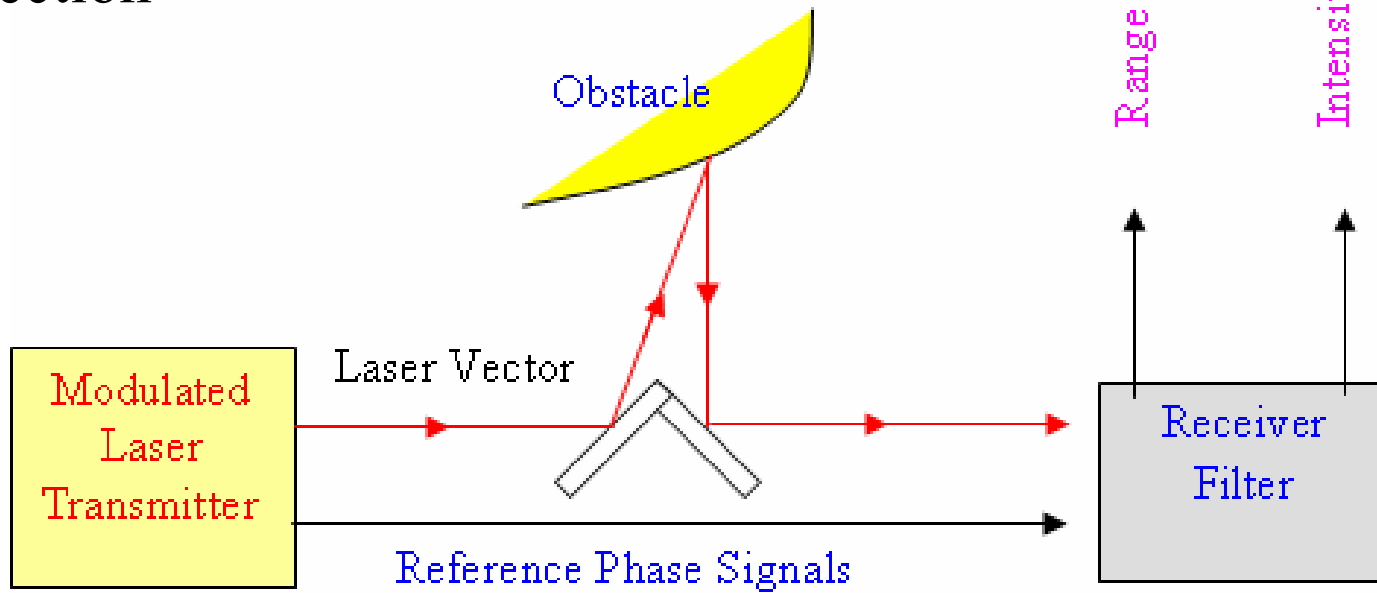
# Simulation Results

<i>Time</i> [s]	<i>Speed</i> [m/s]	<i>Direction Change</i> Degree
0	0	0
4	1	0
8	0.25	-15
12	1	-15
16	1	-15
20	0.25	0
24	0.5	0
28	1	15
32	1	15
36	0.5	15
40	0	0



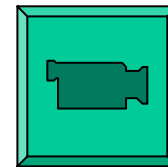
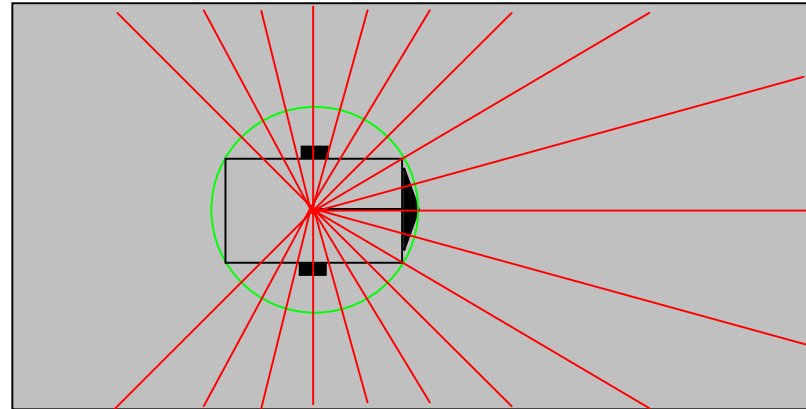
# Ranging System

Laser Range Finder is considered reliable in measuring distance and direction



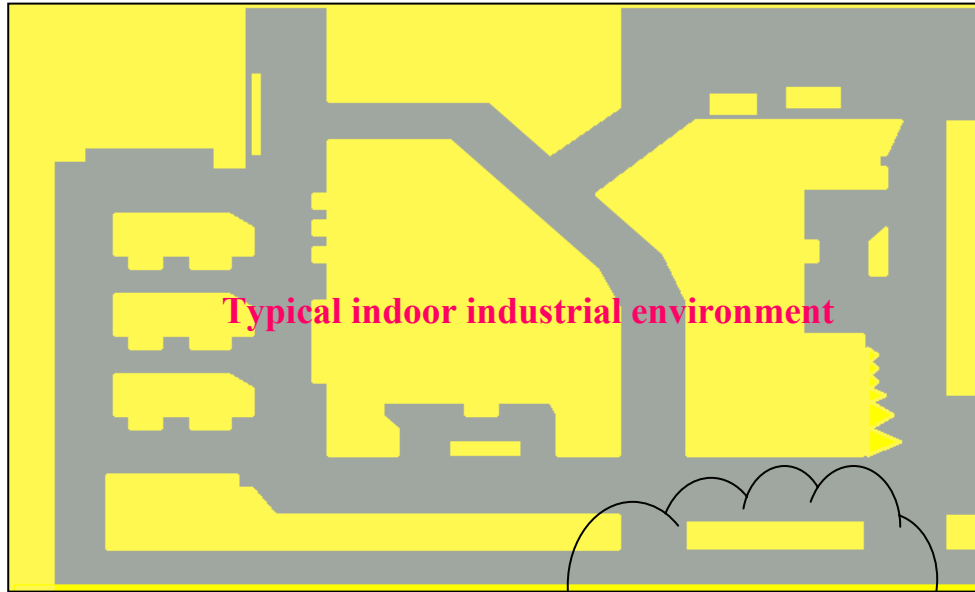
# Distance Measurement

- Distribution of measurements on the robot body
- *19 directions regularly distributed with a separation angle of  $15^\circ$  ( $-135^\circ$  to  $135^\circ$ )*
- Measured distances are relative to the MR circle diameter, which surrounds the robot body

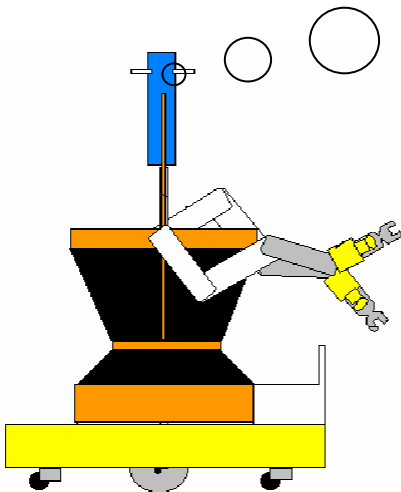


**Model - Frame**

# You Need



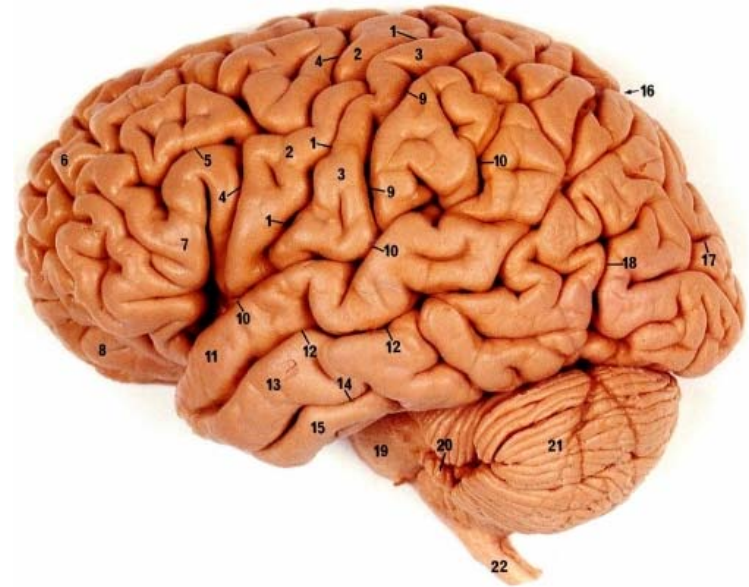
How do I  
move there ?



- Navigation system to guide the robot and avoid obstacles, defining:
  - Location
  - Direction
  - Speed
  - Path
  - Travel distance

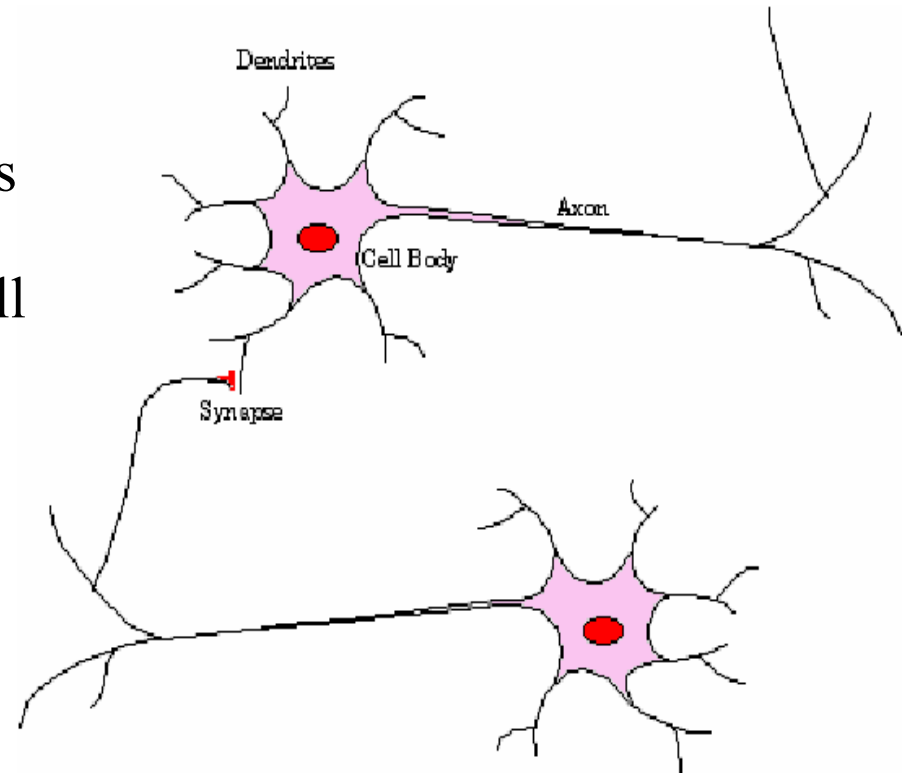
# Human Brain

- The human brain is unique
- It gives us the power to:
  - Think
  - Learn
  - Predict
  - Adapt
  - etc.



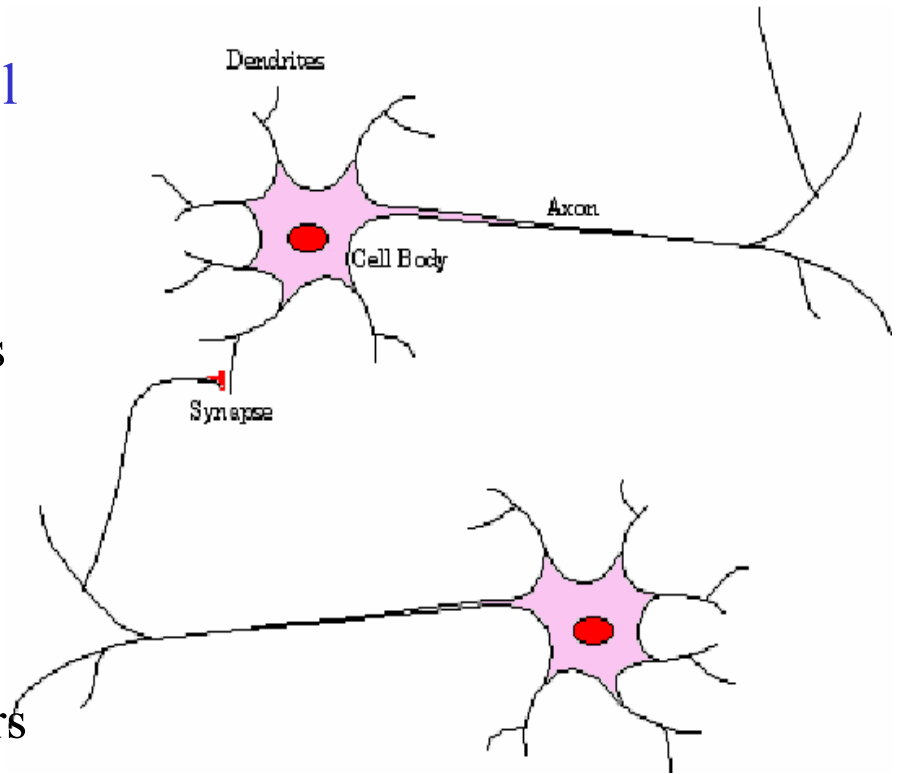
# The Neuron

- **The dendrites:**
  - Tree-like receptive networks of nerve fibers that carry electrical signals into the cell body
- **The cell body:**
  - Effectively sums and thresholds these incoming signals
- **The axon:**
  - A single long fiber that carries the signal from the cell body out to other neurons



# The Neuron

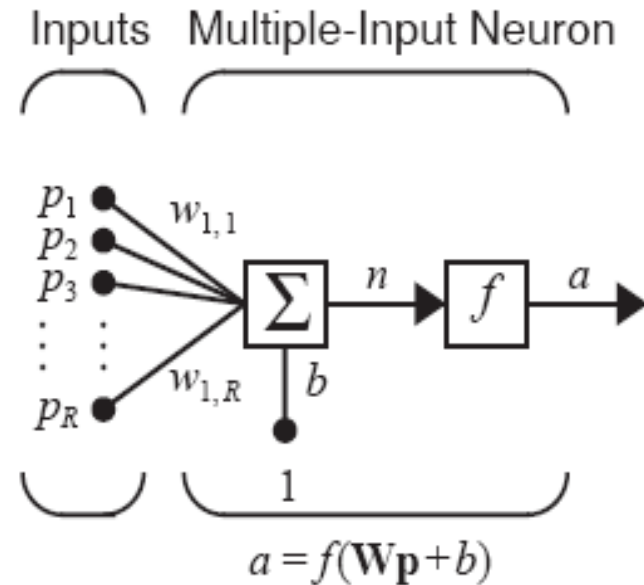
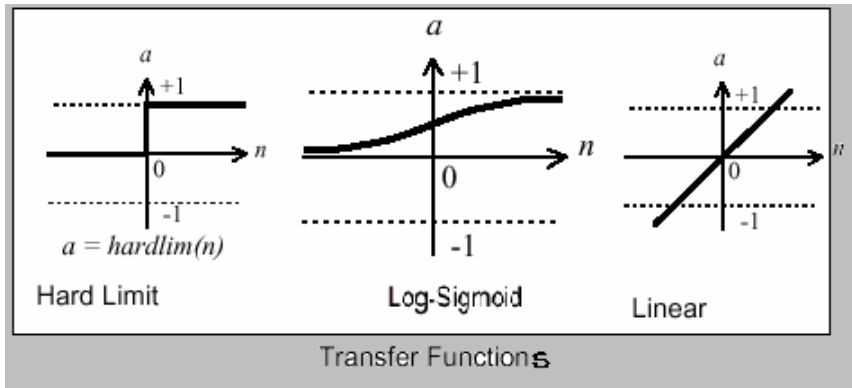
- The brain uses massively parallel computation that has
  - Approximately  $10^{11}$  neurons in the brain
  - Approximately  $10^4$  connections per neuron
  - Some of the neural structure is defined at birth.
  - Other parts are developed through learning, as new connections are made and others waste away



# Similarities: Biological & Artificial

- The characteristics of brain function have motivated the development of the **artificial neural networks**
- Simple computational devices that are highly interconnected
- Connections between neurons determine the **function of the network**
- These can be trained by **learning rules** (training algorithm): a procedure for modifying the **weights** and **biases** of a network

# Artificial Neuron

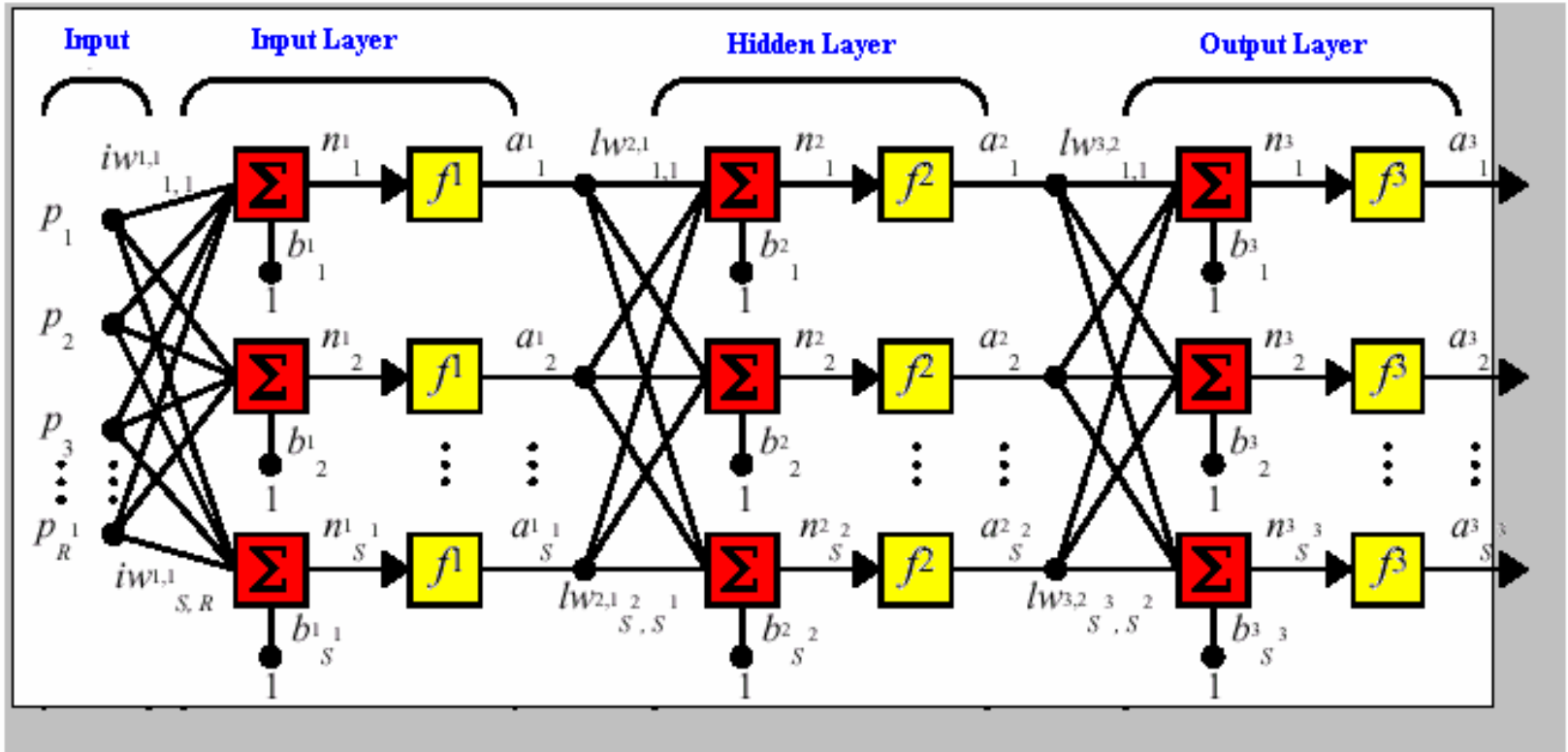


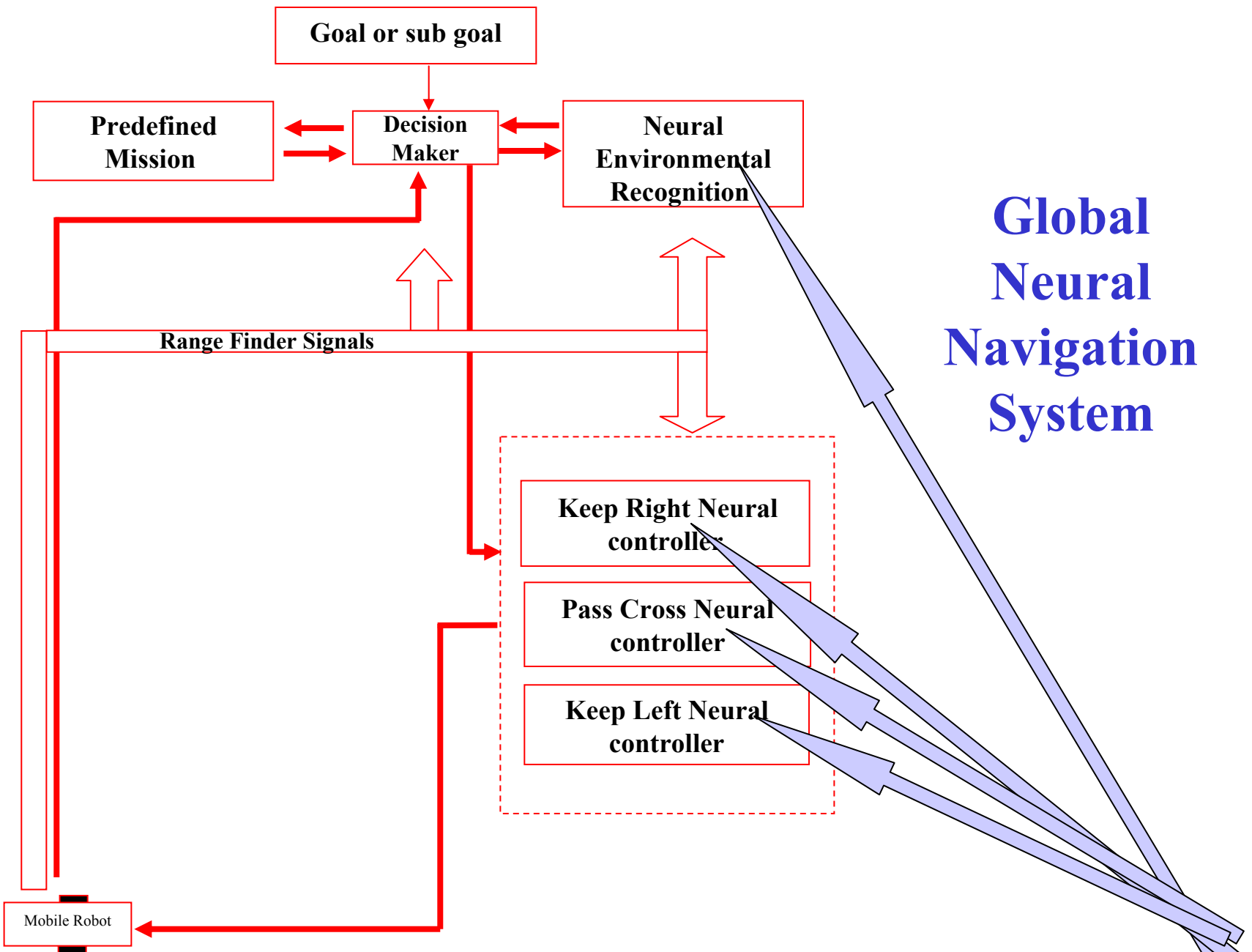
The weighted sum of the input is biased and then activated by a transfer function

$$n = w_{1,1}p_1 + w_{1,2}p_2 + \dots + w_{1,R}p_R + b = \mathbf{W} \times \mathbf{P} + b$$

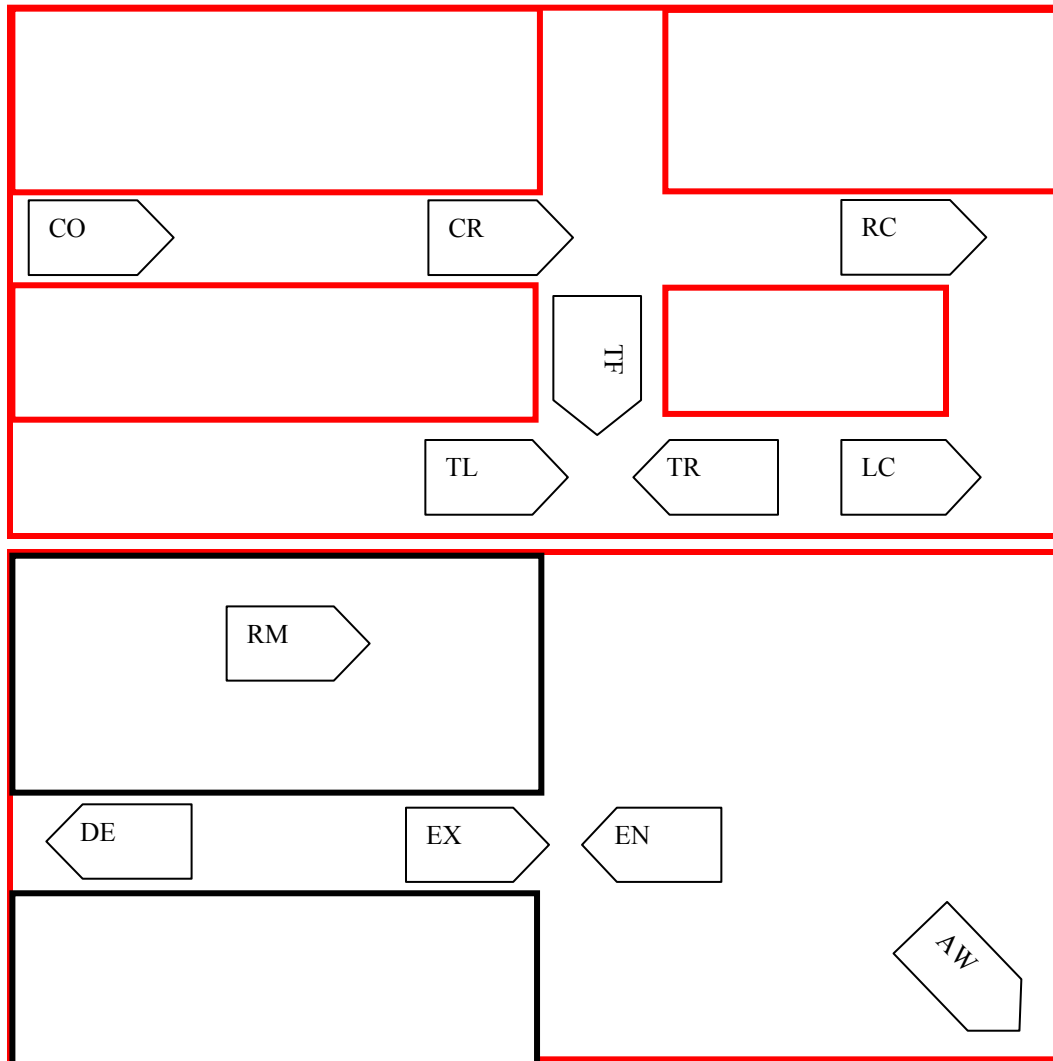
$$a = f(n)$$

# Multi Layer Neural Network





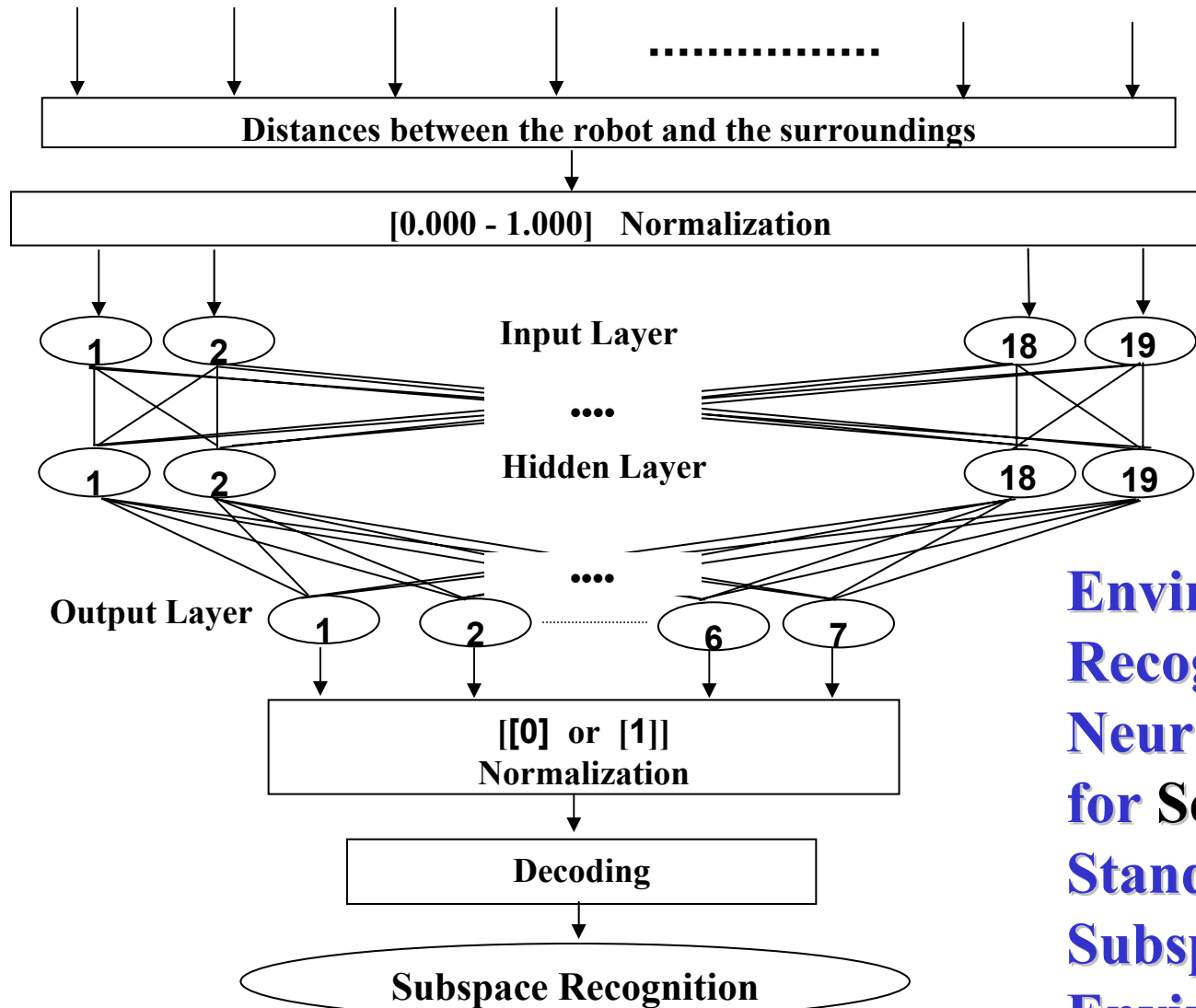
# Environmental Classification



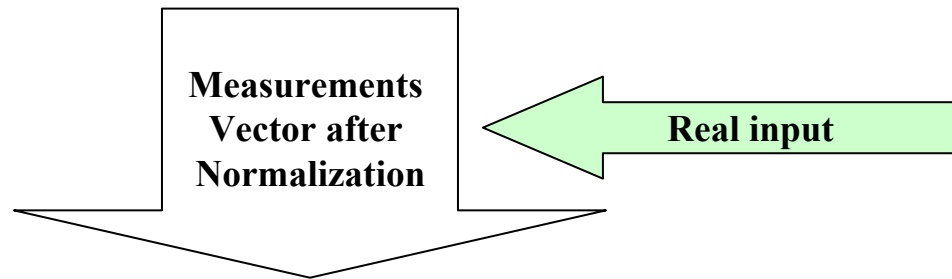
- **Related seven subspaces of the first neural network**

- **Related five subspaces of the second neural network**

# Measurements

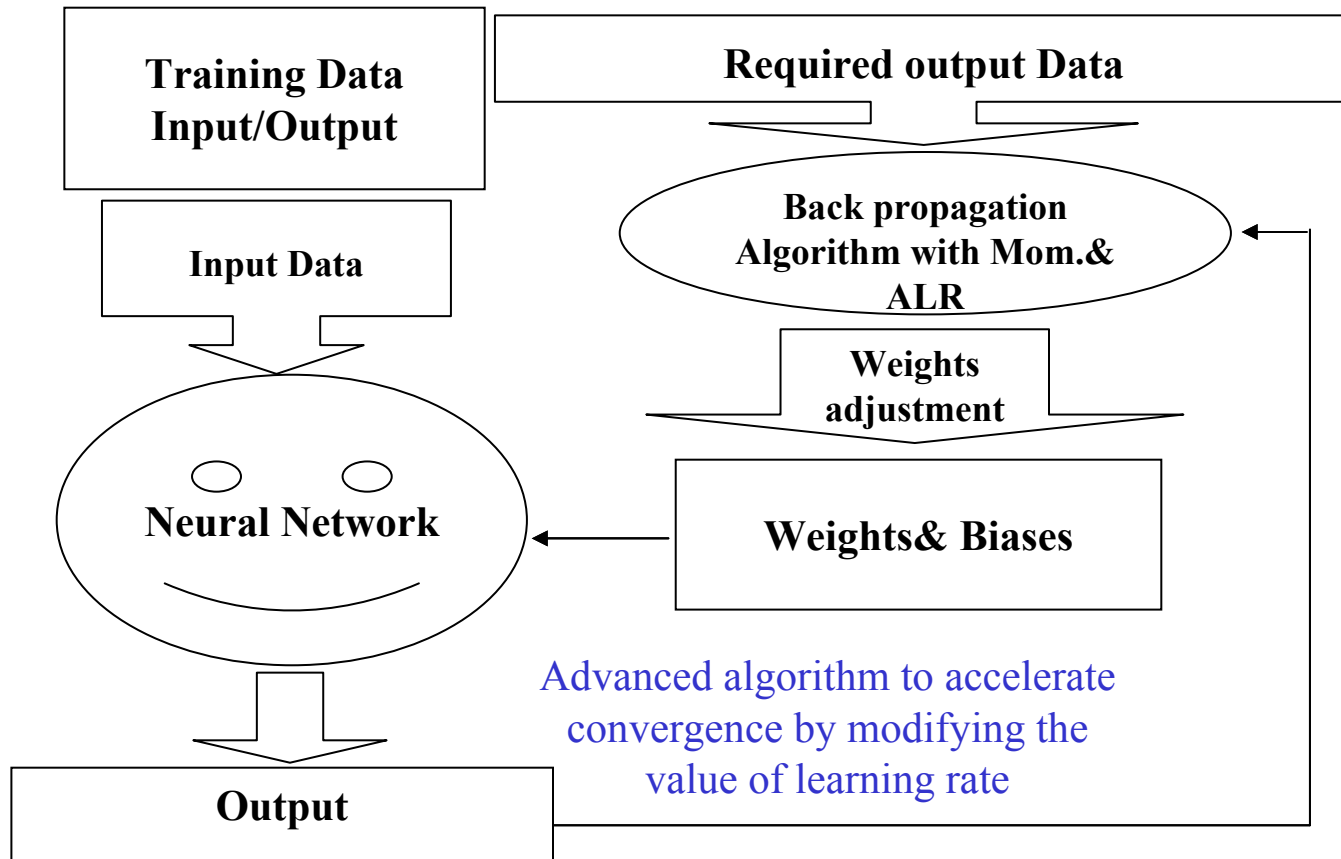


**Environmental  
Recognition  
Neural Network  
for Seven  
Standard  
Subspace  
Environment**

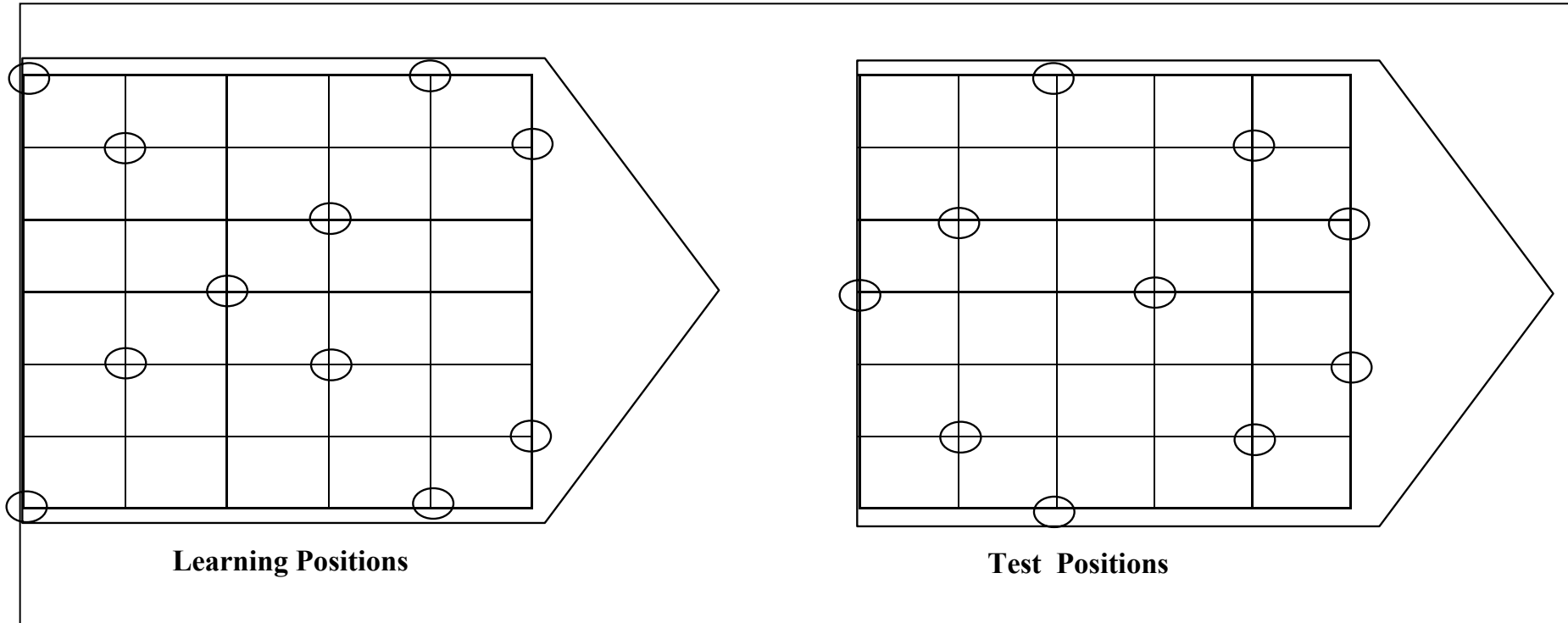


0.110	0.114	0.128	0.154	0.218	0.626	1.000	0.626	0.218	0.154	0.126	0.114	0.110	0.114	0.128	0.154	0.178	0.160	0.154					
<b>0.05</b>		<b>0</b>			<b>0.9</b>			<b>0</b>		<b>0</b>			<b>Decimal output</b>										
0		0			1			0		0			<b>Binary output</b>										
<b>16</b>																		<b>Decimal number</b>					
<b>RC</b>																		<b>Subspace</b>					

# Training Technique

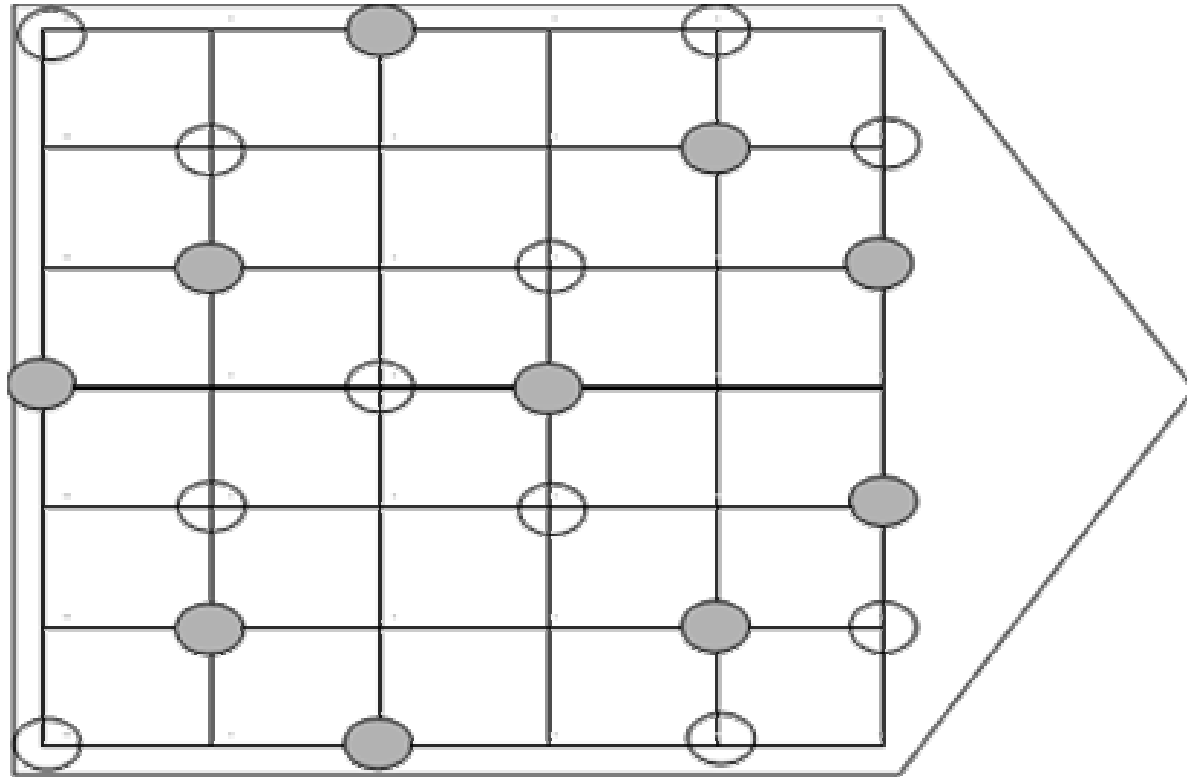


# Database for Network Training

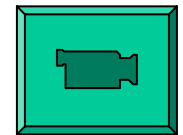
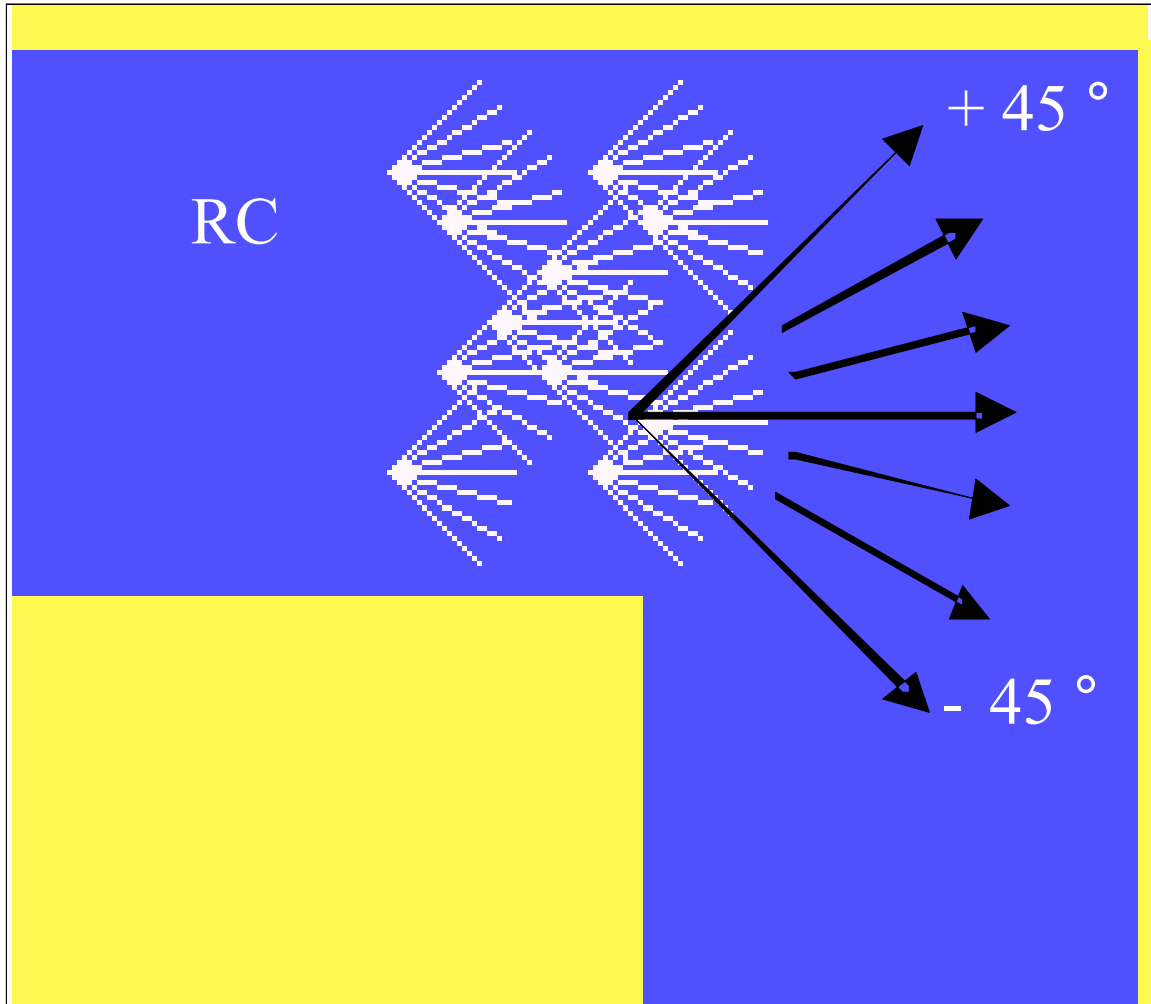


# Database for Network Training

$$\begin{pmatrix} P_1 \\ \cdot \\ P_i \\ \cdot \\ P_Q \end{pmatrix} \quad \begin{pmatrix} Y_1 \\ \cdot \\ Y_i \\ \cdot \\ Y_Q \end{pmatrix}$$



# Location and front direction of a robot at RC subspace

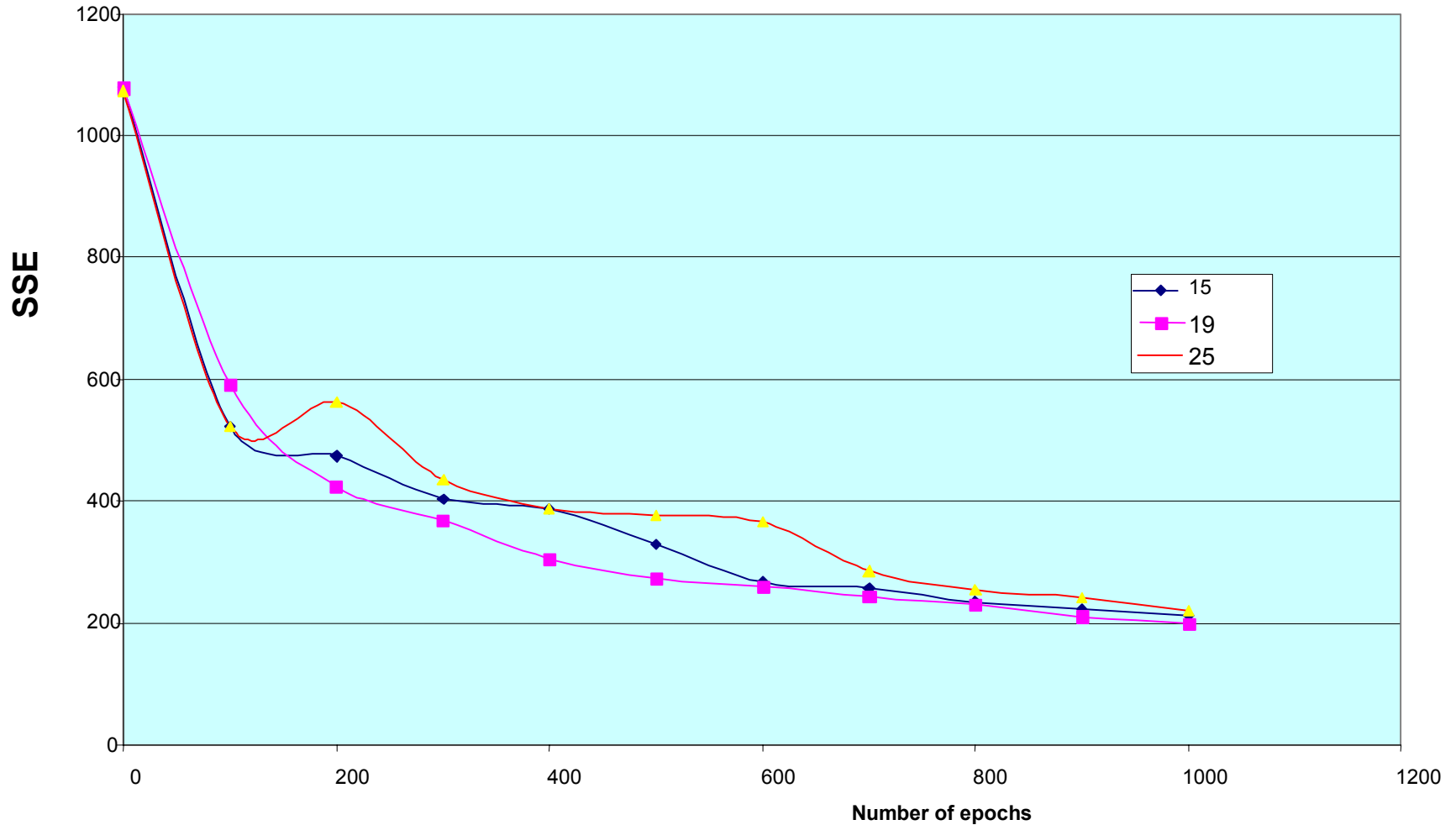


Model-all

# Learning results due to the number of the neurons in the hidden layer

Sum Squared Errors			Hidden layer neurons ◀
15	19	25	Number of epochs ▼
1080.21	1078.33.	1072.84.	0
522.614	592.367.	522.592	100
473.571	423.184.	562.261	200
404.212	367.908.	435.649	300
386.555	305.341.	387.444	400
328.819	272.872.	376.484	500
266.447	260.748.	365.736	600
257.014	242.501.	285.238	700
234.723	230.837.	253.906	800
223.216	210.28.	241.86	900
213.324	197.384	219.604	1000

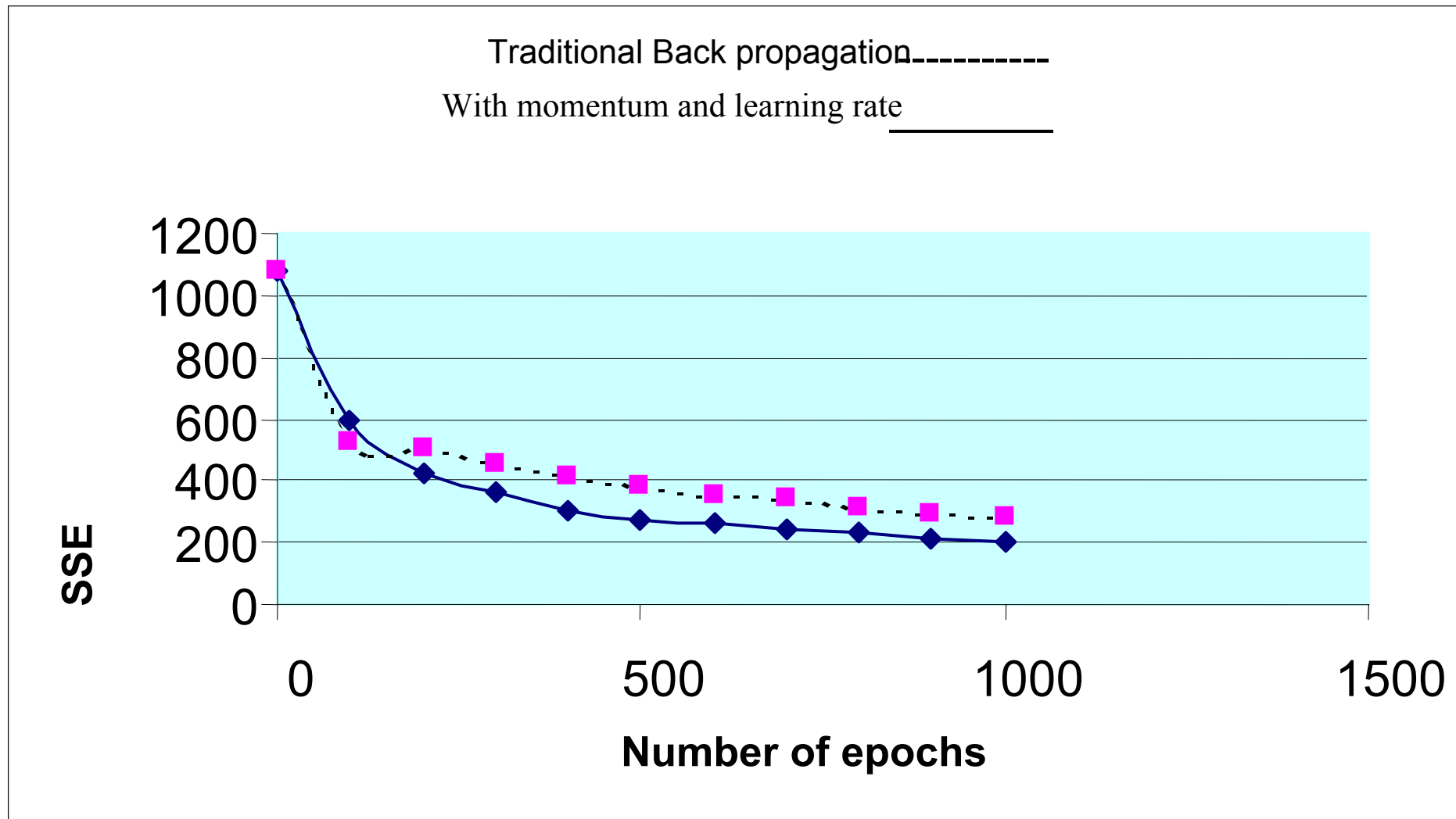
# Learning results due to the number of the neurons in the hidden layer



# Learning results due to training algorithm

Sum Squared Errors based on Hidden 19 Neurons		Training Method
BP with momentum with learning rate	Back Propagation (BP)	Training Epochs ▼
1075.75	1075.75	0
592.367	519.522	100
423.184	501.947	200
367.908	454.966	300
305.341	413.312	400
272.872	384.617	500
260.748	355.192	600
242.501	340.953	700
230.837	310.109	800
210.28	293.21	900
197.384	278.313	1000

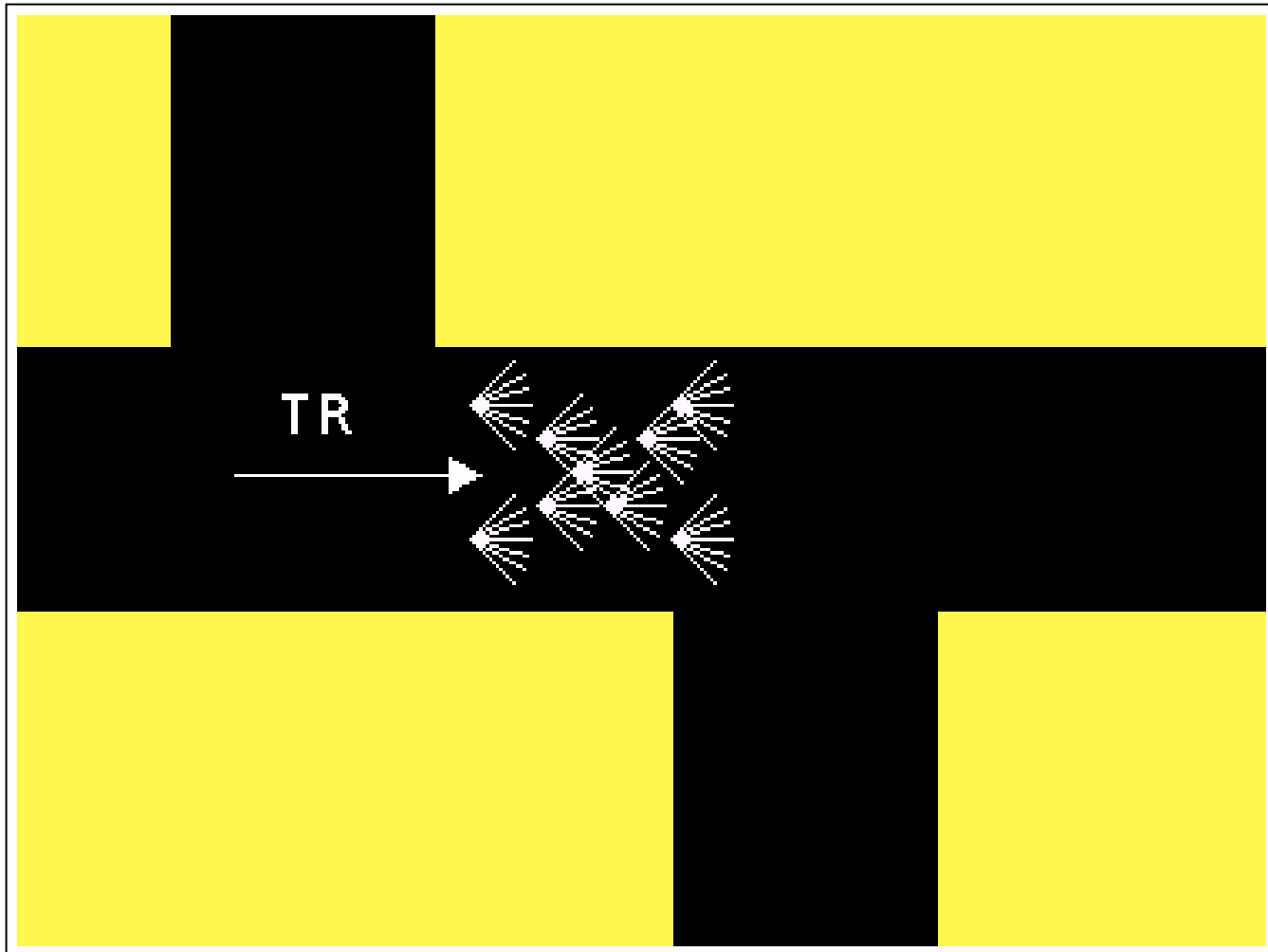
# Learning results due to training algorithm



# First NN testing results

Subspace	Actual output							Desired output						
	7	6	5	4	3	2	1	7	6	5	4	3	2	1
CO	0	0	0	0	0	0	0.25	0	0	0	0	0	0	1
CR	0	1	0	0	0	0.01	0	0	0	0	0	0	1	0
CR	0	0.08	0.01	0	0	0.29	0	0	0	0	0	0	1	0
CR	0	0	0.13	0	0	0.43	0	0	0	0	0	0	1	0
CR	0	1	0	0	0	0	0	0	0	0	0	0	1	0
CR	0	0.99	0	0	0	0.05	0	0	0	0	0	0	1	0
RC	0	0	0	0.15	0.28	0	0	0	0	0	0	1	0	0
TR	0	0	0.58	0	0	0.14	0.01	0	0	1	0	0	0	0
TR	0	0.75	0	0	0	0	0	0	0	1	0	0	0	0
TR	0	0	0.05	0	0	0.01	0	0	0	1	0	0	0	0
TL	0	0.09	0	0	0	0	0	0	1	0	0	0	0	0
TL	0	0.02	0	0	0	0	0	0	1	0	0	0	0	0

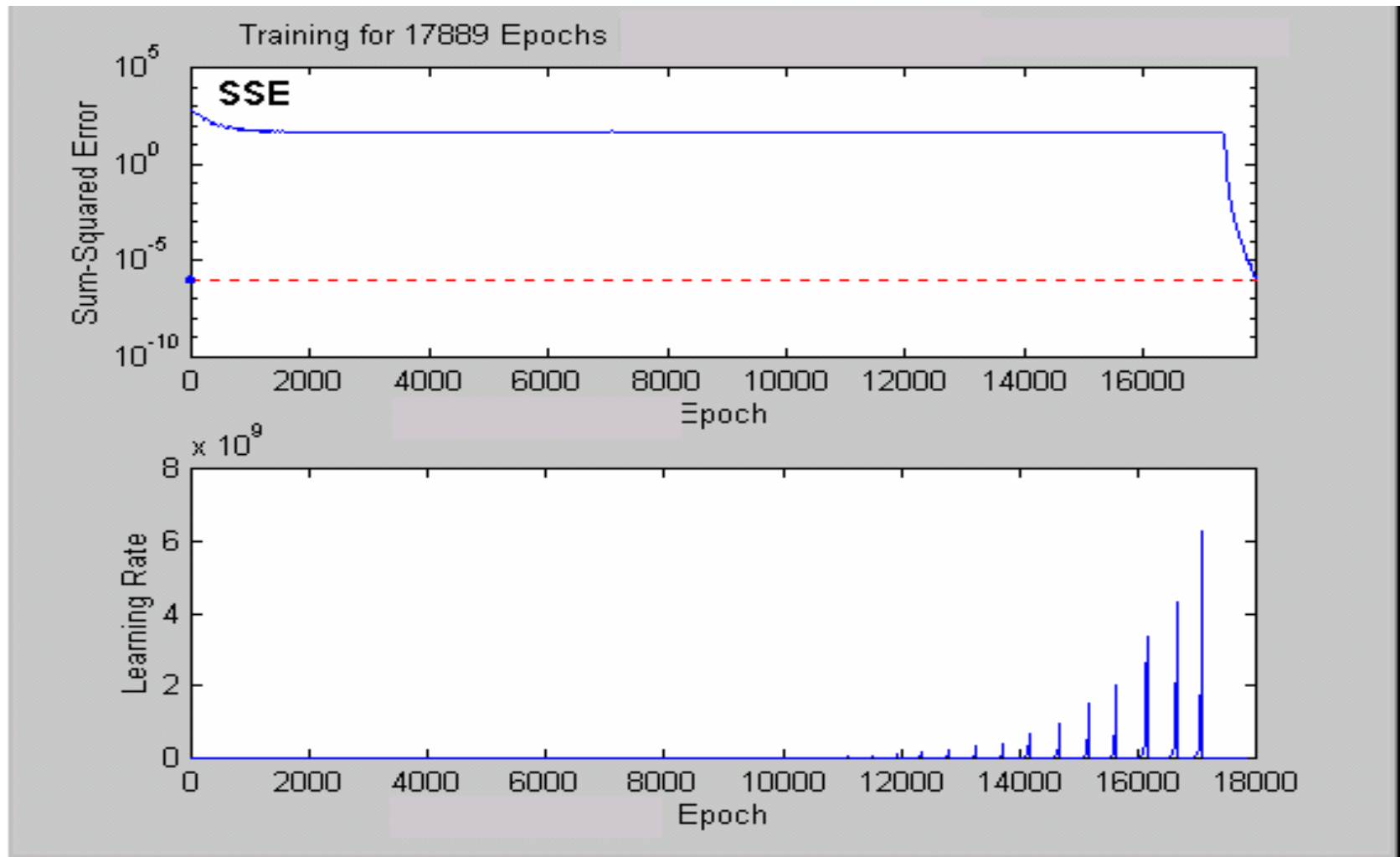
Training was repeated for the first NN at 628 vectors



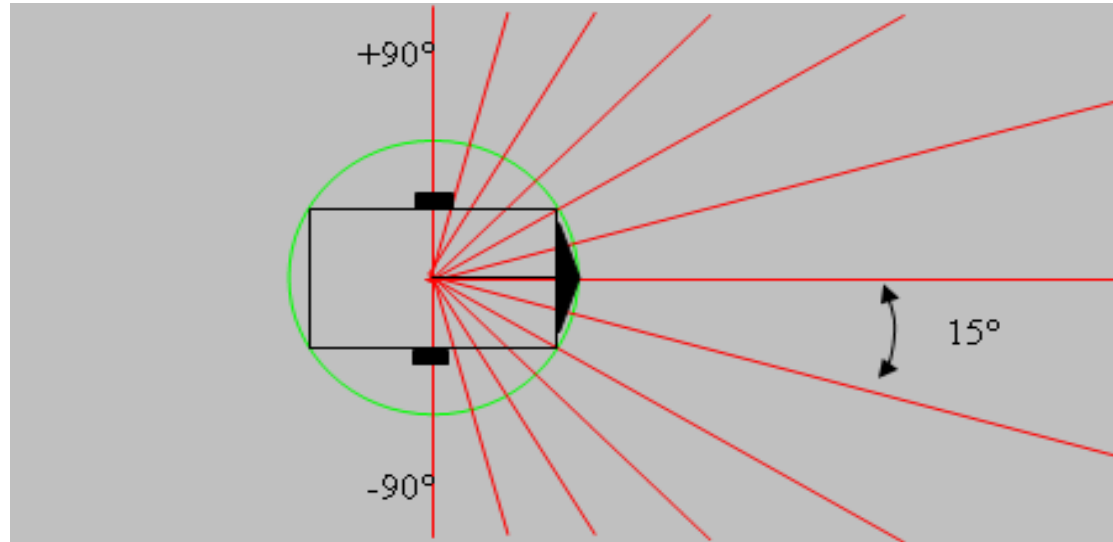
# 19\*19\*5 NN Learning

Training Epochs	Learning Rate	SSE
0	0.0001	591.874
1000	0.0549472	53.2159
2000	0.440266	45.2581
3000	62.9633	45.0052
4000	1.77814	45.0009
5000	23.879	45.0001
6000	224.472	45
7000	191648	45
8000	762568	45
9000	1.14743e+006	45
10000	767349	45
11000	325820	45
12000	9.06048	45
13000	637.057	45
14000	205680	45
15000	324959	45
16000	1.73277e+006	45
17000	7.73546e+007	45
17889	51457.5	9.99353e-007

# 19\*19\*5 NN Learning

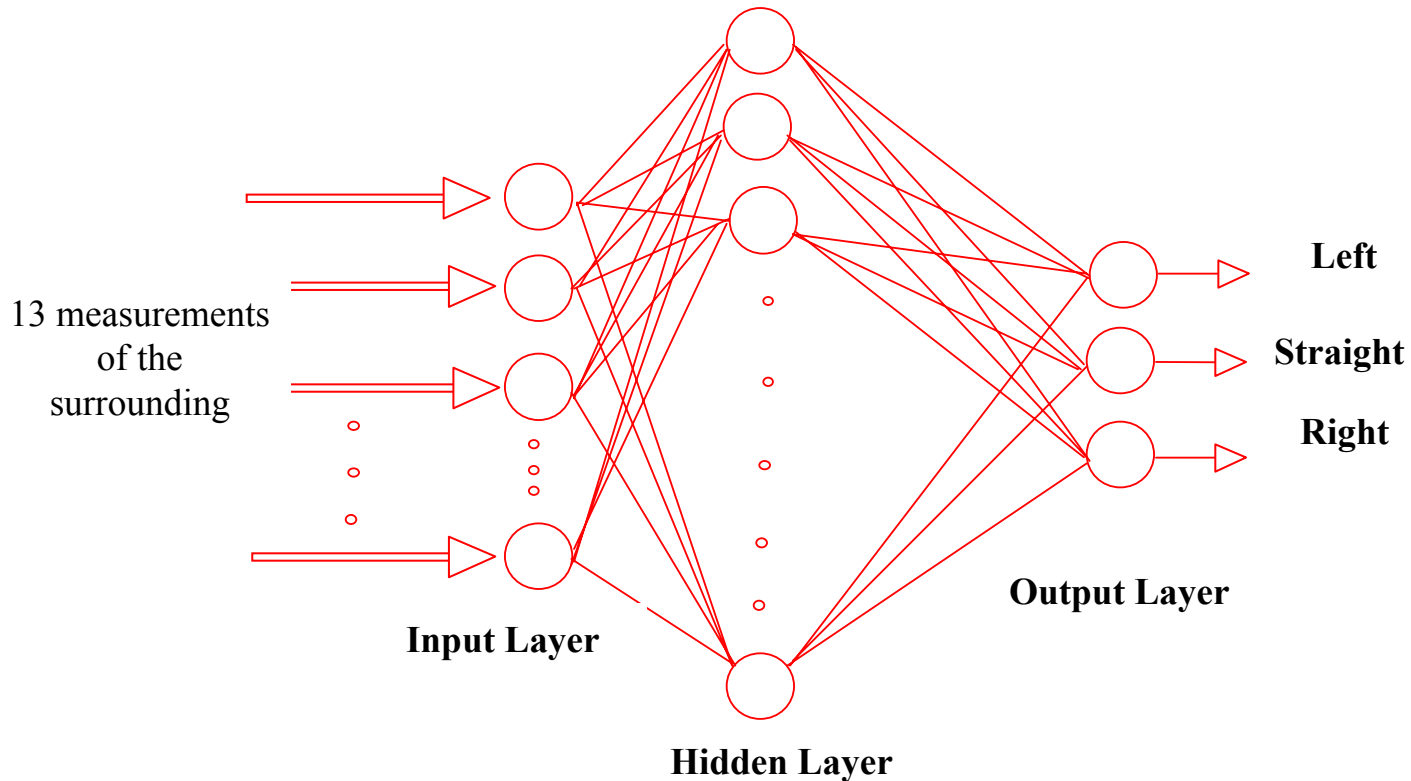


# Design of the Local Navigation Neural Controllers

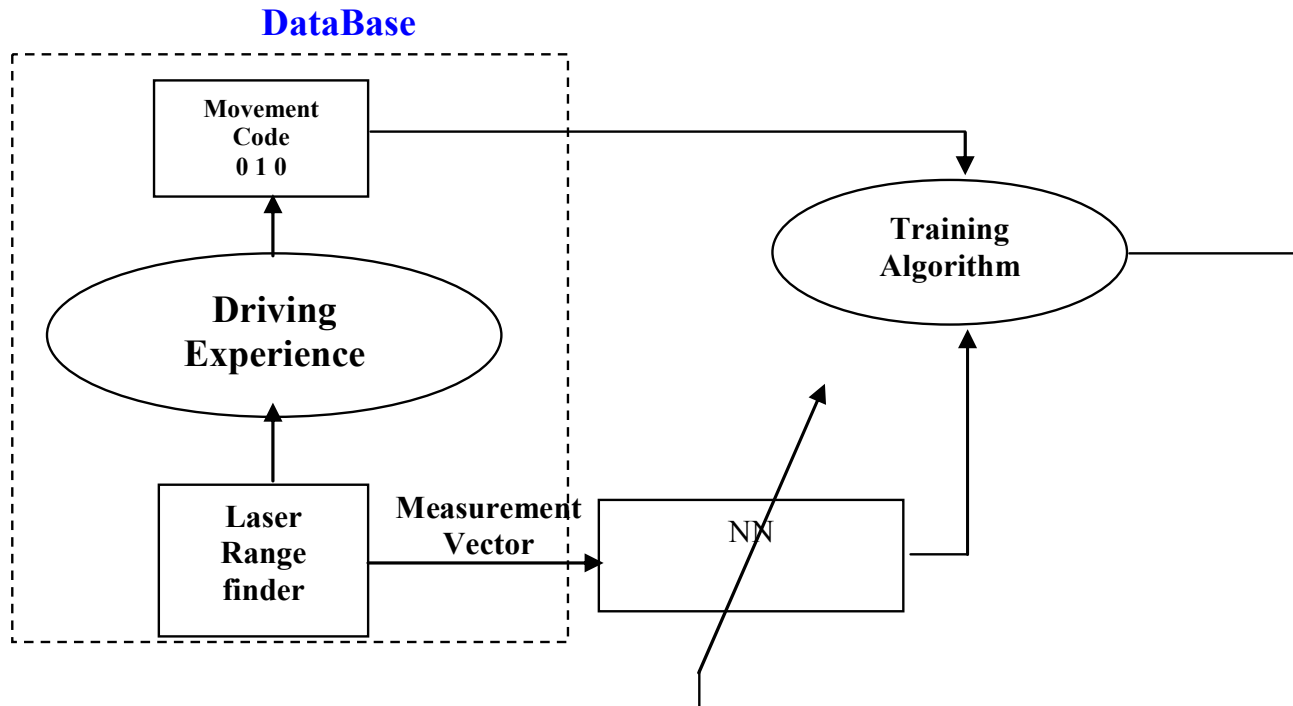


- The measuring range of the **laser range-finder** was limited to **3.5 meters**, to achieve higher resolution in distance measurement
- This helped in distinguishing the **nearby obstacles**, when the measurement vector was normalized to a value **between 0 and 1**
- Measurements were taken within a measuring **sector of  $-90^\circ$  to  $90^\circ$**  to the frontal direction of the robot, by intervals of  $15^\circ$ .

# Design of the Local Navigation Neural Controllers

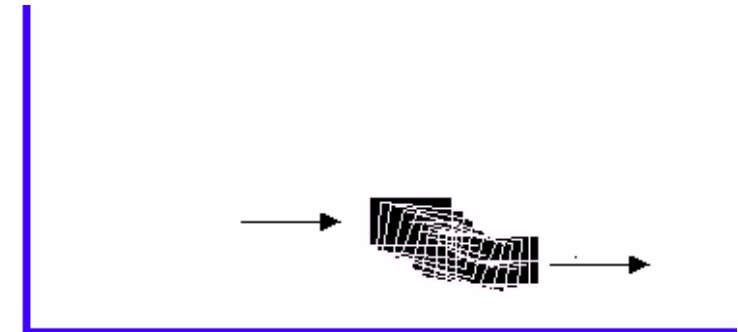
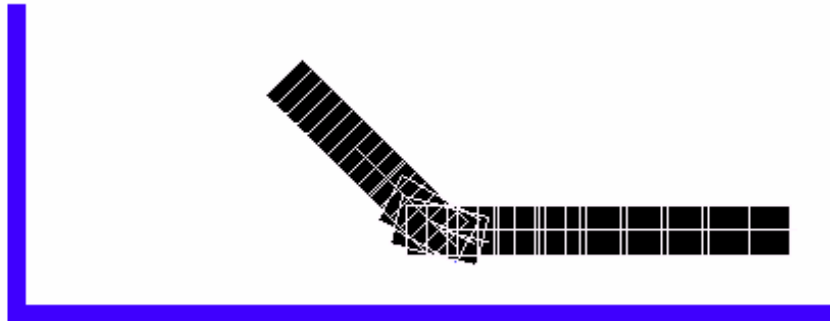
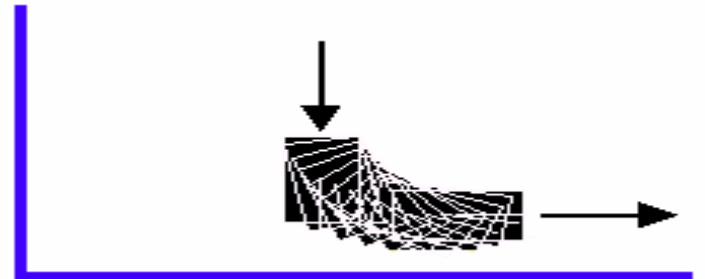
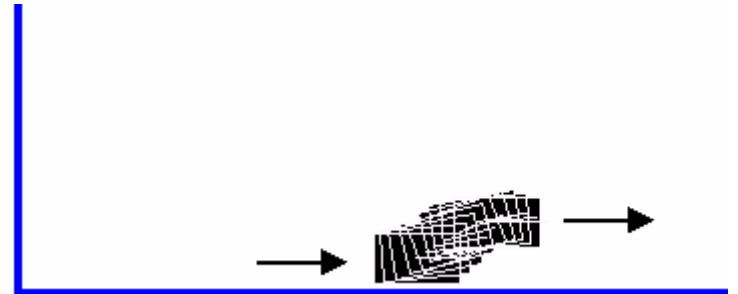
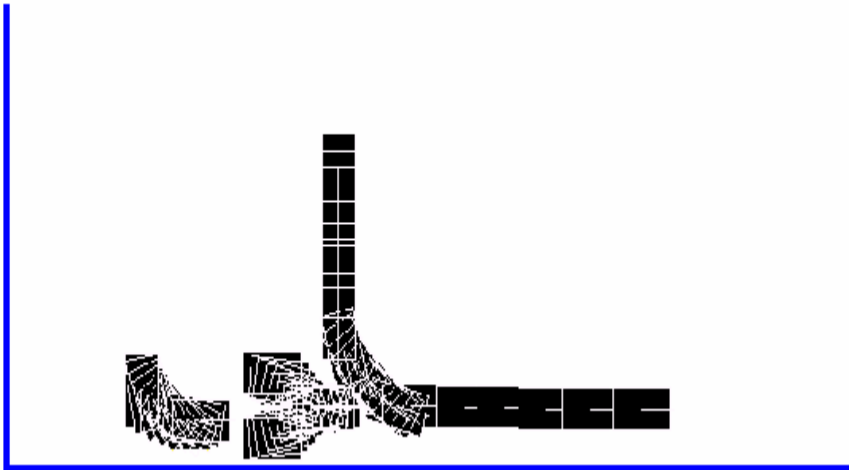


# Design of the Local Navigation Neural Controllers

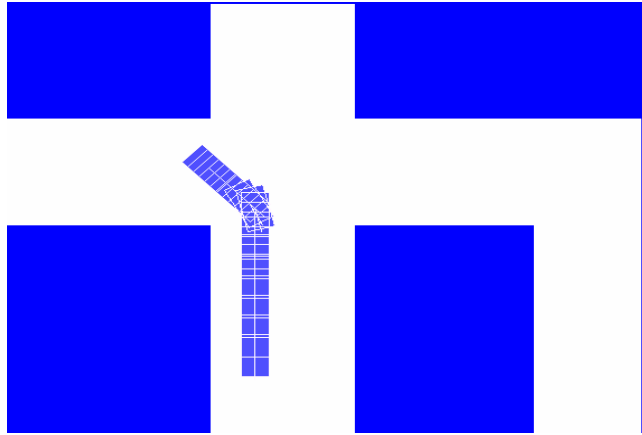


## Training Technique

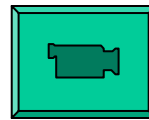
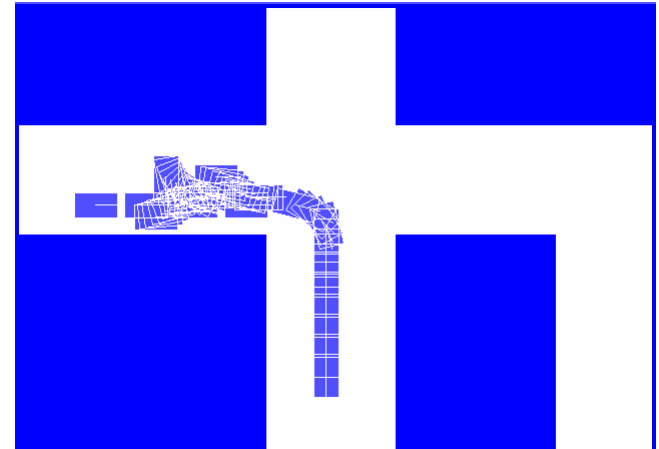
# Keep Right Training



# Keep Right Training



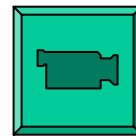
Training to  
turn and  
keep right at  
a cross road



Local-

Training

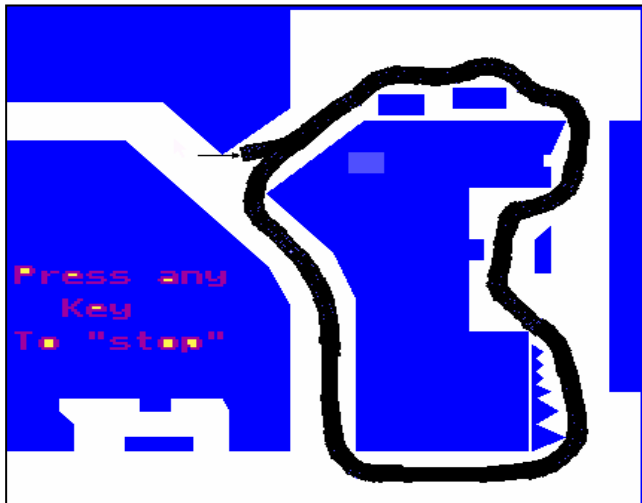
Hybrid-T1



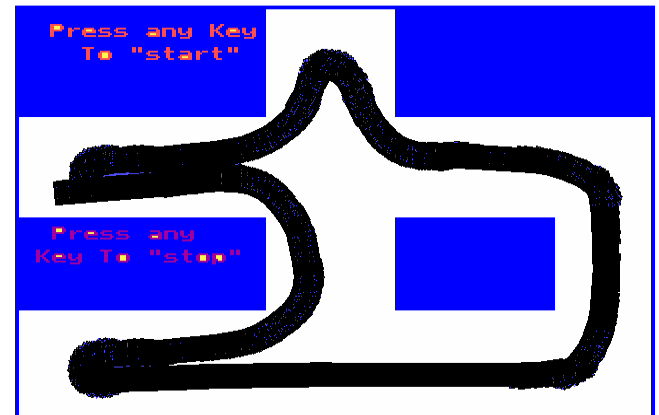
MovmG

Factory

Keep\_R



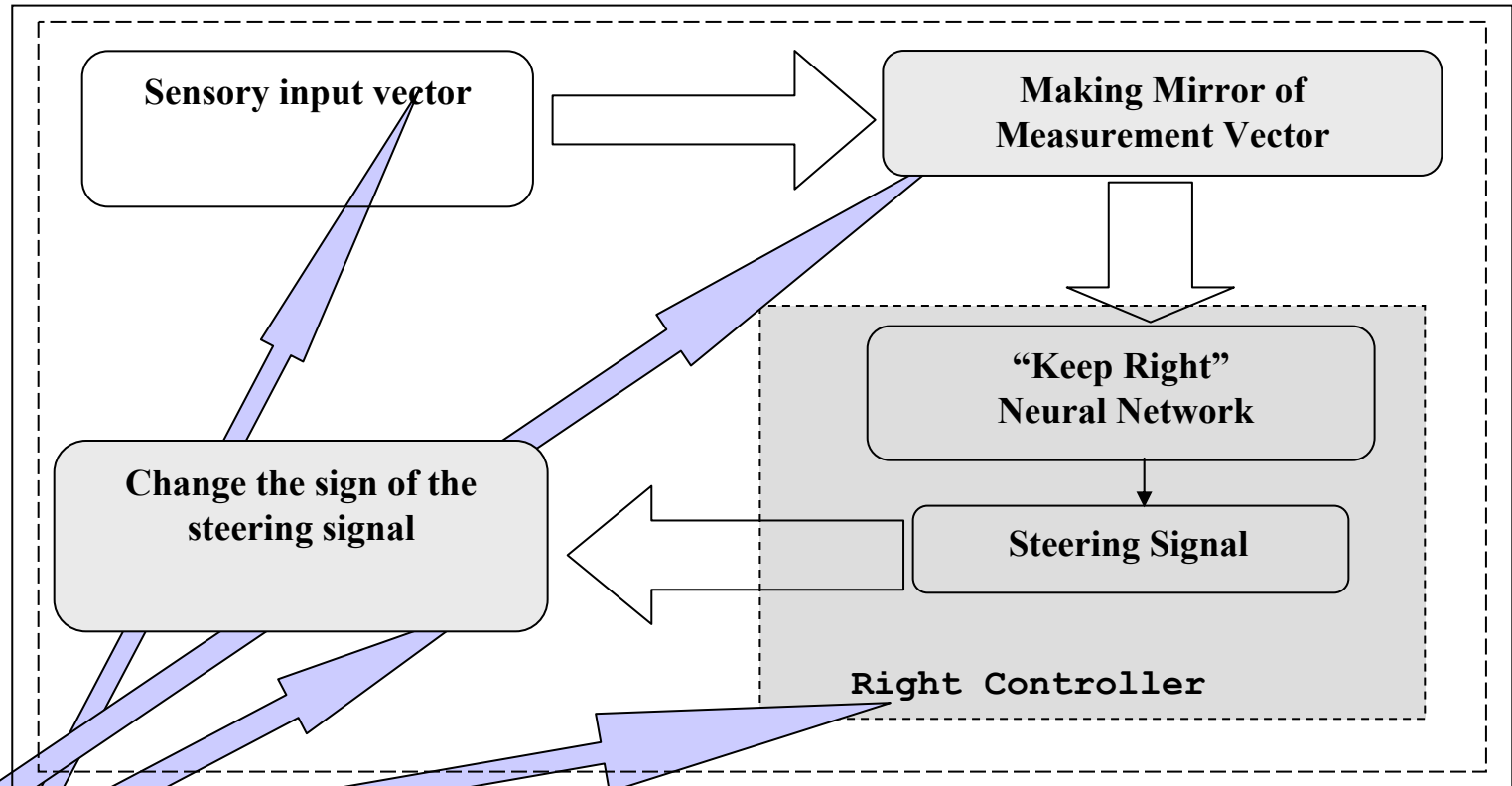
“Keep Right” Neural Controller Performance  
at a non- Standard Environment



“Keep Right” Neural Controller  
Performance at a Standard Environment

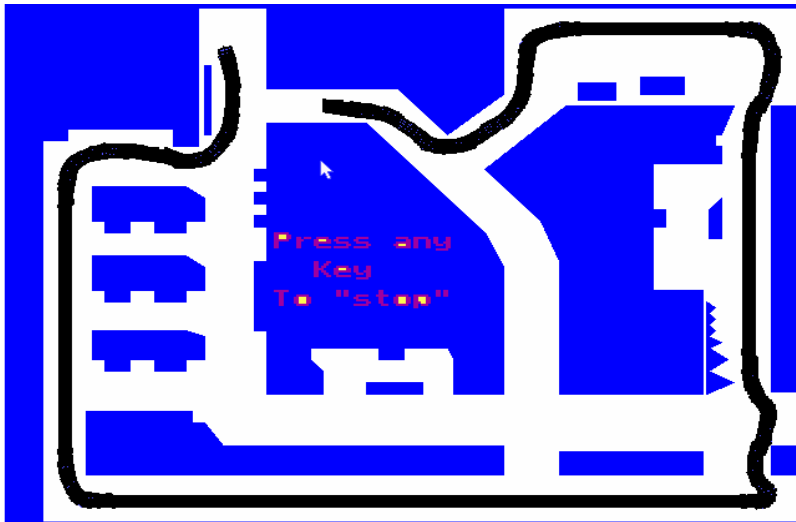


# Keep Left Techniques

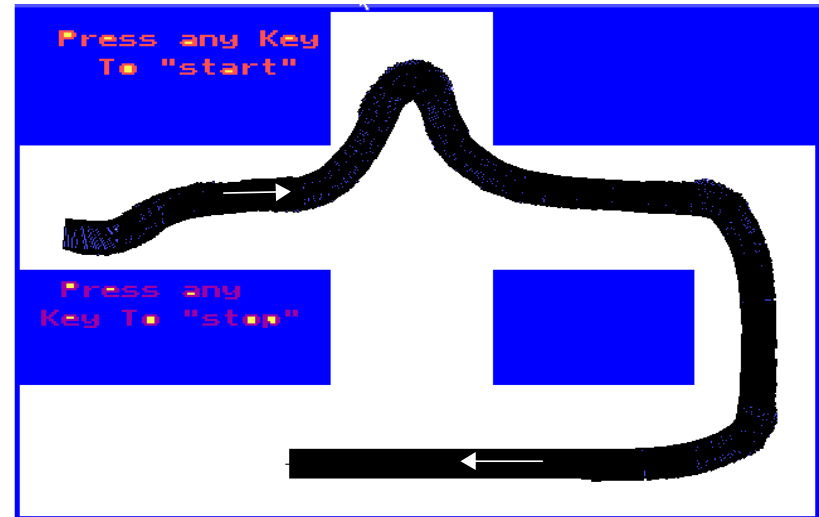


Algorithm of operating “Keep Left” controller

# Keep Left Techniques

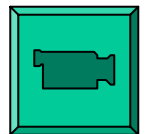


“Keep Left” Neural Controller Performance at non-standard environments



“Keep Left” Neural Controller Performance at standard environments

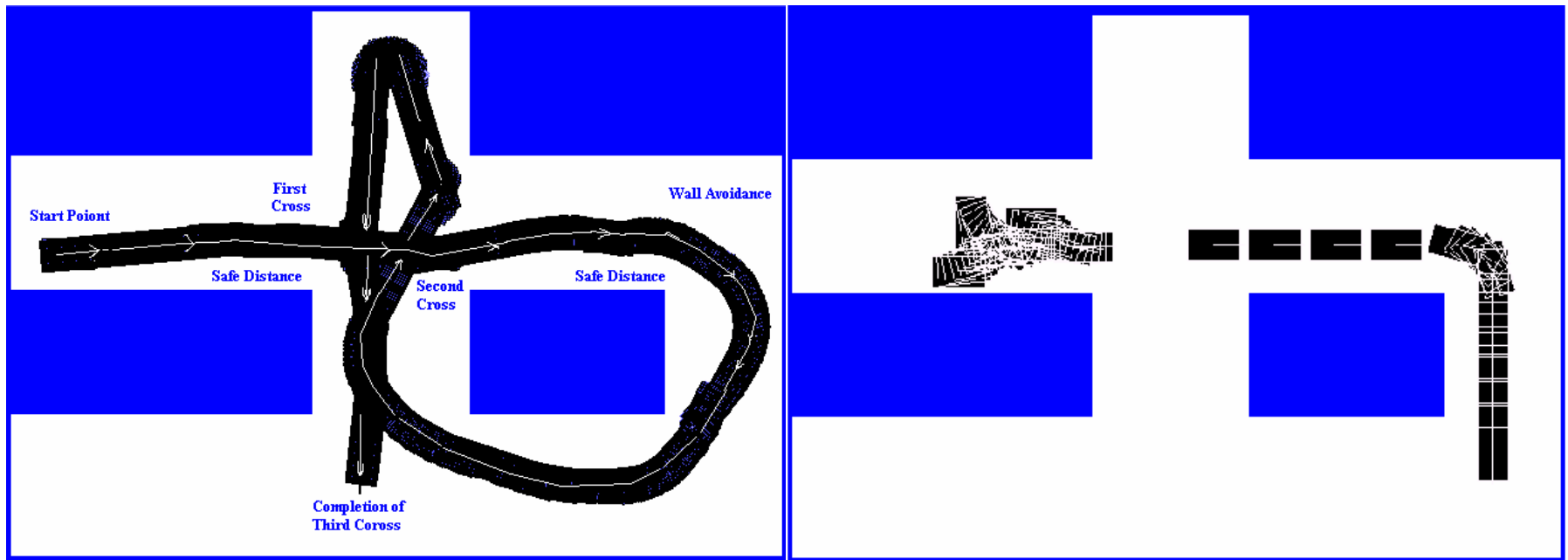
“Keep Left” Neural Controller Test



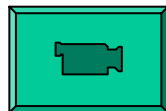
MovmG

Factory  
Keep\_L

# Pass-A-Cross Training



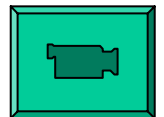
“Pass a cross” neural controller test



**MovmG**

**P\_Cross**

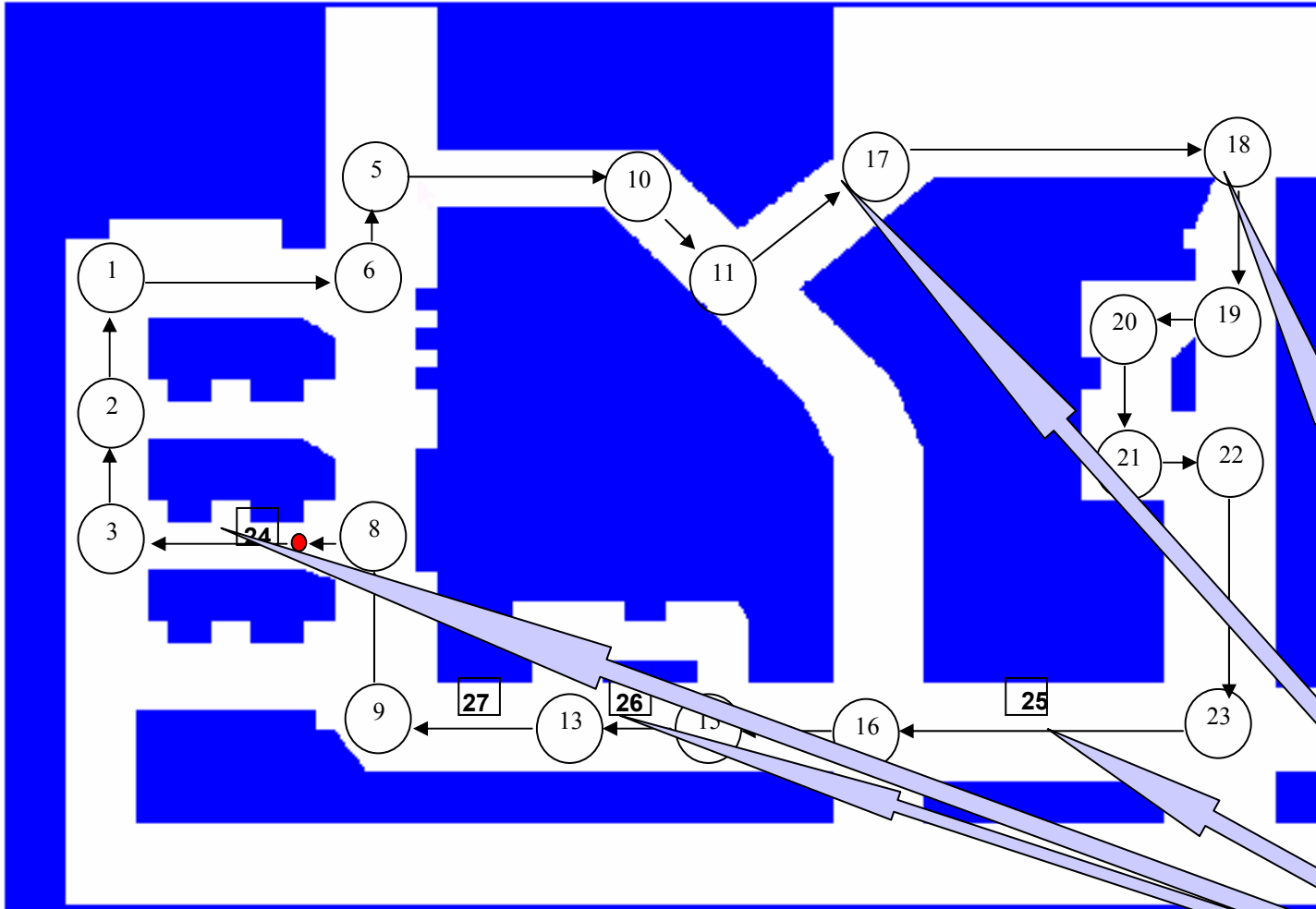
“Pass a cross” neural controller training



**Local-Training**

**Hybrid-T3**

# Path Planning and Tracking

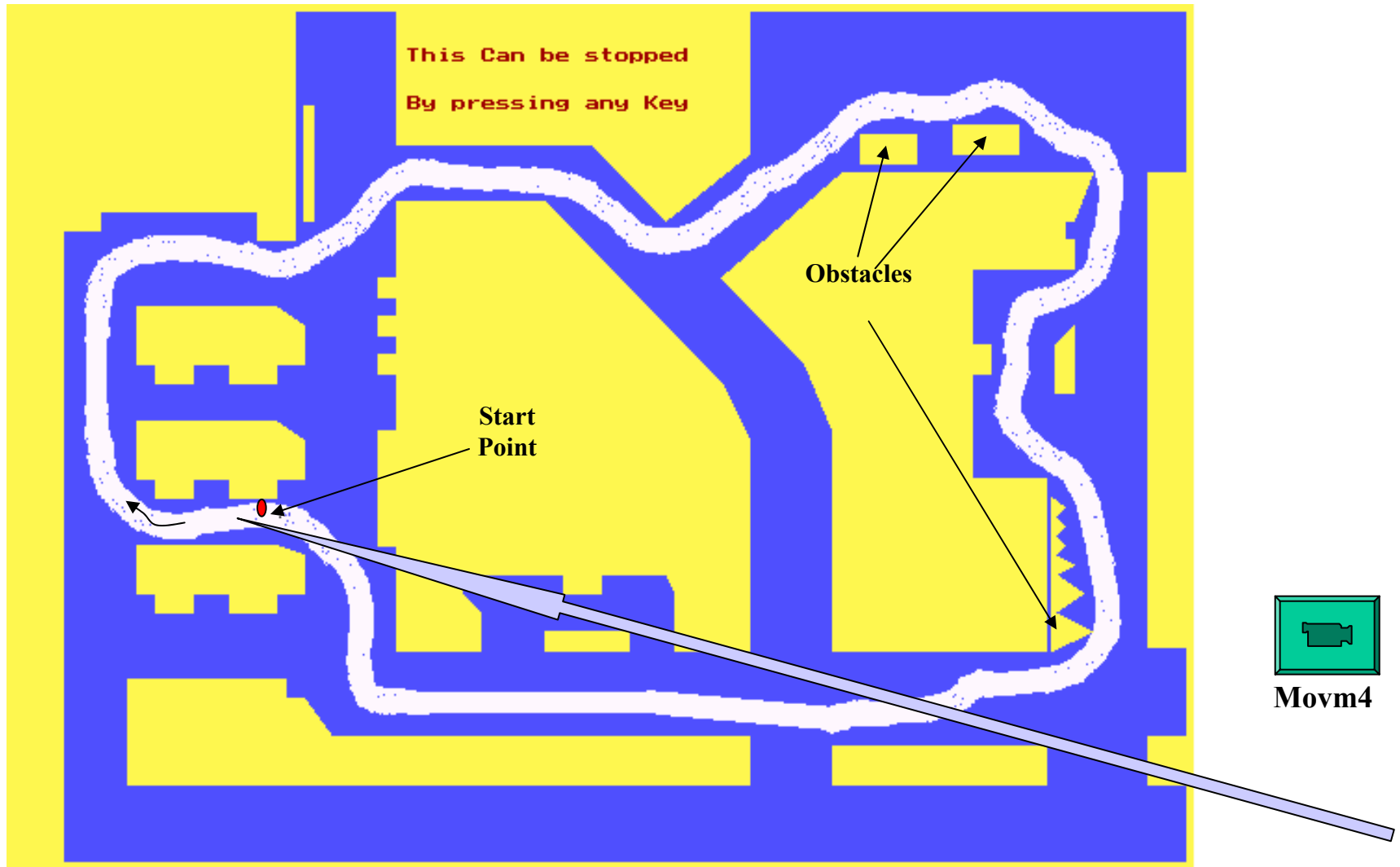


**Predefined path of a start position, sub goals, and a main goal**

# Path Planning and Tracking

LNNC	Selected ERNN	Subspace No. in Fig. 18	Subspace	Path
R	1	24	CO	1
R	1	3	TF	2
L	1	2	TR	3
R	1	1	RC	4
L	1	6	TF	5
R	1	5	TR	6
R	1	10	RC	7
L	1	11	TL	8
R	2	17	EX	9
R	2	18	EN	10
R	1	19	TR	11
R	1	20	LC	12
R	1	21	LC	13
R	1	22	TF	14
R	1	23	CR	15
R	1	25	CO	16
L	1	16	CR	17
S	1	15	TR	18
L	1	26	CO	19
L	1	13	TR	20
L	1	27	CO	21
L	1	9	TR	22
R	1	8	TL	23
L	1	24	CO	24

# Neural system performance at a factory environment based in the predefined path

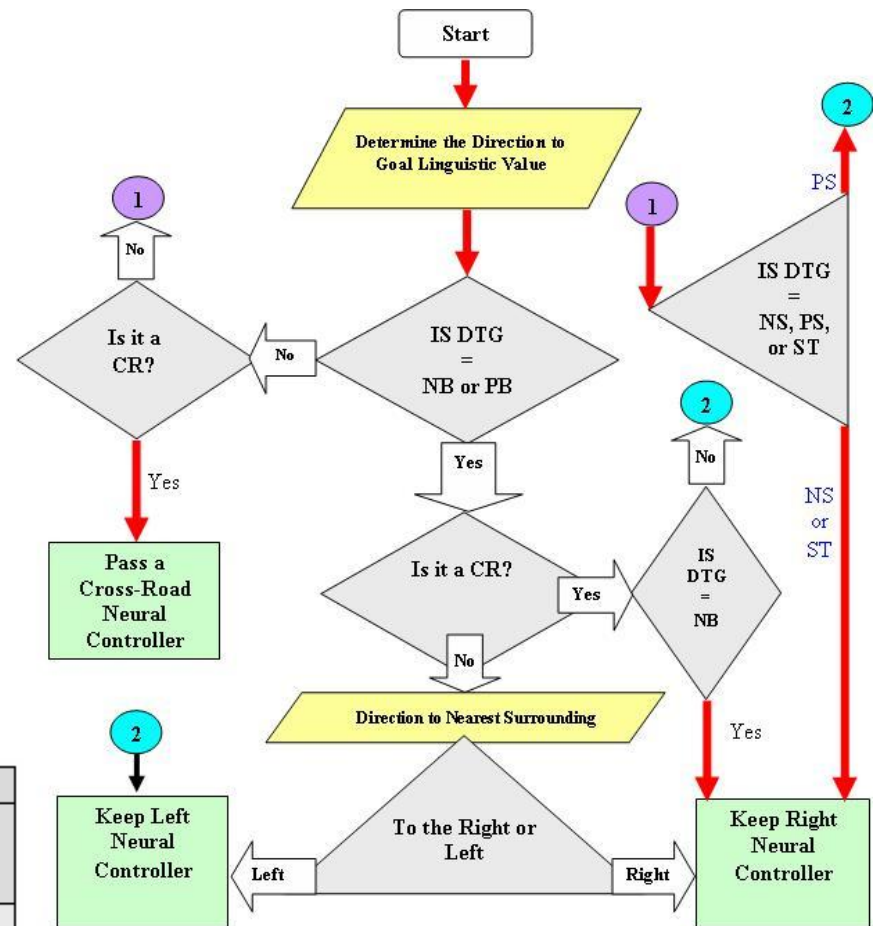


# Algorithm for Autonomous Path Planning

- Autonomously get to the main goal from the present position of the MR
- The decision maker used our ERNN1 to check if the MR faced a crossroad
- Calculates its direction to the goal (DTG) angle with respect to the current frontal direction of the MR

Linguistic values of the DTG angle

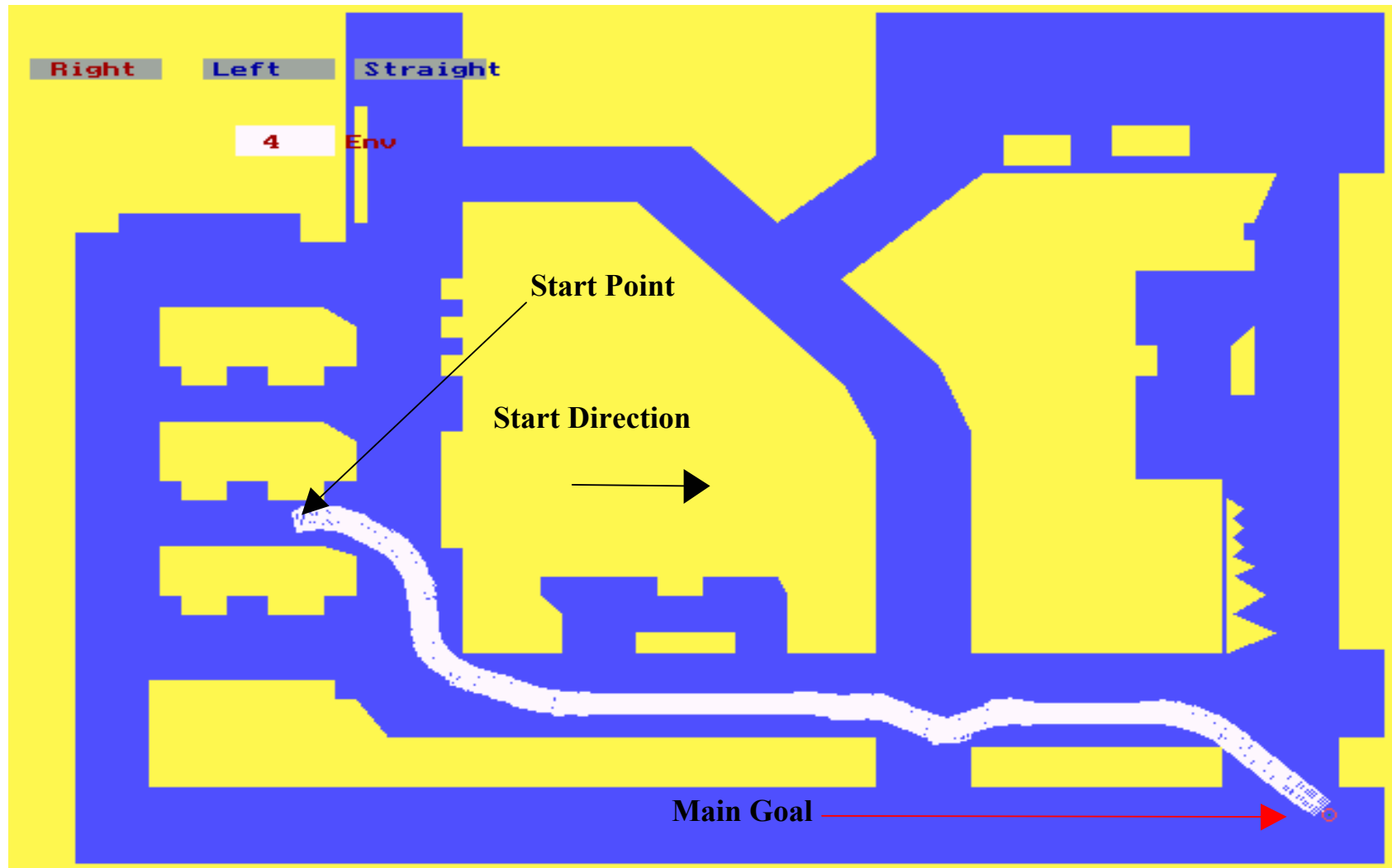
Linguistic values of									
Negative Big (NB)		Negative Small (NS)		Strait (ST)		Positive Small (PS)		Positive Big (PB)	
-180	-91	-90	-31	-30	30	31	90	91	180



# Signal Receiver

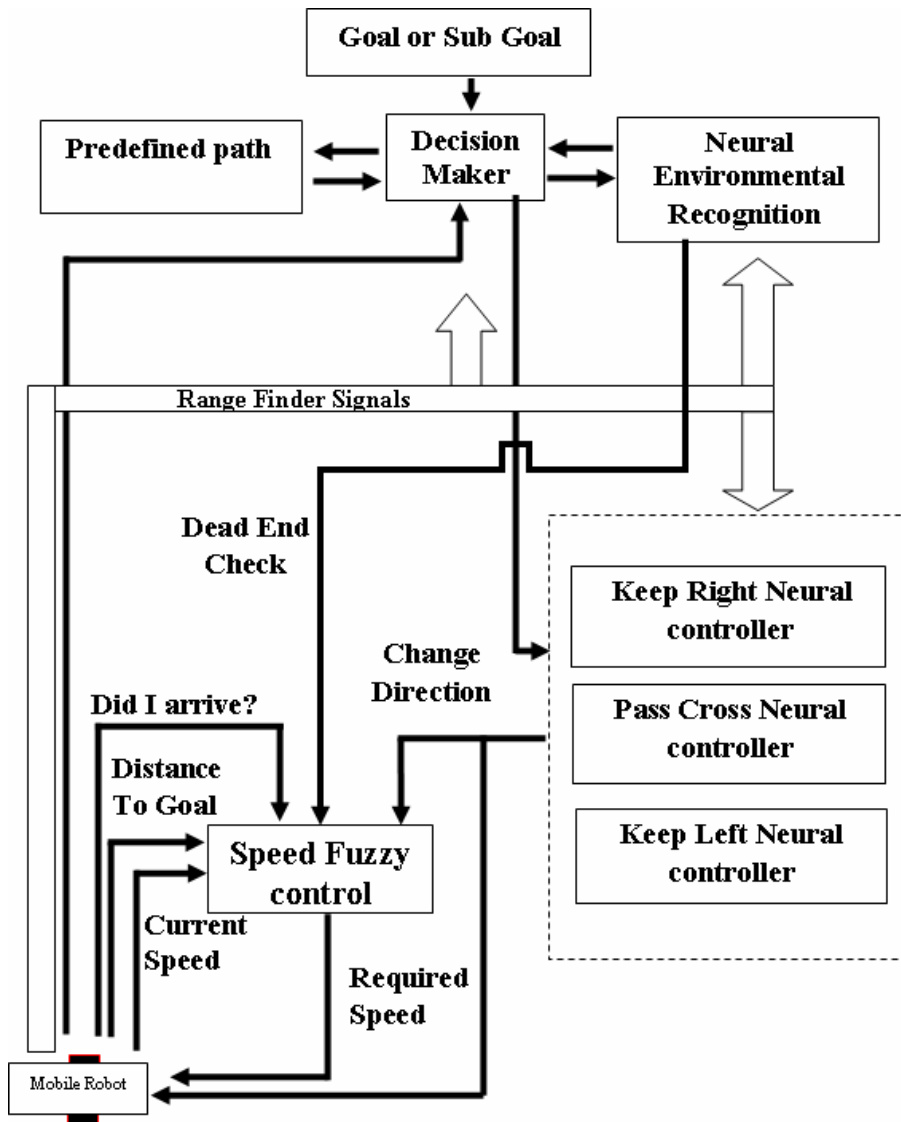
We considered that the MR was equipped with a signal receiver that can detect **coded signals** transmitted from a *goal location*, or a *MR's base* (in the case of returning back to its base location after executing its mission) and **decoded** them to determine the direction to the main goal or to its base

# Neural System Performance at a Factory Environment (Autonomously)





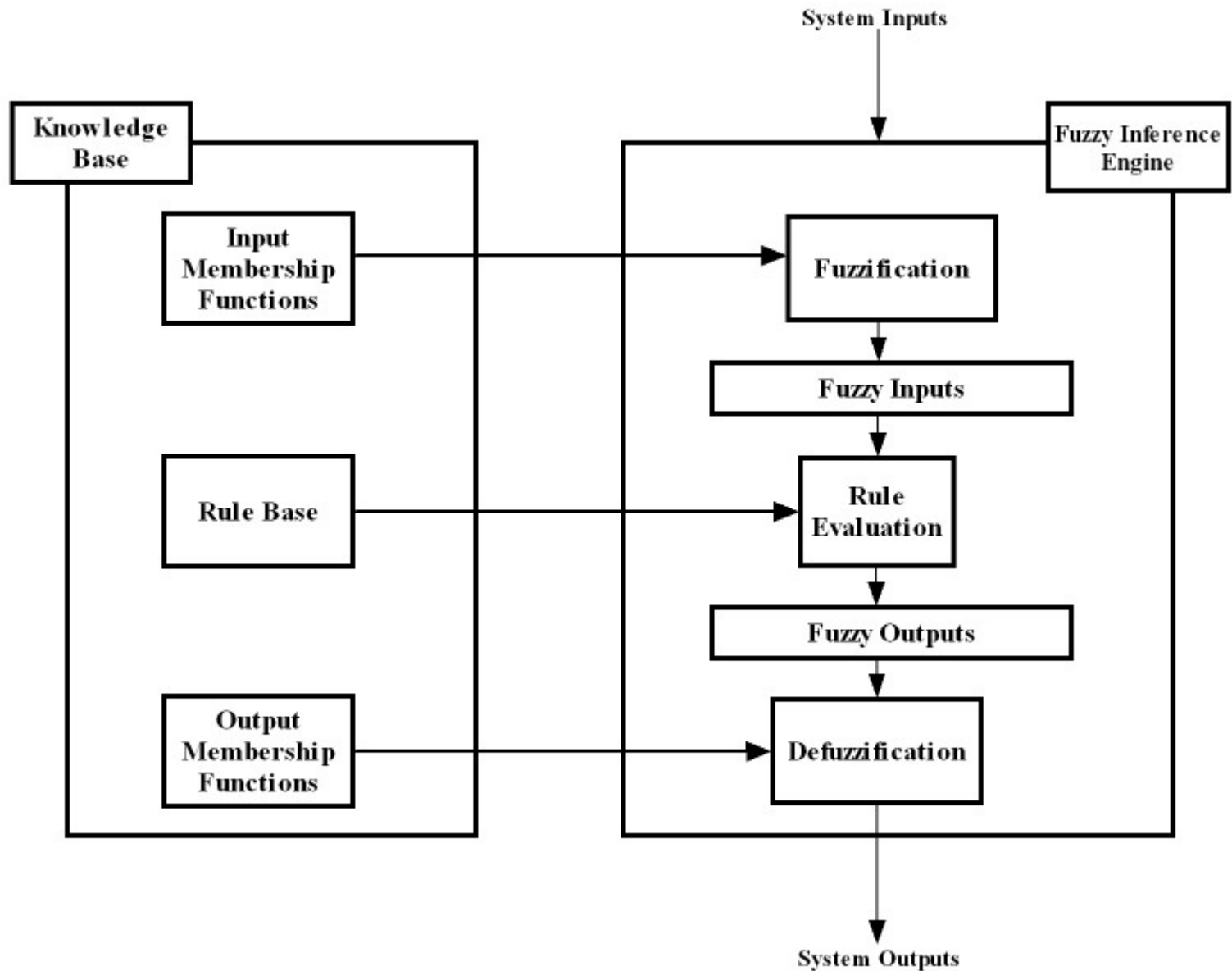
# Incorporated NNs and Fuzzy Logic-Speed Control



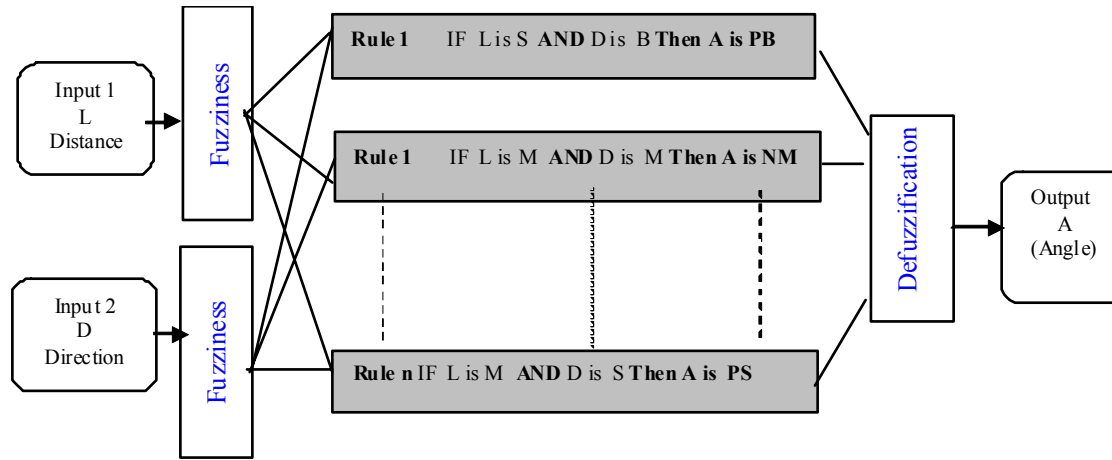
- It was found that the use of the fuzzy logic is more suitable than others to control the speed of the mobile robot
- A set of fuzzy rules block was designed to achieve the **speed control** of the MR

# Fuzzy Logic Introduction

- What is Fuzzy Logic?
  - How can a logic which is "fuzzy" be useful?
  - Professor Lotfi Zadeh, the inventor of fuzzy logic, contends that a computer cannot solve problems as well as human experts unless it is able to think in the characteristic manner of a human being
  - As humans, we often rely on imprecise expressions like "slow", "expensive", or "far". But the comprehension of a computer is limited to a *black-white, everything-or-nothing*, or *true-false* mode of thinking



# Fuzzy Logic Introduction

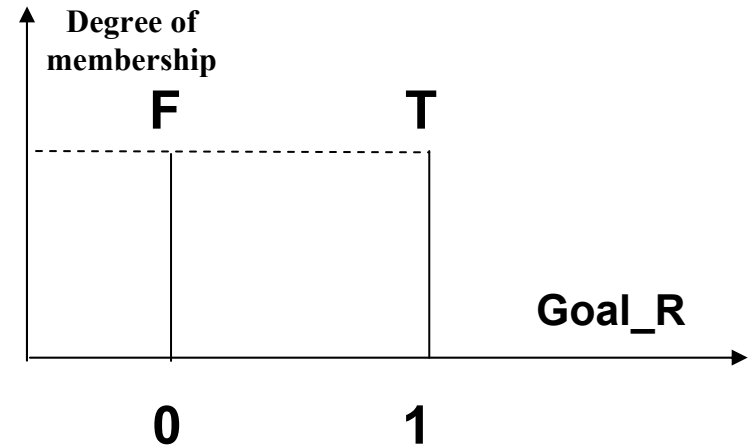


Block Diagram for a Fuzzy Logic System

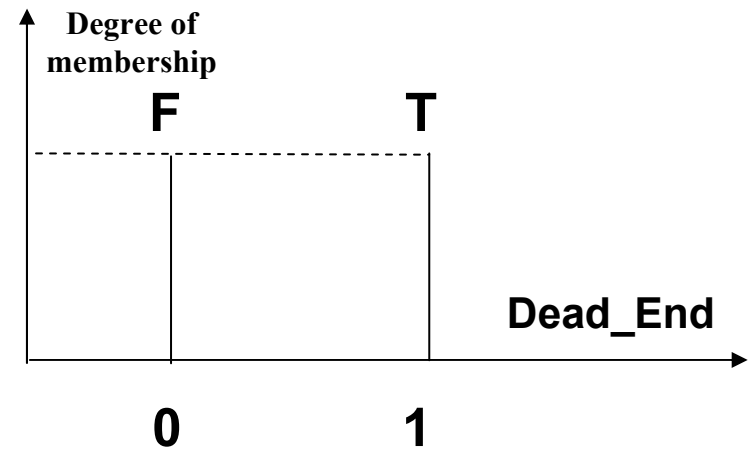
$A(x_1)$	$B(x_2)$	Min ( $A(x_1), B(x_2)$ )	$A(x_1)$	$B(x_2)$	Max ( $A(x_1), B(x_2)$ )	$A(x_1)$	$1 - A(x_1)$
0.5	0.5	0.5	0.5	0.5	0.5	0.7	0.3
0.3	0.5	0.3	0.3	0.5	0.5	0.3	0.7
0.5	0.3	0.3	0.5	0.3	0.5		
0.3	0.3	0.3	0.3	0.3	0.3		
<b>AND</b>			<b>OR</b>			<b>NOT</b>	

# Speed Control Fuzzy Logic Block

Linguistic Fuzzy Terms	Linguistic input variables
Very Near, Near, Medium and Far	<b>Distance to Goal</b>
2 Complementary Crisp Terms True and False	<b>Main Goal reached</b>
Positive Big, Positive Small, Zero, Negative Small, and Negative Big	<b>Turn angle</b>
Zero, Slow, Medium and High	<b>Current speed</b>
2 Complementary Crisp Terms True and False	<b>The dead-end status</b>



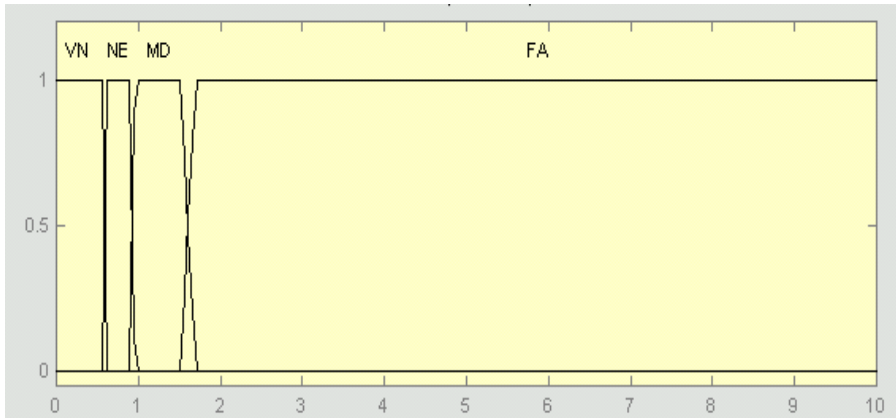
2 complementary linguistic terms of **Goal\_R**



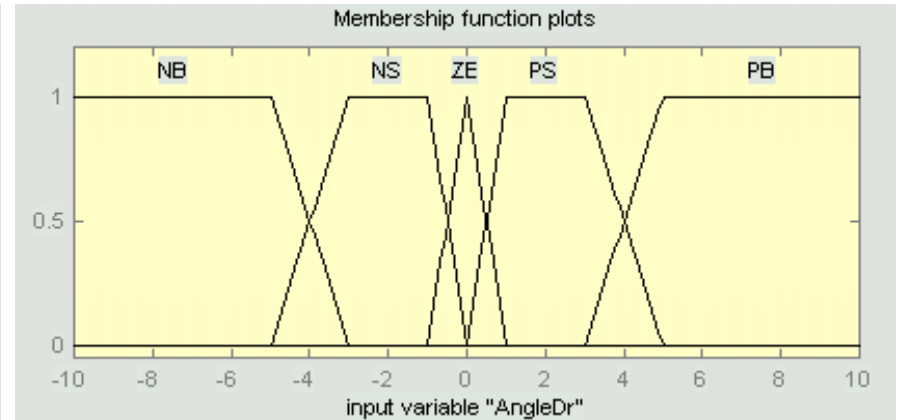
2 complementary linguistic terms of **Dead\_End**

# Speed Control Fuzzy Logic Block

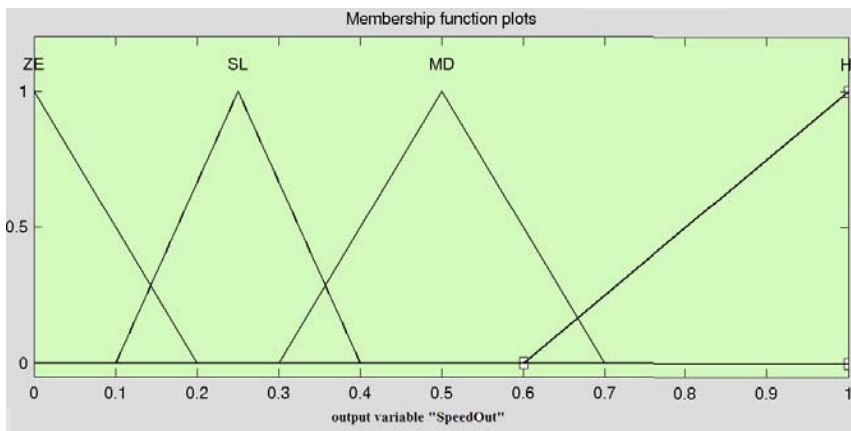
Trapezoid  
Triangle



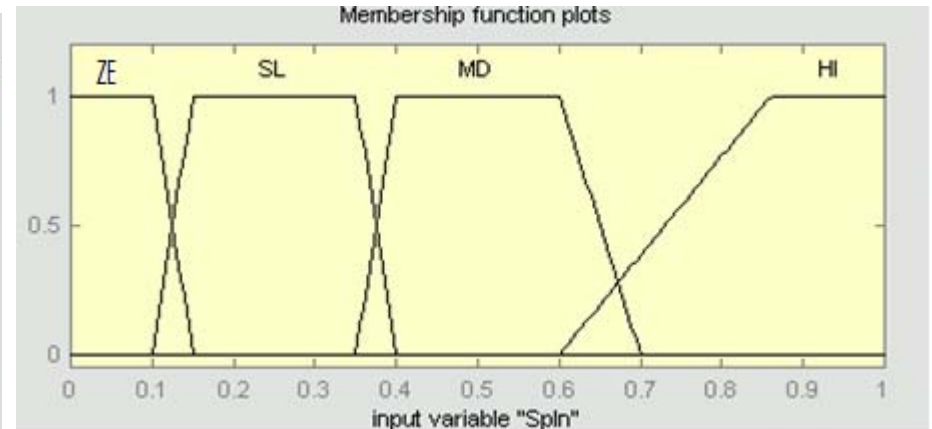
Linguistic terms representing distance to Goal  $Gdist$



Linguistic terms of required turning angle  $AngleDr$



Linguistic terms representing MR's Output speed  $SPO$



Linguistic terms representing MR's current speed  $SPI$

# Speed Control Fuzzy Logic Block

15 If/Then rules of Mamdani type: fuzzy inference system for speed control

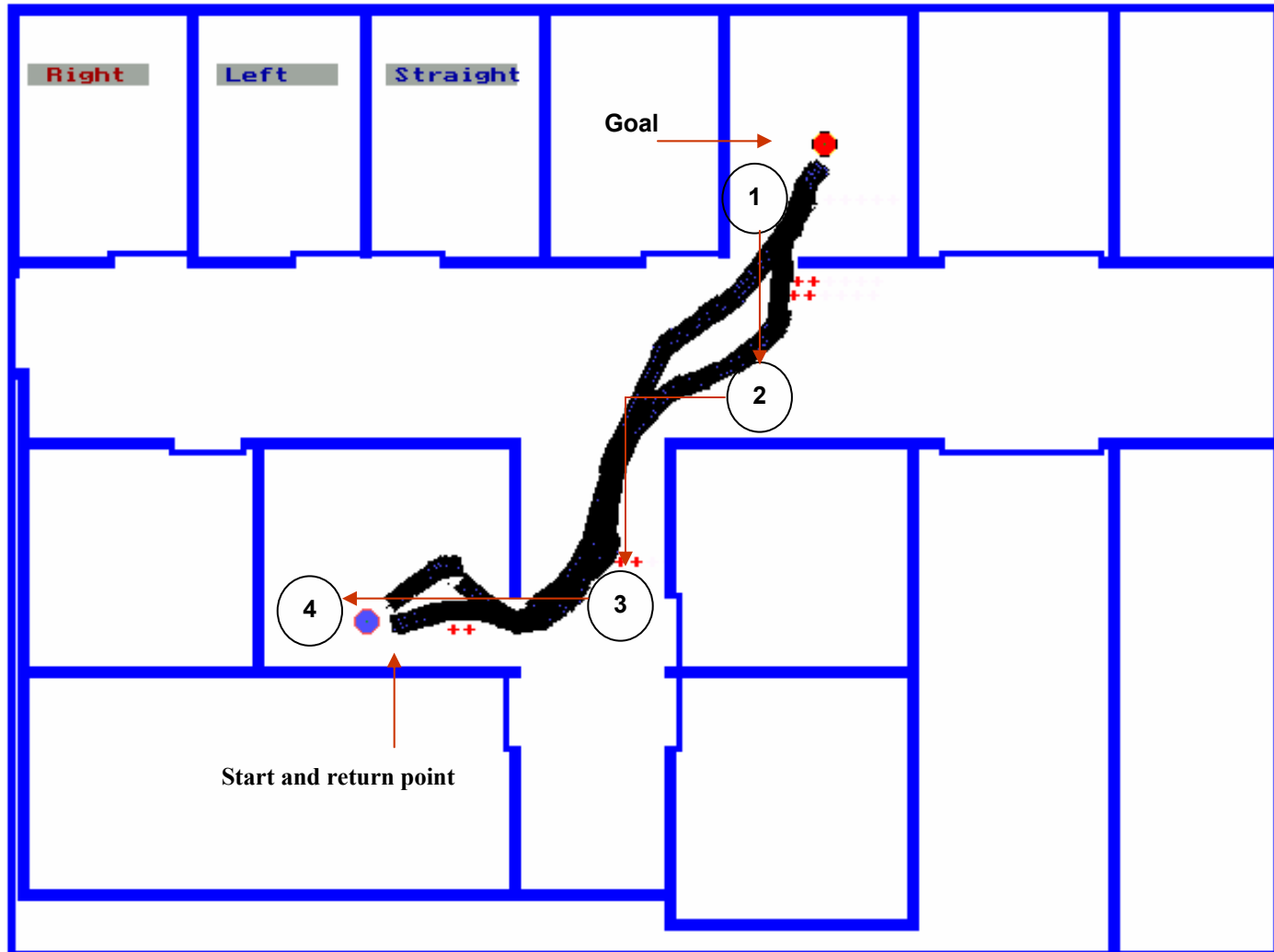
Rule No.	Antecedent	Consequent
(*1*)	<b>If</b> Goal_R.F <b>And</b> AngleDr.Ze <b>And</b> SPI.ZE <b>And</b> Dead_end.F	<b>Then</b> SPO. SL
(*2*)	<b>If</b> GDist.FA <b>And</b> Goal_R.F <b>And</b> AngleDr.Ze <b>And</b> SPI.SL <b>And</b> Dead_end.F	<b>Then</b> SPO. MD
(*3*)	<b>If</b> GDist.FA <b>And</b> Goal_R.F <b>And</b> AngleDr.Ze <b>And</b> SPI.MD <b>And</b> Dead_end.F	<b>Then</b> SPO. HI
(*4*)	<b>If</b> GDist.ME <b>And</b> Goal_R.F <b>And</b> AngleDr.Ze <b>And</b> SPI.SL <b>And</b> Dead_end.F	<b>Then</b> SPO. MD
(*5*)	<b>If</b> GDist.FA <b>And</b> Goal_R.F <b>And</b> AngleDr.Ze <b>And</b> SPI.HI <b>And</b> Dead_end.F	<b>Then</b> SPO. HI
(*6*)	<b>If</b> GDist.ME <b>And</b> Goal_R.F <b>And</b> AngleDr.Ze <b>And</b> SPI.MD <b>And</b> Dead_end.F	<b>Then</b> SPO. MD
(*7*)	<b>If</b> GDist.NE <b>And</b> Goal_R.F <b>And</b> AngleDr.Ze <b>And</b> SPI.SL <b>And</b> Dead_end.F	<b>Then</b> SPO. SL
(*8*)	<b>If</b> GDist.ME <b>And</b> Goal_R.F <b>And</b> AngleDr.Ze <b>And</b> SPI.HI <b>And</b> Dead_end.F	<b>Then</b> SPO. MD
(*9*)	<b>If</b> GDist.NE <b>And</b> Goal_R.F <b>And</b> AngleDr.Ze <b>And</b> SPI.MD <b>And</b> Dead_end.F	<b>Then</b> SPO. SL
(*10*)	<b>If</b> GDist.VN <b>And</b> Goal_R.T	<b>Then</b> SPO. ZE
(*11*)	<b>If</b> AngleDr.NB <b>And</b> Dead_end.F	<b>Then</b> SPO. ZE
(*12*)	<b>If</b> AngleDr.PB <b>And</b> Dead_end.F	<b>Then</b> SPO. ZE
(*13*)	<b>If</b> Goal_R.F <b>And</b> AngleDr.NS <b>And</b> Dead_end.F	<b>Then</b> SPO. SL
(*14*)	<b>If</b> Goal_R.F <b>And</b> AngleDr.PS <b>And</b> Dead_end.F	<b>Then</b> SPO. SL
(*15*)	<b>If</b> Dead_end.T	<b>Then</b> SPO. ZE

- The Min Implication Method, (the AND operator)
  - Is used to select the output of the used fuzzy rules
- Then the Max aggregation method, (the OR operator)
  - Is used to determine the degree of the membership of every linguistic term of the fuzzy output variable

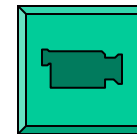
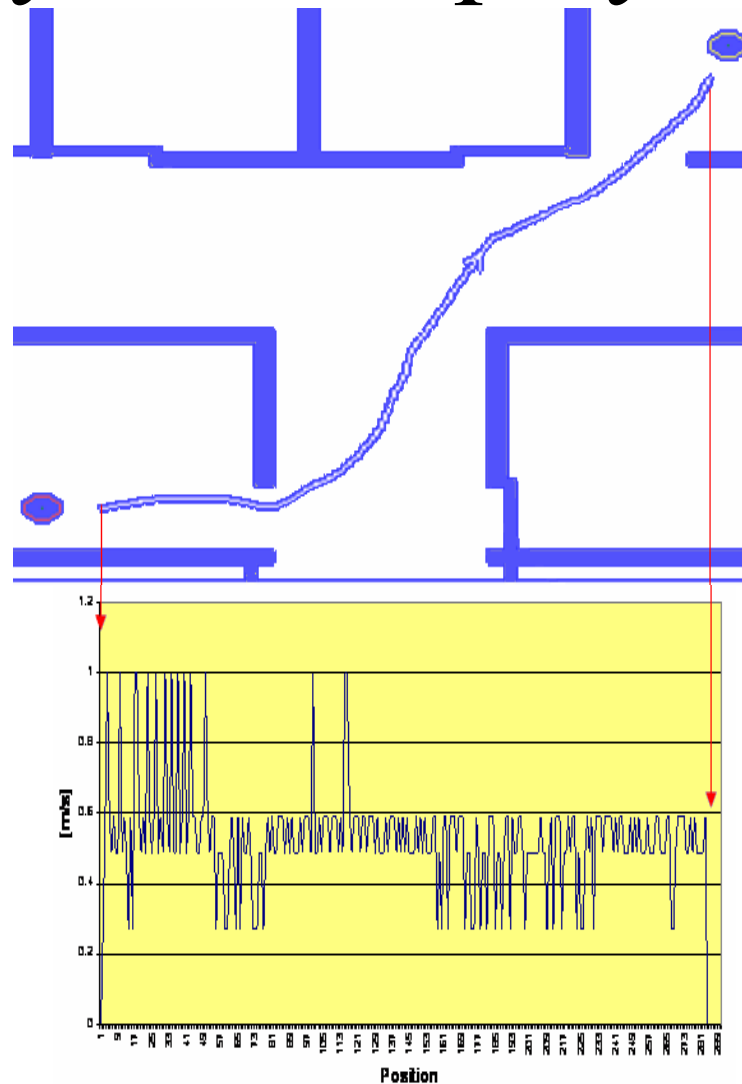
# System Deployment

Path				
Arriving to the main Goal <i>Autonomously</i>				
Local Navigation NC	Working ERNN	Subspace No.	Subspace	Sub Goals
L	2	1	EX	1
S	1	2	TF	2
L	1	3	CO	3
R	2	4	RM	4
Getting back to the return point				

# System Deployment



# System Deployment



MOVCS1

Fuz\_Neu  
Tablet 5

# Conclusion

- The global navigation of the MR controlled MR's direction and speed, tracked its planned and unplanned path, reached Sub-Goals and Main Goal, and avoided unexpected obstacles.
- That will help to navigate in an industrial Environment