Genetic algorithms and evolutionary programming

- Inspired by the Darwin's theory of evolution
- A solution is represented as an instance, a "chromosome".
- Evaluation (fitness) function is required
- Generic optimization techniques.
- Many applications

Industrial Application of Fuzzy Logic Control

- Fuzzy Logic Primer
  - History, Current Level and Further Development of Fuzzy Logic Technologies in the U.S., Japan, and Europe
  - Types of Uncertainty and the Modeling of Uncertainty
  - The Basic Elements of a Fuzzy Logic System
  - Types of Fuzzy Logic Controllers

History, State of the Art, and Future Development

- 1965 Seminal Paper ¡°Fuzzy Logic¡± by Prof. Lotfi Zadeh, Faculty in Electrical Engineering, U.C. Berkeley, Sets the Foundation of the ¡°Fuzzy Set Theory¡±
- 1970 First Application of Fuzzy Logic in Control Engineering (Europe)
- 1975 Introduction of Fuzzy Logic in Japan
- 1980 Empirical Verification of Fuzzy Logic in Europe
- 1985 Broad Application of Fuzzy Logic in Japan
- 1990 Broad Application of Fuzzy Logic in Europe
- 1995 Broad Application of Fuzzy Logic in the U.S.
- 2000 Fuzzy Logic Becomes a Standard Technology and Is Also Applied in Data and Sensor Signal Analysis. Application of Fuzzy Logic in Business and Finance.

Types of Uncertainty and the Modeling of Uncertainty

- Stochastic Uncertainty:
  - The Probability of Hitting the Target Is 0.8
- Lexical Uncertainty:
  - "Tall Men", "Hot Days", or "Stable Currencies"
  - We Will Probably Have a Successful Business Year.
  - The Experience of Expert A Shows That B Is Likely to Occur. However, Expert C Is Convinced This Is Not True.

– **University of Otago, September 22, 1999 INTELLIGENT SYSTEMS FOR A KNOWLEDGE-BASED SOCIETY Prof. Nikola Kasabov Department of Information Science University of Otago nkasabov@otago.ac.nz**

What is a Knowledge-based Society?

- The society is knowledge-, not commodity- driven
- Intellectual achievements are highly ranked
- Global connectivity
- Global Market
- Creating, using, and trading information and knowledge.

The World of Information: Information and knowledge

- **Data, information and knowledge**
- **The “macro” world of information:**
  - medical and health information
  - business and economic information
  - geographic information
  - etc.
- **The “micro” world of information:**
  - the brain
  - genetic information
  - quantum information
- **Exponential information increase with time**
- **Transforming information into knowledge, managing and utilizing it is the major challenge for a KB society.**

Information Science

- The area of science that develops methods and systems for information and knowledge processing regardless of the domain area.
- The Information Science subject areas
- The emergence of Information Sciences
- Information Sciences versus Information Technologies.
- Information AND Computer Science.

Artificial Intelligence and Knowledge Engineering

- **What defines intelligence?**
  - Learning and adaptation
  - Generalisation
  - Knowledge acquisition and processing
  - Reasoning
  - Problem solving and decision making
  - Using language
  - Multi-modal communication: speech, vision, etc.
  - Creativity
  - etc.

Artificial Intelligence (AI)

- Some AI methods are inspired by the way the human brain works - ‘brain-like computing’
- AI also develops and applies its own methods and principles
- Often AI methods combine the two approaches

Symbolic AI systems

- Logic systems, e.g. propositional logic (Aristotle, 4th century BC)
- Rule-based systems that use IF-THEN rules
- Expert systems
- Rule-based systems are universal computational mechanisms but they only use two categories: true and false.

What is Fuzzy Logic?

- In traditional logic, statements can be either true or false, and sets can either contain an element or not.
- These logic values and set memberships are typically represented with number 1 and 0.
- Fuzzy logic generalizes traditional logic by allowing statements to be somewhat true, partially true, etc.
- Likewise, sets can have full members, partial members, and so on.
- For example, a person whose height is 5’ 9” might be assigned a membership of 0.6 in the fuzzy set “tall people”.
- The statement “Joe is tall” is 60% true of Joe is 5’9”.
- Fuzzy logic is a set of “if--then” statements based on combining fuzzy sets. (Beale & Demuth..Fuzzy Systems Toolbox.)

Fuzzy Sets, Statements, and Rules

- A crisp set is simply a collection of objects taken from the universe of objects.
- Fuzzy refers to linguistic uncertainty, like the word "tall".
- Fuzzy sets allow objects to have membership in more than one set (e.g. 6' 0") has grade 70% in the set "tall" and grade 40% in the set "medium".
- A fuzzy statement describes the grade of a fuzzy variable with an expression (e.g. Pick a real number greater than 3 and less than 8.)

Fuzzy Logic Control

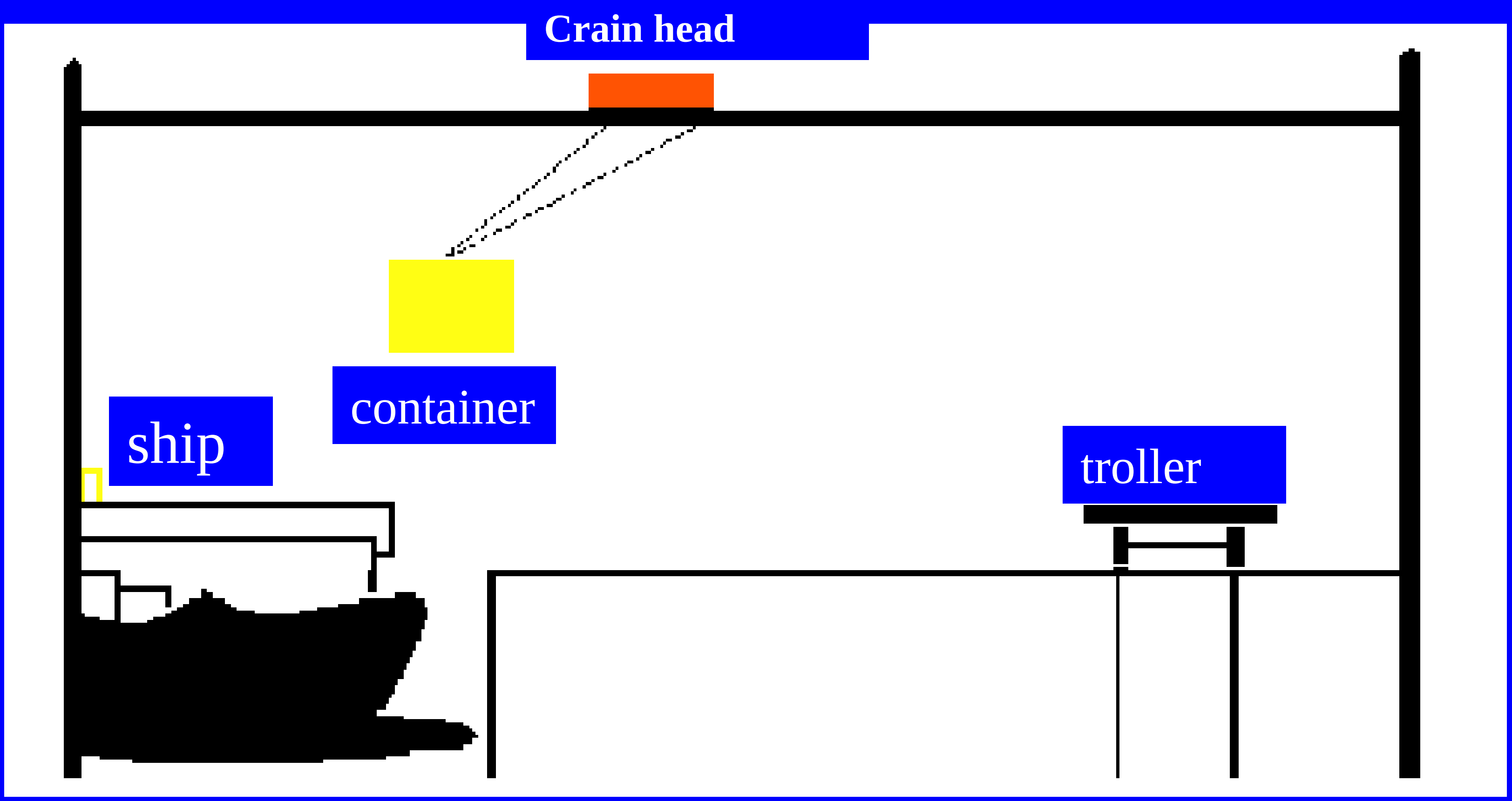
- Fuzzy controller design consist of turning intuitions, and any other information about how to control a system, into set of rules.
- These rules can then be applied to the system.
- If the rules adequately control the system, the design work is done.
- If the rules are inadequate, the way they fail provides information to change the rules.

Example 1

Container Crane Case Study

Basic Elements of a Fuzzy Logic System

Container Crane Case Study:



Fuzzy Set Definitions

Discrete Definition:

$\mu_{SF}(35°) = 0$

$\mu_{SF}(36°) = 0$

$\mu_{SF}(37°) = 0$

$\mu_{SF}(38°) = 0.1$

$\mu_{SF}(39°) = 0.35$

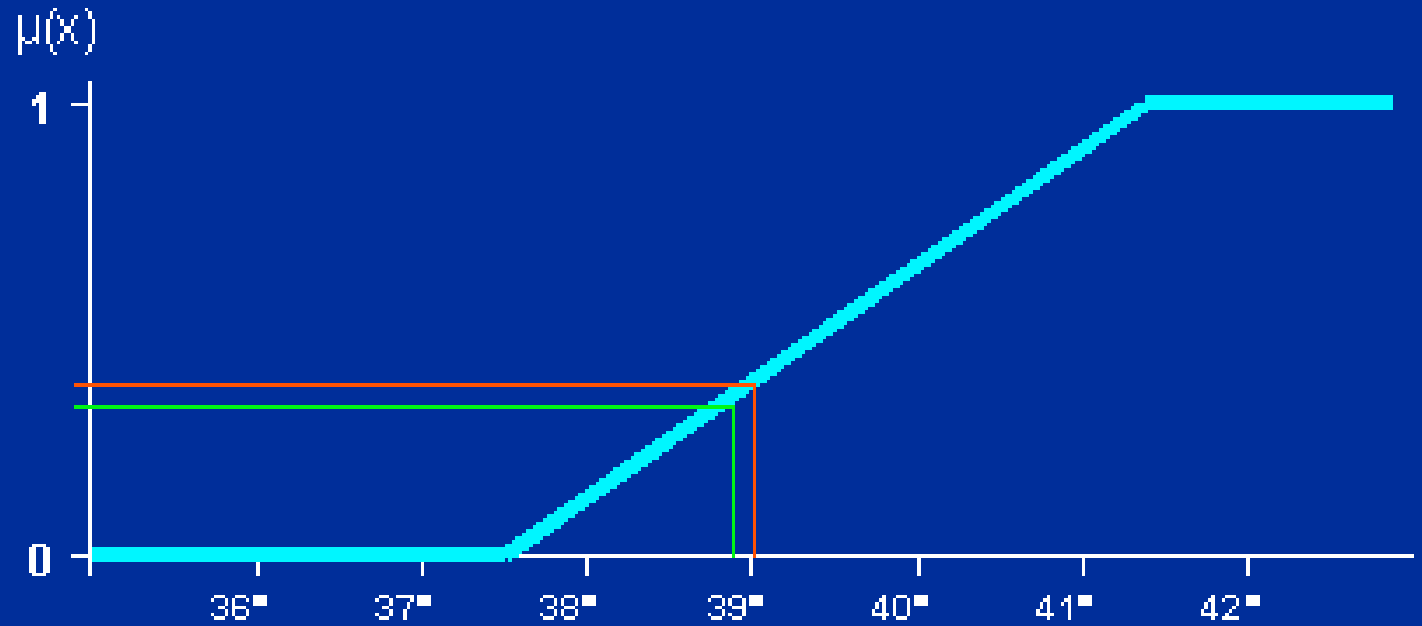
$\mu_{SF}(40°) = 0.65$

$\mu_{SF}(41°) = 0.9$

$\mu_{SF}(42°) = 1$

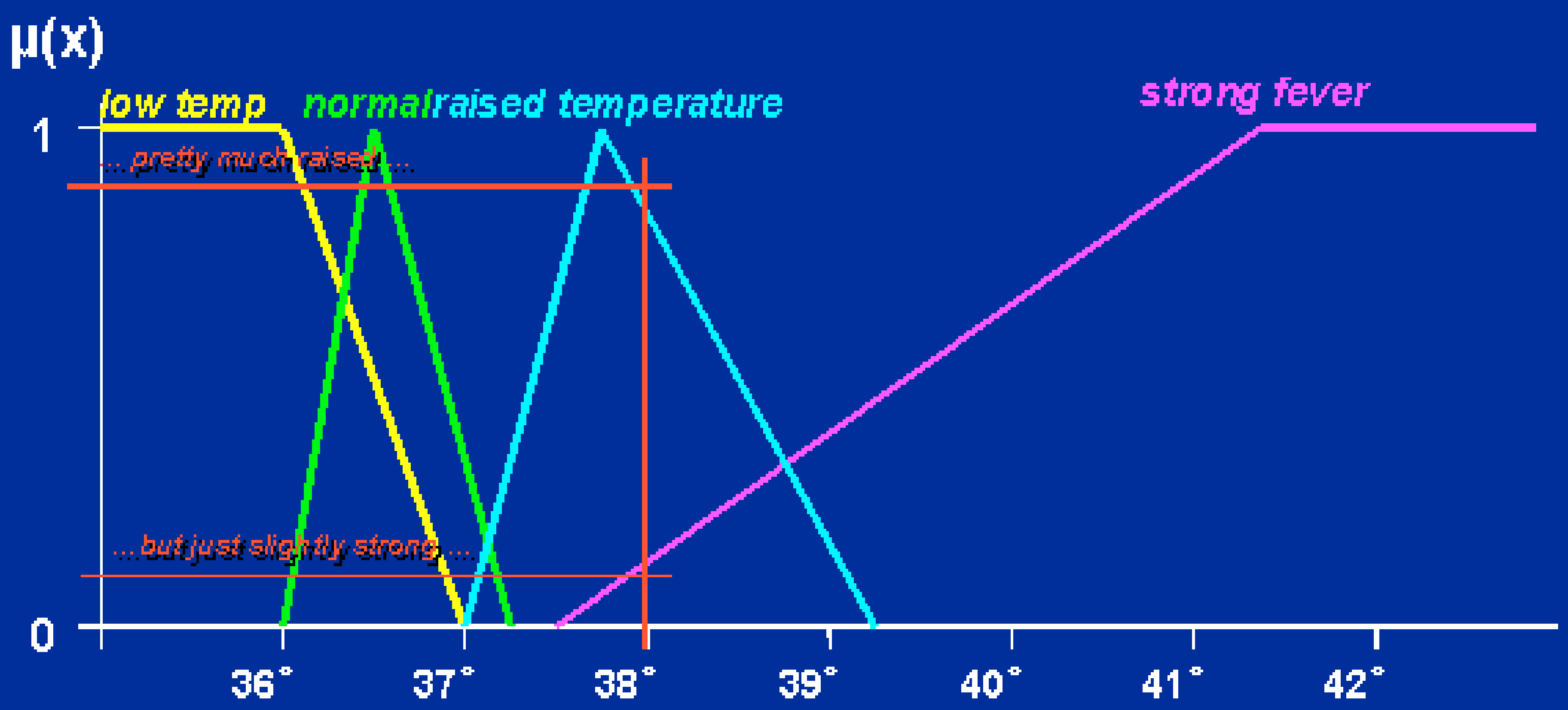
$\mu_{SF}(43°) = 1$

Continuous Definition:



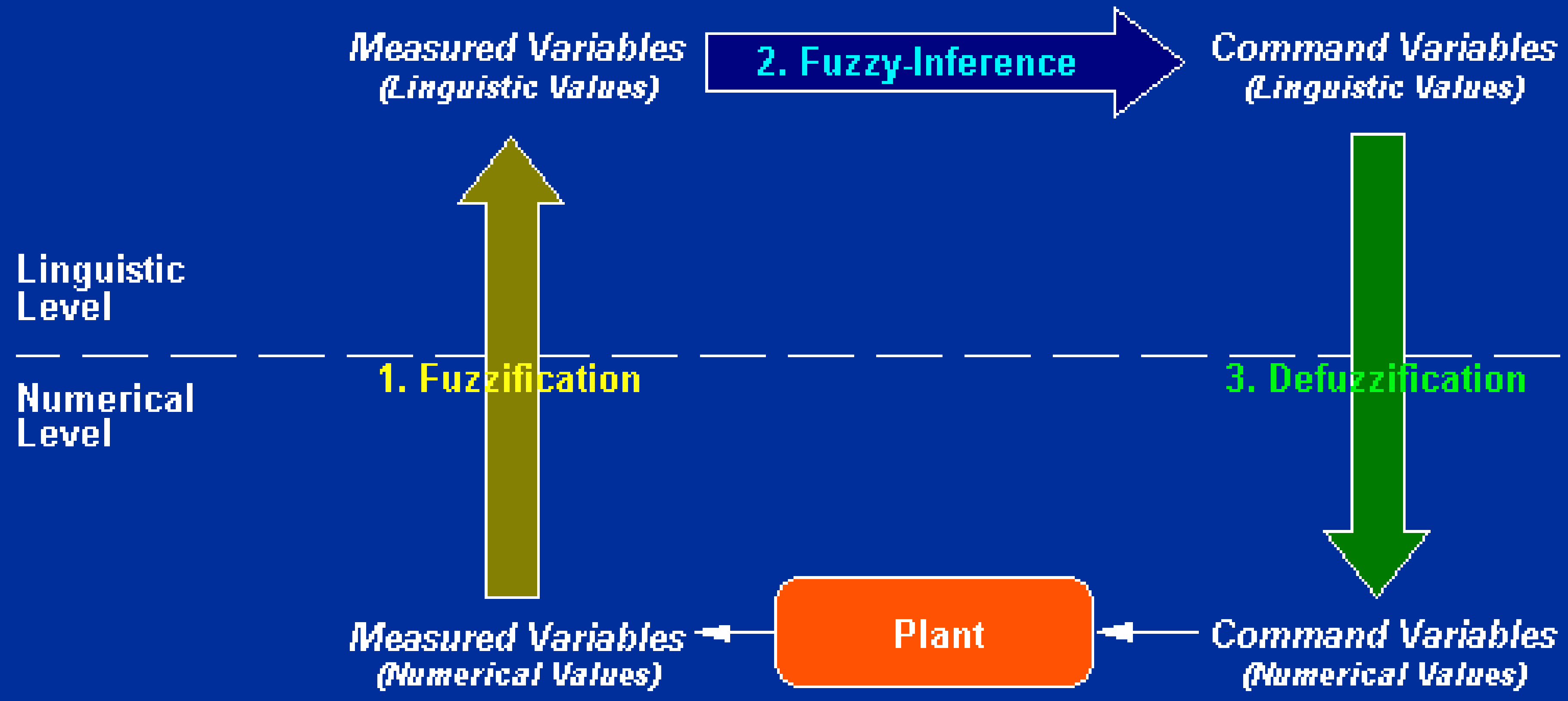
Linguistic Variable

- …. Terms, Degree of Membership, Membership Function, Base Variable…..



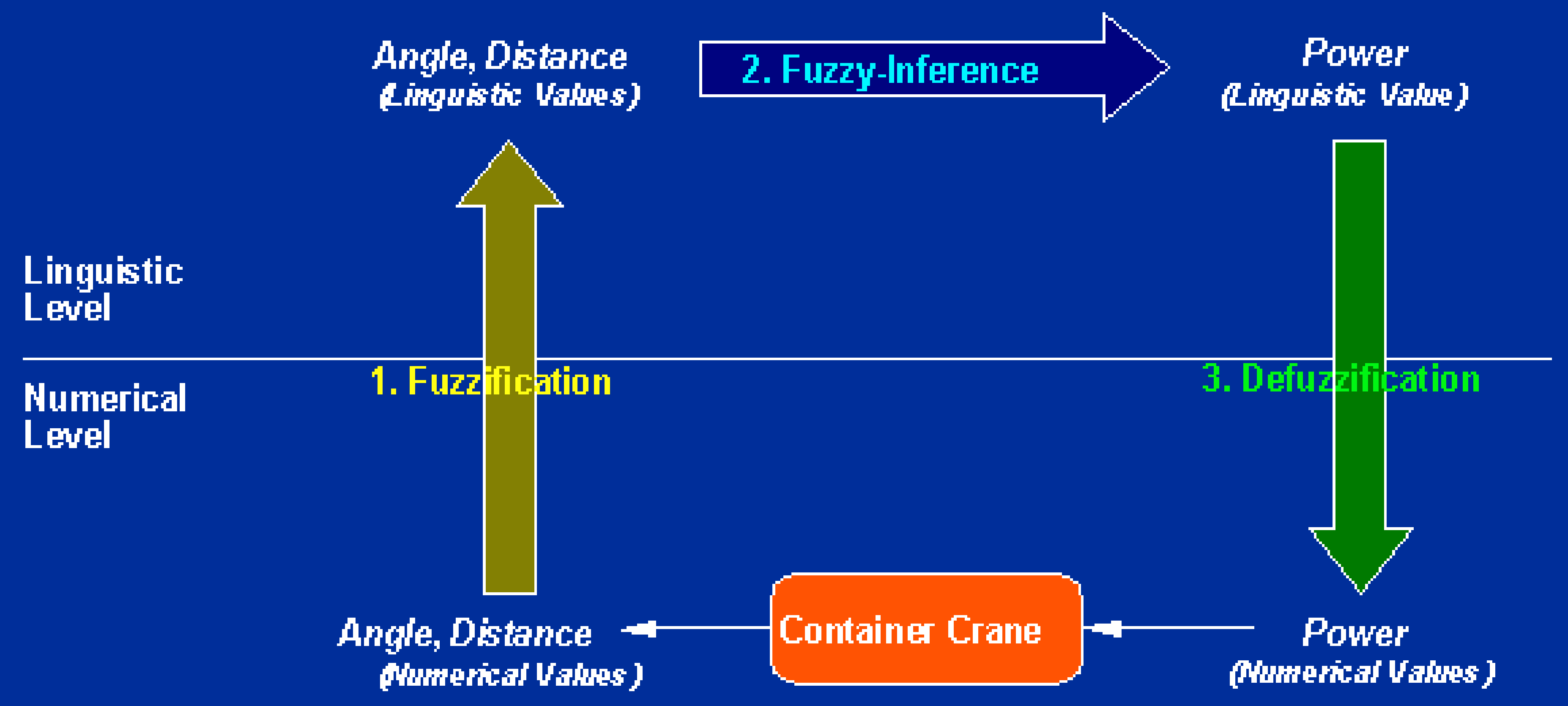
Basic Elements of a Fuzzy Logic System

- Fuzzification, Fuzzy Inference, Defuzzification



Basic Elements of a Fuzzy Logic System

- Control Loop of the Fuzzy Logic Controlled Container Crane:



1. Fuzzification: - Linguistic Variables -

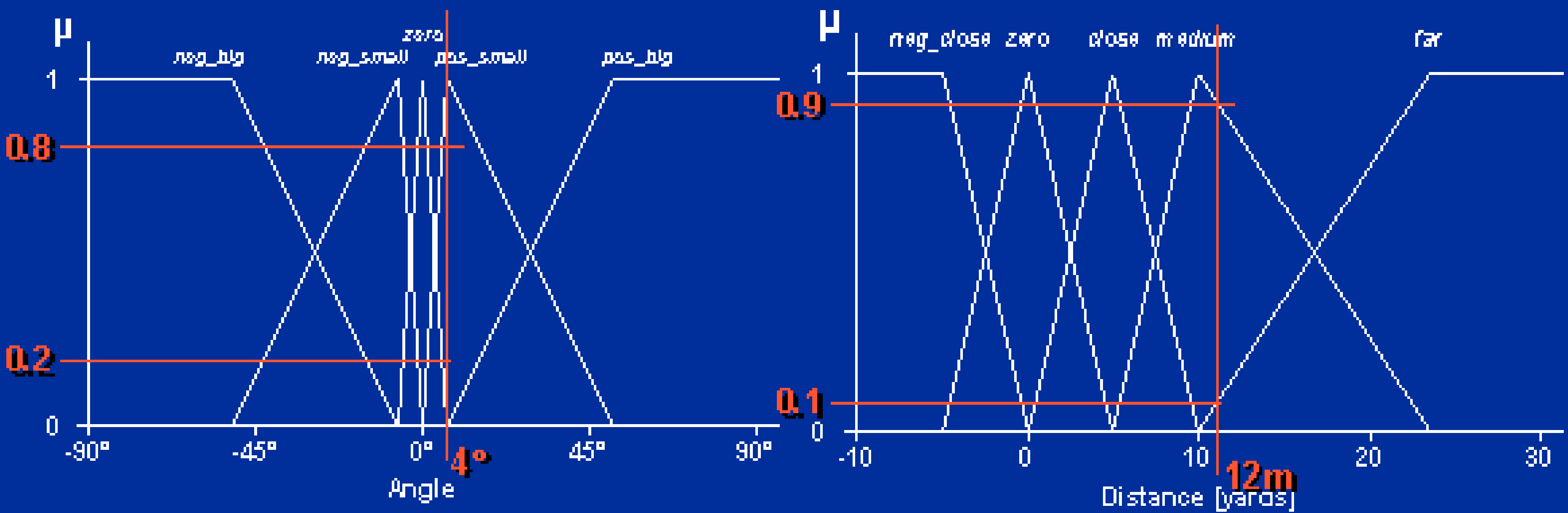
Term Definitions:

Distance := {far, medium, close, zero, neg_close}

Angle := {pos_big, pos_small, zero, neg_small, neg_big}

Power := {pos_high, pos_medium, zero, neg_medium, neg_high}

Membership Function Definition:



2. **Fuzzy Inference:** "IF-THEN-ELSE" Rules

Computation of the "IF-THEN"-Rules:

#1: IF Distance = medium AND Angle = pos_small THEN Power = pos_medium

#2: IF Distance = medium AND Angle = zero THEN Power = zero

#3: IF Distance = far AND Angle = zero THEN Power = pos_medium

- Aggregation: Computing the "IF"-Part
- Composition: Computing the "THEN"-Part

2. Fuzzy Inference: - Aggregation -

Boolean Logic Only
Defines Operators for 0/1:

A	B	A^B
0	0	0
0	1	0
1	0	0
1	1	1

Fuzzy Logic Delivers
a Continuous Extension:

- AND: $\mu_{A \wedge B} = \min\{\mu_A; \mu_B\}$
- OR: $\mu_{A+B} = \max\{\mu_A; \mu_B\}$
- NOT: $\mu_{\neg A} = 1 - \mu_A$

Aggregation of the “IF”-Part:

#1: $\min\{0.9, 0.8\} = 0.8$

#2: $\min\{0.9, 0.2\} = 0.2$

#3: $\min\{0.1, 0.2\} = 0.1$

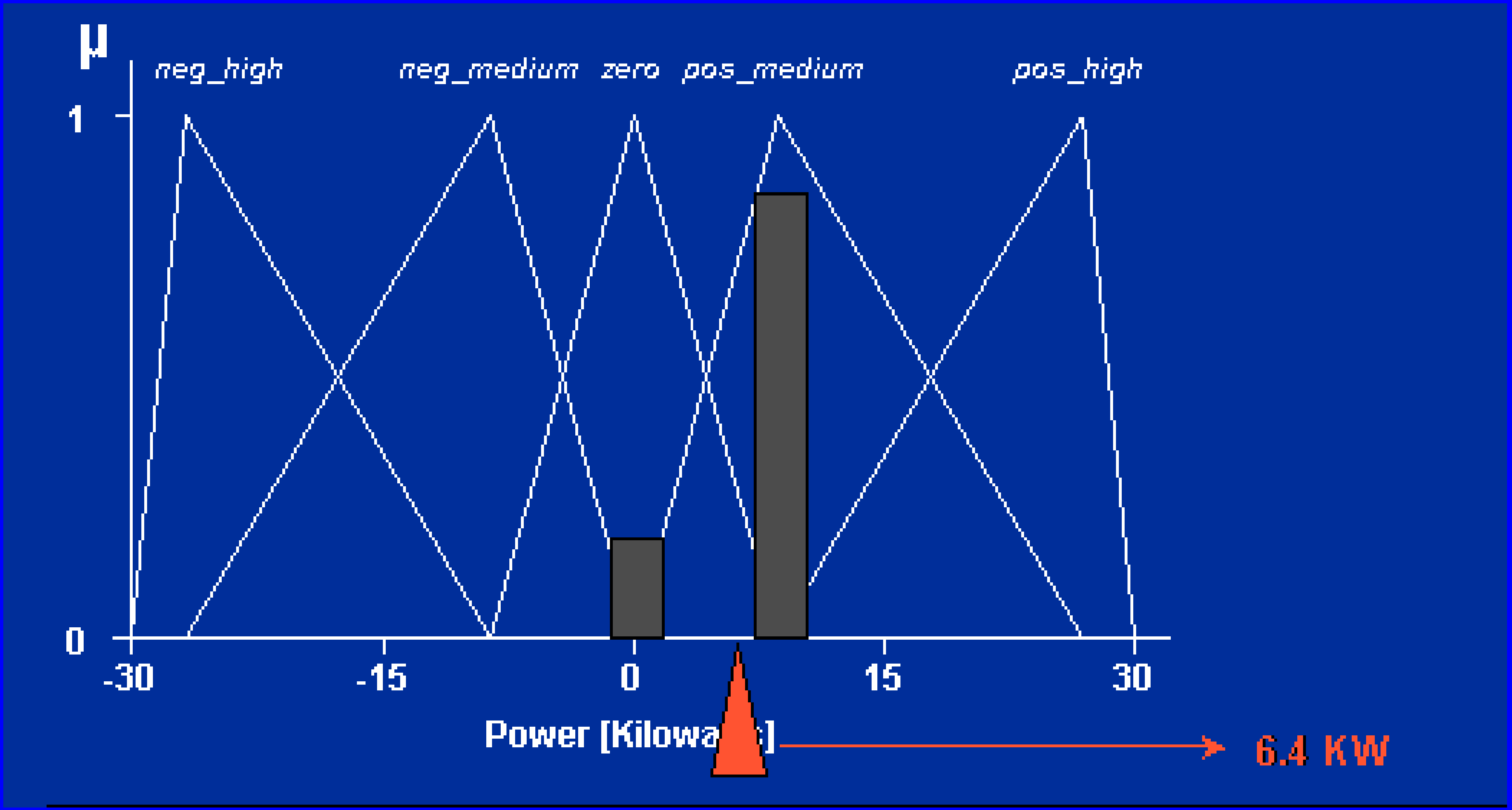
2. Fuzzy-Inference: Composition

- Result for the Linguistic Variable "Power":

pos_high	with the degree 0.0	
pos_medium	with the degree 0.8	(= max{ 0.8 , 0.1 })
zero	with the degree 0.2	
neg_medium	with the degree 0.0	
neg_high	with the degree 0.0	

3. Defuzzification

- Finding a Compromise Using “Center-of-Maximum”:



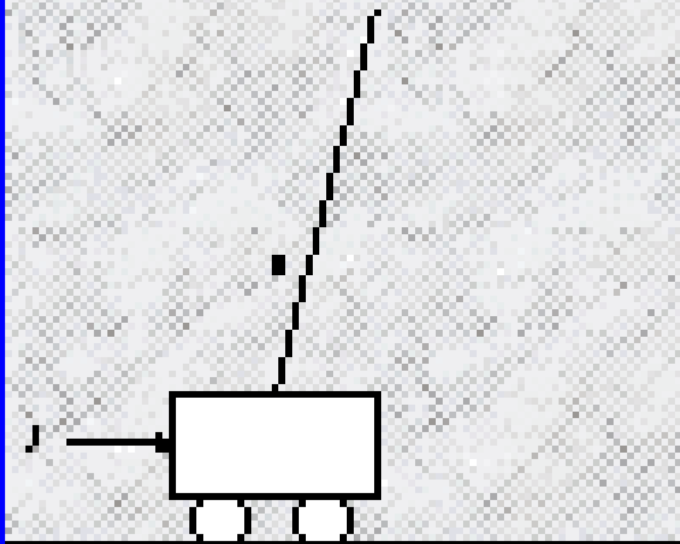
Example 2

Neuro-Fuzzy Inverted Pendulum Control System

Introduction to Pendulum

- Inverted Pendulum Problem
- Neural Network Architecture
- Fuzzy Logic Target Generator
- Balance the Inverted Pendulum
- Conclusion

Inverted Pendulum System



– State Variables

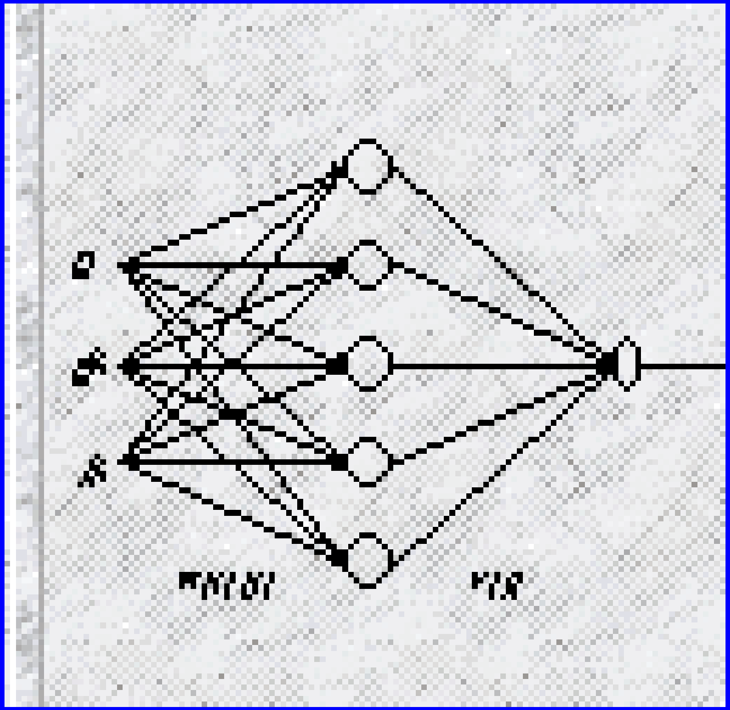
$$\begin{pmatrix}\theta & \dot{\theta} & x & \dot{x}\end{pmatrix}$$

– Dynamic Equations

$$\ddot{x} = \frac{f - ml\left[\ddot{\theta}\cos\theta - \dot{\theta}^2\sin\theta\right]}{M+m}$$

$$\ddot{\theta} = \frac{(M+m)g\sin\theta - \cos\theta\left[f + ml\dot{\theta}^2\sin\theta\right]}{\frac{4}{3}(M+m)l - ml\cos^2\theta}$$

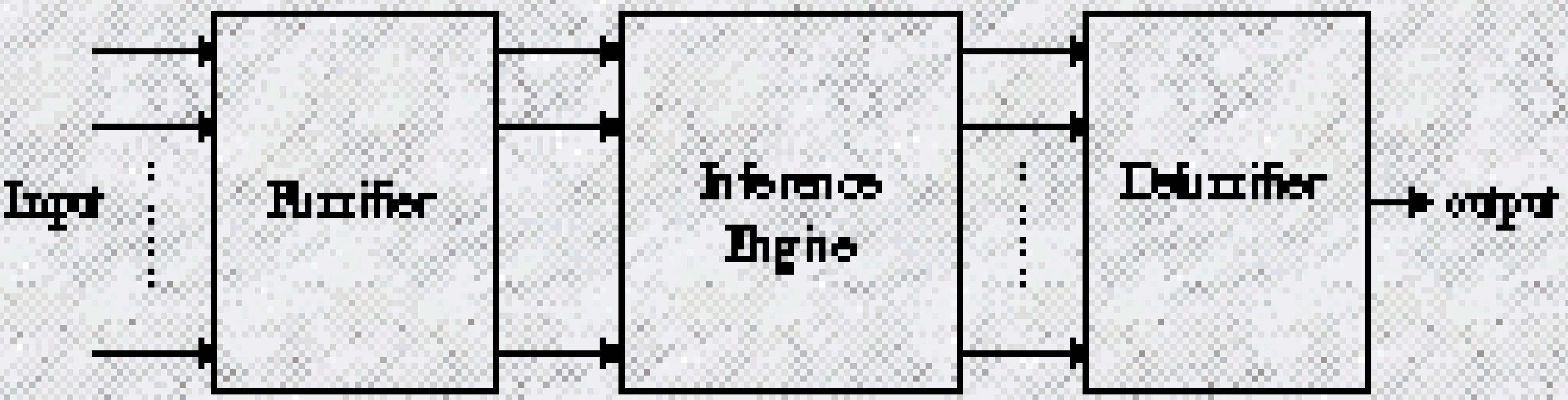
Neural Network Architecture



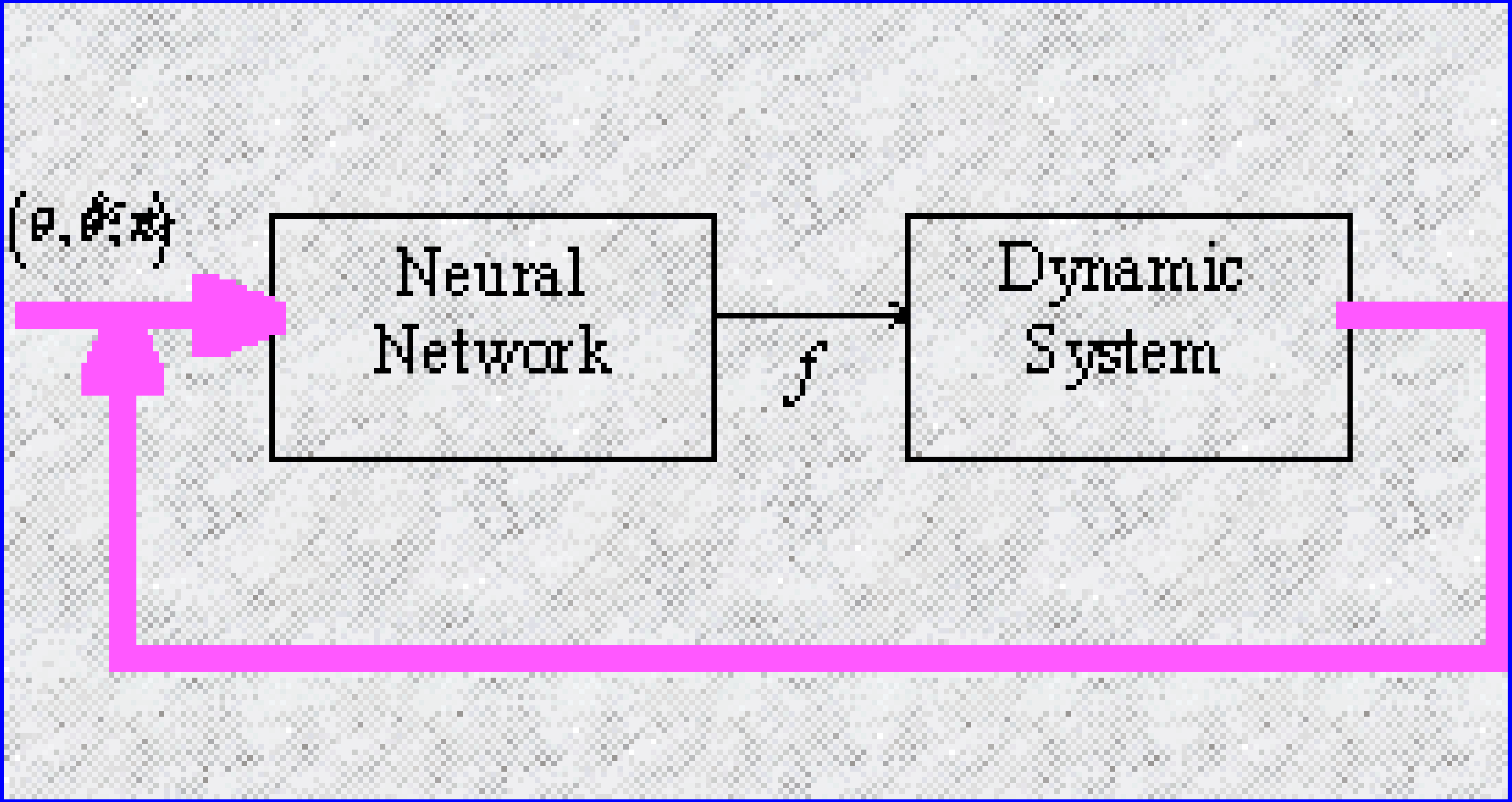
- **Two Architectures**
  - Regular Neurons
  - Linear Combination at output layer
- **Back-Propagation Algorithm**
- **Batch Training**

Fuzzy Logic Target Generator

- Singleton Fuzzifier
- Mamdani Max-min Inference Engine
- Centroid of Area Defuzzifier



Balance the Inverted Pendulum



Conclusion on Applying Fuzzy Logic to Inverted Pendulum

- **ANN can solve this problem efficiently**
- **Advantage of Fuzzy Logic Target Generator (FTLG)**
- **Further Studying**
  - Optimize FLTG using another ANN
  - 3-D Inverted Pendulum Problem

Example 3

Neural Net versus Fuzzy Logic

Neural Versus Fuzzy

- **Recap:** Neural Networks
- What is Fuzzy Logic?
- Fuzzy Sets, Statements, and Rules
- Fuzzy Logic Control
- Used in Expert System and Decision Systems

Neural Networks

- the problem of knowledge acquisition bottleneck
- computers with architectures and processing capabilities that simulate the human brain
- features
  - knowledge representations based on massively parallel processing
  - fast retrieval of large amounts of information
  - ability to recognize patterns based on **experience**

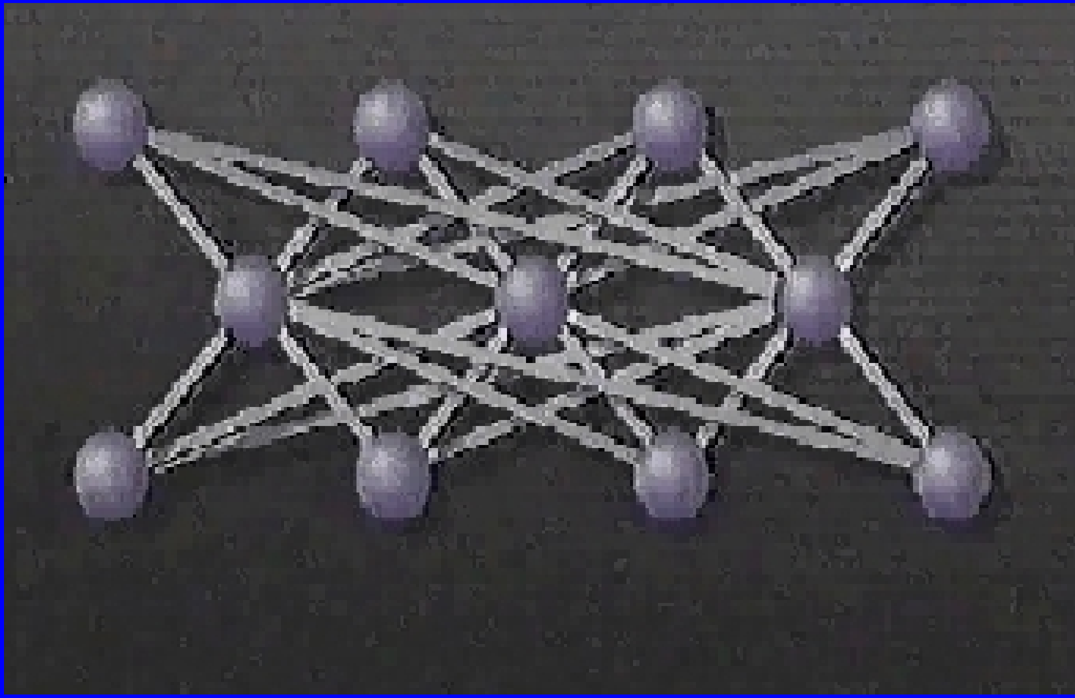
Artificial Neural Networks (ANN)

- model that emulates a biological neural network
- consists of
  - processing elements or neurons; each receives an input, provides processing, produces output
  - neurons are grouped together in layers
  - several topologies possible

Artificial neural networks (ANN)

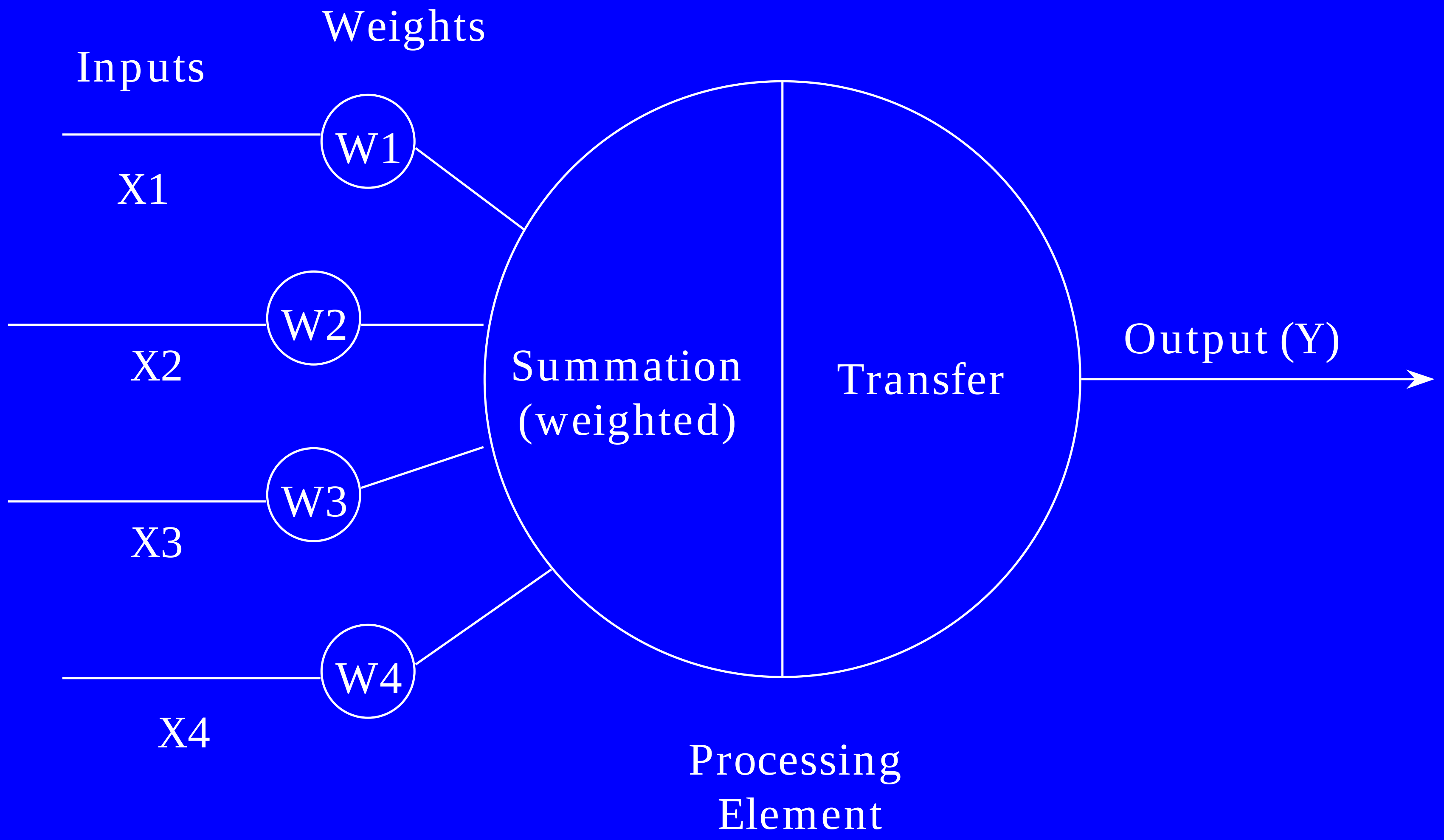
– (connectionist systems)

- Computational models that mimic the nervous system in its main function of adaptive learning.
- ANN are universal computational models

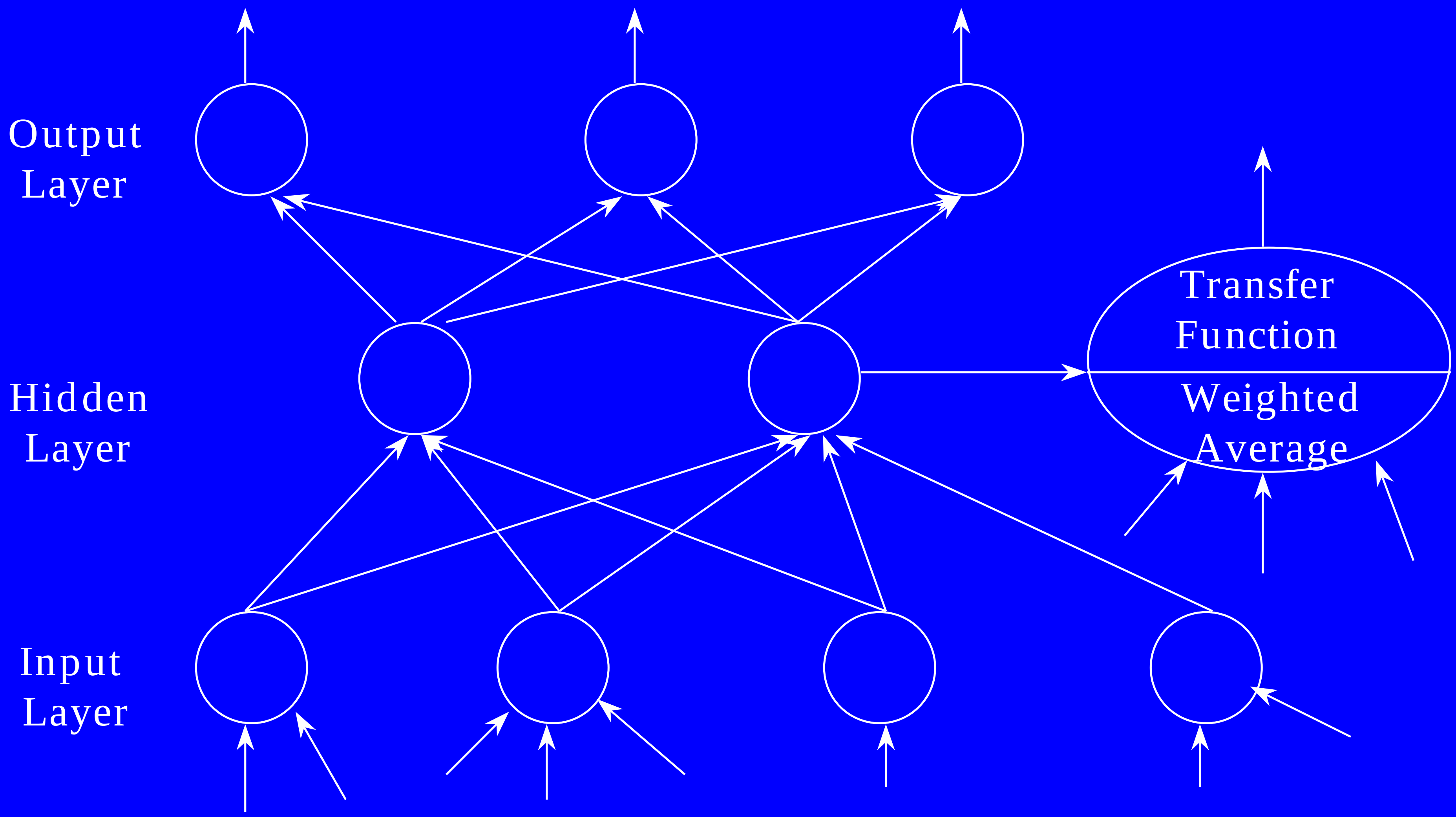


- ANN could learn to “speak” (e.g. NetTalk by T.Sejnowsky) or to do some other intelligent tasks
- Limitations

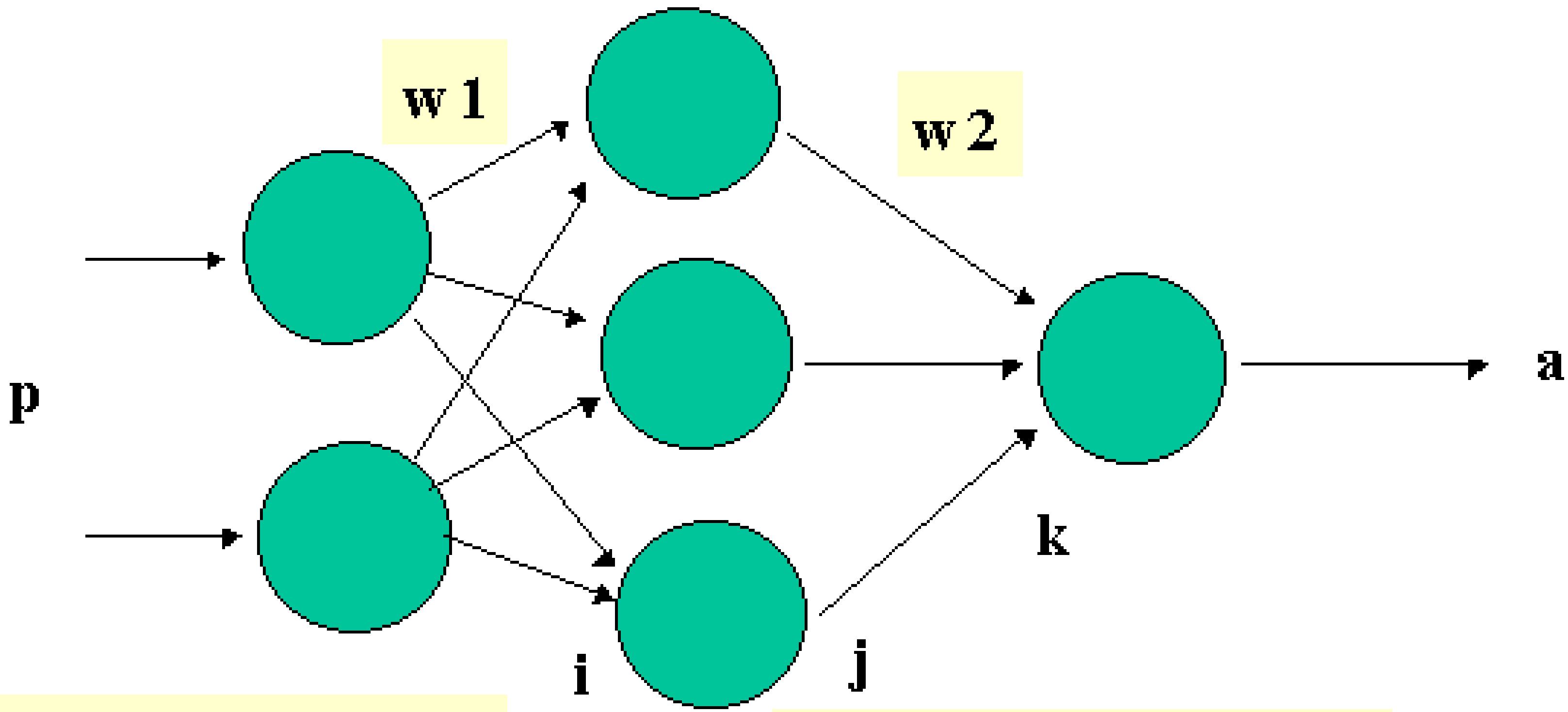
ANN Elements



A Sample Network



Recap: A Layered Neural Network



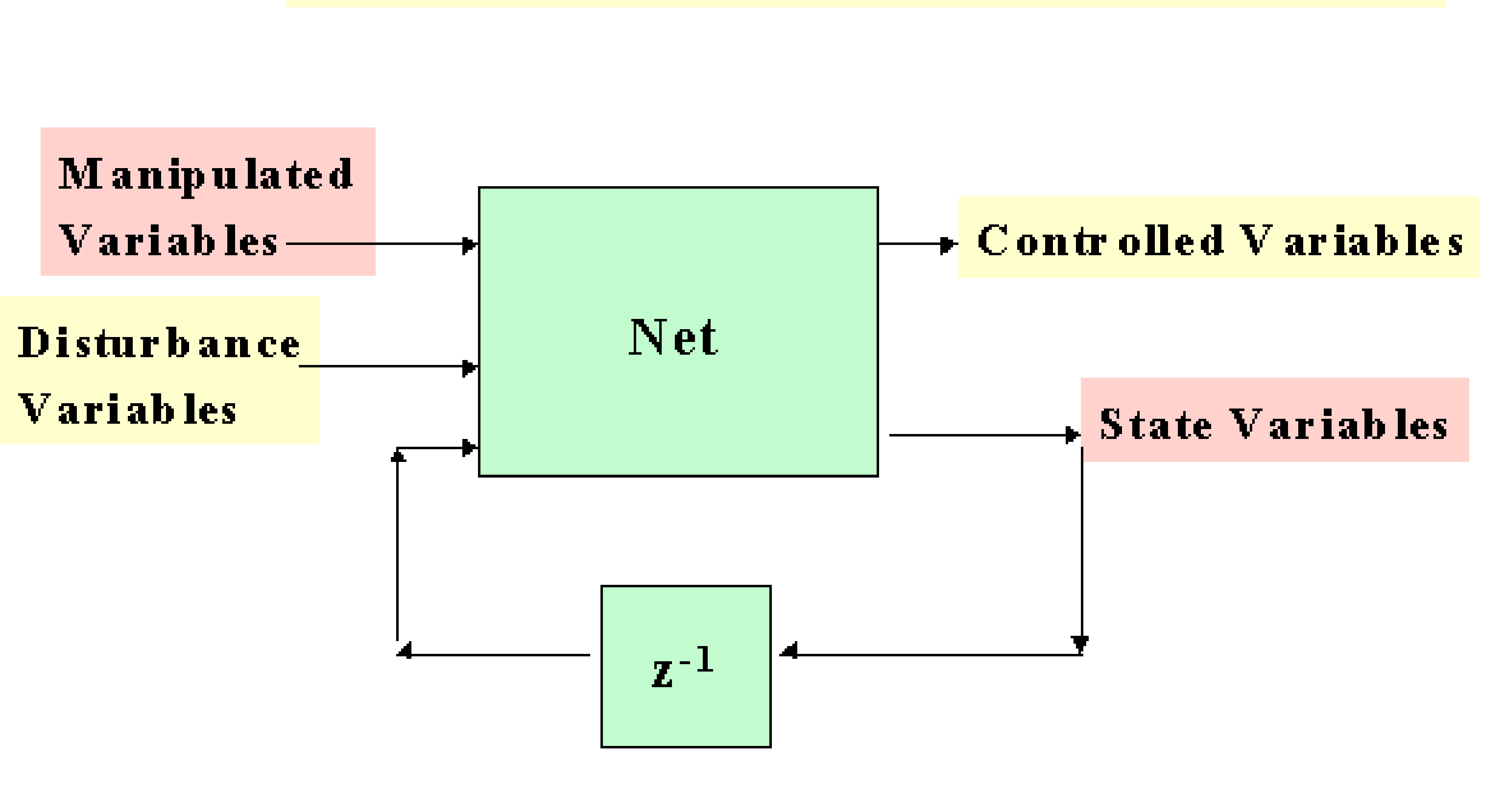
$$\underset{(3\text{x}1)}{i} = \underset{(3\text{x}2)}{w1} * \underset{(2\text{x}1)}{p}$$

$$j = F(i)$$

$$\underset{(1\text{x}1)}{k} = \underset{(1\text{x}3)}{w2} * \underset{(3\text{x}1)}{j}$$

$$a = F(k)$$

Recap: Recursive Neural Networks for Dynamic Simulation and Model Predictive Control



Elements of an ANN

- **inputs:** each corresponds to one <u>attribute</u>
- **output:** correspond to the <u>solution</u> to the problem
- **weights:** <u>relative strength</u> of the initial entering data or the various connections that transfer data from one layer to another
- **summation function:** weighted sum of all input elements entering each PE
- **transfer function:** relationship between internal state and output

Learning in an ANN

- **Three tasks:**
  - compute outputs
  - compare output with desired target
  - adjust weights and repeat
- **learning algorithms:** minimizing delta
  - discrete and continuous
    - type of input data
  - supervised and unsupervised
    - uses inputs for which desired outputs are known
    - only input stimuli shown; self-organizing

Building an ANN

- design choices
  - size of training and test data
  - learning algorithms
  - network topology: number of PE’s, their configuration
  - transfer function to be used
  - learning rate for each layer
  - diagnostic and validation tools

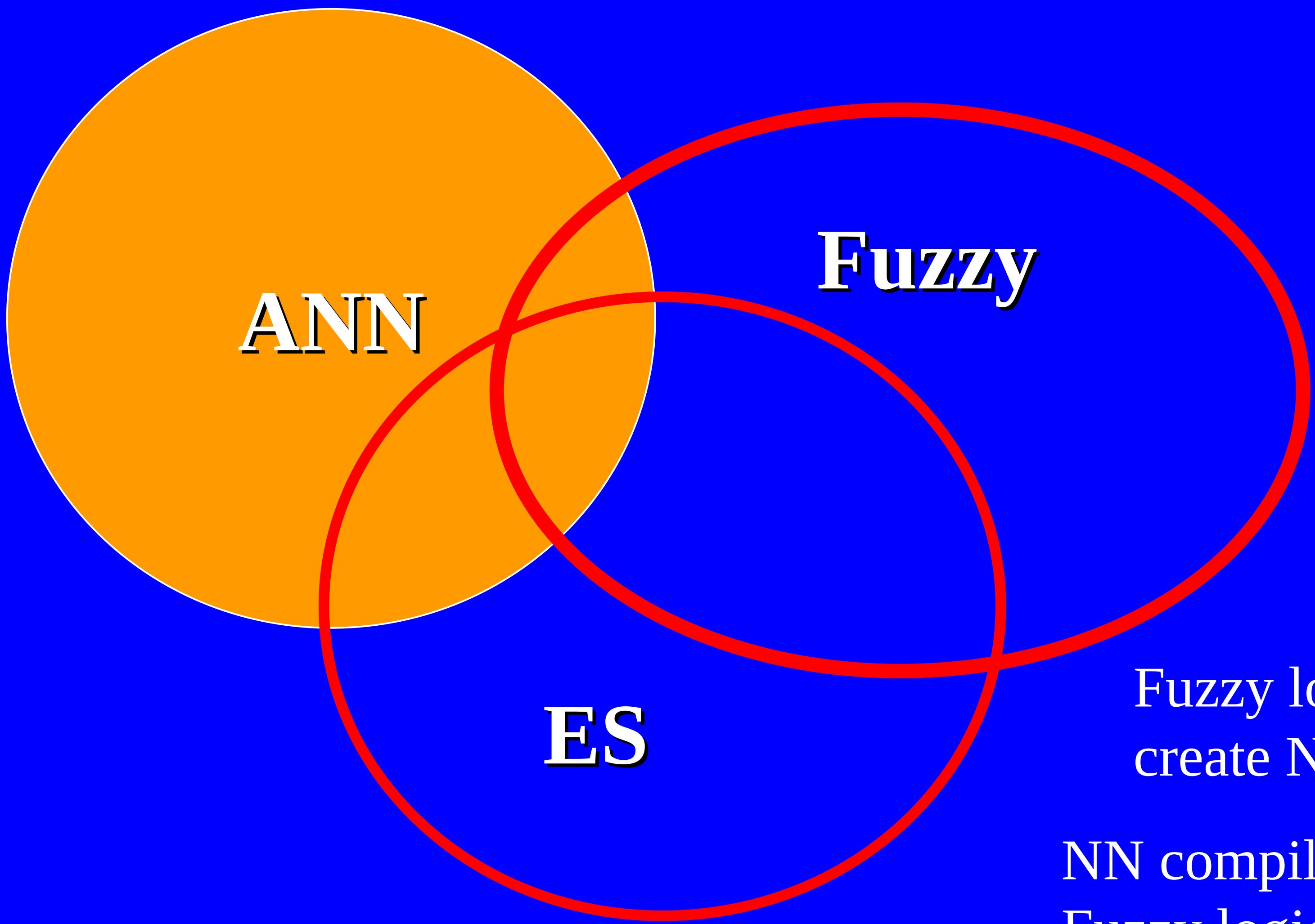
ANN Application Areas

- in marketing
  - unsolicited catalogs to physicians and dentists
  - on purchase, name added to customer db
  - the problem: dormants and how to assign telemarketing time
  - inputs: stat and demographic data; output: rating for each customer
- in stock market prediction
  - input data: economic data
  - output: TOPIX buying and selling timing

ANN and ES

- resource requirements advisor
  - analyzes experiential data in amount of time and effort required to complete previous database projects
- personnel requirements advisor
  - projects personnel resource reqmts. for maintaining networks & workstations at NASA
- diagnosis of airline malfunction
  - critical piece of avionic equipment; Singapore airlines

ANN , Fuzzy, ES



Fuzzy logic to create NN

NN compiled to Fuzzy logic

Fuzzy Logic

- Representation of descriptive or qualitative variables
- Supports gradual transition between variable values
- Mathematical foundation
  - Degree of set membership
- Not the same as probability

Fuzzy Logic in Rule Based Systems (RBS)

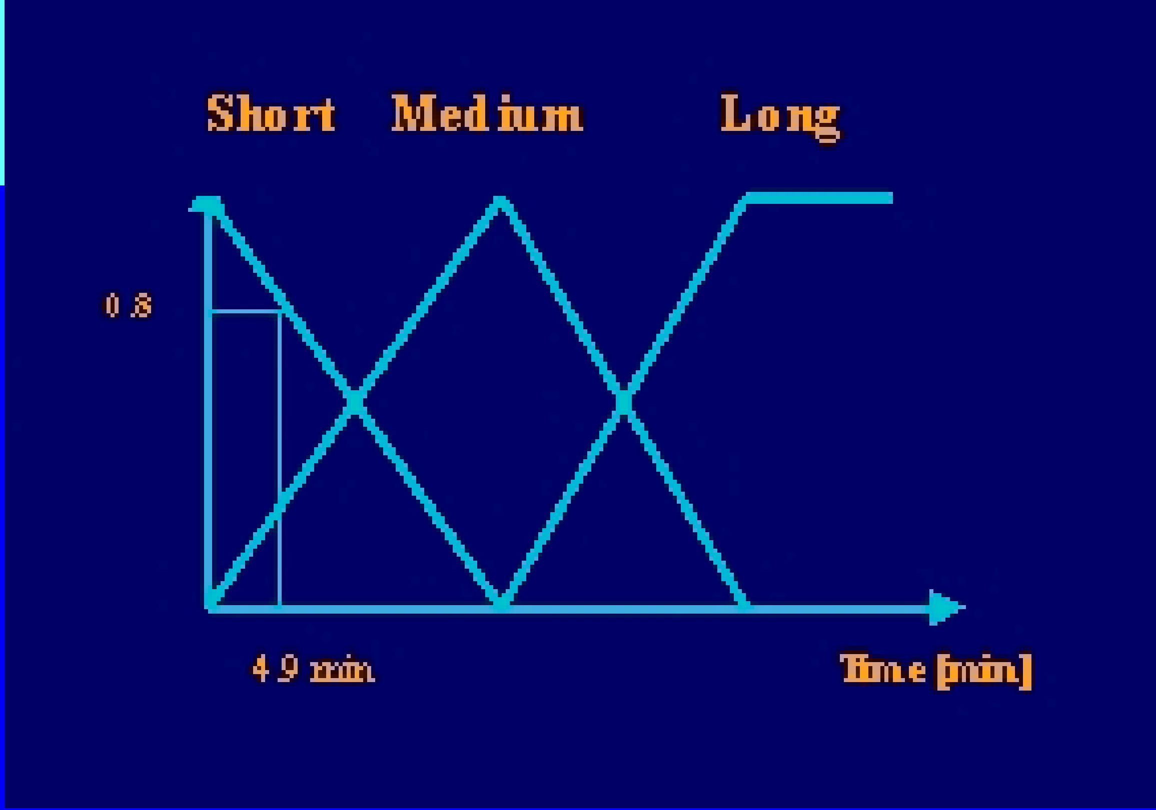
- production rule not “fired” until antecedent condition is met
- fuzzy RBS: all rules executed in each pass with variable strengths
- if antecedent fuzzy proposition is satisfied very well, result matches consequent of rule
- if antecedent fuzzy proposition only partially satisfied, assertion resembles consequent, but only “vaguely”

Benefits of Fuzzy Logic

- provide flexibility and more options: matches human reasoning
- allows for observation and avoids “literal minded” computers
- can be combined with more “exact” schemes

Fuzzy Logic

- Fuzzy logic is an extension of propositional logic (L.Zadeh, 1965)

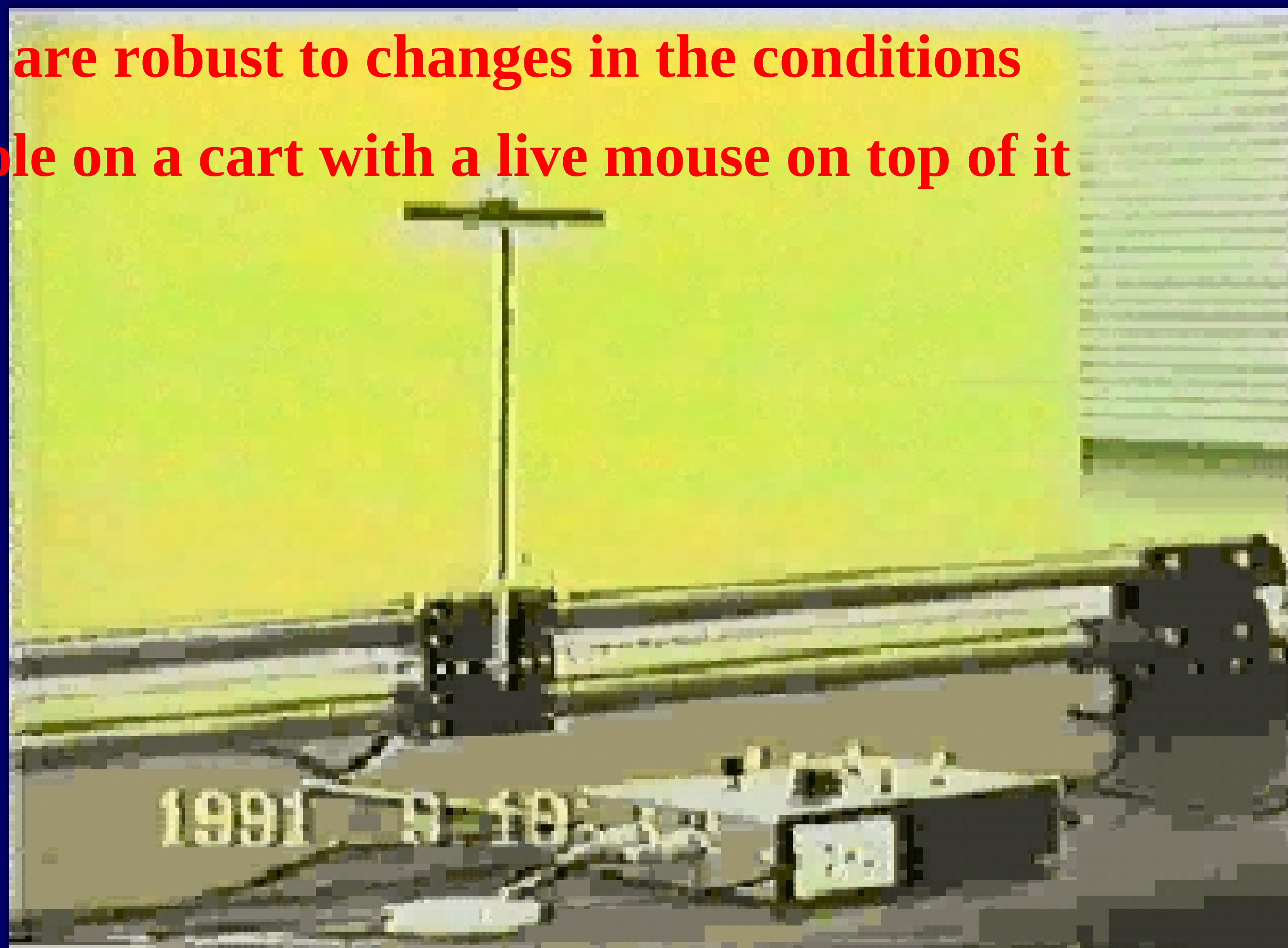


Fuzzy propositions can have truth values between true (1) and false (0), e.g. the proposition “washing time is short” is true to a degree of 0.8 if the time is 4.9 min

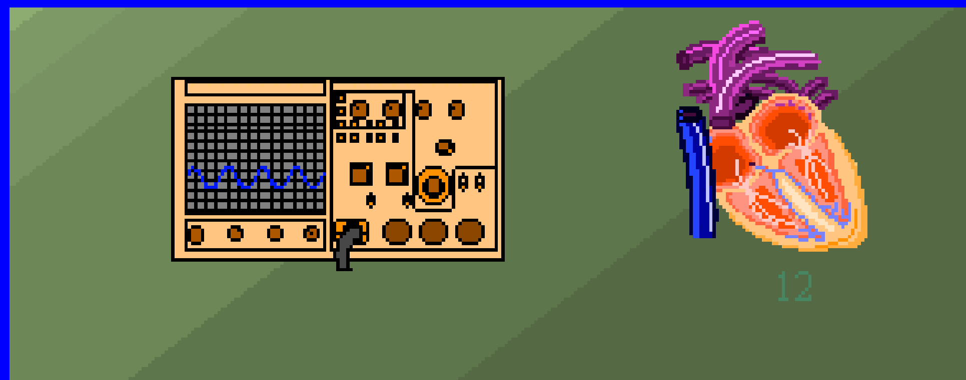
Fuzzy rules represent human knowledge, e.g. “IF wash load is small THEN washing time is short”

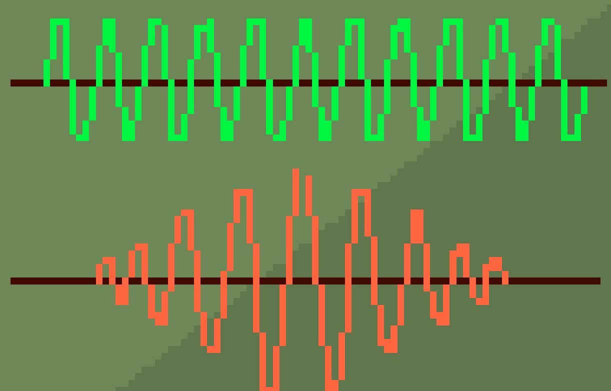
Fuzzy systems

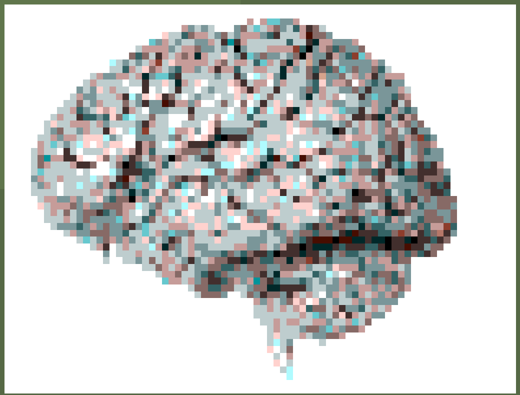
- Fuzzy systems contain fuzzy rules and fuzzy reasoning mechanism
- Fuzzy systems are robust to changes in the conditions
- Balancing a pole on a cart with a live mouse on top of it



Health Monitoring Applications





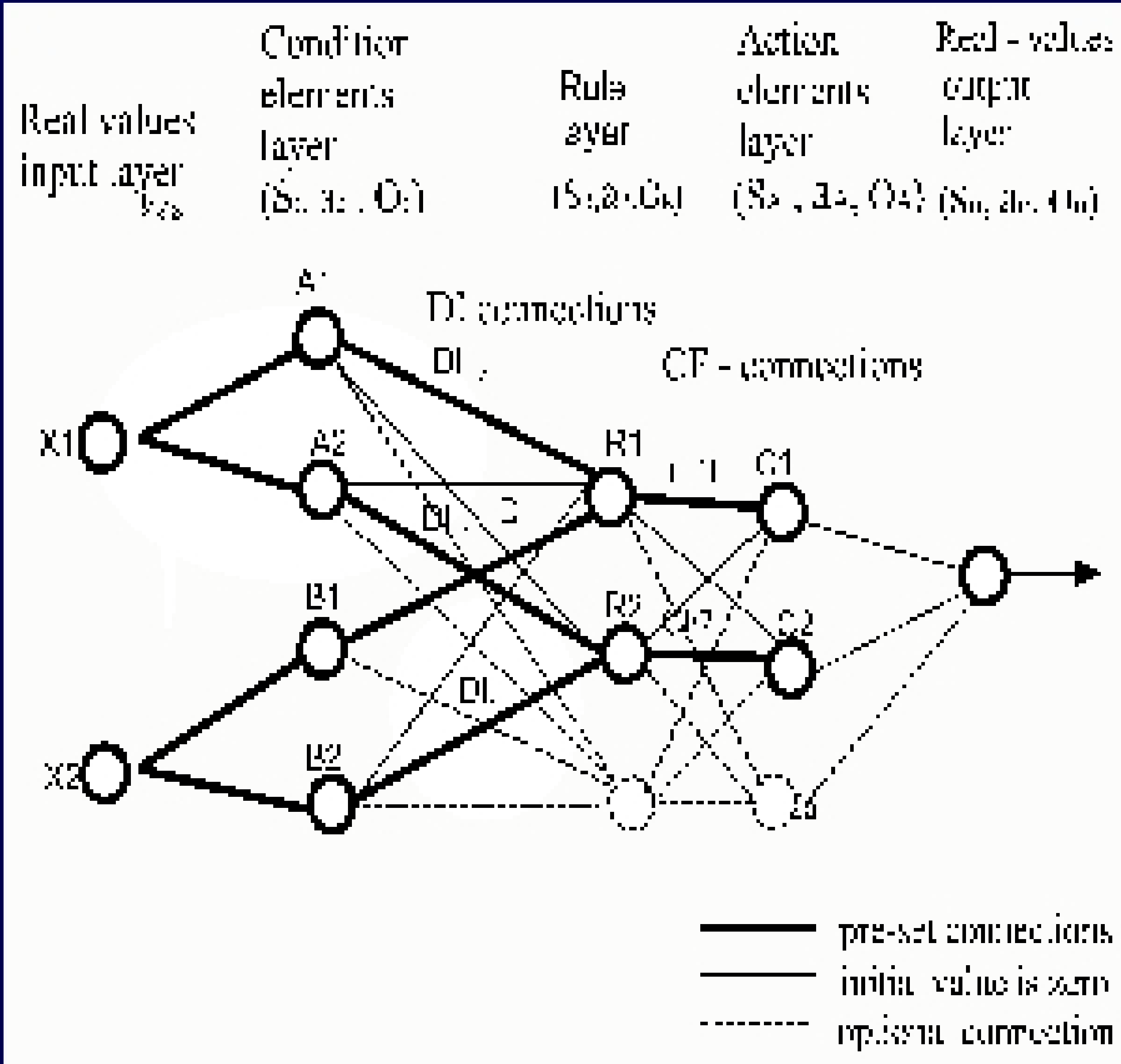


Example 4:

Hybrid neuro-fuzzy- genetic systems

Hybrid neuro- fuzzy- genetic systems

- Combine the strengths of different AI techniques
- FuNN - developed in KEL
- Learning from data
- Rule representation:
    - R1: IF x1 is A1(DI11) and x2 is B1 (DI21) THEN y is C1 (CF1),
    - R2: IF x1 is A2 (DI12) and x2 is B2 (DI22) THEN y is C2 (CF2).



Real values input layer
Condition elements layer
Rule layer
Action elements layer
Real - values output layer
DI connections
CF - connections
X1
X2
A2
B1
B2
R1
C1
C2
pre-set connections
initial value is zero
optional connection

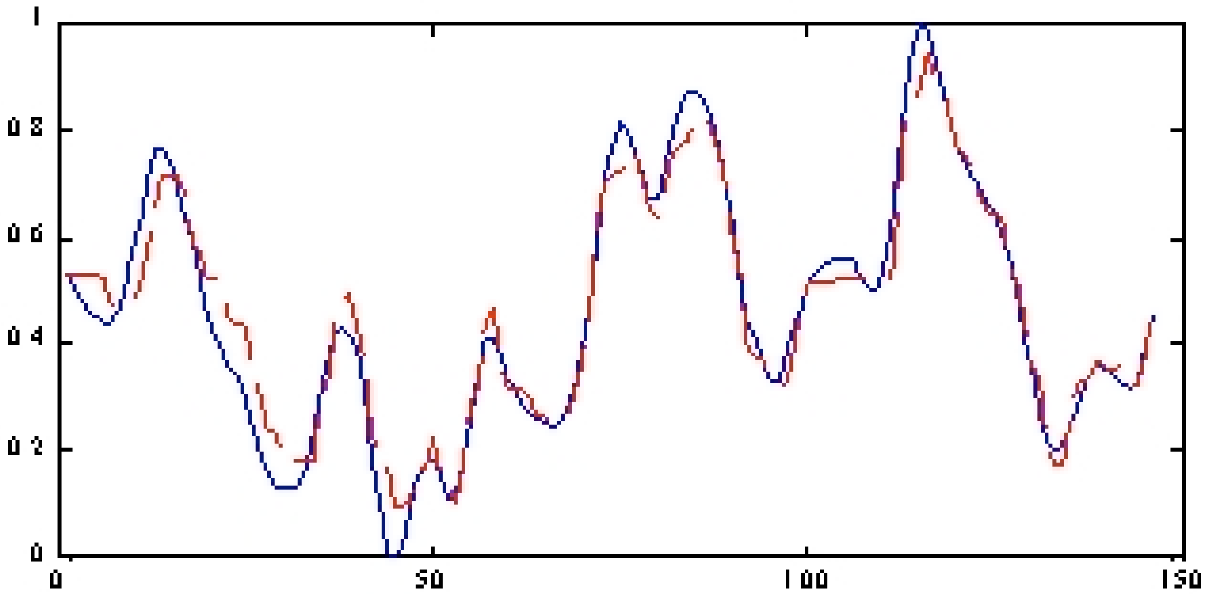
Soft Computing and other AI paradigms

- Soft Computing
- Case-based reasoning
- Data mining
- Cellular automata
- Intelligent distributed systems
- Intelligent agent-based systems
- Artificial life
- Virtual reality
- Emerging computing and AI paradigms:
  brain computers; DNA computing; evolving systems

Evolving Connectionist Systems (ECOS) – a New AI Paradigm?

- ECOS are systems that:
- do “life-long” learning from data
- learn through interaction with the environment
- create and evolve their own algorithm based on the data that has been presented to them
- are self-programmable systems

ECOS applied to a bench-mark data



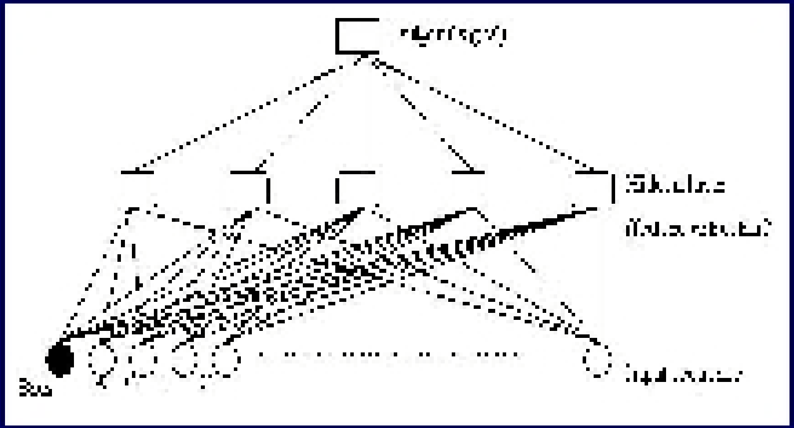
Systems that learn to recognize speech and languages

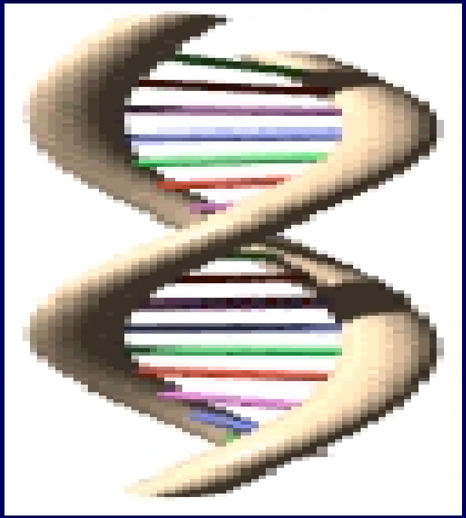
- The Otago Speech Database of NZ English and Maori
- Bilingual English and Maori Web dictionaries
- Voice recognition in a noisy environment
- Computer modeling of speech and language acquisition
- Multi-modal systems (using both auditory and visual information)

Systems that learn to recognise patterns in images

Systems that discover genes in DNA

- ANN are used:
- to identify genes in DNA
- to identify exon/intron splice junctions in mRNA



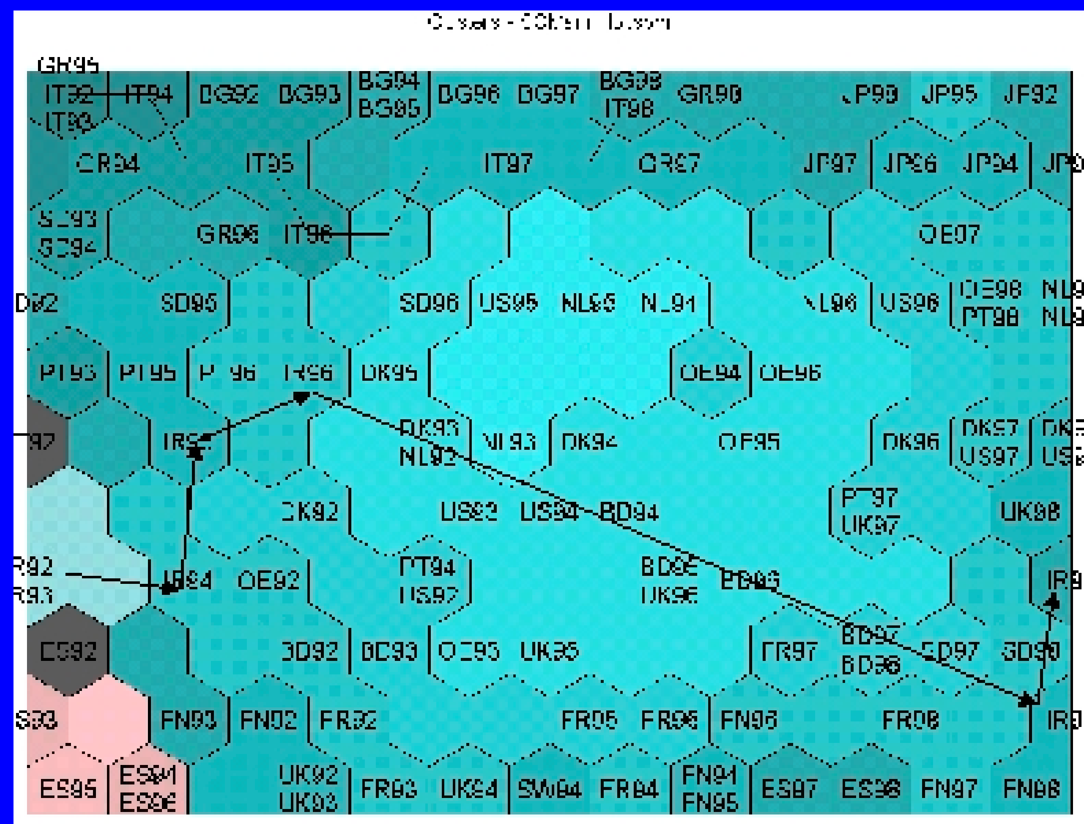


Interactive robots that learn

- Continuous learning of the environment
- Interactive communication in a spoken language
- Rokel - an experimental robot at KEL

Decision support systems that learn from economic and financial data

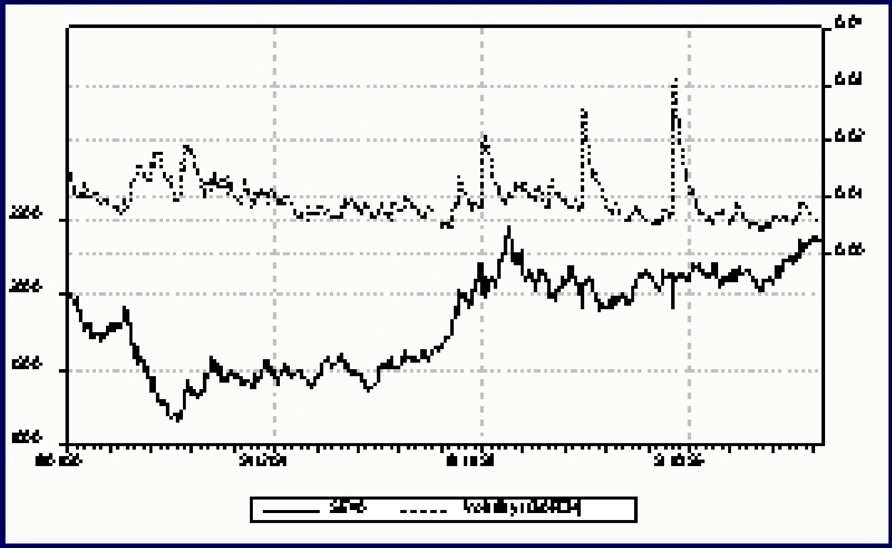
- Information is collected form the Internet
- Trends in economic development are found
- Clusters of countries whose economic performance is similar are indicated

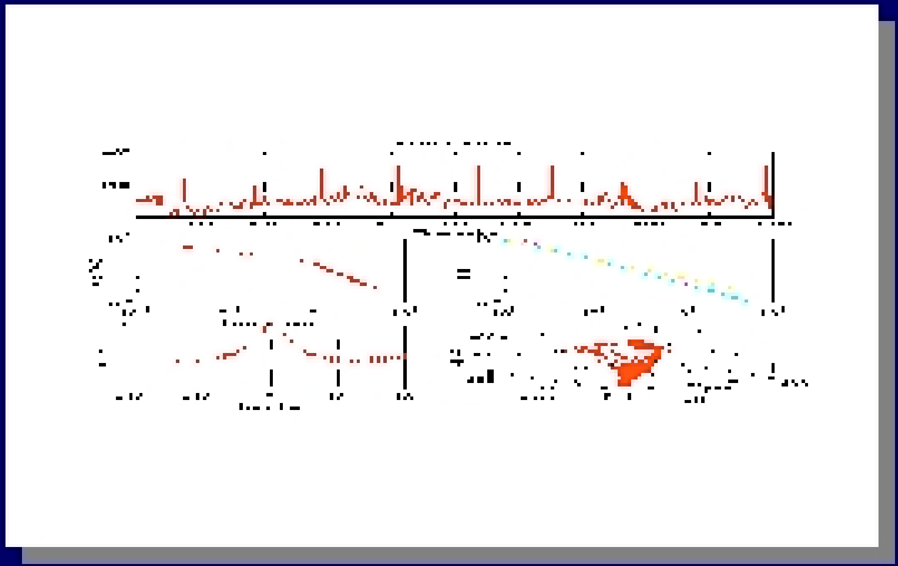


Intelligent systems in horticulture

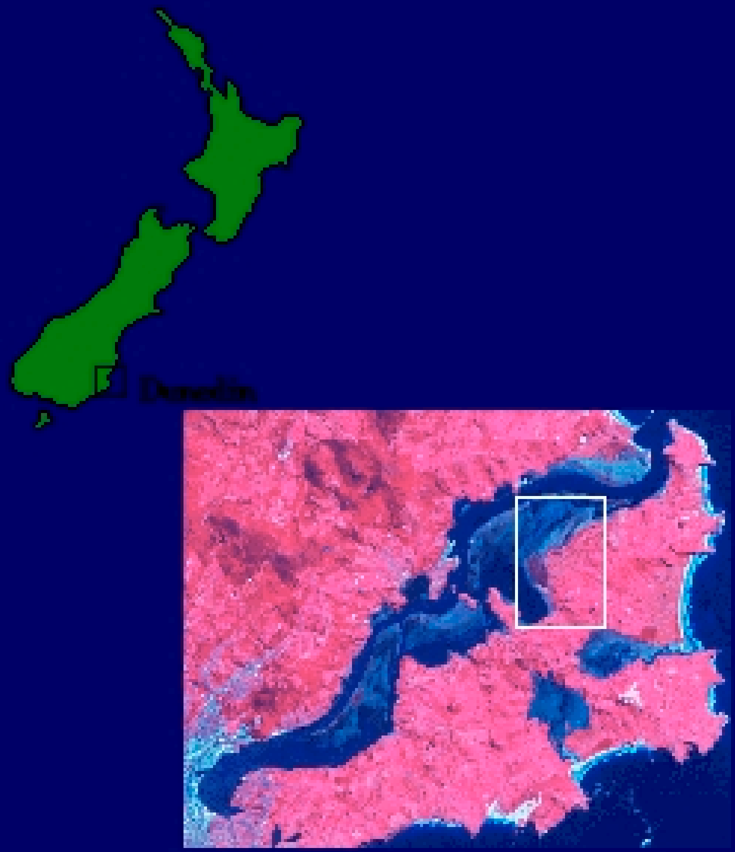
Intelligent systems for the analysis of Chaotic Time Series

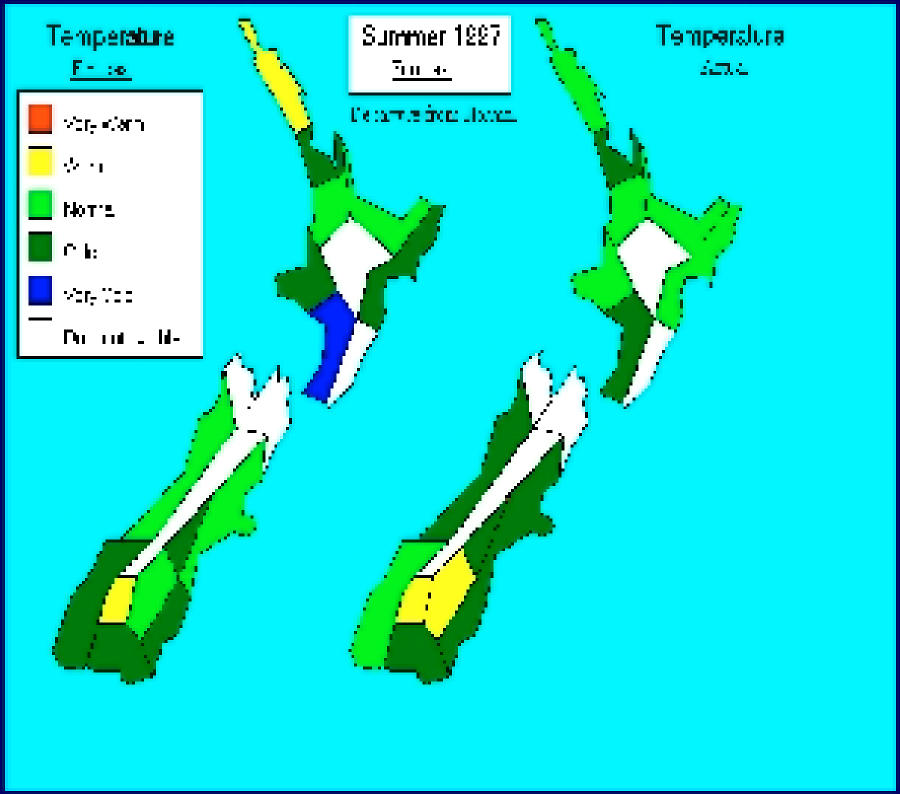
- Medical data: heart rate variability
- Financial data: SE40
- Environmental data: waste water flow





Systems that learn from geographic data





Necessity of Sign Language Recognition
Research for Intelligent System
•Implementing expert knowledge
•Management vagueness in linguistic information
Rehabitation for Hearing Disabled Person
•KSL Recognition System +
•Text to Sign Generation System
Sign Language Interpretation System
Systematic Approach to Gesture Recognition
•Gesticulation
Language-like Gesture
Pantomimes
Emblems
Sign Languages

- Gesticulation
  - » Language-like Gesture
  - » Pantomimes
  - » Sign Languages
  - » Emblems
- Implementing expert knowledge
- Management vagueness in linguistic information
  - » Sign Language
  - » Interpretation System
- KSL Recognition System +
- Text to Sign Generation System
  - » Necessity of Sign Language Recognition

EXAMPLE 5

Autopilot : Performance Tuning in Distributed Computing Environments

– Pablo and Autopilot: Performance Tuning in Distributed Computing Environments
  - Pablo Research Group
  - Department of Computer Science
  - University of Illinois at Urbana-Champaign
  - http://www-pablo.cs.uiuc.edu

Autopilot Toolkit

- Provides a framework for the capture and analysis of real-time application and infrastructure data in a multi-threaded distributed environment
- Offers the ability to control volume of performance data through
  - selective registration and property matching
  - analysis and data reduction at point of collection
  - constant, periodic, or on-demand transmission of data
  - ability to dynamically enable/disable data collection
- Includes a control interface to allow steering of infrastructure policies and applications, either interactively or via automated decision procedures

Basic Autopilot Concepts

- **Sensors: provide data to remote processes, allowing real-time monitoring**
  - intrinsic (procedural - push)
  - extrinsic (threaded - push)
  - transfer data when requested by remote process (pull)
- **Sensor Attached Functions: transform sensed data via user-defined functions before it is recorded by the sensor, providing an important data-reduction technique**

Basic Autopilot Concepts

- **Actuators: provide remote processes the ability to invoke local functions or update data, allowing remote steering**
  - synchronous (application controls when updates are made; requests may be held in pending buffer)
  - asynchronous (updates are made when request received from external agent)
- **Properties: key-value pairs that are associated with and used to identify a sensor or actuator, allowing remote processes to be selective about the sensors and actuators they connect to**

Basic Autopilot Concepts

- **Sensor Client: a process that connects to one or more sensors with matching properties and receives data from those sensors**
- **Actuator Client: a process that connects to one or more actuators with matching properties and sends data to those actuators, causing application variables controlled by the actuators to be updated or functions to be invoked**

Basic Autopilot Concepts

- **Autopilot Manager: a daemon process that is responsible for handling registration requests from sensors and actuators, and matching sensor client and actuator client requests to registered sensors and actuators.**
  - AutopilotManager daemons may be run on multiple hosts throughout the computational grid, allowing sensors, actuators, and clients to tailor data transfer volumes to appropriate levels for local and distant tasks.

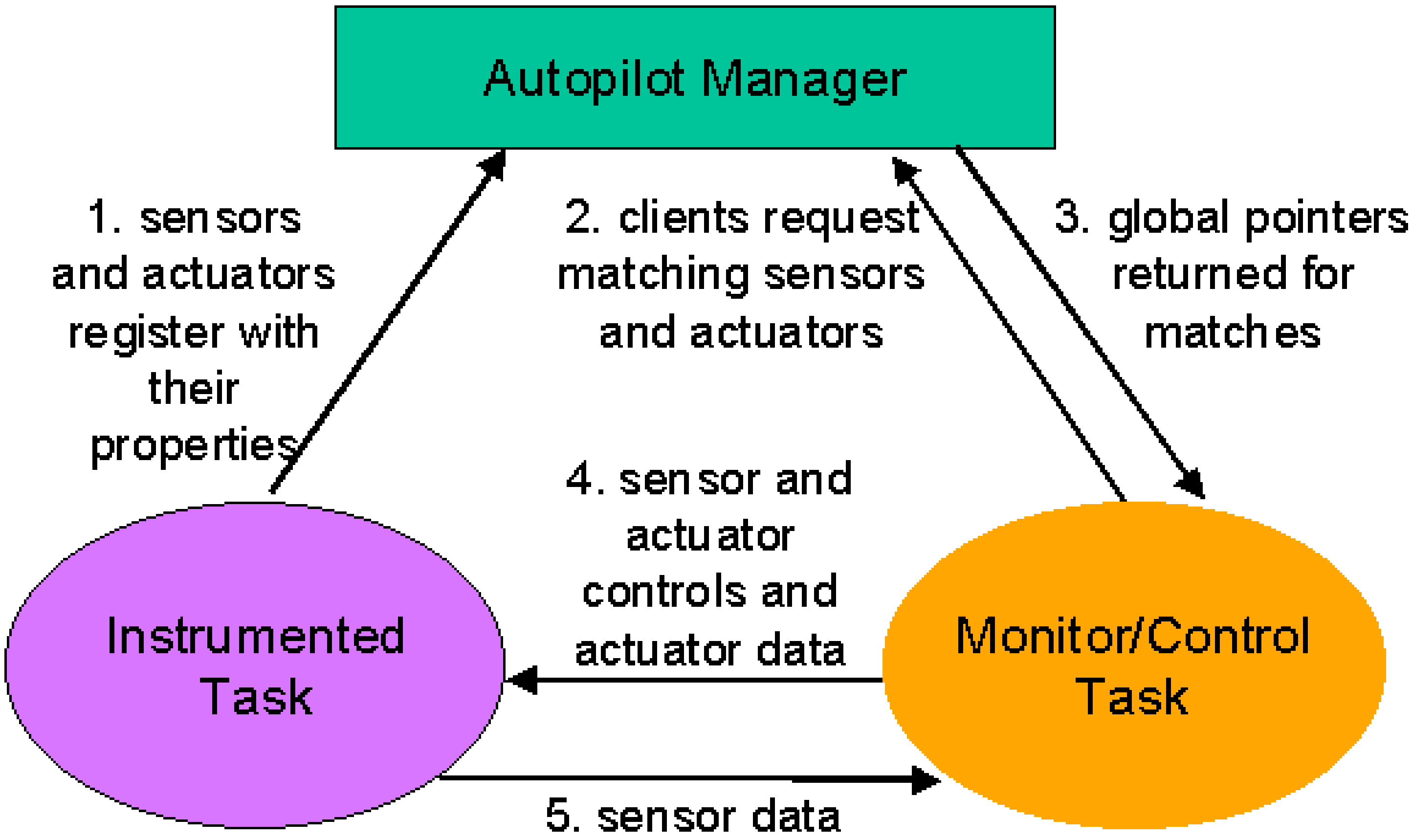
Tagged Sensors, Actuators, Clients

- Information about the structure of the data is forwarded when a client first connects to a matching sensor or actuator, allowing the client to perform verification checks and ignore unwanted data.
- Tagged data sets map naturally into what we normally think of as event trace records.
- Sometimes called “SDDF-enabled” because the buffer contents can easily be translated to SDDF

Autopilot and Nexus/Globus

- **Autopilot uses the Nexus component of the Globus toolkit (http://www-globus.org) to provide...**
    - communication substrate & multithreading capabilities
- **Nexus creates a global address space that encompasses all processes executing on a distributed network**
- **Nexus Remote Service Requests (RSRs) used by Autopilot classes to transmit messages, insuring optimal underlying transfer protocol**
- **Nexus multi-threaded handlers used by Autopilot classes to process RSRs**
- **Most Nexus details hidden by Autopilot classes**

Autopilot Component Interactions



Instrumented Tasks

- May contain multiple sensors and/or actuators
- Many instrumented tasks may be active at any given time
- May register sensors and actuators with multiple Autopilot Managers running on different hosts

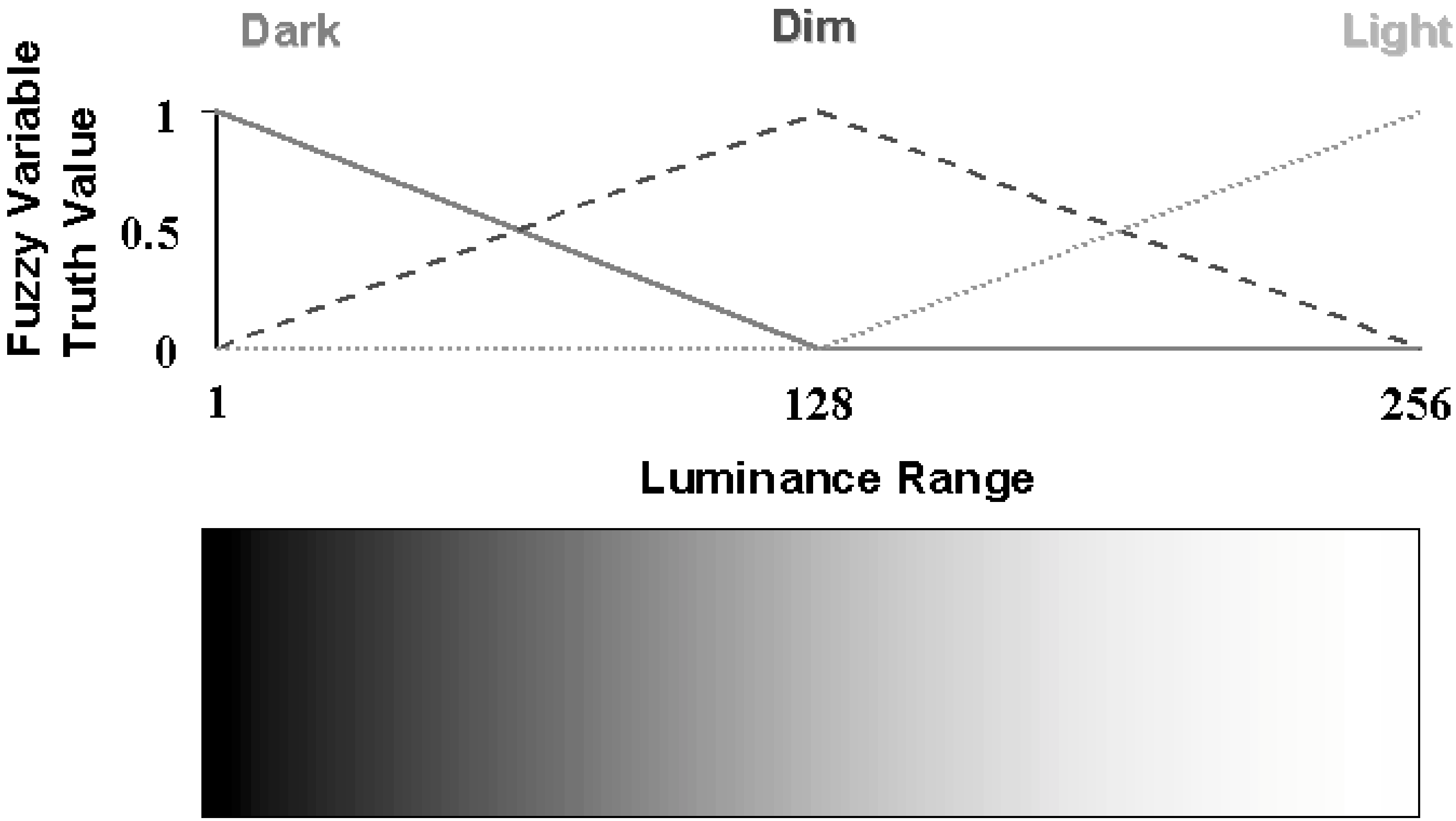
Monitor/Control Tasks

- May contain multiple sensor clients and/or actuator clients
- Many monitor/control tasks may be active at any given time
- May query multiple Autopilot Managers running on different hosts
- May implement "human in the loop" (Autodriver, Virtue) or automated fuzzy logic decision server (PPFS II)
- May be monitor only, writing collected data to a file or displaying it

Fuzzy Logic Rationale

- **Humans rely on qualitative rules**
  - If the system is BUSY, backups should be POSTPONED
- **Fuzzy logic expresses these rules formally**
  - Elegantly integrates qualitative data
  - prefetch for small, sequential, read only requests
  - Supports conflicting goals
  - Processes “gray” statements
- **Well-developed theory and software base**

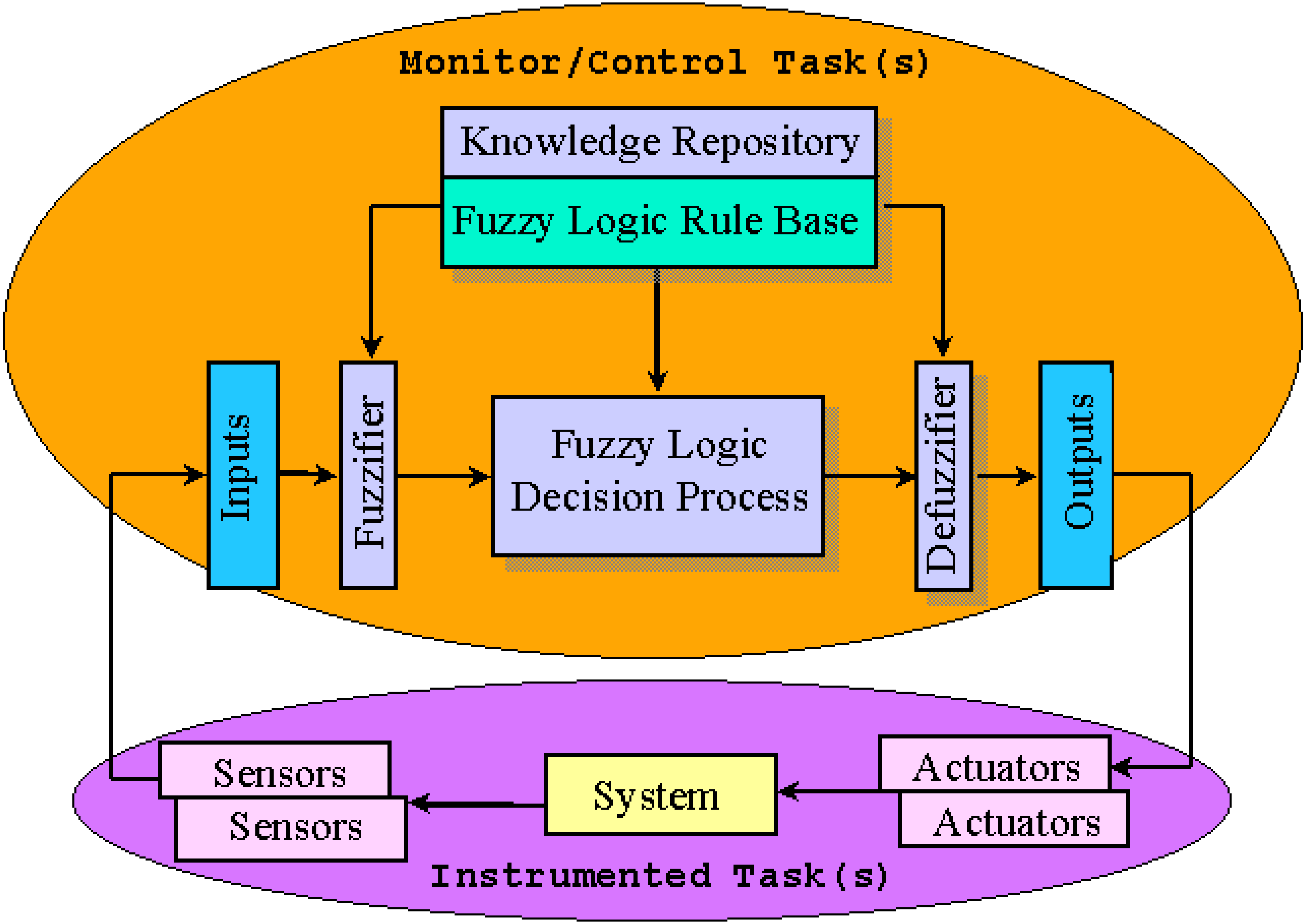
Fuzzy Variables: Degrees of Truth



Fuzzy Controller Structure

- **Fuzzifier**
  - scales and maps input variables to fuzzy sets
- **Inference mechanism**
  - approximate reasoning block
  - deduces the control action
  - Compositional Rule of Inference (CRI)
- **Defuzzifier**
  - converts fuzzy outputs into control signals
  - several defuzzification methods

Fuzzy Logic Decision Infrastructure



Fuzzy Inference Process

- **Interpreting an IF-THEN rule**
  - evaluate the antecedent (fuzzify inputs)
  - apply the result to the consequent
  - if the antecedent (premise) is true to some degree then the consequent is true to the same degree
- **Multiple rules active for the same input**
  - multiple outputs combined
  - combination can yield complex behavior
- **Defuzzification yields final output(s)**

Adaptive File Striping Example

- **Files striped across disks based on utilization**

- disk parallelism is intra-request disk stripe count
- if the system is highly utilized
- decrease individual request parallelism
- this decreases contention
- if the system is under utilized
- increase individual request parallelism
- this increases throughput

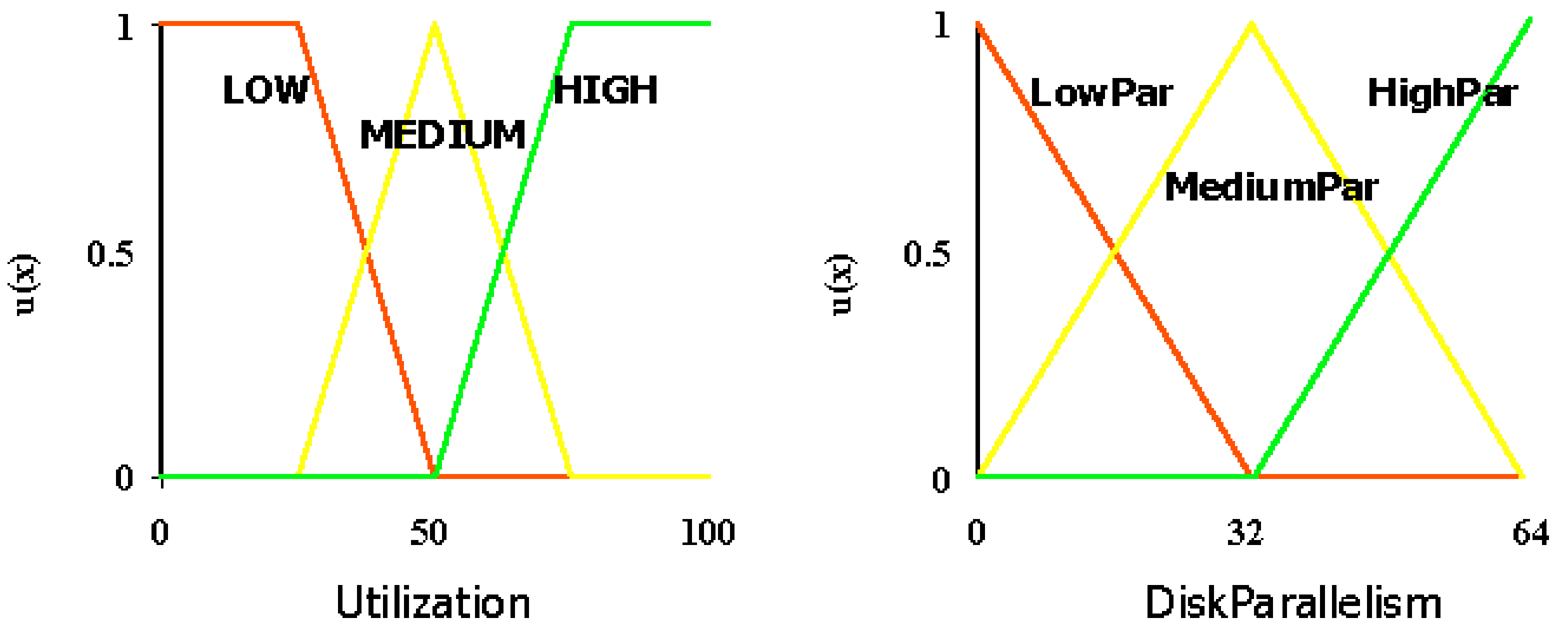
Fuzzy Rule Set Example

- if (Utilization == LOW)
- {DiskParallelism = HighPar;}
- if (Utilization == MEDIUM)
- {DiskParallelism = MediumPar;}
- if (Utilization == HIGH)
- {DiskParallelism = LowPar;}
- One can also associate certainties with rules!

Fuzzy Variable Definitions

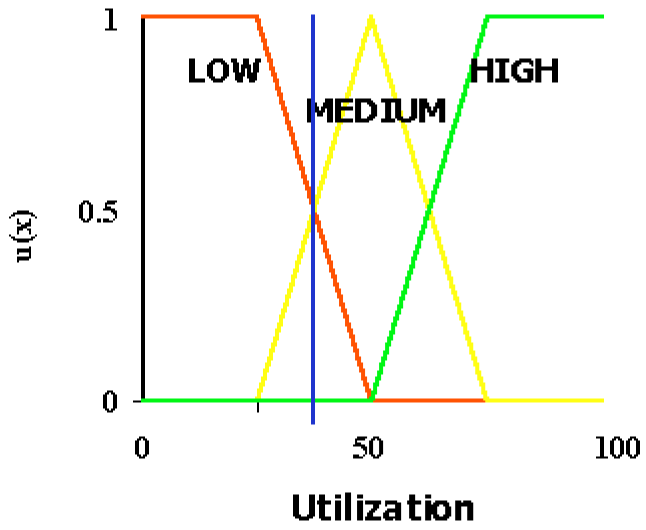
```
rulebase StripingRB;
var Utilization(0,100) {
set trapez LOW (0, 25, 0, 25);
set triangle MEDIUM (50, 25, 25);
set trapez HIGH (75, 100, 25, 0);
}
var DiskParallelism(0,64) {
set triangle LowPar ( 0, 0, 32);
set triangle MediumPar (32, 32, 32);
set triangle HighPar (64, 32, 0);
}
```

Fuzzy Variables (Membership)



- The truth of any statement is a matter of degree
- Membership function is the curve that defines how true a given statement is for a given input value

Fuzzification Process

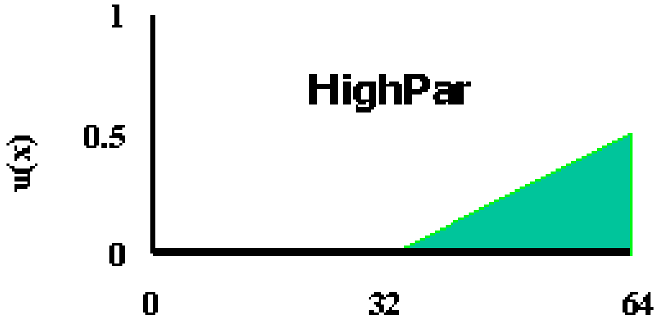


Utilization	=	37.5
LOW	~	.5
MEDIUM	~	.5
HIGH	~	0

Fuzzy Inference

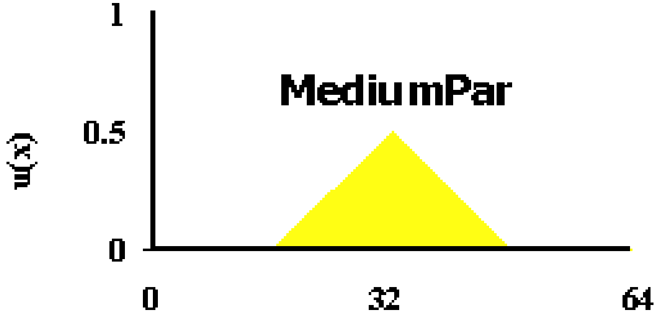
```
if (Utilization == LOW)
  {DiskParallelism = HighPar;}
```

LOW has truth value = .5



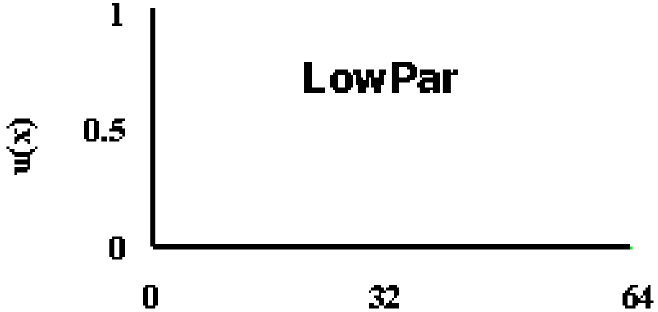

```
if (Utilization == MEDIUM)
  {DiskParallelism = MediumPar;}
```

MEDIUM has truth value = .5

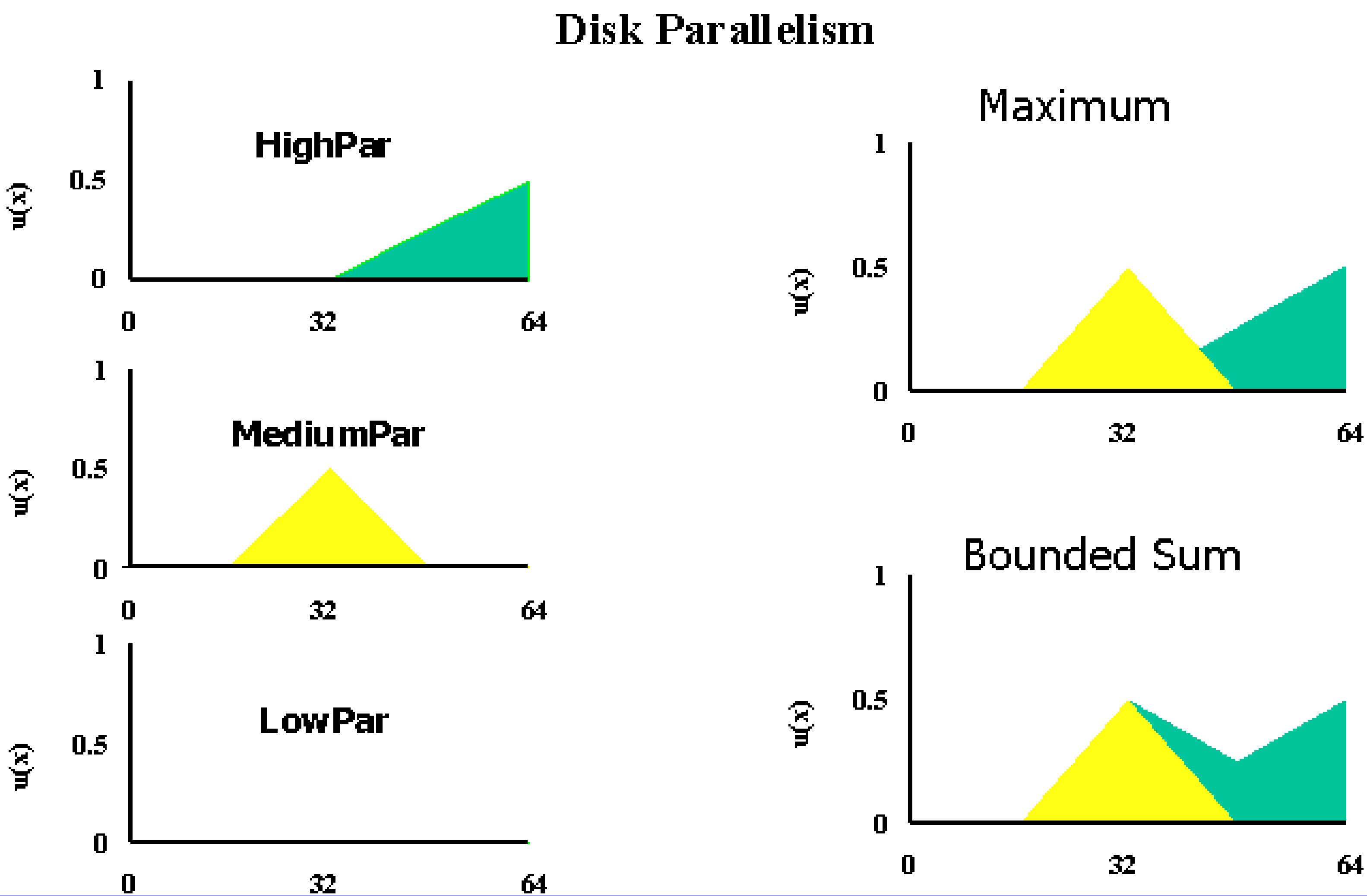



```
if (Utilization == HIGH)
  {DiskParallelism = LowPar;}
```

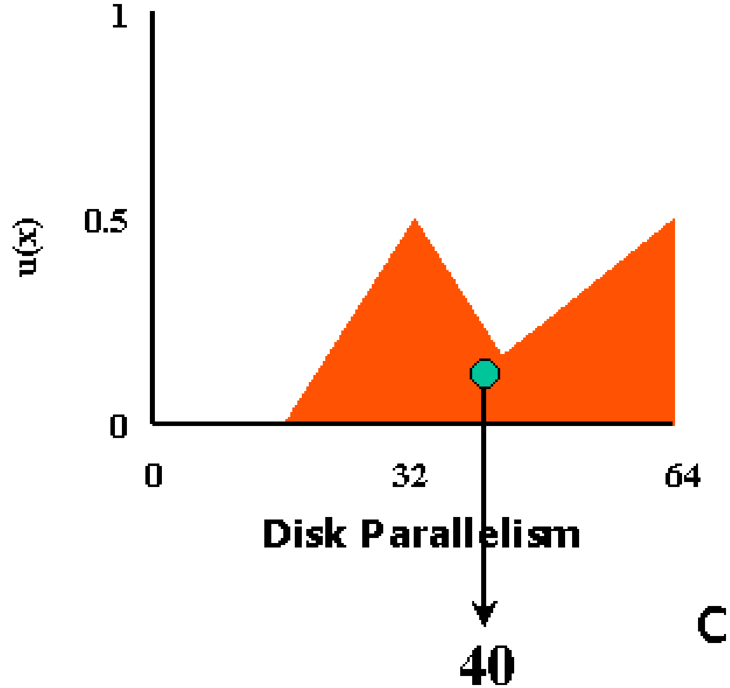
HIGH has truth value = 0



Fuzzy Composition



Defuzzification



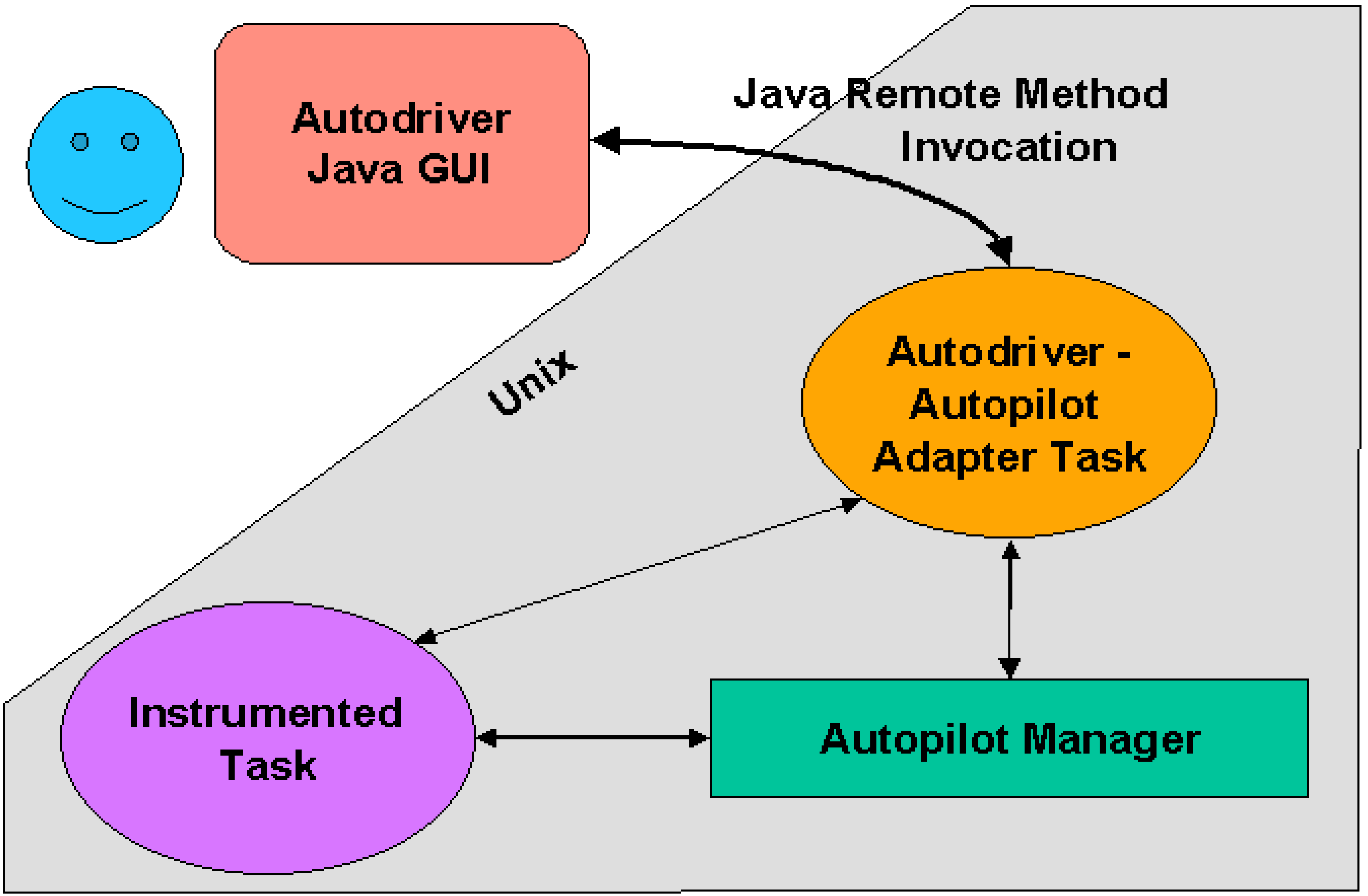
- Defuzzification methods
  - center of gravity
  - first maximum
  - mean of maxima

$$\text{center of gravity} = \frac{\int \mu(x)\,dx}{\int x\,dx}$$

Decision Infrastructure Summary

- **Autopilot sensors provide streams of measurements. After fuzzification, these streams define the values of the input fuzzy variables.**
- **Rules whose conditions are non-zero all contribute to determining the value of the output fuzzy variables. After defuzzification, the value of the output fuzzy variables define the actionstaken by the Autopilot actuators.**
- **Fuzzy logic handles noisy data and conflicting goals.**
- **Fuzzy logic separates data sets (definition of fuzzy variables) and rules (assertions and consequents) allowing each to be independently adjusted for a particular computing environment without re-coding the decision procedure**

Autodriver Monitor and Control Architecture



Example 6

Prediction of Customer’s Response to Advertisement

So how might you do better?

- Build STATISTICAL models to predict response?
- YES but its simultaneously
  - complex
  - non automated
  - hard to build
  - even harder to explain
  - highly assumption dependent
  - performance is questionable

Use a Neural Network

- often more accurate than a statistical model
- easier to apply
- insensitive to data noise
- highly non-linear
- has few assumptions
- offers good levels of performance
- <u>easy to describe BUT impossible to explain WHAT the model is</u>

BUT

YOU should now be seriously considering greater use of Neural Networks

embed them in a statistical research design

Genetic Programming and Evolutionary Computing Targeters

- Offers prospect of even BETTER levels of performance than Neural Networks
- Its more FLEXIBLE technology
- Highly adaptive

- **BUT** it may require High Performance Computing to get the best results

High Performance Computing

- Highly parallel computers offer vast amounts of computational power
- Leading edge machines will soon reach teraflop speeds with hundreds of gigabytes of RAM
  - e.g. a 50,000 times faster than a PC

BUT

How many of YOU are either already USING or PLANNING to use High Performance Computers?

NO...

- The aim is not just to make existing methods go 1,000's of times faster
- It is to **EXPLOIT the NEW OPPORTUNITIES** that HPC **provides YOU with** for doing database marketing in a more computationally orientated and machine intelligent manner

Fuzzy Logic Modeling is also worth considering!

- As good as neural networks
- Unlike neural networks it can incorporate your knowledge, skills, and marketing experience and intuition
- Easy to apply
- The results can be described in Plain English

An Example

- You want to mail clients who meet the following criteria..
  - middle aged
  - recent customers
- You can convert these linguistic statements into a mathematical prediction model
- You can optimize performance subsequently

An Example

- **You want to mail clients who meet the following criteria..**
  - middle aged
  - recent customers
- **You can convert these linguistic statements into a mathematical prediction model**
- **You can optimize performance subsequently**

A Case Study

- Model a database of several million customers
  - A simple Fuzzy Logic Model achieved an accuracy level of 46%
  - A Genetic Algorithm optimized version of the same FLM achieved 85%

Fuzzy Logic Model based Segmentation

- If you then applied this FLM to mail 50,000 best prospects you would achieve a response rate of **25%**
- Not quite as good as before but this reflects
  - use of only 3 variables
  - no search for good variables
  - a desire for an easily explanation
  - still 5 times better than a standard geodemographics

What Else?

- What about an Exemplar Machine Learning technique?
- On previous example it achieved an accuracy of over 85% after seeing a random sample of only 550 cases
- Yet, it is
  - fully automated
  - works well with very low response rates

Building Intelligent Hybrid Computational (Marketing) Systems

- <u>Need to think in terms intelligent adaptive database marketing systems that incorporate feedback loops</u>
- Intelligence isn't just due to the use of individual AI tools, it needs to be built into the design of the marketing process itself

Summary

- Fuzzy logic captures intuitive, human expressions.
- Fuzzy sets, statements, and rules are the basis of control.
- The technique is extremely powerful, and appears in mills at a growing rate.
- Many mixtures possible
- Many systems practically used in various applications