Biomimicry for Optimization, Control, and Automation

Kevin M. Passino

Dept. Electrical Engineering
The Ohio State University
2015 Neil Avenue
Columbus, OH 43210-1272
(614) 292-5716, passino@ee.eng.ohio-state.edu
http://eewww.eng.ohio-state.edu/~passino

T · H · E OHIO STATE UNIVERSITY

Copyright

INTRODUCTION

Focus: Control design methodology, biomimicry scientific foundations.

➙ Control systems are a key part of automation.

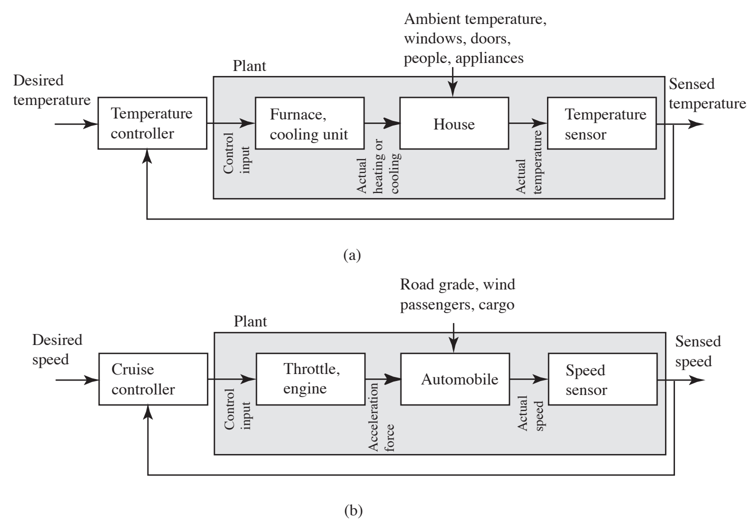

Figure 1: Example control systems: (a) temperature control in a home, and (b) cruise control for an automobile.

➙ Design objectives: Reducing the effects of adverse conditions and uncertainty, behavior in terms of time responses (stability, rise-time, overshoot, settling time, steady state tracking, etc.)

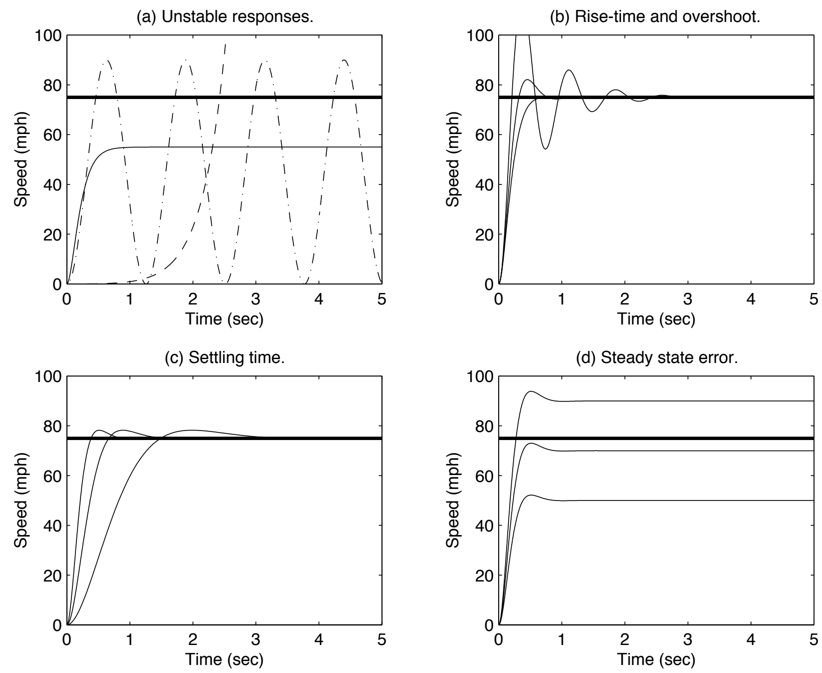

Figure 2: Desired speed (thick solid line at 75 mph) and possible closed-loop time responses resulting from different controller designs.

- Robustness is often the key design objective.
- Engineering goals/constraints (often ignored in the university): Cost, computational complexity, manufacturability, reliability, maintainability, adaptability (to similar applications), expandability (for new-improved versions), understandability, politics.

Control System Design Methodology

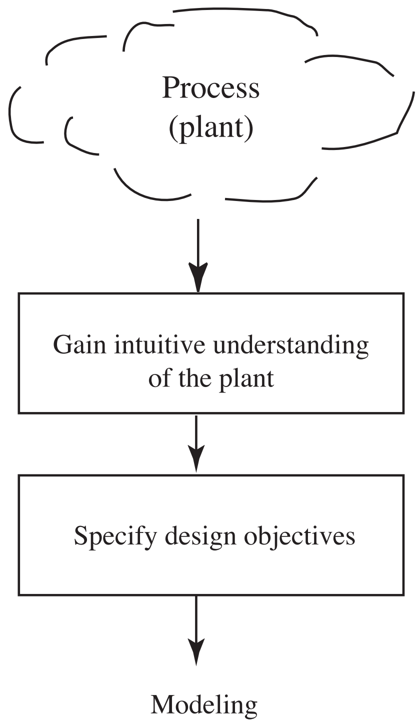

Figure 3: Early steps in the control system design procedure.

- Modeling forms the foundation for control design methodology. No model is perfect; but, even uncertainties can be represented.

- Construct model(s) (physics, system identification) and representations for uncertainty

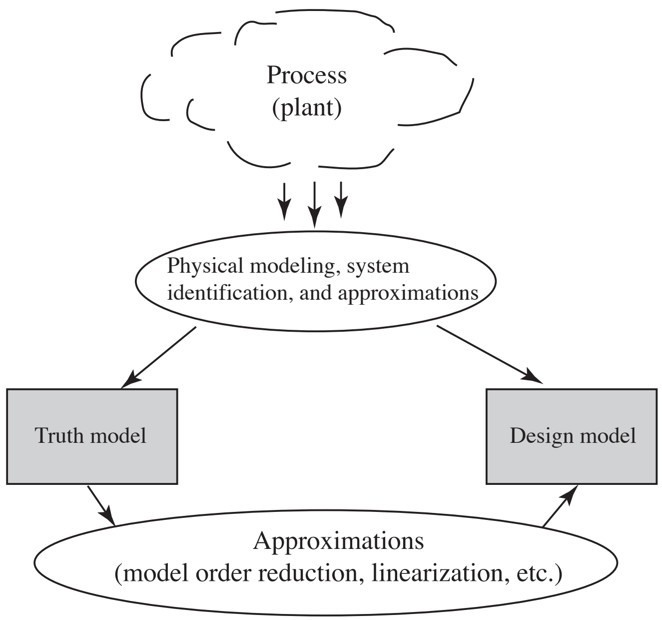

Figure 4: The modeling procedure.

- Analyze model accuracy and system properties

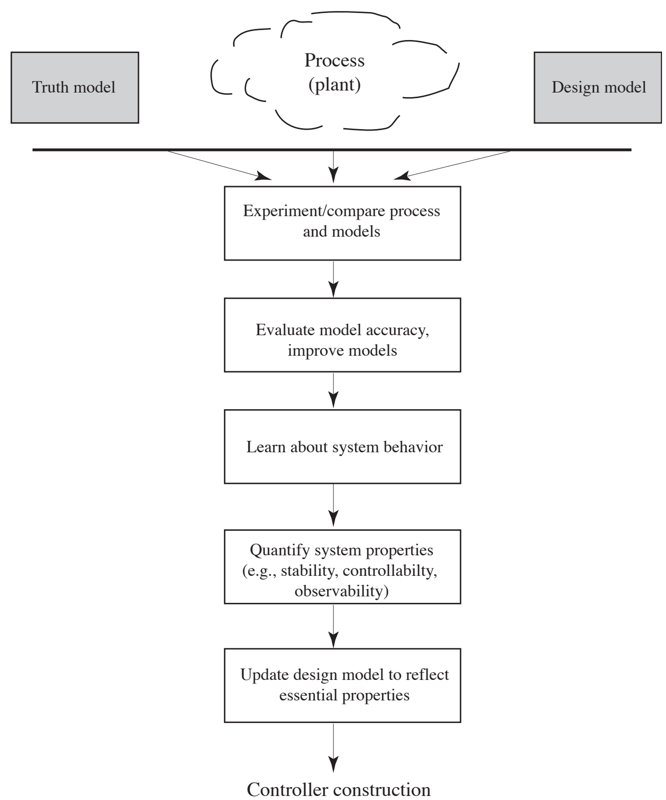

Figure 5: Model evaluation/adjustment process.

➙ A model should be simple, but accurate enough to succeed in control design.

- Models help give intuition on how the plant behaves (e.g., stability, controllability, observability, rates, etc.).
- Construct and evaluate the control system

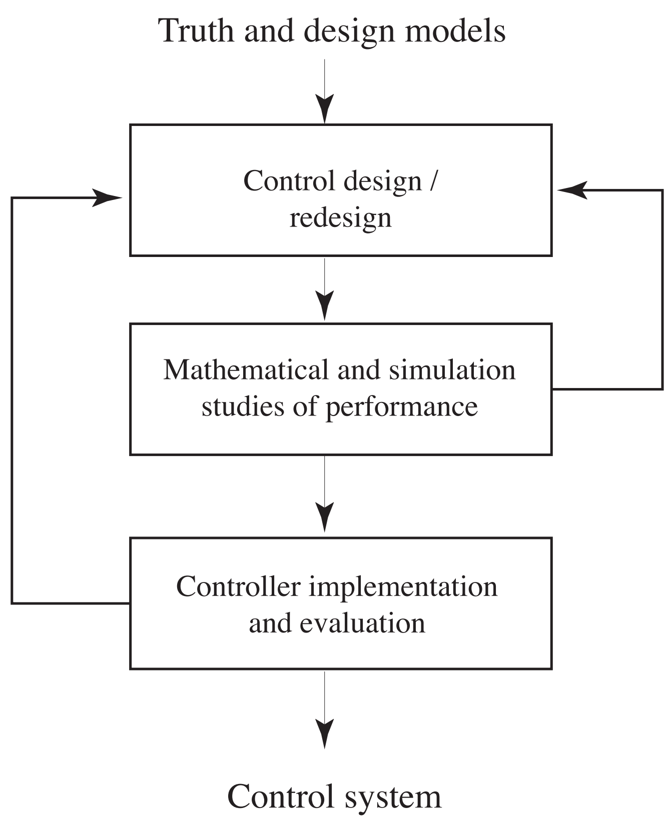

Figure 6: Controller construction and evaluation process (for convenience, it ignores the possibility of iterative improvement of models).

- Controller synthesis: "conventional methods" (PID, classical control, state space methods, optimal control, robust control, adaptive control, nonlinear control, stochastic control, discrete event and hybrid systems.

➙ Analysis of closed-loop control system performance (analysis entails the use of mathematics, simulations, and experimentation; each has its own advantages and disadvantages):

  - You prove stability of a model not the system.
  - Simulations are abstract representations of physical systems.
  - Experiments are typically for *one* plant, not the class of all possible ones that the controller could eventually be applied to.

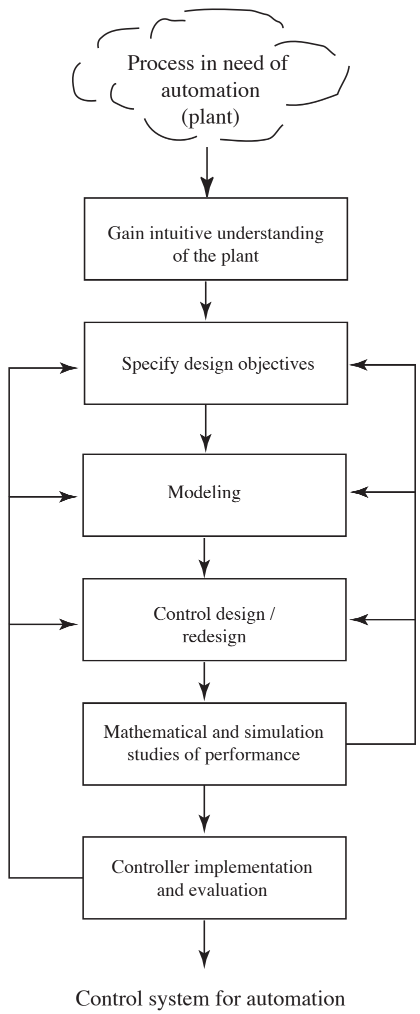

Figure 7: Flowchart of control system design steps.

Methodology Without Mathematical Models: The Use of Heuristics

- In many practical control problems it is difficult or impossible to obtain a mathematical model and in this case we rely on the use of heuristics for control design (e.g., in PID control).
- Heuristic control design methodology?

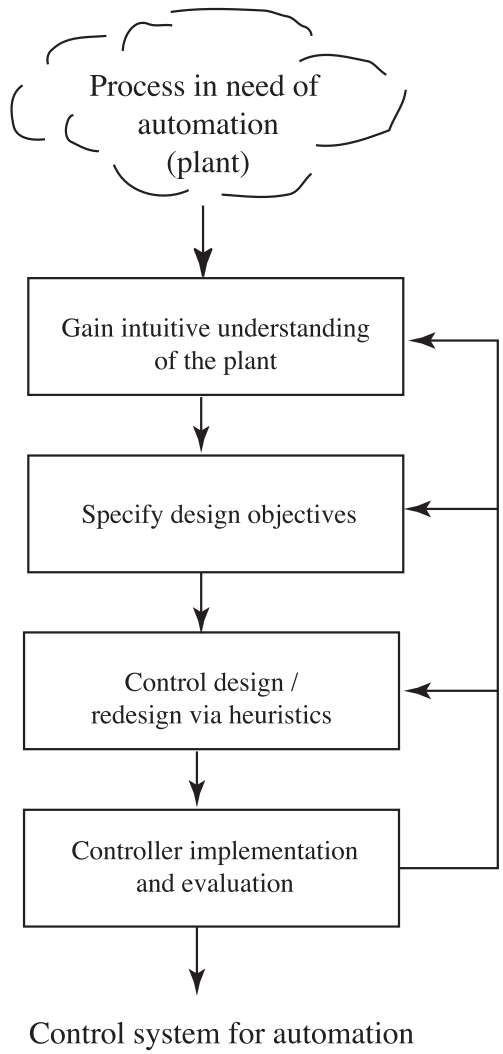

Figure 8: Heuristic control design methodology.

- Close relationships with traditional design methodology (e.g. for PID control).
- Performance evaluation? Only possible via implementation studies? No model for simluation or mathematical analysis?
- Some use a truth model in the heuristic control design methodology, one that would not be useful in a design methodology beyond in simulation/tuning (e.g., not useful for mathematical analysis).

Complex Hierarchical Control Systems for Automation

- Computer decision-making can be very complex, hierarchical, and distributed, and be designed to solve complex automation problems, even when the “plant” contains intelligent human adversaries.

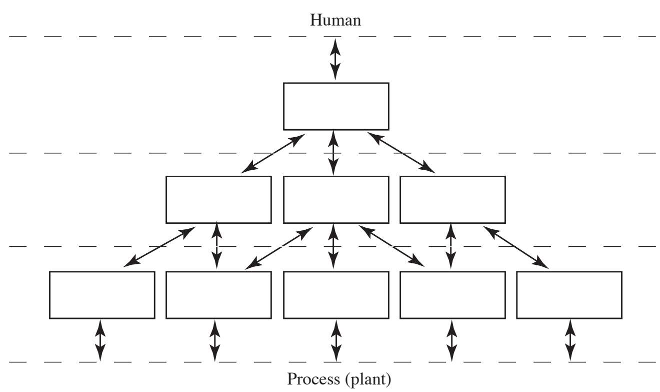

Figure 9: Hierarchical control system.

- Examples: Robots, autonomous vehicles (ground, air, or underwater), manufacturing and process control, networks of intelligent agents

Functional Architectures

- What functions are needed for control and automation?
- Examples: Interface to humans, interface to other systems, to control parts of the system by sequencing operations and guiding the overall behavior of the system, to handle exceptions to normal operation of the control system , to cope with special operating conditions.
- Why do hierarchies and distribution arise?:
  - Need to "divide and conquer"
  - Goals and priorities often split the performance of tasks into different parts (a "behavioral hierarchy")
  - Components physically distributed.
  - Complexity dictates need for distribution.
  - Human interface unique, relative to plant interface.

– Algorithm developers often view problems hierarchically.

- Organizing the controller into a "functional architecture"

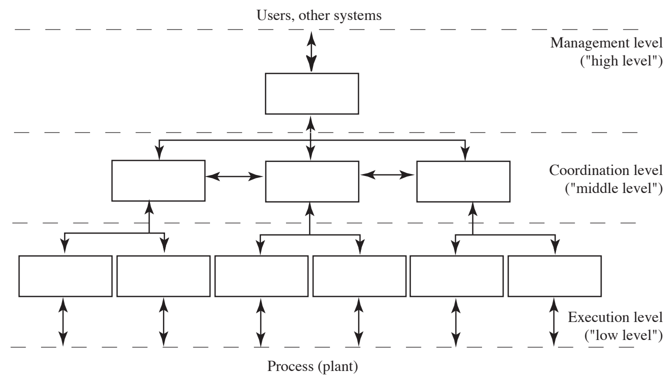

Figure 10: A typical hierarchical distributed control system ("levels").

- Fundamental Operational Characteristics: Task division, time scales, models.
- Examples: Temperature control in large building, coordination of multiple autonomous vehicles.

Design Objectives/Methodology for Automation

➙ More sophisticated than for traditional single-loop control systems.

➙ Examples: Dynamically changing composition of the specifications of the traditional control systems, proper sequencing of events, rate of operation, proper parsing and use of information, orderly and efficient operation, the ultimate goal—autonomy.

- Design methodology via software engineering for complex control systems
- Implementation of complex control systems presents many challenges (e.g., platform differences, communications, etc.)

Scientific Foundations for Biomimicry

"Intelligent control" is the study of how to achieve control automation via the emulation of biological intelligent systems (bio-functionality or behavior).

Intelligent control seeks to develop/exploit biology as a foundation.

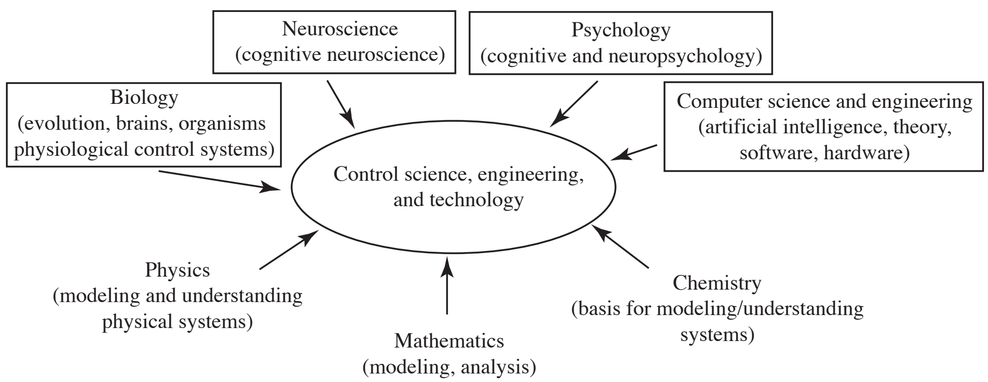

Figure 11: Mathematical, physical, and social sciences that impact control science, engineering, and technology.

Control Systems in Biology

- There is a type of hierarchy in biology (cells, tissue, organs, organisms, populations).

➙ Control systems at the cellular and organ levels:
  - In the (single-cell) *E. coli* (*Escherichia coli*) bacterium there is sensing and locomotion involved in seeking nourishment and avoiding harmful chemicals.
  - Tracking of light and nutrient sources by plants.
  - Homeostasis (e.g., thermoregulation)
  - Immune systems recognize foreign substances and take actions to help the animal survive by controlling the density of antigens.
  - The pancreas is involved in the regulation of blood sugar levels.
  - Motor functions in two-legged animals that provide for

balancing while standing.

  – In the human brain there is the supervision of motor control for voluntary movement, supervision of the attentional system, and others.

- At the organism and population levels:
  – Humans control systems (e.g., in process control).
  – Evolution acts to shape every biological system (notice feedback).

Nervous Systems

Sensory, Motor, and Brain Processes

- The computer is a useful metaphor for the brain. Long term memory is a hard disk, and short term memory is RAM. The computer processes inputs and generates outputs.

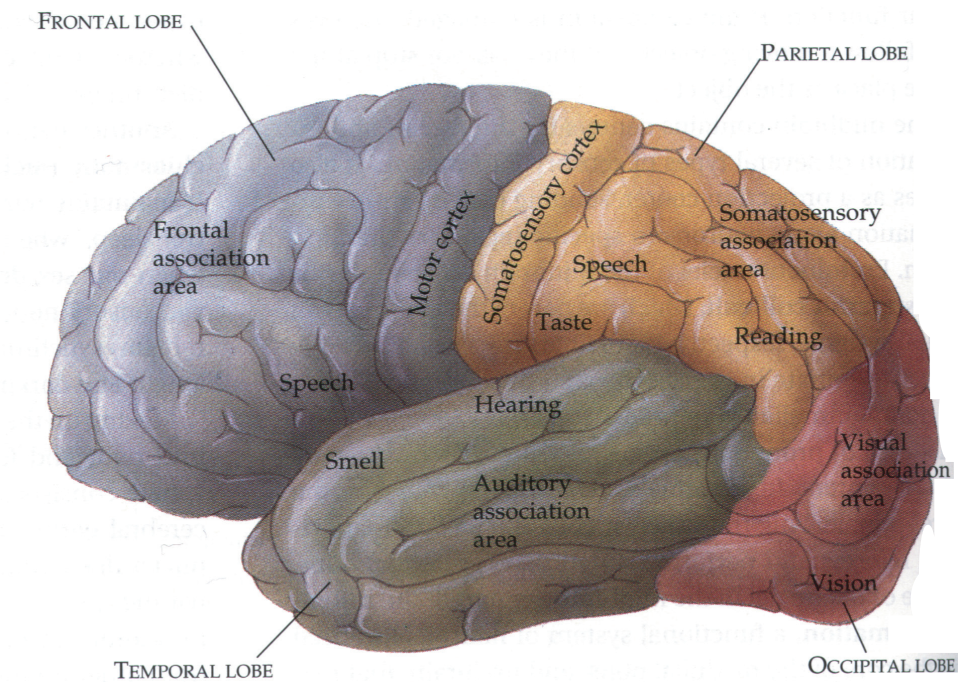

Figure 12: Functional areas of the cortex (figure taken from [1]).

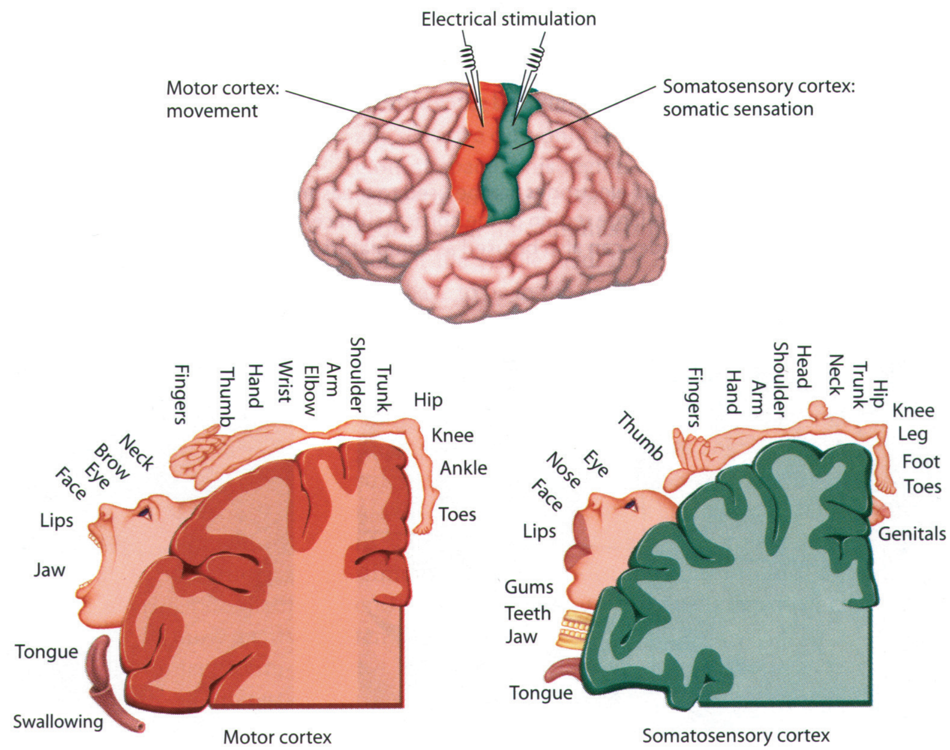

Figure 13: Functional map of the primary motor and somatosensory cortex (figure taken from [3]).

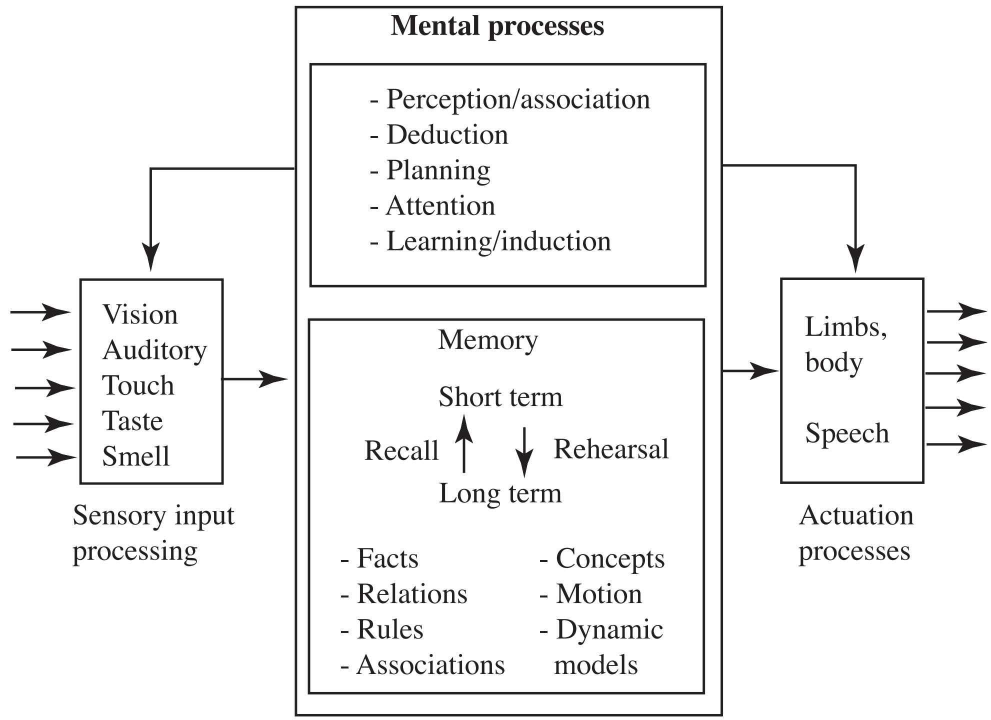

Figure 14: Functional block diagram of *some* brain functions.

The Neurophysiological Level

- A network of neurons provides for information processing, and generating responses for specific patterns of stimuli.

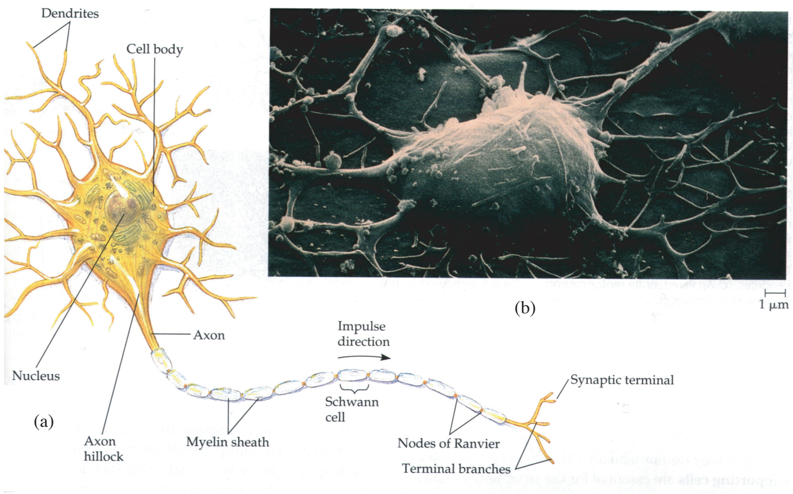

Figure 15: The neuron (nerve cell). In (a) a vertebrate motor neuron is shown, and in (b) a scanning electron micrograph of a neuron is shown (figure taken from [1].

- Motor control is a type of neural hierarchical distributed learning control system.
- Learning takes place via modifications to individual neurons, and changes to interconnectivity of a neural network.

Hierarchical Neural Organization

- Neural network structures are sometimes organized in a hierarchical fashion.

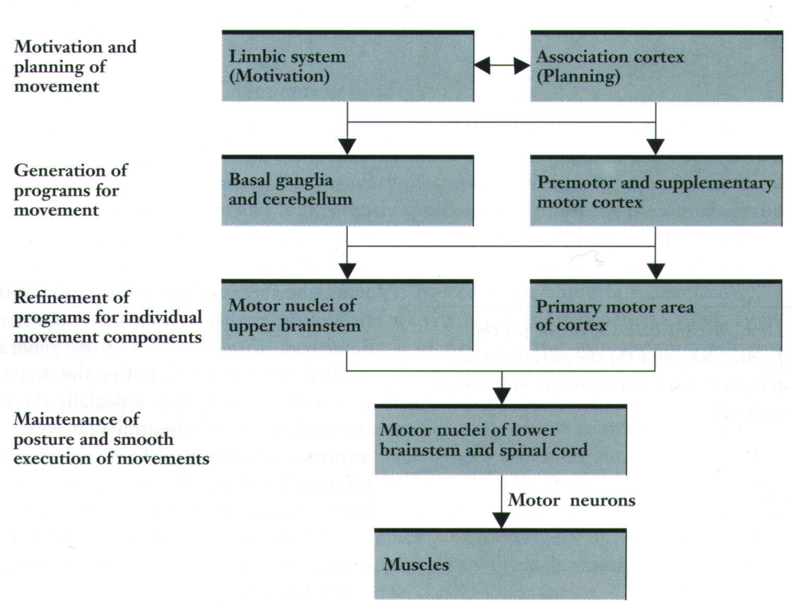

Figure 16: Hierarchy of motor control (figure taken from [4]).

- Brain science is an expanding frontier

Biomimicry for Control and Automation: Cognition

- Neural networks
- Deduction
- Planning
- Attention
- Learning

Organisms

➙ Organisms vs. computers for control—which is better?

- Example: B.F. Skinner's pigeons for missile guidance

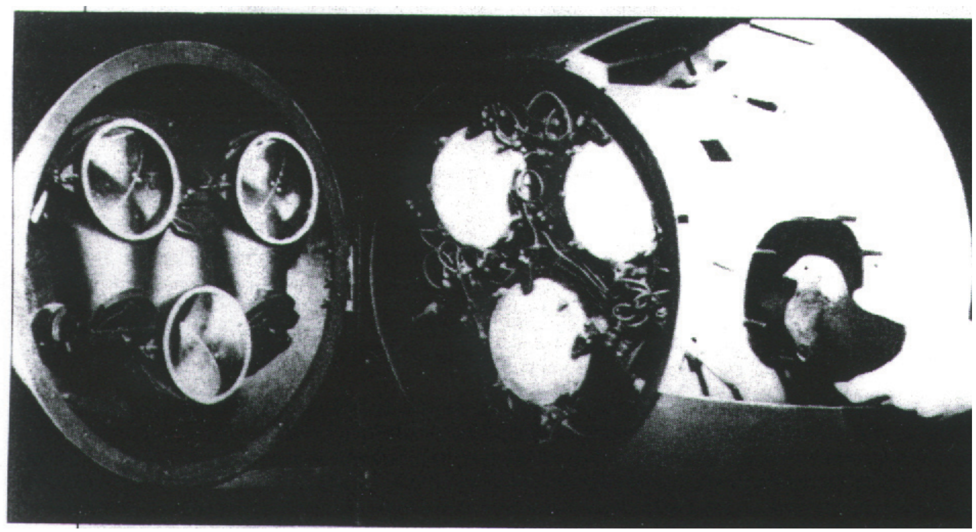

Figure 17: Pigeon being placed in nose cone of the Pelican missile testbed (figure taken from [2]).

➙ Human control expertise and "human-mimicry"

- Perception, deduction, planning, attention, and learning drive the behavior of the human control expertise we seek to emulate.
- Key features to mimic:
  - Rules
  - Deduction over more complex representations
  - Planning using mental models
  - Attention to salient behaviors of the process
  - Learning how to control

Human Operator Control Expertise: Quantity and Quality

➙ Do humans always have the needed control expertise?

- Computers may be able to perform better than humans in some cases, but in others humans may do better.

➙ Do you want to emulate what a human would do?

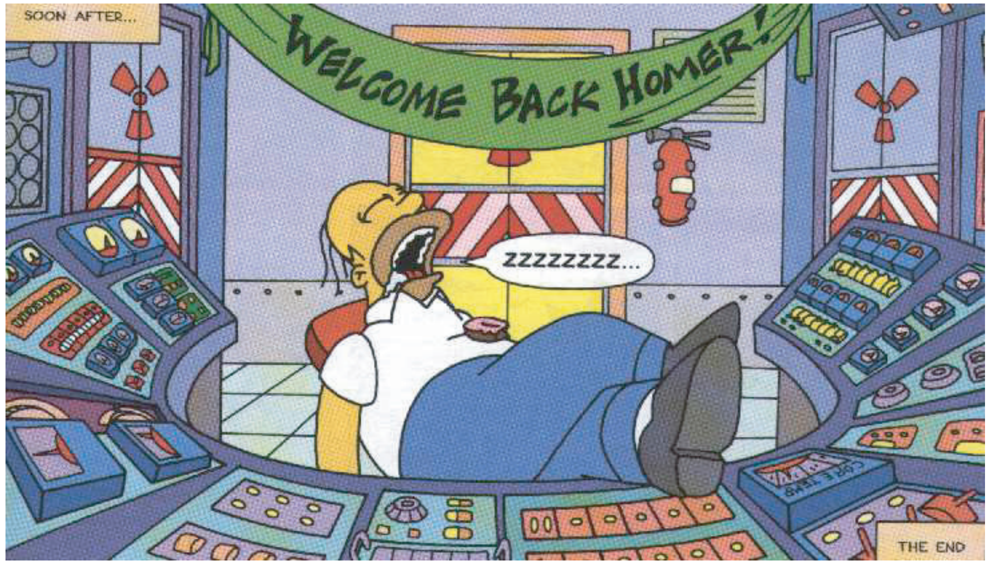

Figure 18: Homer Simpson on the job at a nuclear power plant (figure taken from [5]).

➙ The goal is not emulation of substandard human behavior; it is to design the best control system possible.

Groups of Organisms

- Social foraging and emergent swarm behavior
  - Animals search for and obtain nutrients in a way that maximizes
  $$\frac{E}{T}$$
  where E is energy obtained, and T is time spent foraging.
  - Foraging is an optimization process that has been fine-tuned via evolution.
  - Social foraging: birds, bees, ants, etc.
  - Simple organisms in colonies that obey simple rules can sometimes achieve a type of collective intelligent behavior.
- Example: Bacterial chemotaxis
  - Bacterial foraging involves a type of nutrient "hill-climbing" and hence optimization process.

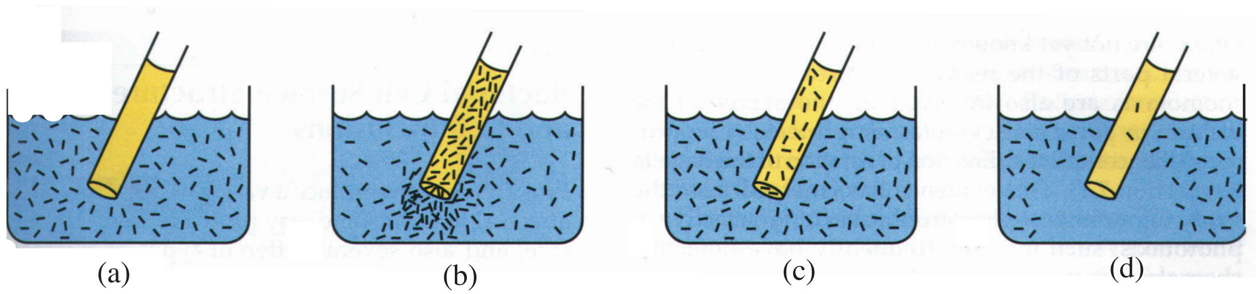

Figure 19: Experiment showing how *E. coli* swarm towards nutrients, and away from noxious substances (figure taken from [8]).

Emulation of Coordinated Behavior of Humans

➙ Groups of humans may be able to coordinate their actions to manage and control some enterprise; it can be useful to mimic their collective behavior for automation.

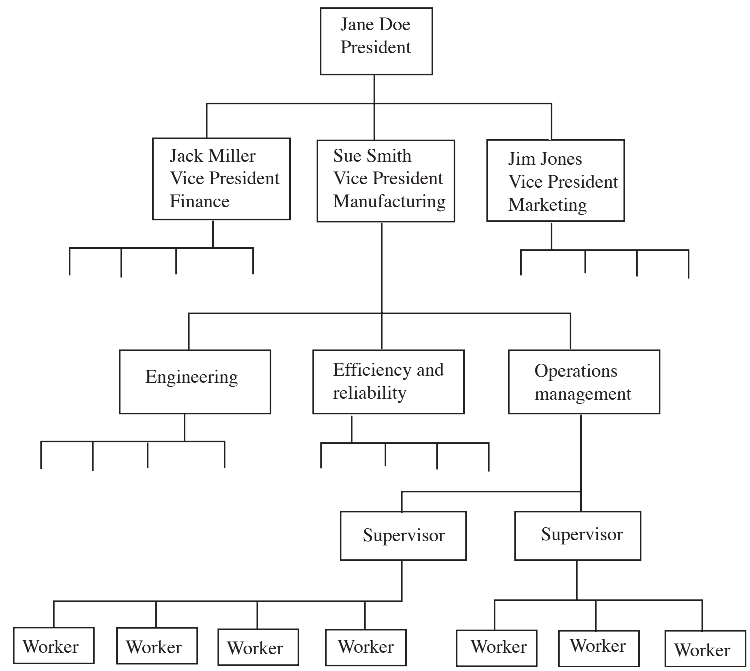

Figure 20: Hypothetical organizational chart for business management.

Biomimicry: Hierarchies and Social Foraging

- Hierarchical biological/cognitive structures and organizations
- Intelligent social foraging

Evolution

- D. Dennett: “biology is engineering”

The Evolutionary Process

- Darwin:
  1. Species have great potential fertility, population sizes are largely fixed but do change with climate changes, and resources are limited.
  2. Life is a struggle and only a relatively few offspring survive.
  3. Individuals in a population vary extensively and that the variation is heritable.
  4. Individuals that possess characteristics that allow them to survive to reproduce leave more offspring than less fit individuals.
  5. This will then lead to a gradual change in a population, and that the favorable characteristics will tend to accumulate in

the population over many generations.

- Example: Polar bear

Figure 21: Polar bear (figure taken from [10]).

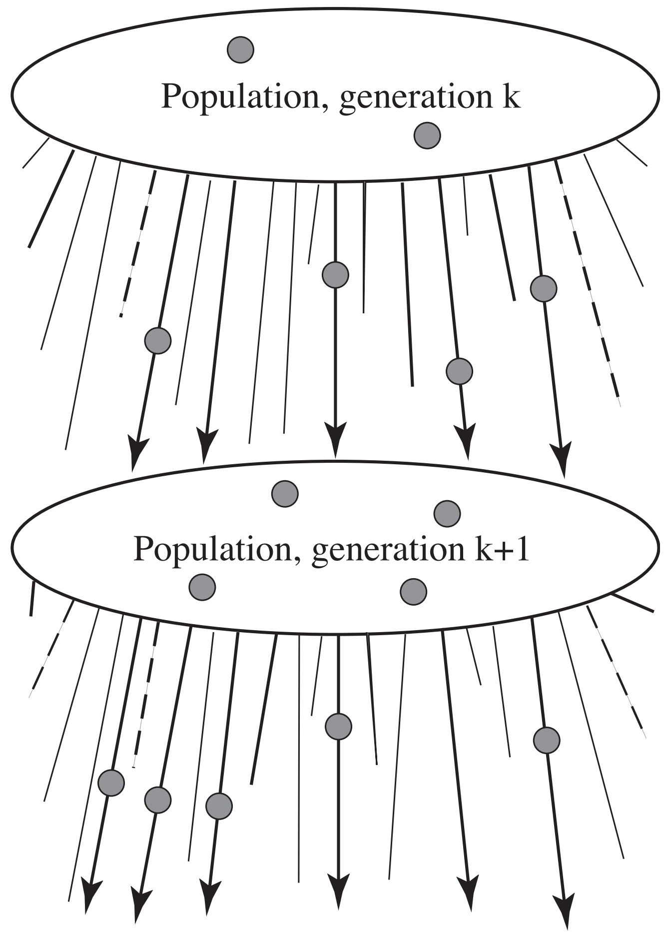

Figure 22: Graphical depiction of the biological evolutionary process.

- Evolution has shaped biology at all levels, from cells to organisms, behavior, and intelligence.
- Example: Selective breeding for behavior and intelligence
- "Selective breeding," is a practice used for thousands of years by plant and animal breeders

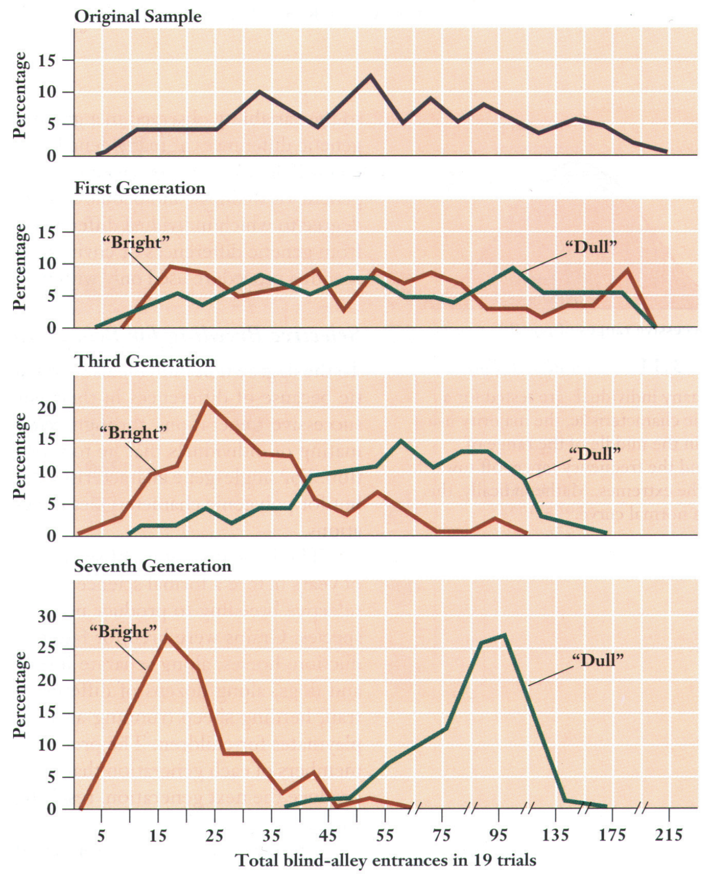

Figure 23: Results of selective breeding in rats for learning capability (figure taken from [4]).

Biomimicry for Control and Automation: Evolution

- “Darwinian” design is stochastic optimization for design
- Evolution and learning—synergies in adaptation.
- Evolution for on-line adaptation
- Evolution of hierarchies and foragers
- Evolution of control technology—a global perspective

A Control Engineering Viewpoint

- Why all the biology? Do I need it?
  - Cohesive framework to think about the development of control and automation systems.
  - Another viewpoint on the dynamics, functionality, design, and operation of high technology control systems for automation.
  - Concepts from biology can hence help to teach the engineering methods.
  - May provide additional ideas (the most successful robust control system in existence is a biological system).
  - Can help us explain what we do to others.
- The focus is engineering, not the foundational sciences (hence, different from an AI focus)

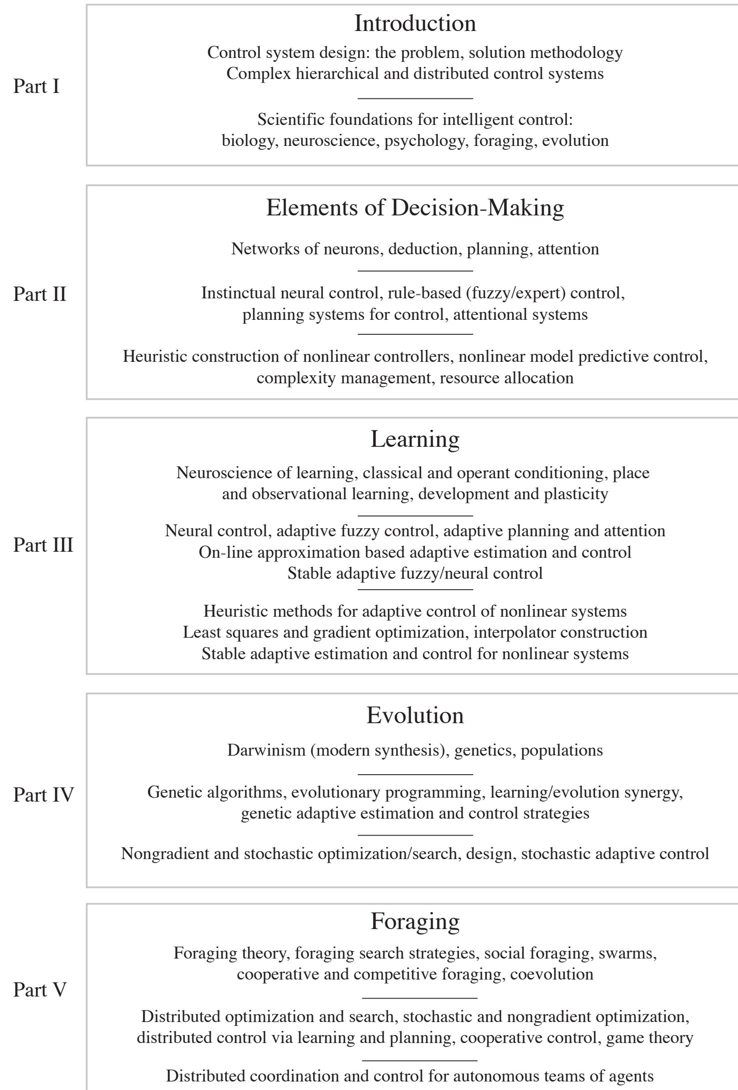

Figure 24: Book overview.

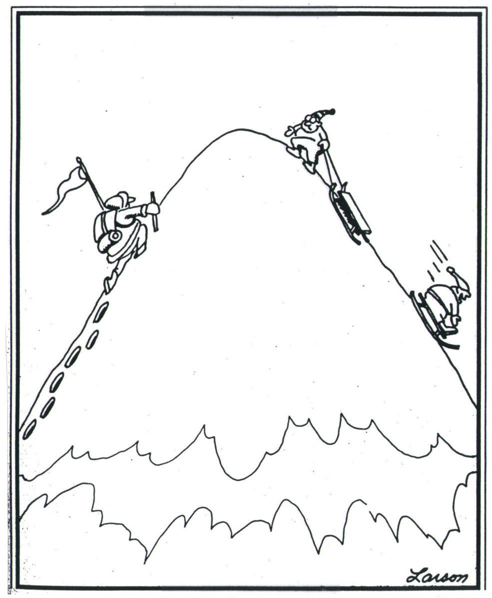

Figure 25: Cartoon to illustrate the importance of choosing the right research problem, one that is not considered child's play by another group (figure taken from [6]).

Central Themes: Optimization, Adaptation, and Decision-Making

➙ Central themes wind throughout...

- Automation via control.
- Decision-making for control.
- Optimization is needed to achieve learning (e.g., we try to find the best information).
- Optimization and adaptation are key features of many decision-making strategies.
- Foraging is an optimization process. Optimization, learning, and decision-making are all used by an intelligent forager.
- Evolution is a design strategy, an optimization process, and an adaptation method. It can be used for design/adaptation.

➙ Intelligent vs. conventional control? No real conflict!

➙ The goal is not "intelligence," it is autonomy

ELEMENTS OF DECISION-MAKING

Focus:

- Instinctual neural networks for control
- Rule-based control (human-mimicry)
- Planning (guidance, model predictive control)
- Attention (dynamic focusing on multiple predators/prey)

Neural Network Substrates for Control Instincts

Neurons and Neural Networks

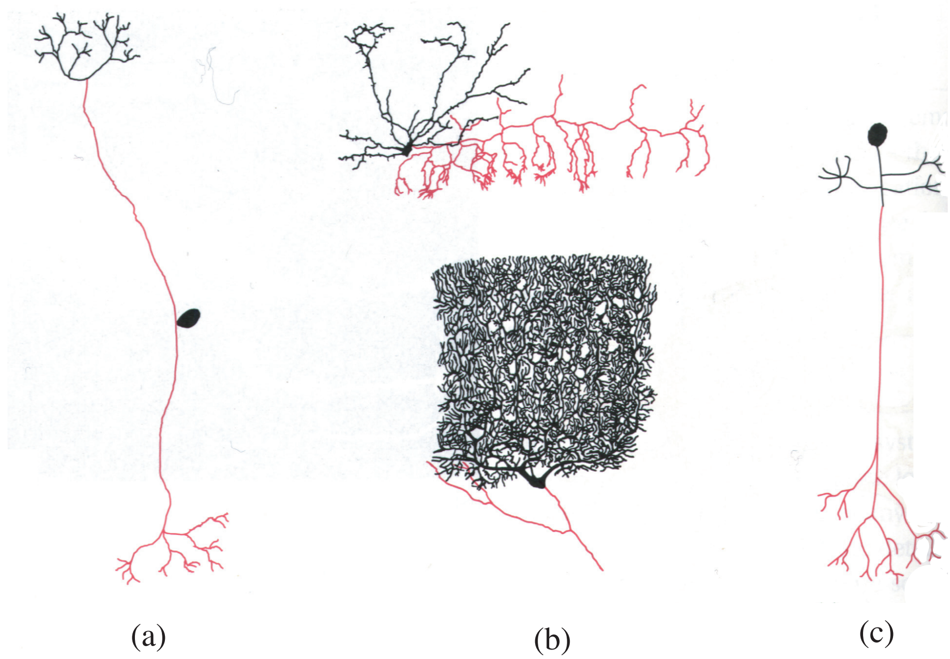

Figure 26: Different forms of neurons (figure taken from [1]).

- Processing characteristics of individual neurons and network interconnections and hence topology of the network change via learning.

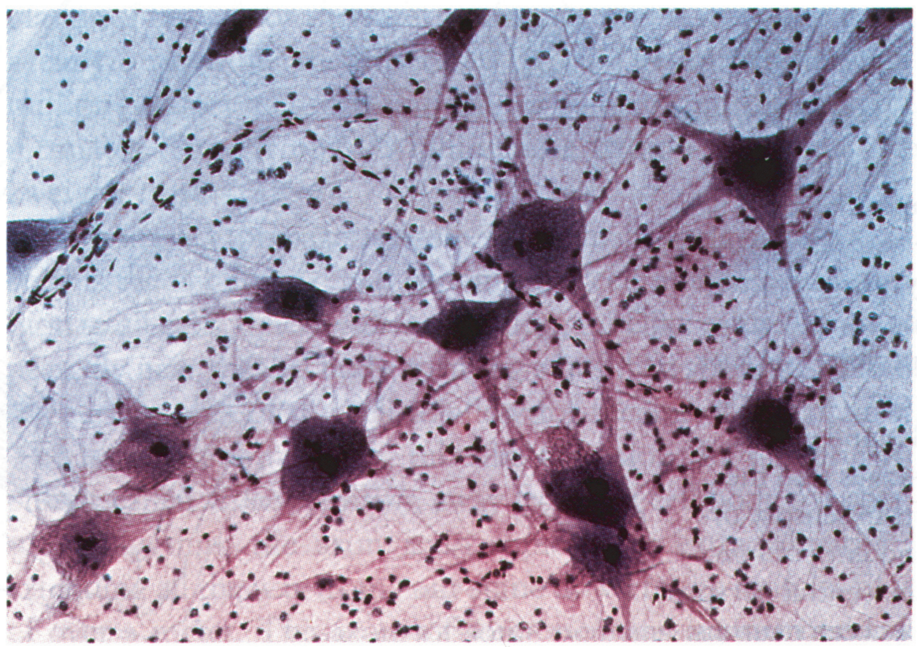

Figure 27: Network of motor neurons in the spinal cord, photograph taken through a microscope (figure taken from [4]).

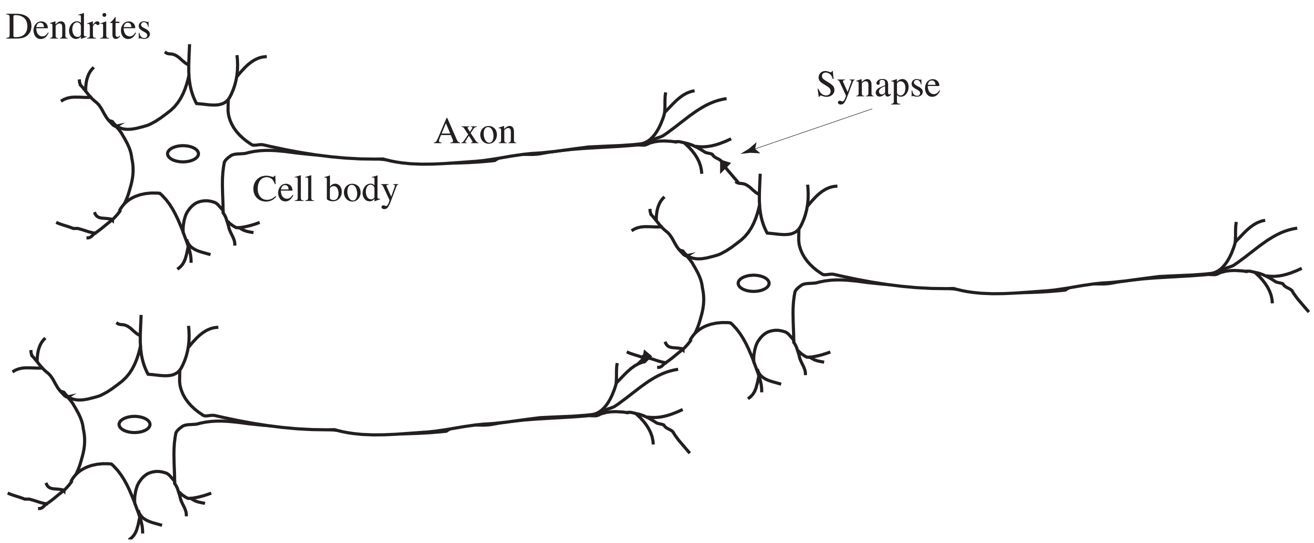

Figure 28: Three connected neurons, a simple biological neural network.

- We study "hard-wired" neural networks that perform "instinctual" control functions

Examples: Instinctual Neural Control Functions in Simple Organisms

- "Command systems" of neurons are used in biological systems for a variety of tasks, such as control of motion, locomotion, digestion, etc.
- Neural network implements a "central pattern generator" that produces a pattern of signals that results in a rhythmic contraction and relaxation of muscles.
- Neurons that Control Swimming in a *Clione*:
- Even very simple networks of neurons can implement controls for actions that are critical for survival.
- There are two neurons for moving each wing, one for "upswing" and the other for "downswing."
- Each neuron has an inhibitory effect on the other so that when one is active, the other is not (i.e., it is inhibited by the other).

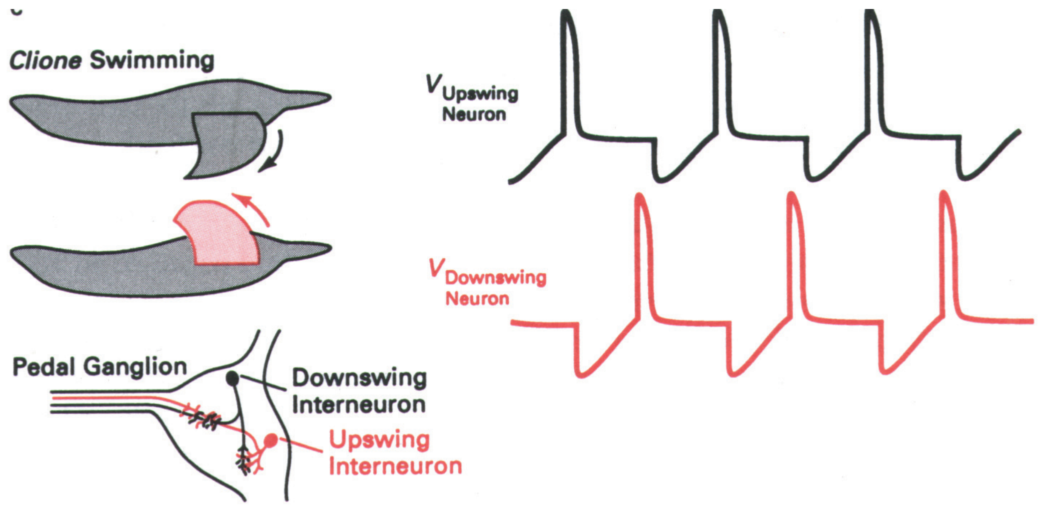

Figure 29: Command system of neurons (neural controller) for swimming in a *Clione* (figure taken from [7]).

- Neuron Stimulus-Response Actions to Achieve Control in a Swimming Leech:
- Rhythmic bursts from a central pattern generator produce a "wave" of contraction that travels from the front to the rear of the leech - swimming.

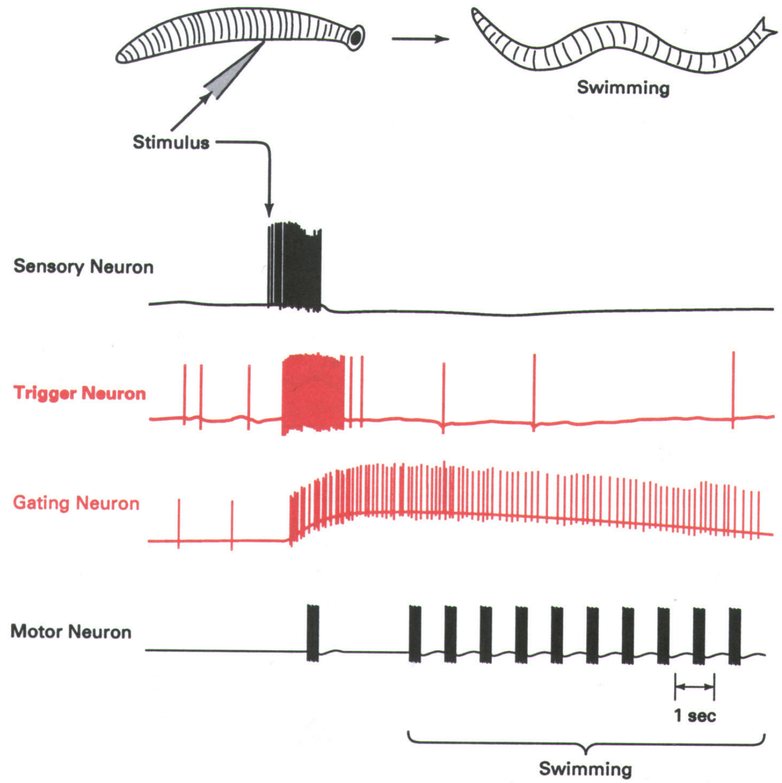

Figure 30: Neuron signaling connecting stimulus to swimming response in a medicinal leech *Hirudo medicinalis* (figure taken from [7]).

Multilayer Perceptrons

- A feed-forward neural network. "Firing rate model".
- It is composed of an interconnection of basic neuron processing units (models of "tuning curves").

The Neuron

- Suppose that we use x_i, $i = 1, 2, \ldots, n$, to denote the neuron's inputs
- Suppose that it has a single output y.
- Figure 31 shows the neuron.

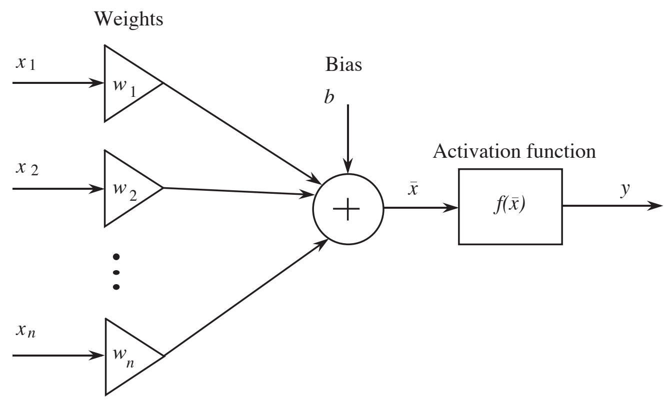

Figure 31: Single neuron model.

- We have

$$\bar{x} = \left(\sum_{i=1}^{n} w_i x_i \right) + b$$

where w_i are the interconnection "weights" and b is the "bias" (these parameters model the interconnections between the cell bodies in the neurons of a biological neural network such as the one shown in Figure 28).

- The signal $\bar{x}$ represents a signal in the biological neuron, and the processing that the neuron performs on this signal is represented with an "activation function" f where

$$y = f(\bar{x}) = f\left(\left(\sum_{i=1}^{n} w_i x_i\right) + b\right) \tag{1}$$

- Basically, the neuron model represents the biological neuron that "fires" (turns on and passes an electrical signal down the axon so that it can go to other neurons as shown in Figure 28) when its inputs are significantly excited (i.e., $\bar{x}$ is big enough).
- There are many ways to define the activation function:
  - Threshold function: We have

$$f(\bar{x}) = \begin{cases} 1 & \text{if } \bar{x} \geq 0 \\ 0 & \text{if } \bar{x} < 0 \end{cases}$$

so if $\bar{x}$ is above zero the neuron turns on.

– Linear function: We have

$$f(\bar{x}) = \bar{x}$$

(it is on when $f(\bar{x}) > 0$ and off when $f(\bar{x}) < 0$).

– Logistic function: For this "sigmoid function," we have

$$f(\bar{x}) = \frac{1}{1 + \exp(-\bar{x})} \tag{2}$$

($\bar{x}$ continuously turns on the neuron as shown in Figure 32).

– Hyperbolic tangent function: Another sigmoid is the hyperbolic tangent function (see Figure 32)

$$f(\bar{x}) = \tanh(\bar{x}) = \frac{1 - \exp(-2\bar{x})}{1 + \exp(-2\bar{x})}$$

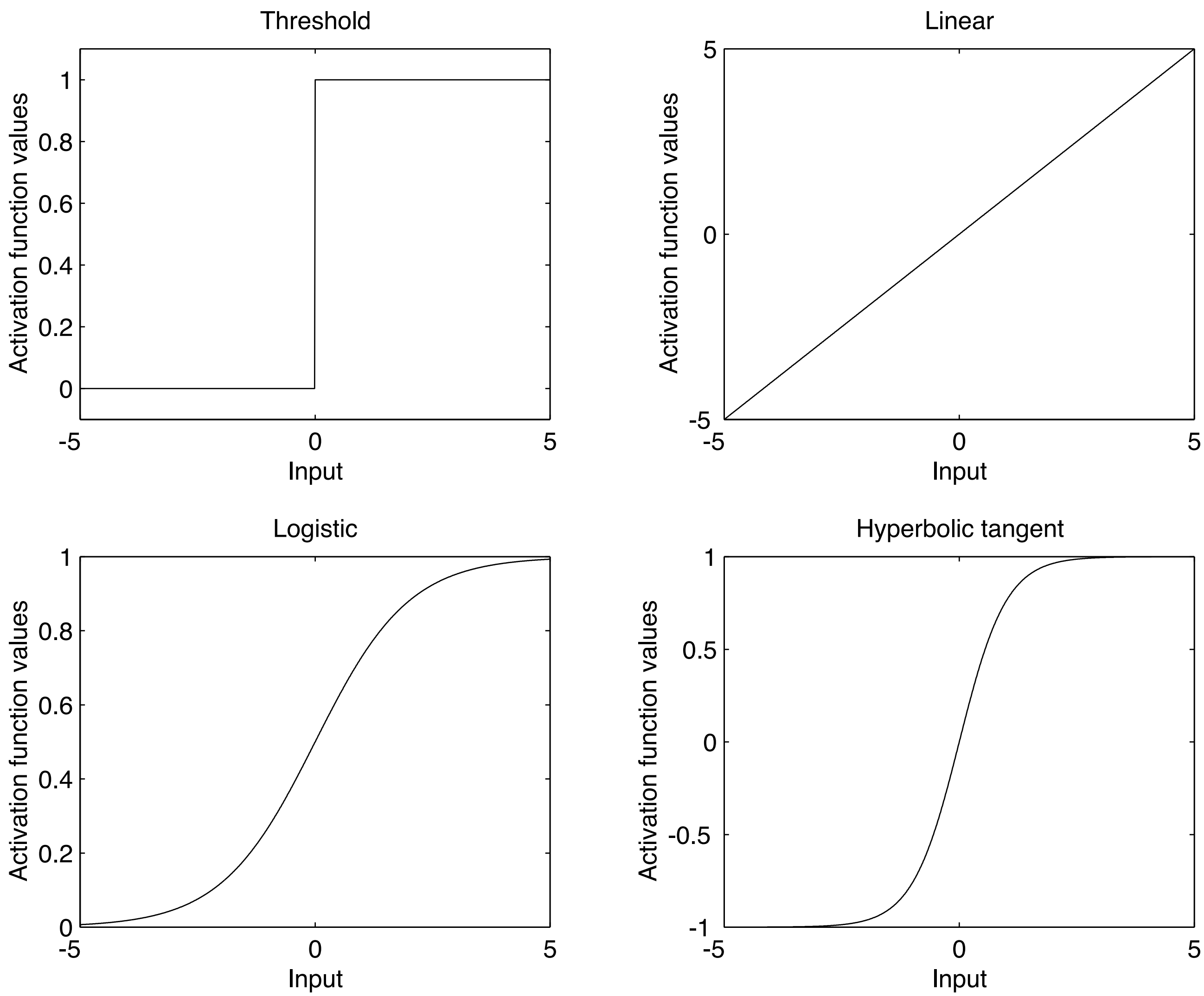

Figure 32: Activation functions for neurons.

Feedforward Network of Neurons

- MLP, $y = F_{mlp}(x, \theta)$ is shown in Figure 33.

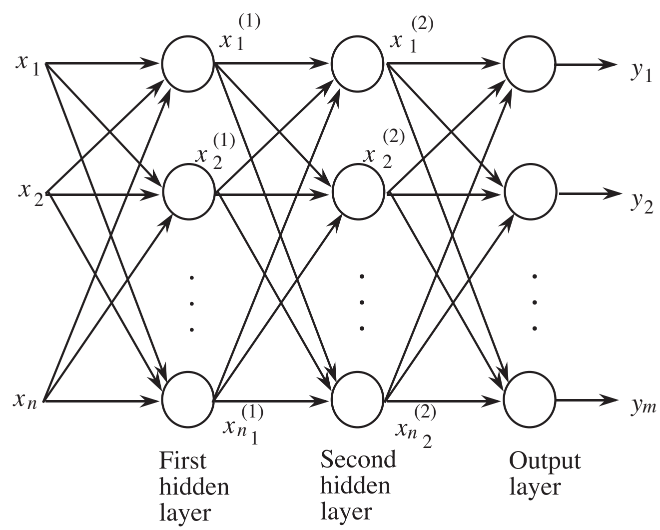

Figure 33: Multilayer perceptron model.

- Circles represent the neurons (weights, bias, and activation function) and lines represent the connections between the inputs and neurons, and layers.

Example: Multilayer Perceptron for Tanker Ship Steering

➙ Tanker ship heading regulation problem

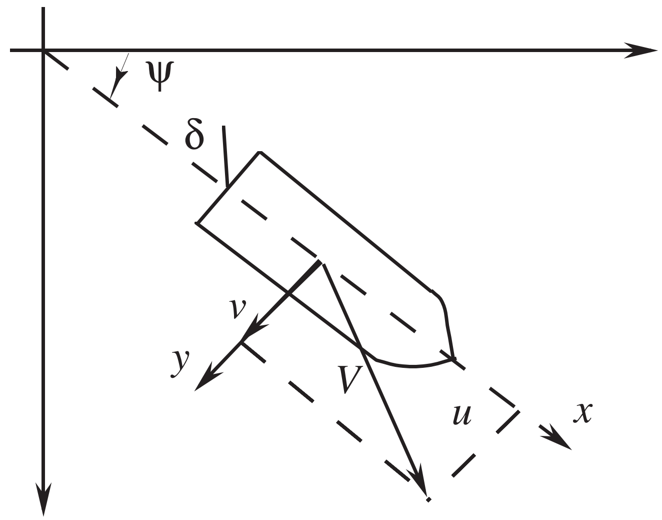

Figure 34: Tanker ship steering problem.

- Tanker ship moves in x direction at a nominal speed u
- ψ denotes the heading angle (in radians)
- δ is the rudder input (in radians).

- ψ_r is the desired ship heading

★ Goal: Want ψ to track ψ_r.

★ Steering performance is affected by:
  - Speed of travel (the rudder becomes less effective at very low speeds)
  - The ship weighs different on different trips (and heavy ships turn slower)
  - Wind hits the side of the tanker and this can affect heading regulation some
  - The sensor for the heading has some noise
  - The rudder can only move between ± 80 degrees.

- Nonlinear model: Can change ship weight from “ballast” to “full” (a lighter ship), can change speed, can add wind effect and sensor noise.
- Simulation issues:
  - Use Runge-Kutta method
  - Simulate a digital controller that updates once every 10 s, but where the integration step size is 1 s (simulation of a digital/analog system).

Construction of a Multilayer Perceptron for Ship Steering

- Controller inputs: ψ_r, ψ

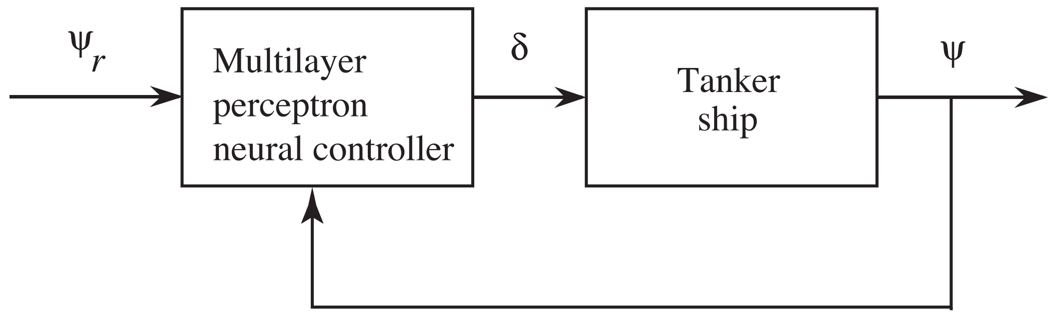

Figure 35: Control system for using a multilayer perceptron for tanker ship steering.

Structure Choice and The First Hidden Layer

➙ MLP is a mapping from ψ_r and ψ to δ, $\delta = F_{mlp}(\psi_r, \psi)$

➙ Controller construction = map synthesis

- Structure choice:

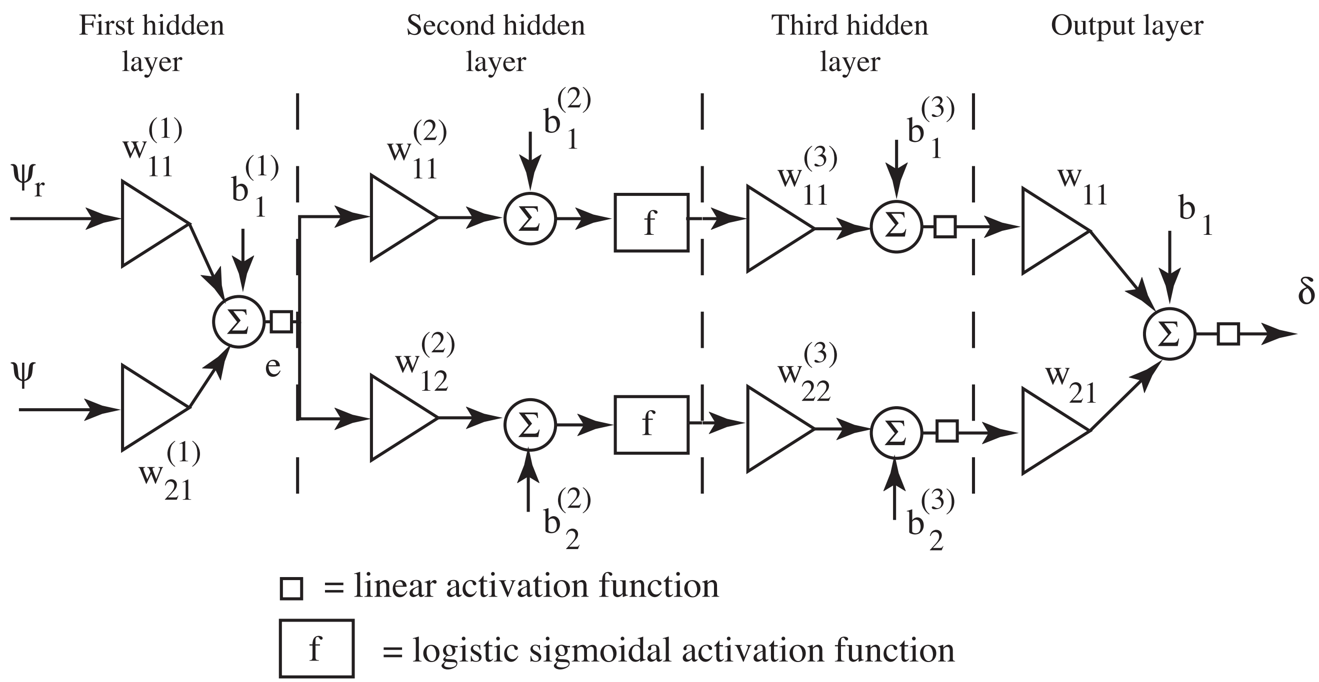

Figure 36: A multilayer perceptron for tanker ship steering.

- Hidden layer: Choose $w_{11}^{(1)} = 1$, $w_{21}^{(1)} = -1$, and $b_1^{(1)} = 0$.

- Output of the first layer is the heading error

$$e = \psi_r - \psi$$

- A neural network can be designed to compare signals for use in "decision-making"

Choosing Weights and Biases: Building Nonlinearities with Smooth Step Functions

- Two "paths" of processing from the signal e
- Remove the path on the bottom.
- Top path used when

$$e = \psi_r - \psi \geq 0$$

- In this case we want to have a *negative* rudder input.
- For larger values of $|e| = |\psi_r - \psi|$ we generally want larger values of δ

- How do we get this?

➙ Neural network construction can be viewed as "building" stimulus-response characteristics from basic neuron building blocks that are deformable via their parameters (here, we build functions from "smooth steps").

- Notice:

$$\delta = w_{11}\left(\frac{w_{11}^{(3)}}{1+\exp(-\bar{x})} + b_1^{(3)}\right) + b_1$$

where

$$\bar{x} = b_1^{(2)} + w_{11}^{(2)} e$$

  - b_1, $b_1^{(3)}$: Shift the mapping up and down.
  - w_{11}, $w_{11}^{(3)}$: Scale the vertical axis.
  - $b_1^{(2)}$: Shifts the smooth step (logistic function) horizontally, with $b_1^{(2)} > 0$ shifting it to the *left*.
  - $w_{11}^{(2)}$: Scale the horizontal axis (you may think of this as a

type of gain for the function, at least locally).

- Using these ideas, choose

$$
\begin{aligned}
b_1 &= b_1^{(3)} = 0 \\
w_{11} &= 1 \\
w_{11}^{(3)} &= -\frac{80\pi}{180} \\
b_1^{(2)} &= -\frac{200\pi}{180} \\
w_{11}^{(2)} &= 10
\end{aligned}
$$

- Continue in this manner for all cases:

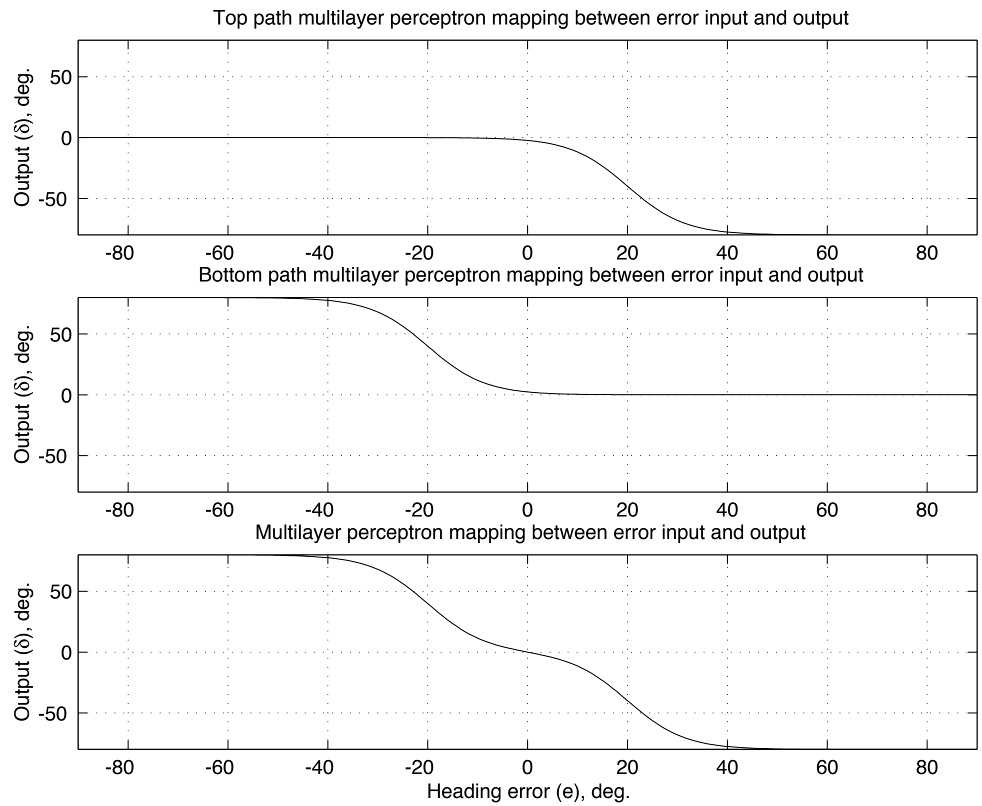

Figure 37: Multilayer perceptron mappings, top plot is for the top path of the perceptron from e to δ, middle plot is for the bottom path of the perceptron from e to δ, bottom plot is for the entire perceptron from e to δ.

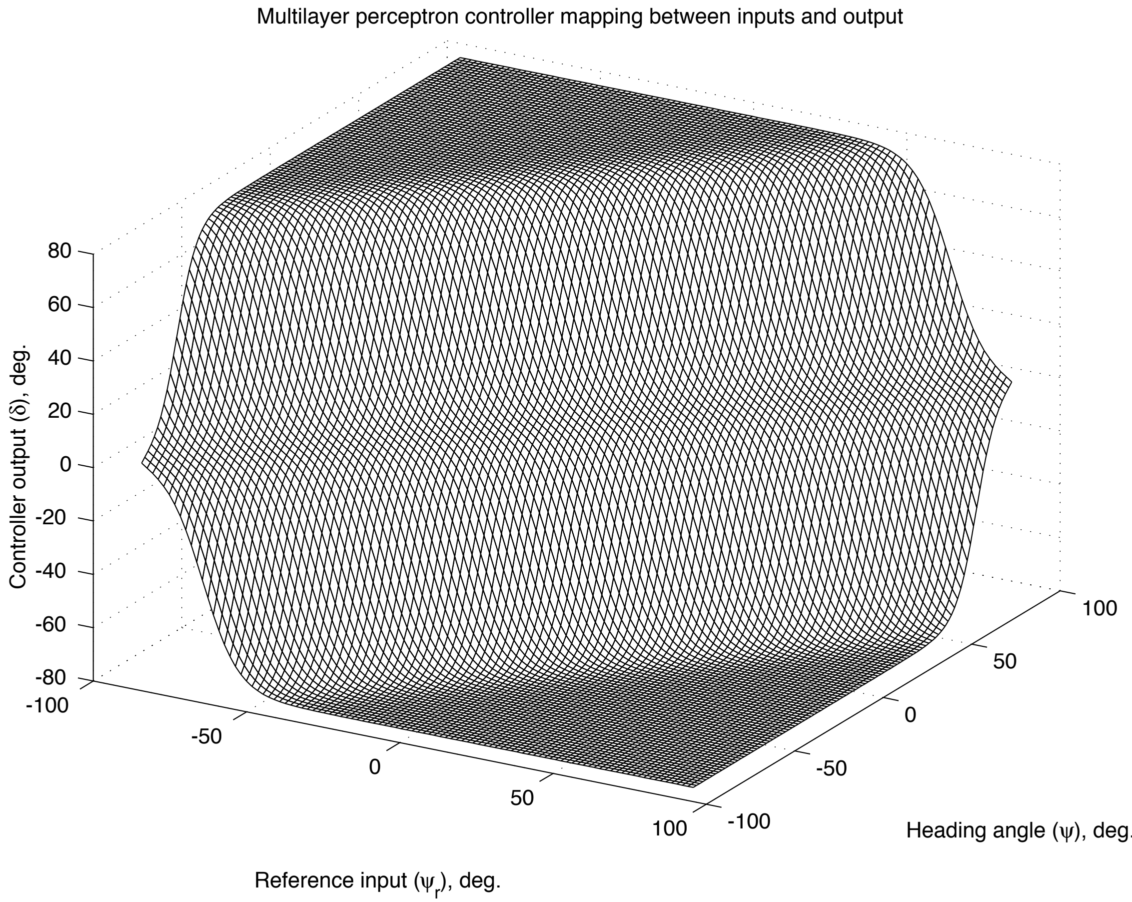

Figure 38: Control surface implemented by the multilayer perceptron for tanker ship steering.

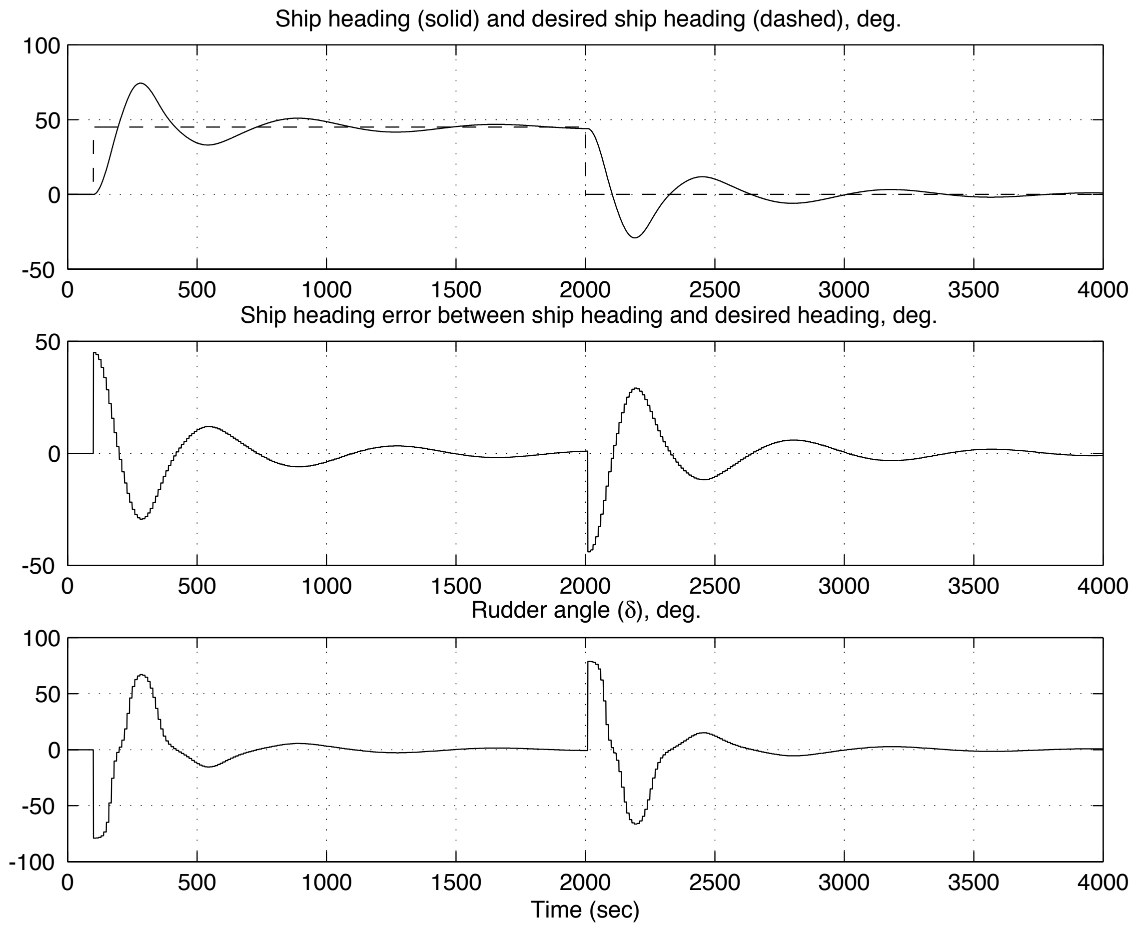

Figure 39: Closed-loop response resulting from using the multilayer perceptron for tanker ship steering.

➙ Tuning to get better performance = change mapping shape.

Effects of Wind on Heading Regulation

- Wind: Hits side of ship, pushes water against rudder.
- We add a disturbance onto the rudder angle input by adding

$$0.5\left(\frac{\pi}{180}\right)\sin\left(2\pi(0.001)t\right)$$

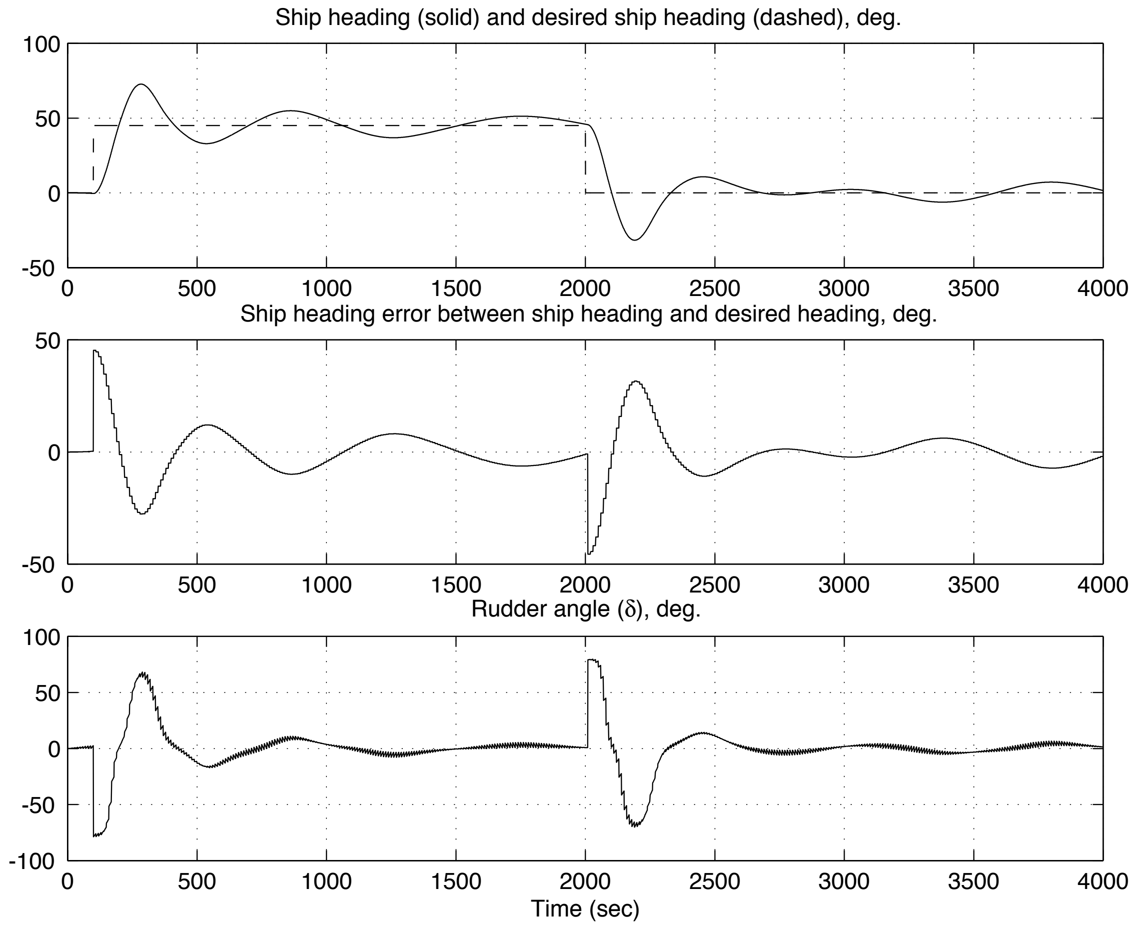

Figure 40: Closed-loop response resulting from using the multilayer perceptron for tanker ship steering, with wind.

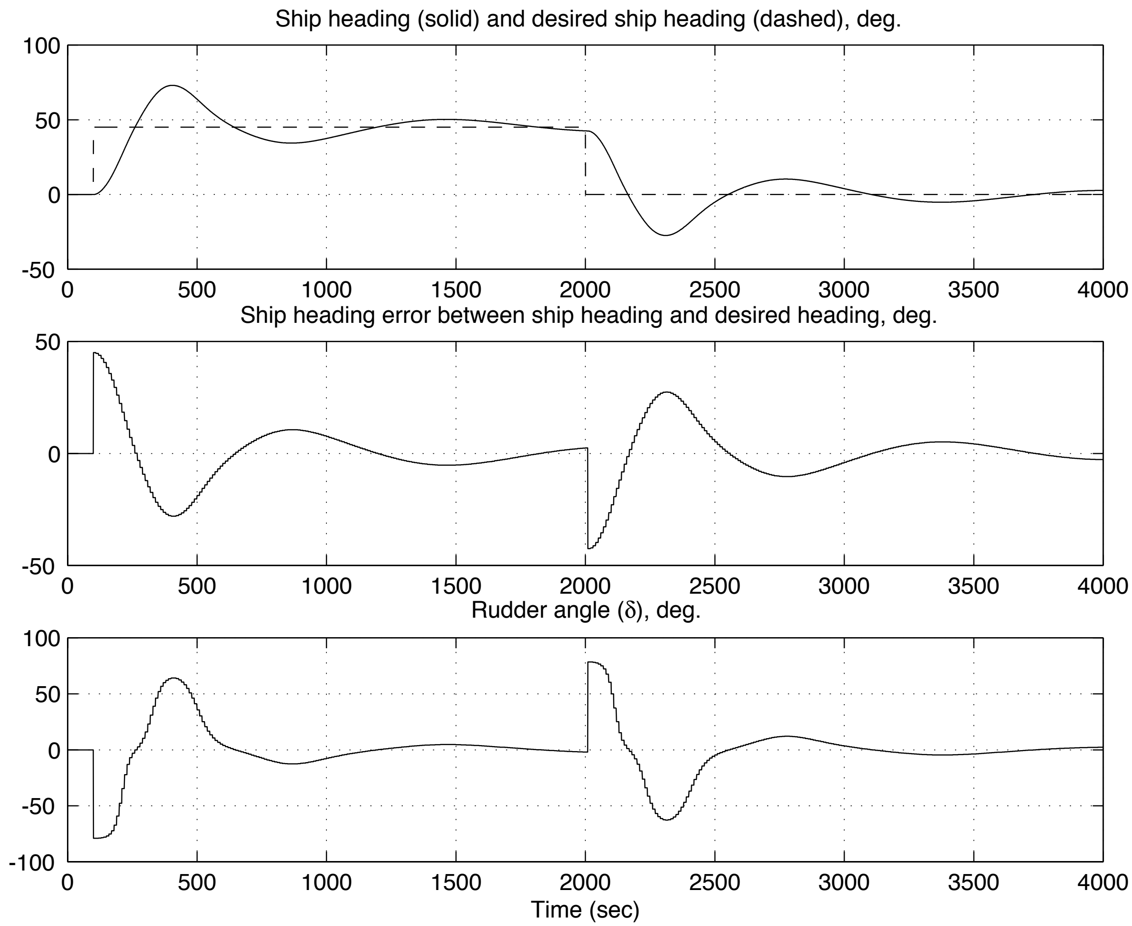

Figure 41: Closed-loop response resulting from using the multilayer perceptron for tanker ship steering with speed of 3 meters/sec.

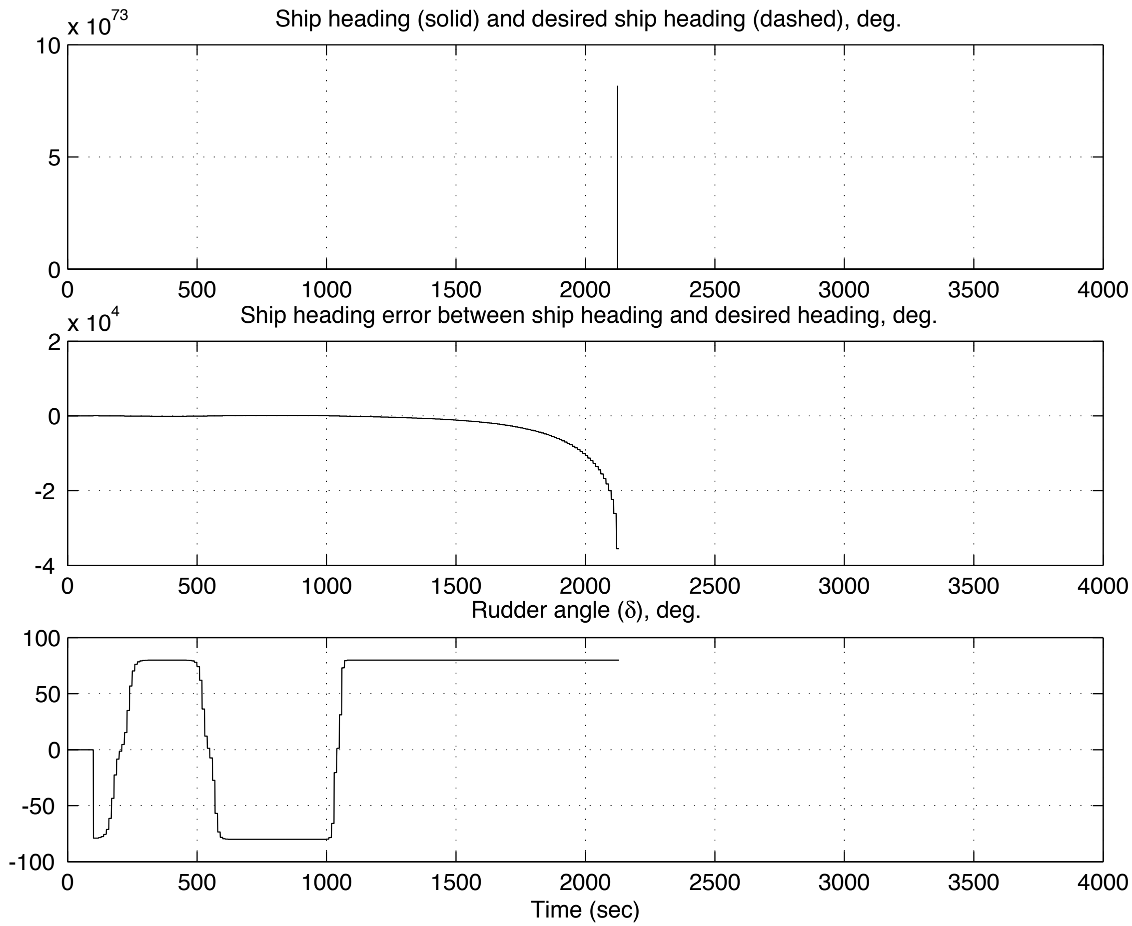

Figure 42: Closed-loop response resulting from using the multilayer perceptron for tanker ship steering, full rather than ballast conditions.

Radial Basis Function Neural Networks

- A locally tuned, overlapping receptive field is found in parts of the cerebral cortex, in the visual cortex, and in other parts of the brain.
- Different "tuning curve" shape, but again connect into network.
- A radial basis function neural network is shown in Figure 43.

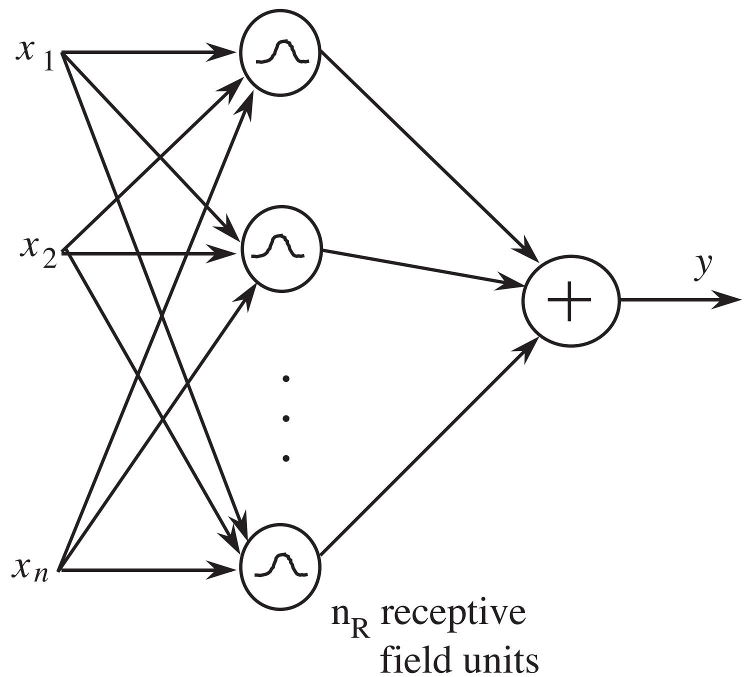

Figure 43: Radial basis function neural network model.

- The inputs are x_i, $i = 1, 2, \ldots, n$, and the output is $y = F_{rbf}(x)$
- Let $x = [x_1, x_2, \ldots, x_n]^\top$.
- The input to the i^{th} receptive field unit is x, and its output is

denoted with $R_i(x)$.

- It has what is called a "strength" which we denote by b_i.
- Assume that there are n_R receptive field units.
- Hence, from Figure 43,

$$y = F_{rbf}(x, \theta) = \sum_{i=1}^{n_R} b_i R_i(x) \tag{3}$$

where θ holds the b_i parameters (and possibly the parameters of the receptive field units).

- There are several possible choices for the "receptive field units" $R_i(x)$:

1. We could choose

$$R_i(x) = \exp\left(-\frac{|x - c_i|^2}{\sigma_i^2}\right)$$

where $c_i = [c_1^i, c_2^i, \ldots, c_n^i]^\top$, σ_i is a scalar, and if z is a vector then $|z| = \sqrt{z^\top z}$.

– For the case where $n = 1$, $c_1 = [c_1^1] = [2]$, and $\sigma_1 = 0.1$, $R_1(x)$ is shown in Figure 44(a).

– As x moves away from c_1^1, $R_1(x)$ decreases, with the rate of decrease dictated by the size of σ_1.

2. We could choose

$$R_i(x) = \frac{1}{1 + \exp\left(-\frac{|x-c_i|^2}{\sigma_i^2}\right)}$$

where c_i and σ_i are defined in choice 1.

– For the case where $n = 1$, $c_1 = [c_1^1] = [2]$, and $\sigma_1 = 0.1$, $R_1(x)$ is shown in Figure 44(b).

– This is similar to the above but "flipped" upside down.

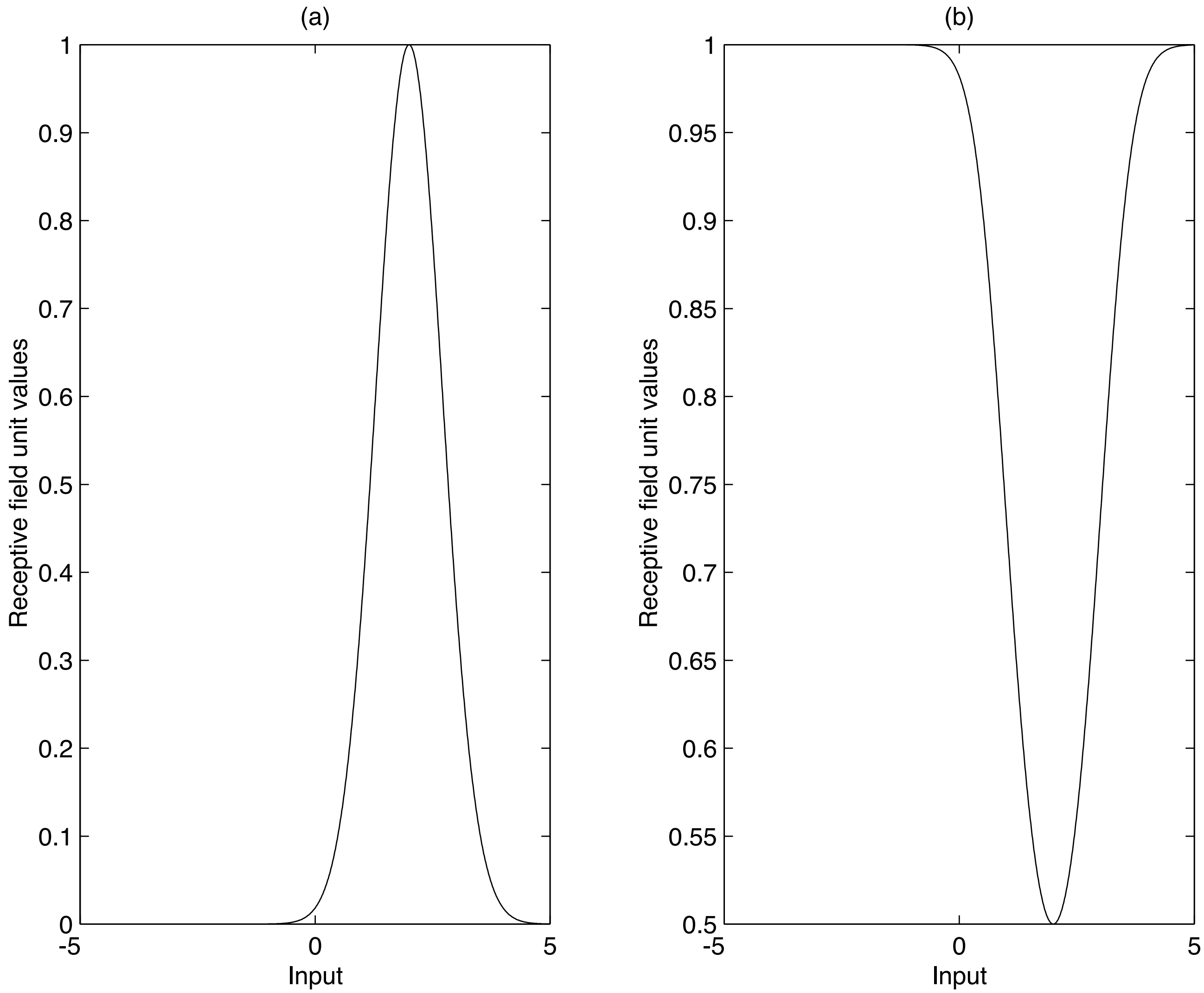

Figure 44: Example receptive field units.

- There are also alternatives to how to compute the output of the radial basis function neural network.
- For instance, you could compute a weighted average

$$y = F_{rbf}(x, \theta) = \frac{\sum_{i=1}^{n_R} b_i R_i(x)}{\sum_{i=1}^{n_R} R_i(x)} \tag{4}$$

- It is also possible to define multilayer radial basis function neural networks.

Example: RBF Neural Network for Ship Steering

Controller Input Choice and Control System Structure

- Inputs:

$$e = \psi_r - \psi$$

and

$$\dot{e} = \dot{\psi}_r - \dot{\psi}$$

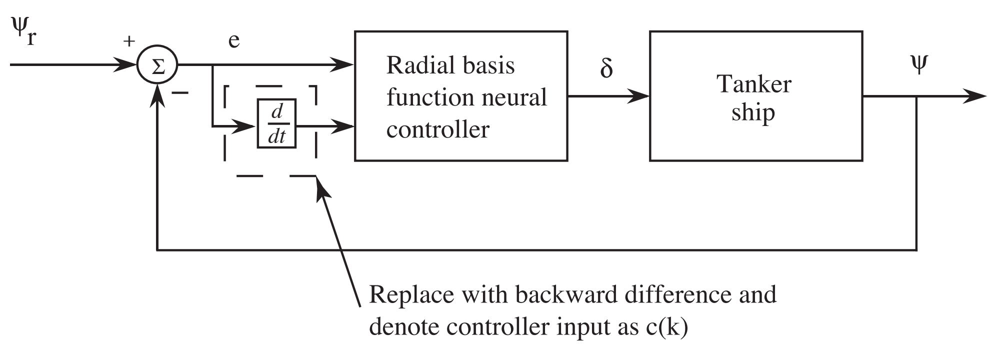

Figure 45: Radial basis function neural network used as a controller for ship heading.

- Need to synthesize the map:

$$\delta(k) = F_{rbf}(e(k), c(k))$$

Design of a Radial Basis Function Neural Network for Steering

- RBF: $n = 2$, $n_R = 121$
- For the $R_i(e(k), c(k))$ create a uniform grid for the c^i centers, $i = 1, 2, \ldots, 121$.
- Assume

$$e(k) \in [-\frac{\pi}{2}, \frac{\pi}{2}]$$

and

$$c(k) \in [-0.01, 0.01]$$

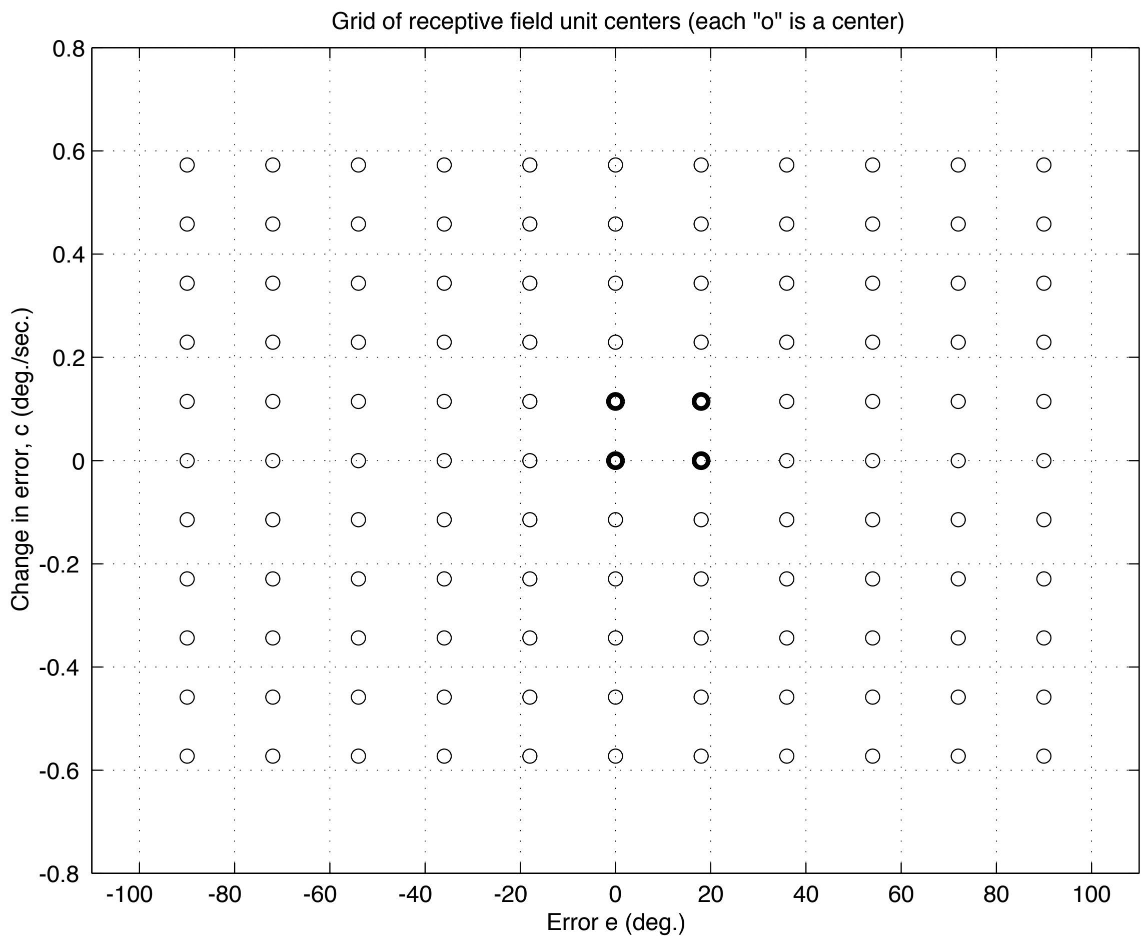

Figure 46: Receptive field unit centers.

- Choose:

$$\sigma_1^i = 0.7\frac{\pi}{\sqrt{n_R}}$$

and

$$\sigma_2^i = 0.7\frac{0.02}{\sqrt{n_R}}$$

- Example: Consider $R_{73}(e, c)$

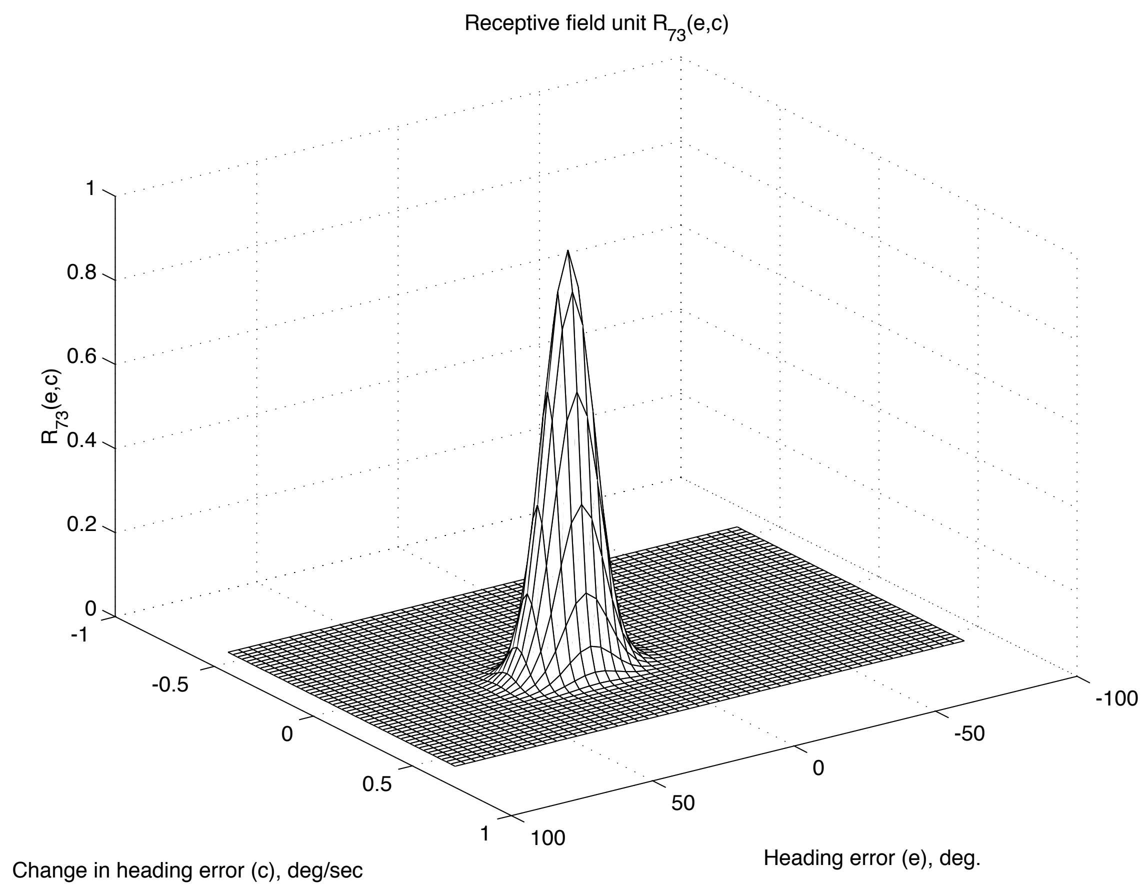

Figure 47: Mapping implemented by receptive field unit $R_{73}(e, c)$.

➙ We can view construction of an RBF NN as building a stimulus-response characteristic from tunable "spatially local" functions.

- "Strengths" just scale heights
- Example:

$$2R_{61}(e,c) + R_{62}(e,c) + 2R_{72}(e,c) + R_{73}(e,c)$$

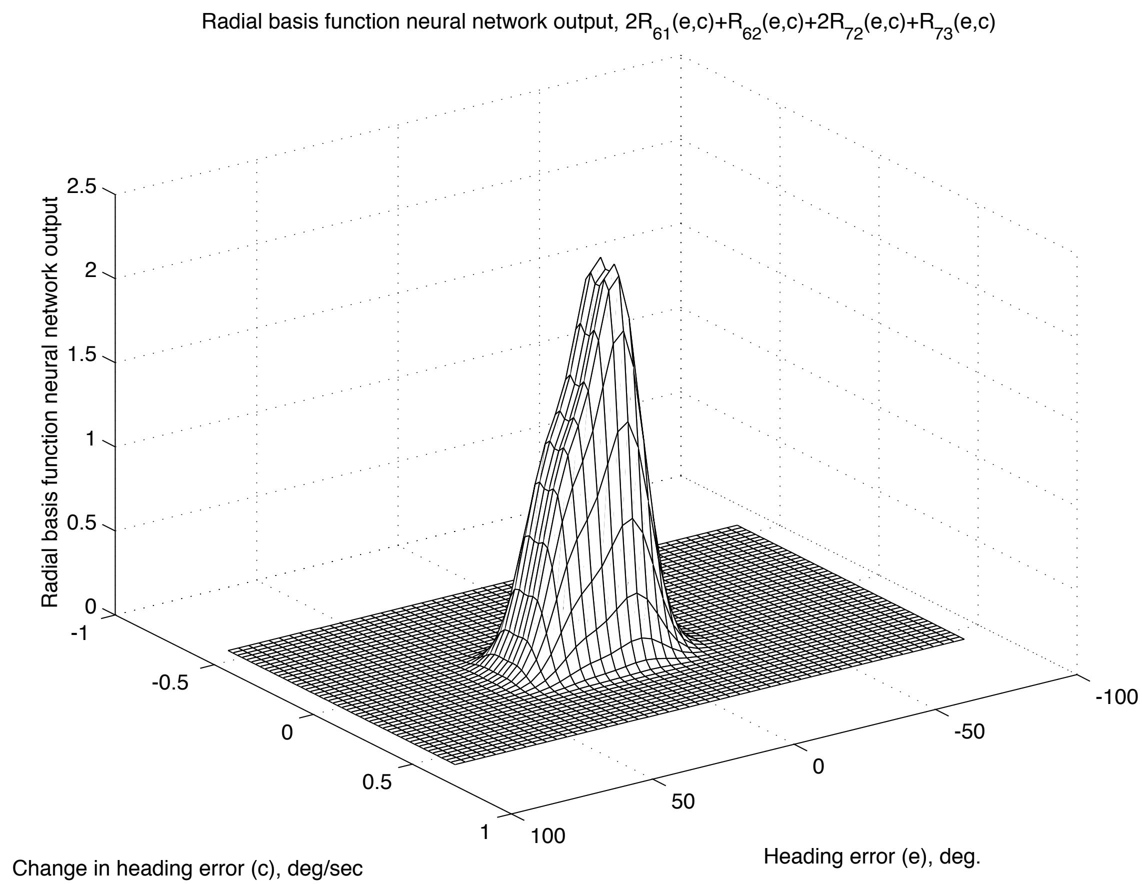

Figure 48: Scaling and addition of several receptive field units (i.e., $2R_{61}(e,c) + R_{62}(e,c) + 2R_{72}(e,c) + R_{73}(e,c)$).

- Need to choose the b_i, $i = 1, 2, \ldots, 121$
- View the parameters as being loaded in a matrix

$$\begin{bmatrix} b_1 & b_{12} & b_{23} & b_{34} & b_{45} & b_{56} & b_{67} & b_{78} & b_{89} & b_{100} & b_{111} \\ b_2 & & & & & \cdots & & & & & b_{112} \\ \vdots & & & & & \cdots & & & & & \vdots \\ b_{11} & b_{22} & b_{33} & b_{44} & b_{55} & b_{66} & b_{77} & b_{88} & b_{99} & b_{110} & b_{121} \end{bmatrix}$$

and then choose this matrix to be

```
Columns 1 through 7

  1.3963    1.3963    1.3963    1.3963    1.3963    1.3963    1.3963
  1.3963    1.3963    1.3963    1.3963    1.3963    1.3963    1.0472
  1.3963    1.3963    1.3963    1.3963    1.3963    1.0472    0.6981
  1.3963    1.3963    1.3963    1.3963    1.0472    0.6981    0.3491
  1.3963    1.3963    1.3963    1.0472    0.6981    0.3491         0
  1.3963    1.3963    1.0472    0.6981    0.3491         0   -0.3491
  1.3963    1.0472    0.6981    0.3491         0   -0.3491   -0.6981
  1.0472    0.6981    0.3491         0   -0.3491   -0.6981   -1.0472
  0.6981    0.3491         0   -0.3491   -0.6981   -1.0472   -1.3963
  0.3491         0   -0.3491   -0.6981   -1.0472   -1.3963   -1.3963
       0   -0.3491   -0.6981   -1.0472   -1.3963   -1.3963   -1.3963
```

```
Columns 8 through 11

    1.0472    0.6981    0.3491         0
    0.6981    0.3491         0   -0.3491
    0.3491         0   -0.3491   -0.6981
         0   -0.3491   -0.6981   -1.0472
   -0.3491   -0.6981   -1.0472   -1.3963
   -0.6981   -1.0472   -1.3963   -1.3963
   -1.0472   -1.3963   -1.3963   -1.3963
   -1.3963   -1.3963   -1.3963   -1.3963
   -1.3963   -1.3963   -1.3963   -1.3963
   -1.3963   -1.3963   -1.3963   -1.3963
   -1.3963   -1.3963   -1.3963   -1.3963
```

➙ Notice the pattern of elements in the matrix.

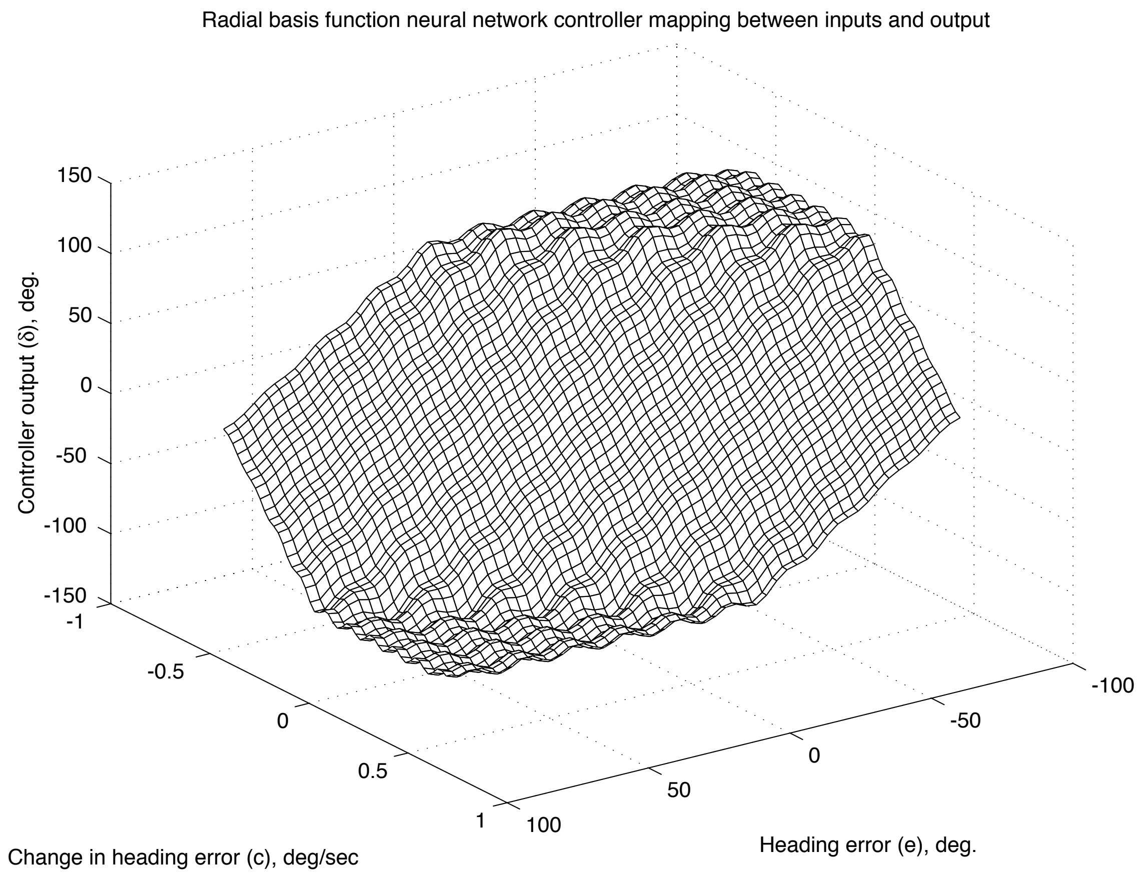

Figure 49: Stimulus-response characteristics of the radial basis function neural network for tanker ship heading regulation.

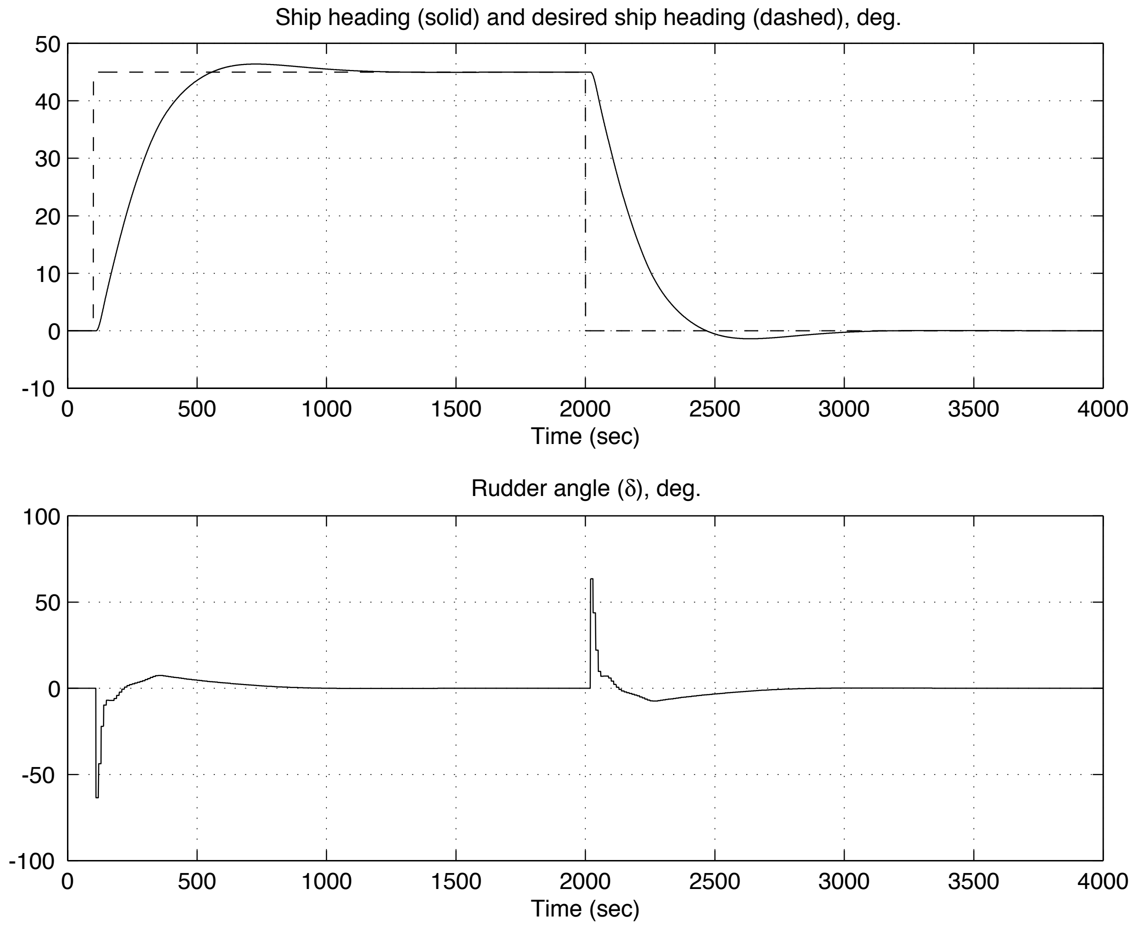

Figure 50: Closed-loop response resulting from using the radial basis function neural network for tanker ship steering.

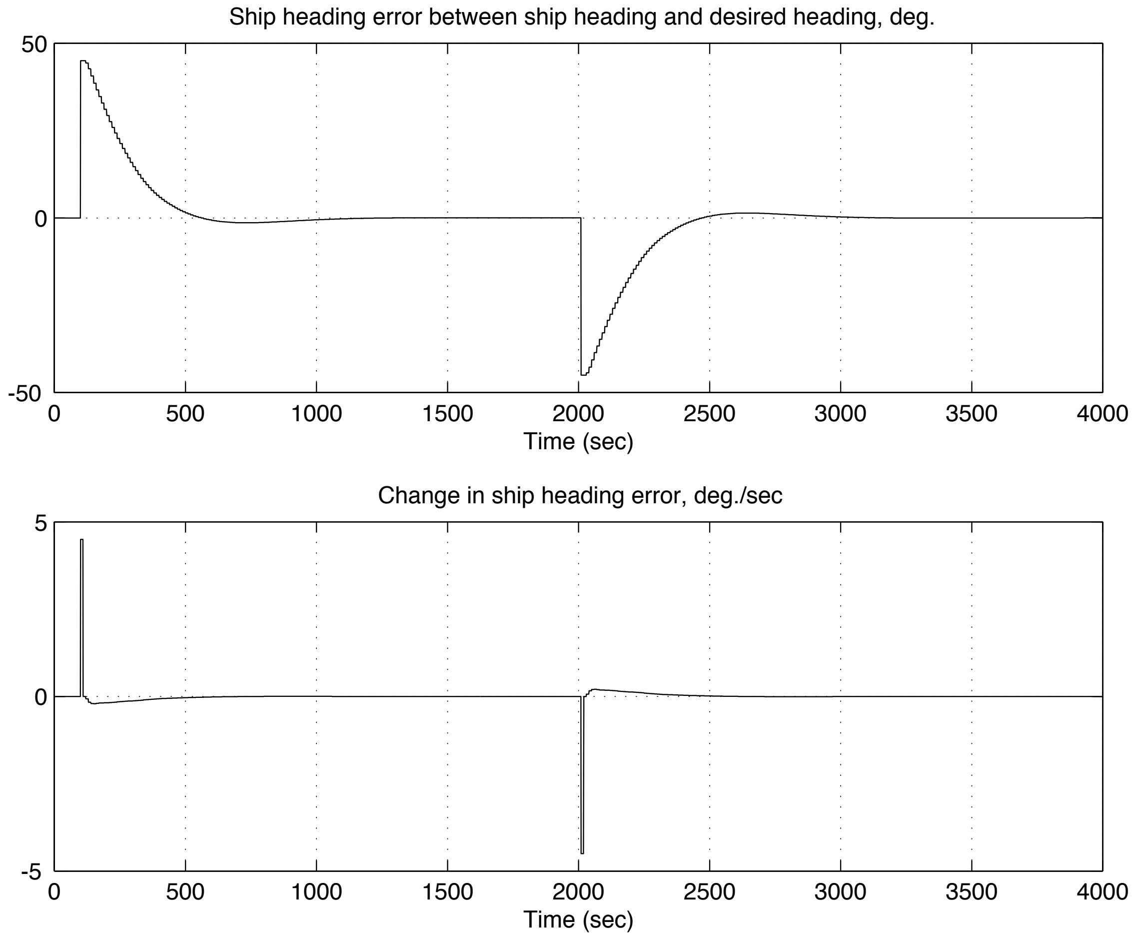

Figure 51: Closed-loop response resulting from using the radial basis function neural network for tanker ship steering.

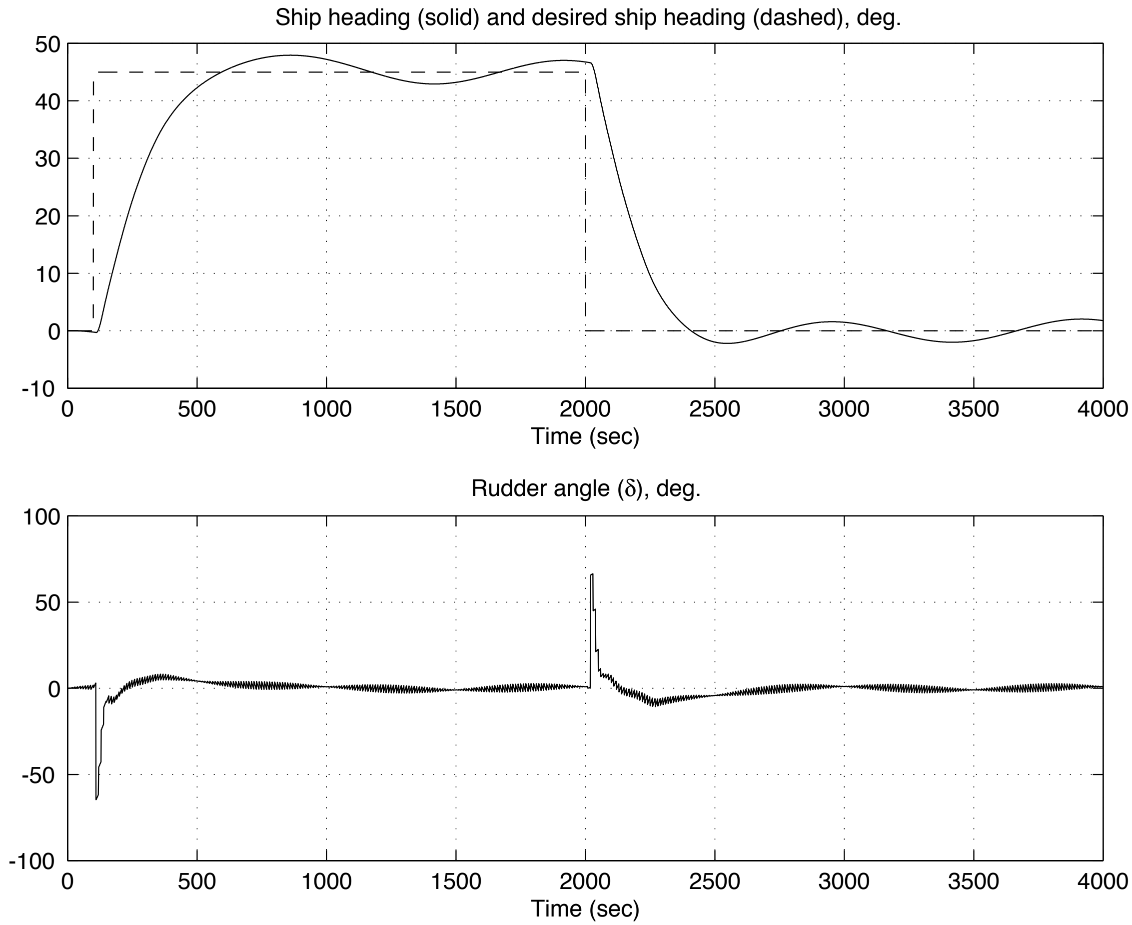

Figure 52: Closed-loop response resulting from using the radial basis function neural network for tanker ship steering, with wind.

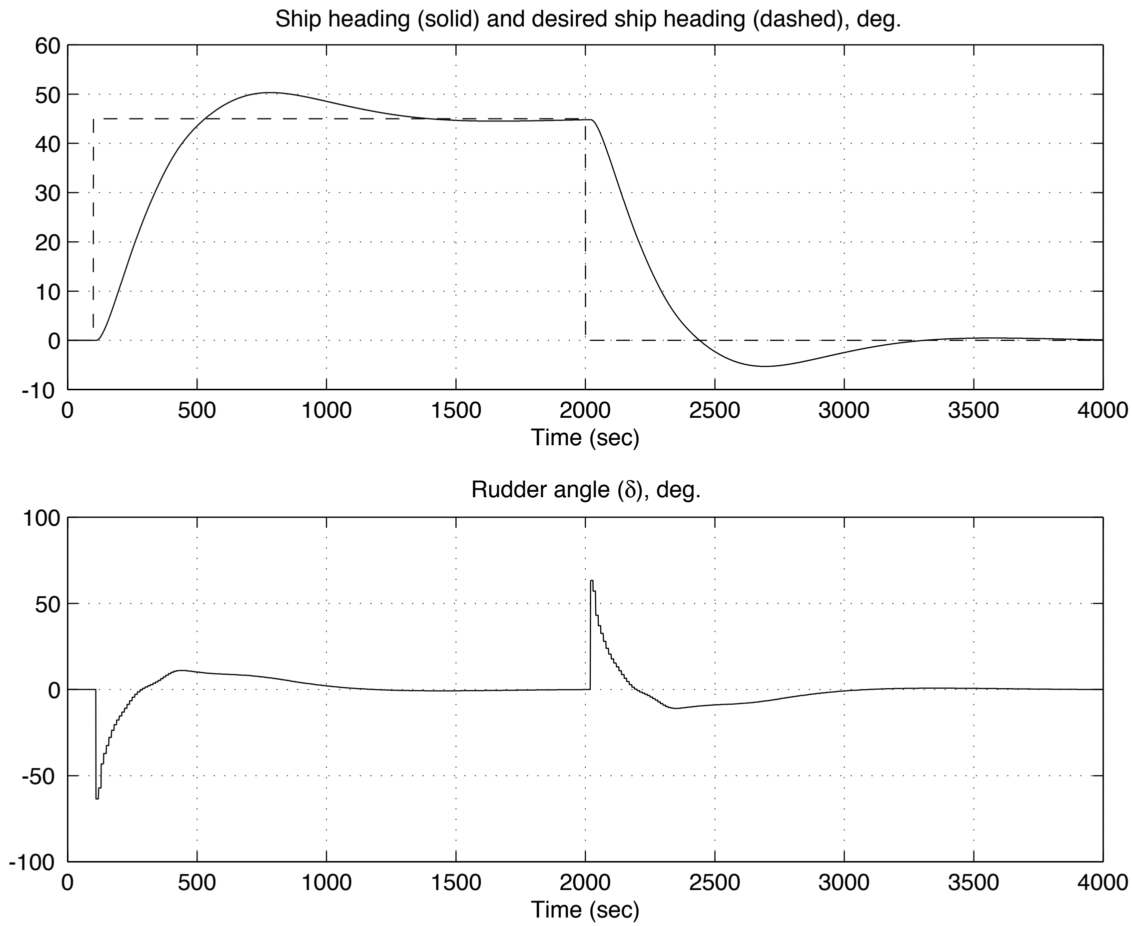

Figure 53: Closed-loop response resulting from using the radial basis function neural network for tanker ship steering, speed of 3 meters/sec.

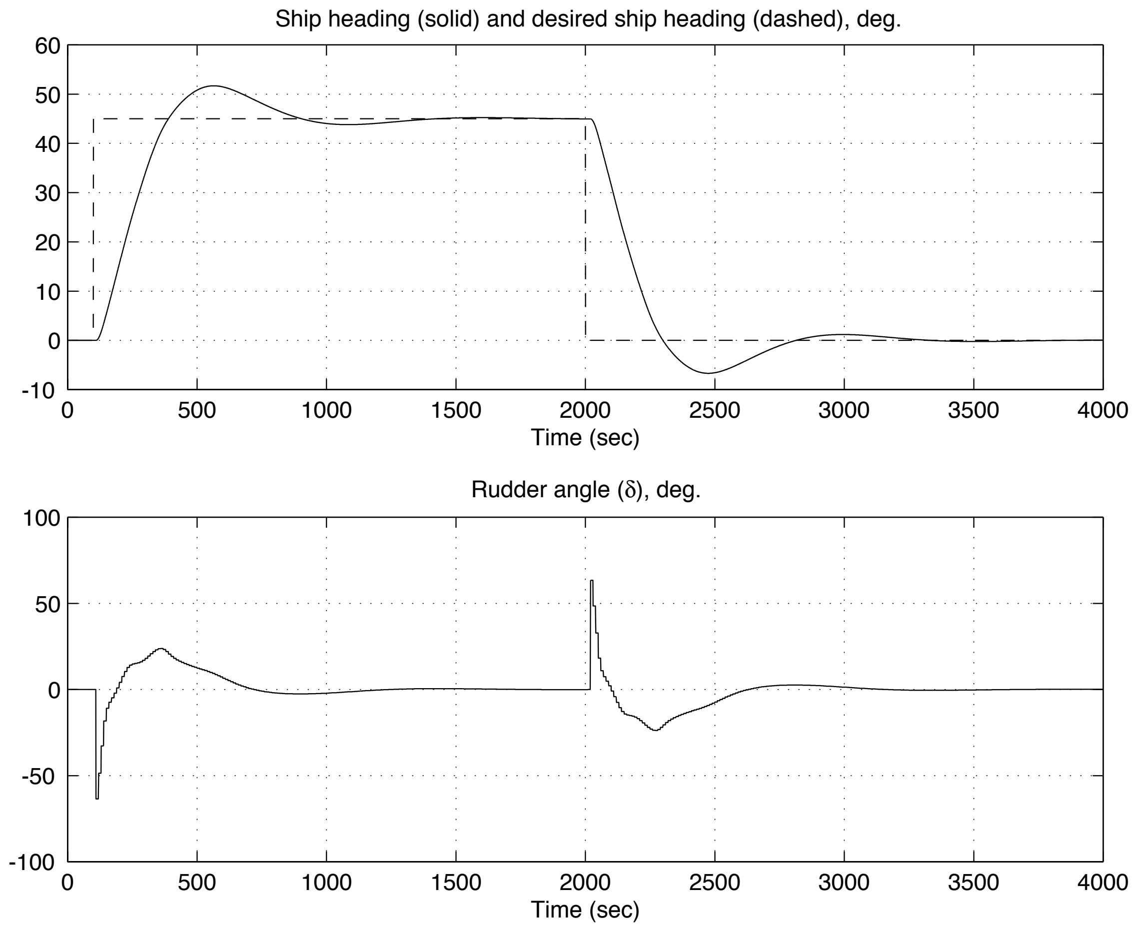

Figure 54: Closed-loop response resulting from using the radial basis function neural network for tanker ship steering, full rather than ballast conditions.

Additional Topics

➙ Stability analysis possible?? Yes.

- Lyapunov stability analysis, circle criterion, describing function analysis, etc.

➙ Hierarchical neural networks useful? Yes.

- Structure sythesis, coping with complexity, etc.

Rule-Based Control: Fuzzy/Expert Control

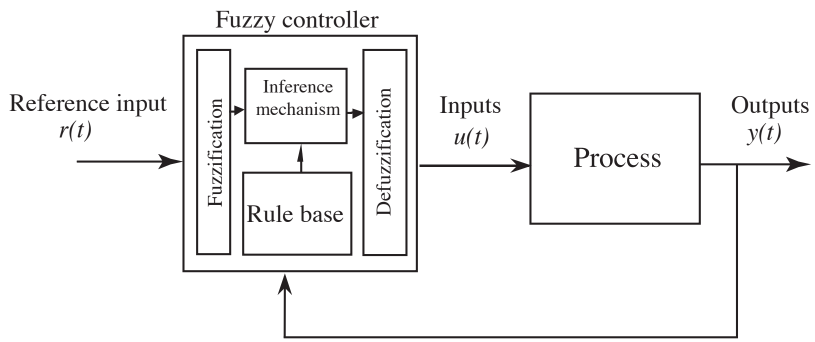

Figure 55: Fuzzy controller.

- A rule-base (a set of If-Then rules) that contains a fuzzy logic quantification of the expert's linguistic description of how to achieve good control
- An inference mechanism which emulates the expert's decision-making in interpreting and applying knowledge about how to do good control
- A fuzzification interface which converts controller inputs into

information that the inference mechanism can easily use to activate and apply rules, and

- A defuzzification interface which converts the conclusions of the inference mechanism into actual inputs for the process.

➙ Consider the tanker ship heading regulation problem, as shown in Figure 56.

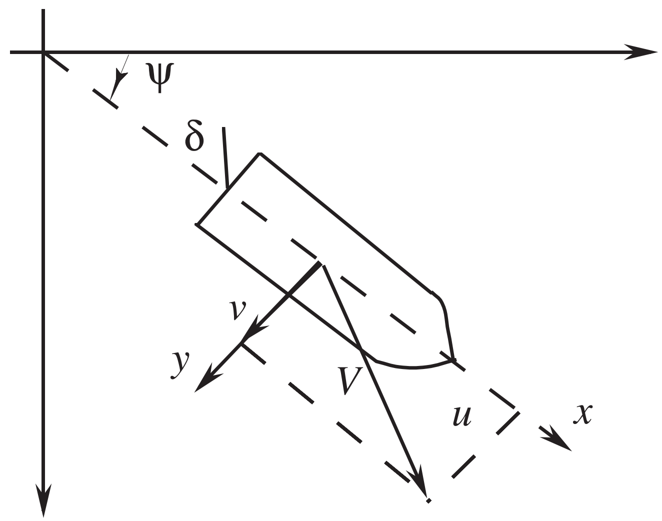

Figure 56: Tanker ship steering problem.

- Tanker ship moves in x direction at a nominal speed u
- ψ denotes the heading angle (in radians)
- δ is the rudder input (in radians).

- ψ_r is the desired ship heading

★ Goal: Want ψ to track ψ_r.

★ Steering performance is affected by:

  – Speed of travel (the rudder becomes less effective at very low speeds)
  – The ship weighs different on different trips (and heavy ships turn slower)
  – Wind hits the side of the tanker and this can affect heading regulation some
  – The sensor for the heading has some noise
  – The rudder can only move between ± 80 degrees.

Choosing Fuzzy Controller Inputs and Outputs

- Consider a human-in-the-loop whose responsibility is to control the tanker ship (i.e., the ship captain), as shown in Figure 57.

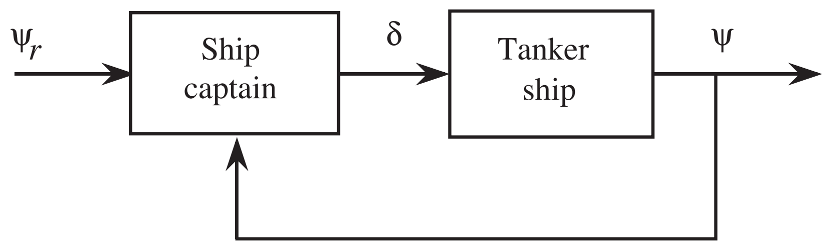

Figure 57: Human controlling a tanker ship.

- The captain tells us what information she or he will use as inputs to decision-making.

- Suppose the captain uses

$$e(t) = \psi_r(t) - \psi(t)$$

and

$$\frac{de(t)}{dt} = \dot{e}(t)$$

as the variables on which to base decisions.

➙ For more complex applications, the choice of the inputs to the controller and outputs of the controller (inputs to the plant) can be more difficult.

- The resulting fuzzy control system for the tanker ship is shown in Figure 58.

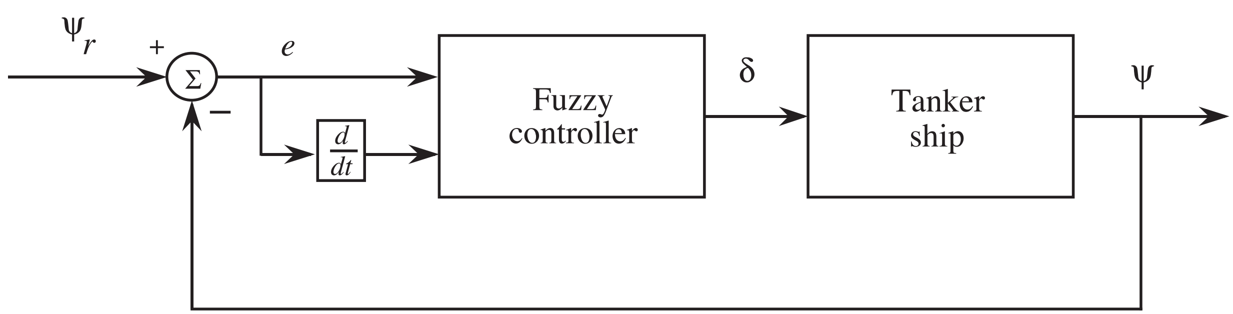

Figure 58: Fuzzy controller for a tanker ship steering problem.

- Within this framework we seek to obtain a description of how to control the process.

Putting Control Knowledge into Rule Bases

- The captain gives us a description of how best to control the plant in some natural language (e.g., English) and we characterize the expert's description with "linguistics."

Linguistic Descriptions

- "Linguistic variables" describe each of the time-varying fuzzy controller inputs and outputs.
- For the tanker ship,

"error" describes $e(t)$
"change-in-error" describes $\frac{de(t)}{dt}$
"rudder-input" describes $\delta(t)$

★ Keep the descriptions short - use whatever you would like!

- Linguistic variables take on "linguistic values."
- For the tanker ship example suppose that "error," "change-in-error," and "rudder-input" take on the following values:

"neghuge"
"neglarge"
"negbig"
"negmed"
"negsmall"
"zero"
"possmall"
"posmed"
"posbig"
"poslarge"
"poshuge"

- For an even shorter description we could use integers:

"-5" to represent "neghuge"
"-4" to represent "neglarge"
"-3" to represent "negbig"
"-2" to represent "negmed"
"-1" to represent "negsmall"
"0" to represent "zero"
"1" to represent "possmall"
"2" to represent "posmed"
"3" to represent "posbig"
"4" to represent "poslarge"
"5" to represent "poshuge"

- We are not, for example, associating "-1" with any particular number of radians of error; the use of the numbers for linguistic descriptions simply quantifies the sign of the error (in the usual way) and indicates the size in relation to the other linguistic values.

- We will call these "linguistic-numeric."

➙ The linguistic variables and values provide a language for the expert to express her or his ideas about the control decision-making process.

- Suppose that for the tanker ship $\psi_r(t) = 45$ deg. ($\psi_r(t) = \frac{45\pi}{180}$ rad.) and $e = r - y$ so that

$$e = \frac{45\pi}{180} - \psi$$

and

$$\frac{de}{dt} = -\frac{d\psi}{dt}$$

since $\frac{d\psi_r}{dt} = 0$.

- For the tanker ship each of the following statements quantifies a different configuration of the ship (refer back to Figure 56):

- The statement "error is poslarge" can represent the situation where the ship heading is at a significant angle counterclockwise to where it should be heading.
- The statement "error is negsmall" can represent the situation where the ship heading is just slightly clockwise of where it should be heading, but not too close to the reference heading ψ_r to justify quantifying it as "zero" and not too far away to justify quantifying it as "negmed."
- The statement "error is zero" can represent the situation where the ship heading is very near the desired heading (a linguistic quantification is not precise, hence we are willing to accept any value of the error around $e(t) = 0$ as being quantified linguistically by "zero" since this can be considered a better quantification than "possmall" or "negsmall").
- The statement "error is poslarge **and** change-in-error is

possmall" can represent the situation where the ship heading is counterclockwise to where it should be and, since $\frac{d\psi}{dt} < 0$, the ship heading is moving *away* from the desired heading (note that in this case the ship is moving counterclockwise).

– The statement "error is negsmall **and** change-in-error is possmall" can represent the situation where the ship heading is slightly clockwise of where it should be heading and, since $\frac{d\psi}{dt} < 0$, the ship heading is moving *toward* the desired heading (note that in this case the ship is moving counterclockwise).

Overall, we see that to do this properly you need to understand the physics of the problem (which can be much more difficult for more complex applications).

Rules

- For the tanker ship in the three positions shown in Figure 59

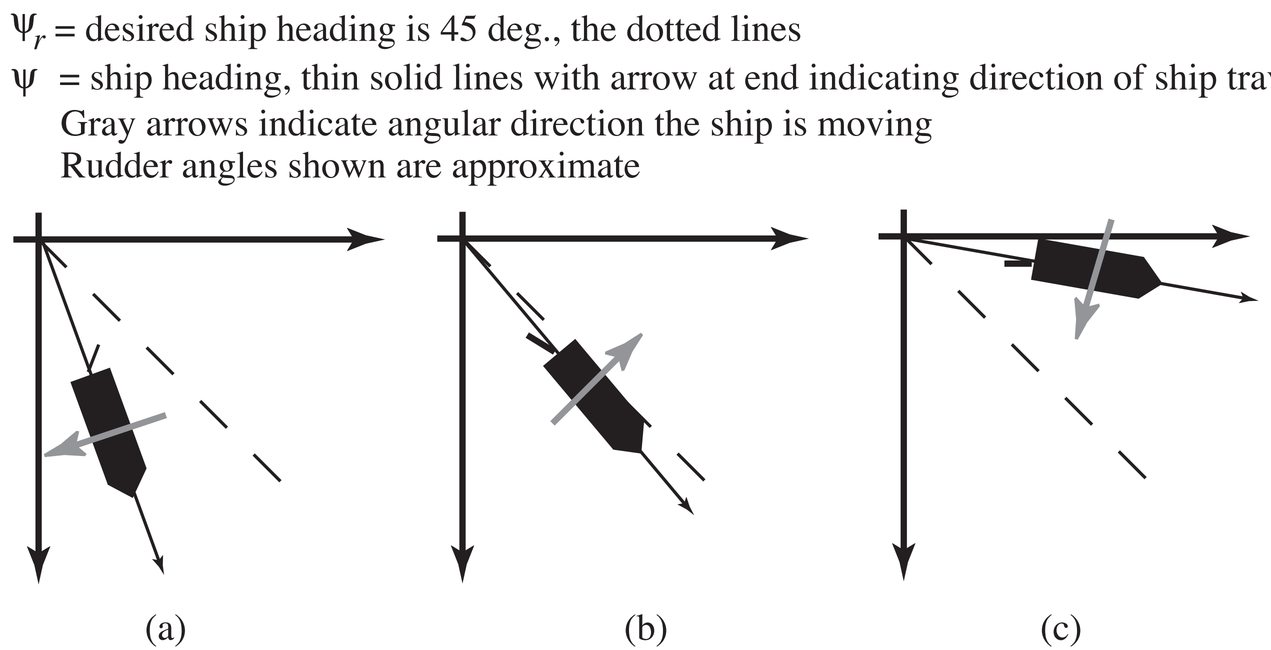

Figure 59: Tanker ship in various positions.

- We have the following "linguistic rules":
  1. **If** error is negsmall **and** change-in-error is negsmall **Then** rudder-input is posmed

     This rule quantifies the situation in Figure 59(a) where the ship has a heading angle that is clockwise of the desired heading and is moving clockwise; hence it is clear that we should apply a medium positive rudder angle so that we can get the ship moving in the proper direction.
  2. **If** error is zero **and** change-in-error is possmall **Then** rudder-input is negsmall

     This rule quantifies the situation in Figure 59(b) where the ship is nearly moving in the proper direction (a linguistic quantification of zero does not imply that $e(t) = 0$ exactly) and is moving counterclockwise; hence we should apply a small negative rudder angle to counteract the movement so that it moves toward zero (a positive rudder angle could result in the ship heading overshooting the desired angle).

3. **If** error is possmall **and** change-in-error is negsmall **Then** rudder-input is zero

   This rule quantifies the situation in Figure 59(c) where the ship is counterclockwise of the desired heading and is moving clockwise; hence we apply a near zero rudder angle since the ship is already moving in the proper direction.

- General form is

If premise **Then** consequent

Rule Bases

➙ There are at most $11^2 = 121$ possible rules (all possible combinations of premise linguistic values for two inputs) if you use both premise terms in every rule.

- A tabular representation of these rules is shown in Table 1.
- Notice that the body of the table lists the linguistic-numeric consequents of the rules, and the left column and top row of the table contain the linguistic-numeric premise terms. Then, for instance, the $(+1, -1)$ position (where the "+1" represents the row having "+1" for a numeric-linguistic value and the "-1" represents the column having "-1" for a numeric-linguistic value) has a 0 ("zero") in the body of the table and represents the rule

  If error is possmall **and** change-in-error is negsmall **Then** rudder-input is zero

  which is rule 3 above.

★ Notice the pattern.

Table 1: Rule Table for the Tanker Ship

"rudder-input" δ		"change-in-error" $\dot{e}$										
		-5	-4	-3	-2	-1	0	1	2	3	4	5
"error" e	-5	5	5	5	5	5	5	4	3	2	1	0
	-4	5	5	5	5	5	4	3	2	1	0	-1
	-3	5	5	5	5	4	3	2	1	0	-1	-2
	-2	5	5	5	4	3	2	1	0	-1	-2	-3
	-1	5	5	4	3	2	1	0	-1	-2	-3	-4
	0	5	4	3	2	1	0	-1	-2	-3	-4	-5
	1	4	3	2	1	0	-1	-2	-3	-4	-5	-5
	2	3	2	1	0	-1	-2	-3	-4	-5	-5	-5
	3	2	1	0	-1	-2	-3	-4	-5	-5	-5	-5
	4	1	0	-1	-2	-3	-4	-5	-5	-5	-5	-5
	5	0	-1	-2	-3	-4	-5	-5	-5	-5	-5	-5

Fuzzy Quantification of Knowledge

- We use fuzzy logic to quantify the meaning of linguistic descriptions.

Membership Functions

- We quantify the meaning of the linguistic values using "membership functions."
- Consider, for example, Figure 60.

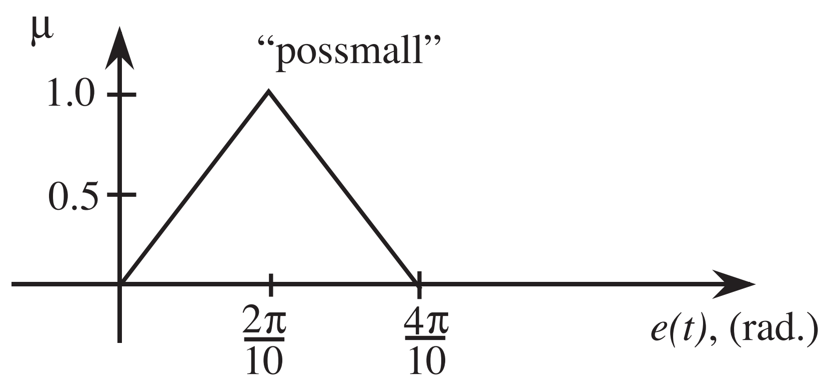

Figure 60: Membership function for linguistic value "possmall."

- The function μ quantifies the certainty that $e(t)$ can be classified linguistically as "possmall."
- For various values of $e(t)$:
  - If $e(t) = -\frac{4\pi}{10}$ then $\mu(-\frac{4\pi}{10}) = 0$, indicating that we are certain that $e(t) = -\frac{4\pi}{10}$ is *not* "possmall" (indeed, it is negative).
  - If $e(t) = \frac{2\pi}{20}$ then $\mu(\frac{2\pi}{20}) = 0.5$, indicating that we are halfway certain that $e(t) = \frac{2\pi}{20}$ is "possmall" (we are only halfway certain since it could also be "zero" with some degree of certainty—this value is in a "gray area" in terms of linguistic interpretation).
  - If $e(t) = \frac{2\pi}{10}$ then $\mu(\frac{2\pi}{10}) = 1.0$, indicating that we are absolutely certain that $e(t) = \frac{2\pi}{10}$ is what we mean by "possmall."

- The membership function quantifies the linguistic statement "error is possmall."

★ There are many different possible membership functions (Figure 61).

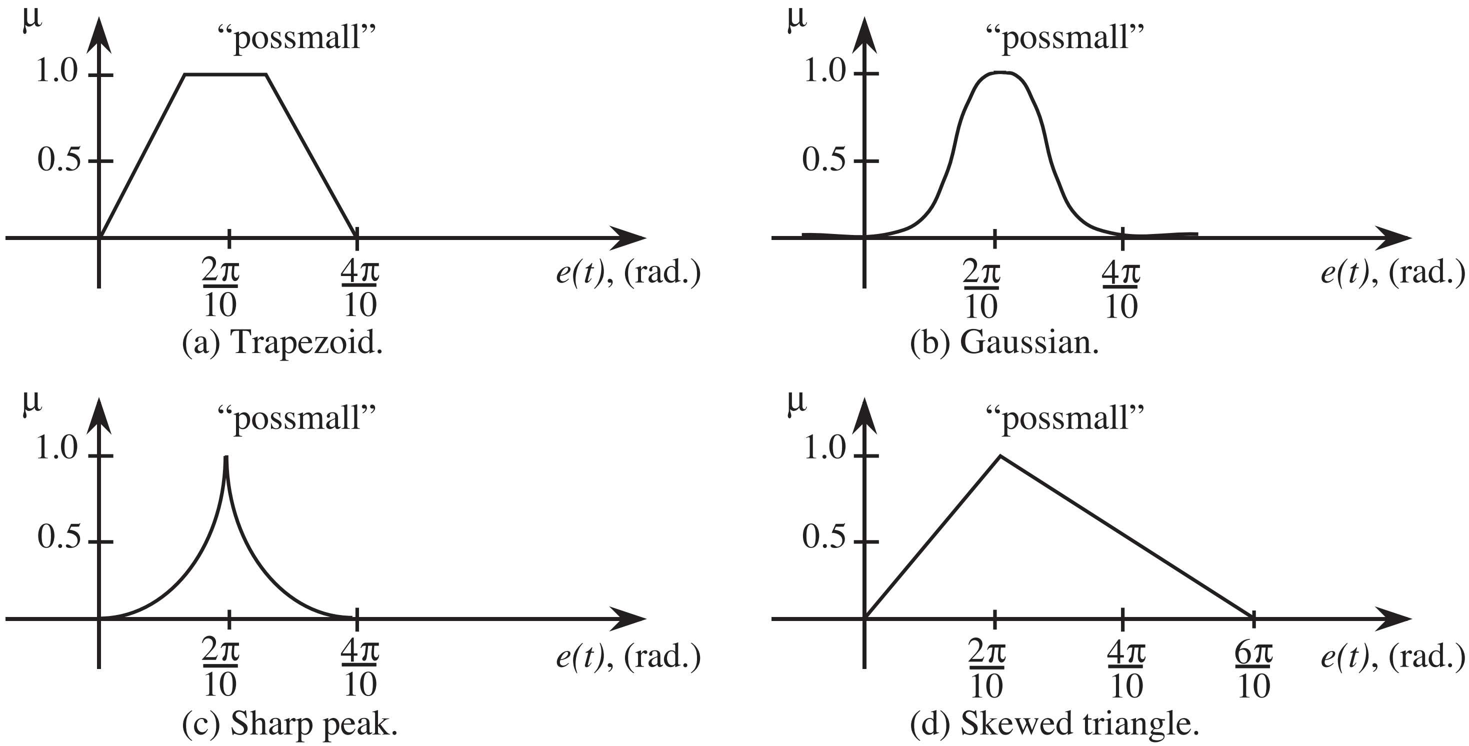

Figure 61: A few membership function choices for representing "error is possmall."

- The set of values that is described by μ as being "positive small" is called a "fuzzy set."

- Let A denote this fuzzy set.
- Notice that from Figure 60 we are absolutely certain that $e(t) = \frac{2\pi}{10}$ is an element of A, but we are less certain that $e(t) = \frac{2\pi}{40}$ is an element of A.
- Membership in the set is fuzzy, hence the term "fuzzy set."
- A "crisp" (as contrasted to "fuzzy") quantification of "possmall" can also be specified, but via the membership function shown in Figure 62.

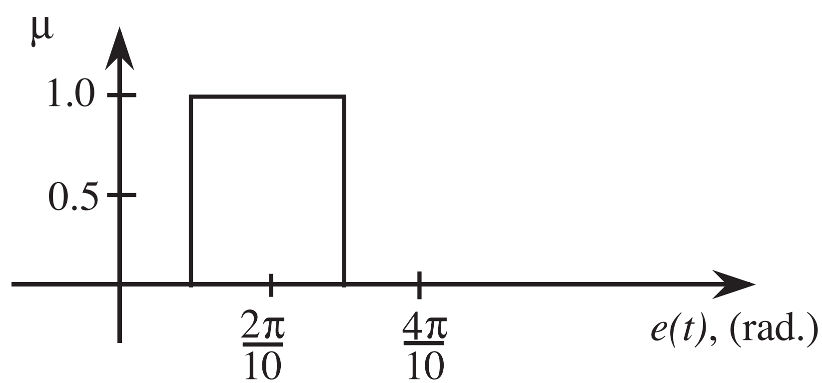

Figure 62: Membership function for a crisp set.

- The horizontal axis in Figure 60 is called the "universe of discourse" for the input $e(t)$ (it is simply the domain).
- For the tanker ship we can define the membership functions as in Figure 63.

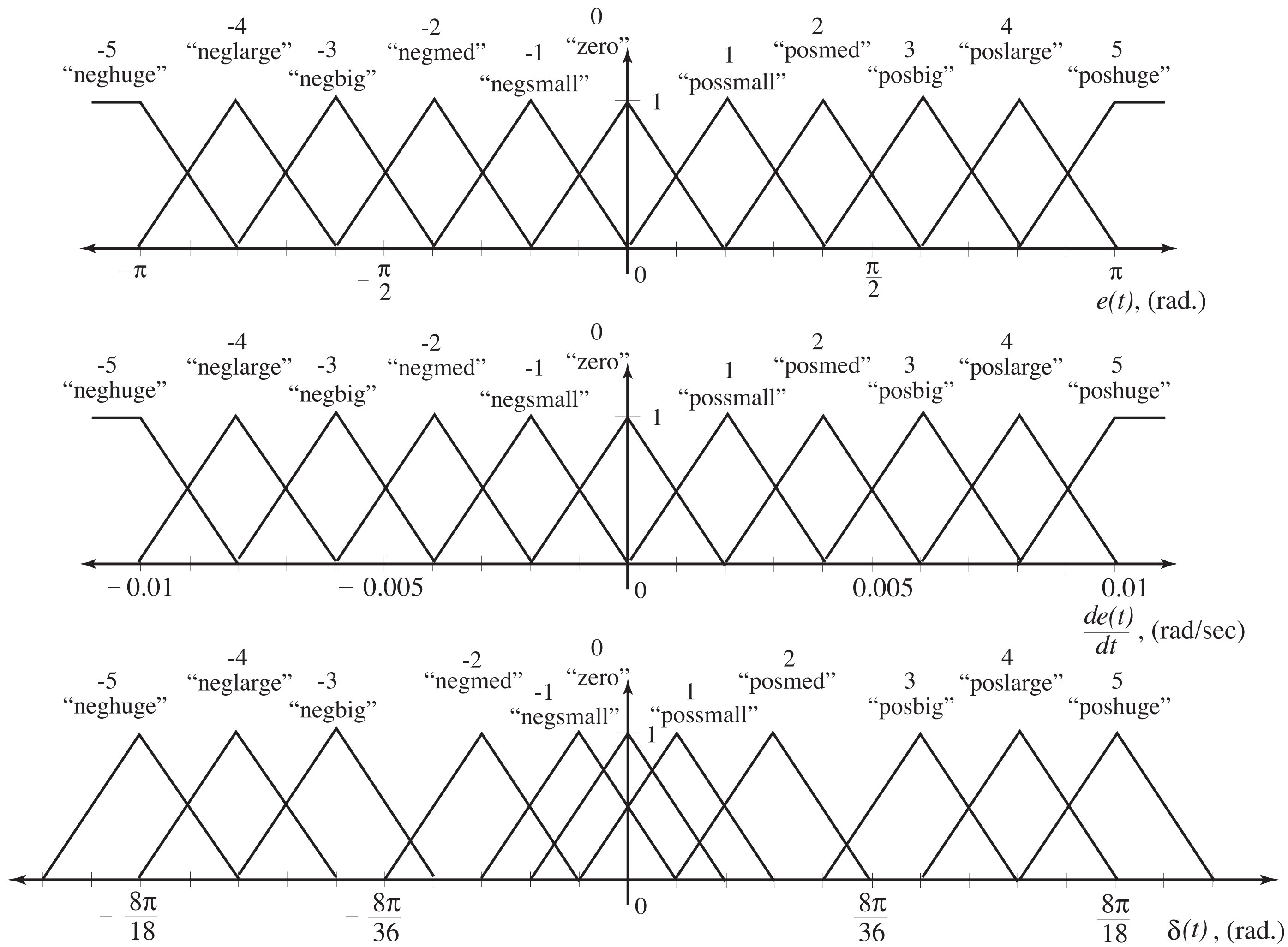

Figure 63: Membership functions for a ship steering example.

- Notice then that the meaning of the linguistics on the $\dot{e}$ universe of discourse is different from those on the e universe of discourse (due to a scale change).
- The human expert would just groups all large values together in a linguistic description such as "poshuge" (or "neghuge").
- They characterize "greater than" (for the right side) and "less than" (for the left side).

★ It should be clear in your mind how different function values arise as the controller inputs vary.

- For the "rudder-input" $\delta(t)$ the horizonal scale was chosen since the rudder input can only be moved between ± 80 degrees.
- Next, note that for the output δ, the membership functions at the outermost edges cannot be saturated for the fuzzy system to be properly defined (more on this later).

The Meaning of Membership Functions and Rules

- Notice that the pattern of center positions for the output membership functions in Figure 63 is not uniform as it is for the input universes of discourse.
- To get a uniform distribution of output membership function centers you can choose the center values, which we denote by b_i where i is the linguistic-numeric index for the corresponding membership function, as

$$b_i = \frac{8\pi}{18}\left(\frac{i}{5}\right)$$

  where $i = -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5$ (and then use the same base widths and rule base).
- Due to the lack of clarity of the meaning of control rules in the linguistic rule base shown in Table 1, schemes are often used which include membership function information in the rule

base table.

- For example, consider Table 2 where the centers of the appropriate output membership functions are listed, up to a scale factor, which in this case is $8\pi/18$ (i.e., to get the actual center from the rule base table you take the entry and multiply it by $8\pi/18$).

Table 2: Rule Table for the Tanker Ship (body of table holds the output membership function centers where each element should be multiplied by $8\pi/18$).

		$\dot{e}$										
		-5	-4	-3	-2	-1	0	1	2	3	4	5
	-5	1	1	1	1	1	1	.8	.6	.3	.1	0
	-4	1	1	1	1	1	.8	.6	.3	.1	0	$-.1$
	-3	1	1	1	1	.8	.6	.3	.1	0	$-.1$	$-.3$
	-2	1	1	1	.8	.6	.3	.1	0	$-.1$	$-.3$	$-.6$
	-1	1	1	.8	.6	.3	.1	0	$-.1$	$-.3$	$-.6$	$-.8$
e	0	1	.8	.6	.3	.1	0	$-.1$	$-.3$	$-.6$	$-.8$	-1
	1	.8	.6	.3	.1	0	$-.1$	$-.3$	$-.6$	$-.8$	-1	-1
	2	.6	.3	.1	0	$-.1$	$-.3$	$-.6$	$-.8$	-1	-1	-1
	3	.3	.1	0	$-.1$	$-.3$	$-.6$	$-.8$	-1	-1	-1	-1
	4	.1	0	$-.1$	$-.3$	$-.6$	$-.8$	-1	-1	-1	-1	-1
	5	0	$-.1$	$-.3$	$-.6$	$-.8$	-1	-1	-1	-1	-1	-1

➙ Coupled with our understanding of the meaning of the linguistic-numeric indices for the error and change in error, all the major components of the captain's knowledge of ship steering are directly evident from Table 2 in the following manner:

1. If the heading error and change in error are both too big (upper left and lower right corners of the rule base shown in Table 2), then use the appropriate maximum rudder input.
2. For zero e and $\dot{e}$, the rudder angle should be zero, but if e and $\dot{e}$ move positive then the rudder should move negative (where if $\dot{e}$ moves significantly positive, then the rudder should move even more negative). Similar reasoning is used for e and $\dot{e}$ negative where we then make the rudder angle positive and for the case where e and $\dot{e}$ have opposite signs, depending on the magnitude of the signals we will make the rudder input either positive or negative.
3. For small e and $\dot{e}$ be conservative in making changes to the rudder position since such corrections may cause heading deviations instead (i.e., lower the "gain" of the controller

near zero so that noise is not amplified). Also, if the ship's angular position is moving sufficiently fast to remove the heading error then be conservative in using the rudder to help move it since this can require unnecessary control energy.

- This provides a summary of the captain's knowledge about ship steering.

Fuzzification

★ The fuzzification process is simply the act of obtaining a value of an input variable (e.g., $e(t)$) and finding the numeric values of the membership function(s)

- Some think of the membership function values as an "encoding" of the fuzzy controller numeric input values.

Matching: Determining Which Rules to Use

- The inference process generally involves two steps:
  1. The premises of all the rules are compared to the controller inputs to determine which rules apply to the current situation. This "matching" process involves determining the certainty that each rule applies, and typically we will more strongly take into account the recommendations of rules that we are more certain apply to the current situation.
  2. The conclusions (what control actions to take) are determined using the rules that have been determined to apply at the current time. The conclusions are characterized with a fuzzy set (or sets) that represents the certainty that the input to the plant should take on various values.

Premise Quantification via Fuzzy Logic

- To perform inference we must first quantify each of the rules with fuzzy logic.

- To do this we first quantify the meaning of the premises of the rules that are composed of several terms, each of which involves a fuzzy controller input.
- Consider Figure 64, where we list two terms from the premise of the rule

If error is zero **and** change-in-error is possmall **Then** rudder-input is negsmall

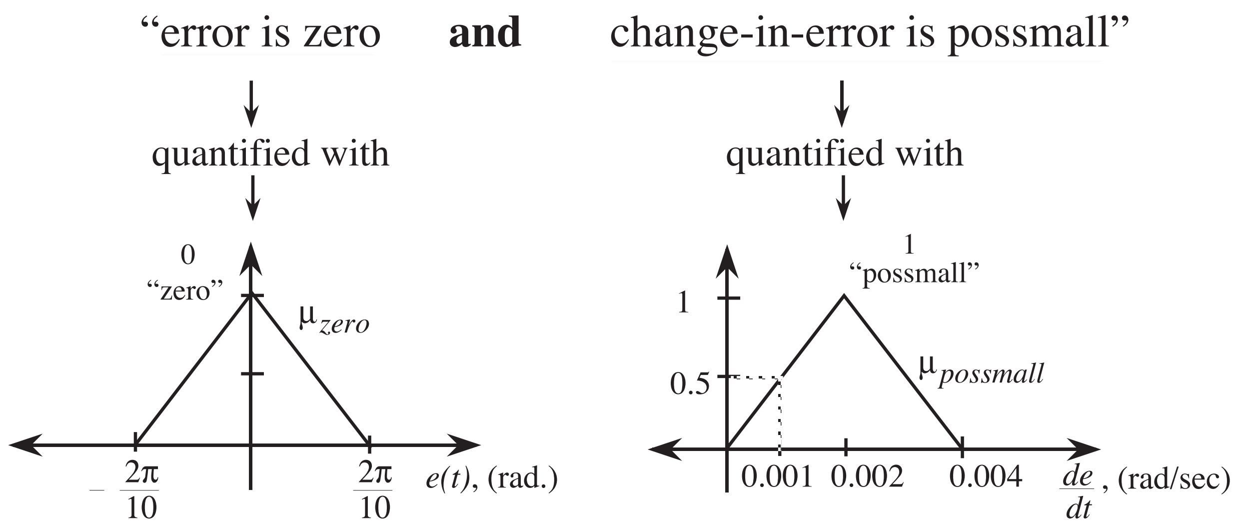

Figure 64: Membership functions of premise terms.

- Above, we had quantified the meaning of the linguistic terms "error is zero" and "change-in-error is possmall" via the membership functions shown in Figure 63.
- Now we seek to quantify the linguistic premise "error is zero **and** change-in-error is possmall."

- To see how to quantify the "and" operation, begin by supposing that $e(t) = \pi/10$ and $\dot{e}(t) = 0.0005$, so that using Figure 63 (or Figure 64) we see that

$$\mu_{zero}(e(t)) = 0.5$$

and

$$\mu_{possmall}\left(\dot{e}(t)\right) = 0.25$$

- What, for these values of $e(t)$ and $\dot{e}(t)$, is the certainty of the statement

"error is zero **and** change-in-error is possmall"

that is the premise from the above rule?

- We will denote this certainty by $\mu_{premise}$.
- There are actually several ways to define it:
  - Minimum: Define $\mu_{premise} = \min\{0.5, 0.25\} = 0.25$, that is, using the minimum of the two membership values.

- Product: Define $\mu_{premise} = (0.5)(0.25) = 0.125$, that is, using the product of the two membership values.

- Notice that both ways of quantifying the "and" operation in the premise indicate that you can be no more certain about the conjunction of two statements than you are about the individual terms that make them up (note that $0 \leq \mu_{premise} \leq 1$ for either case).
- We will obtain a multidimensional membership function $\mu_{premise}(e(t), \dot{e}(t))$ for each rule (Figures 65 and 66).

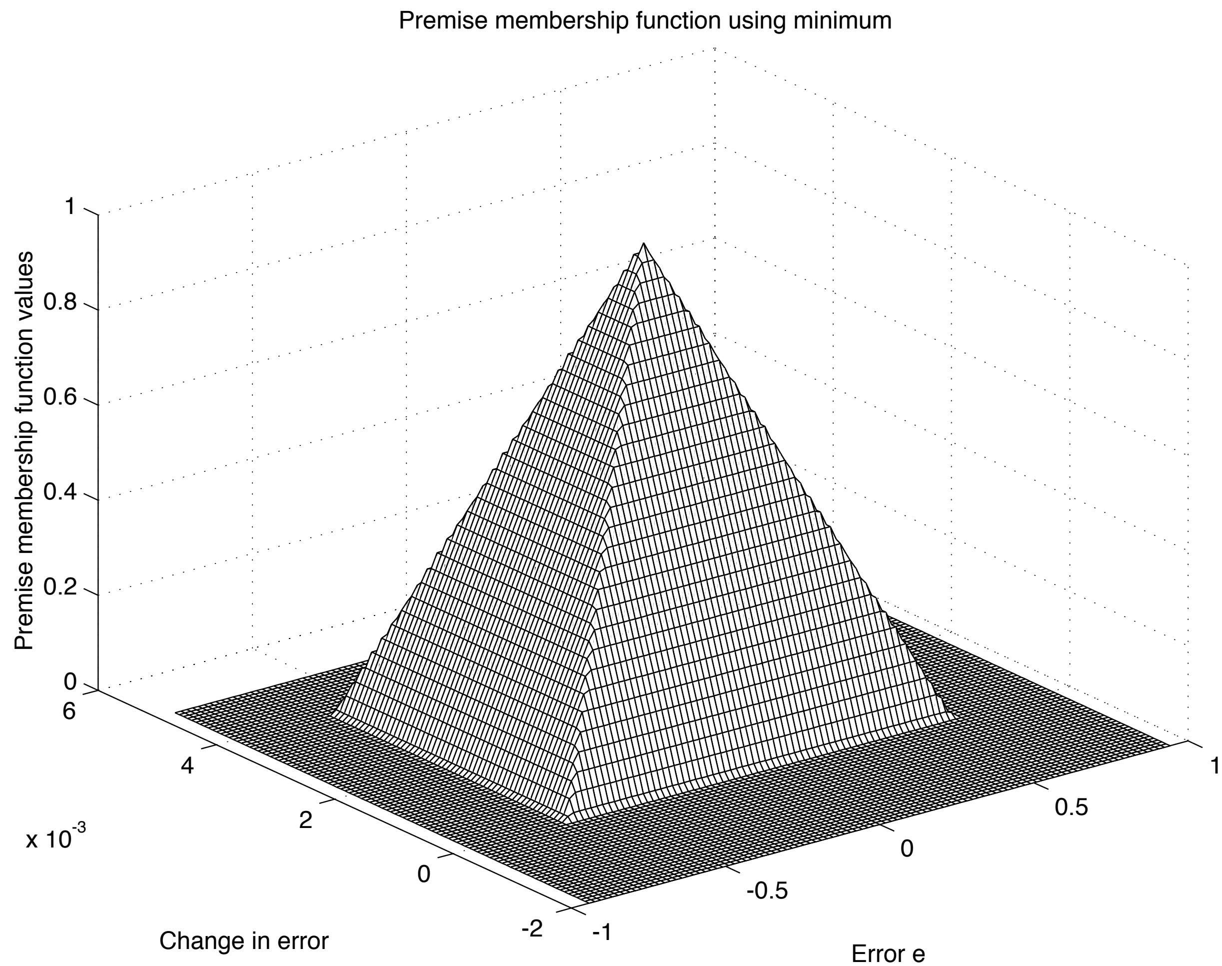

Figure 65: Membership function of the premise for a single rule using minimum to represent the conjunction.

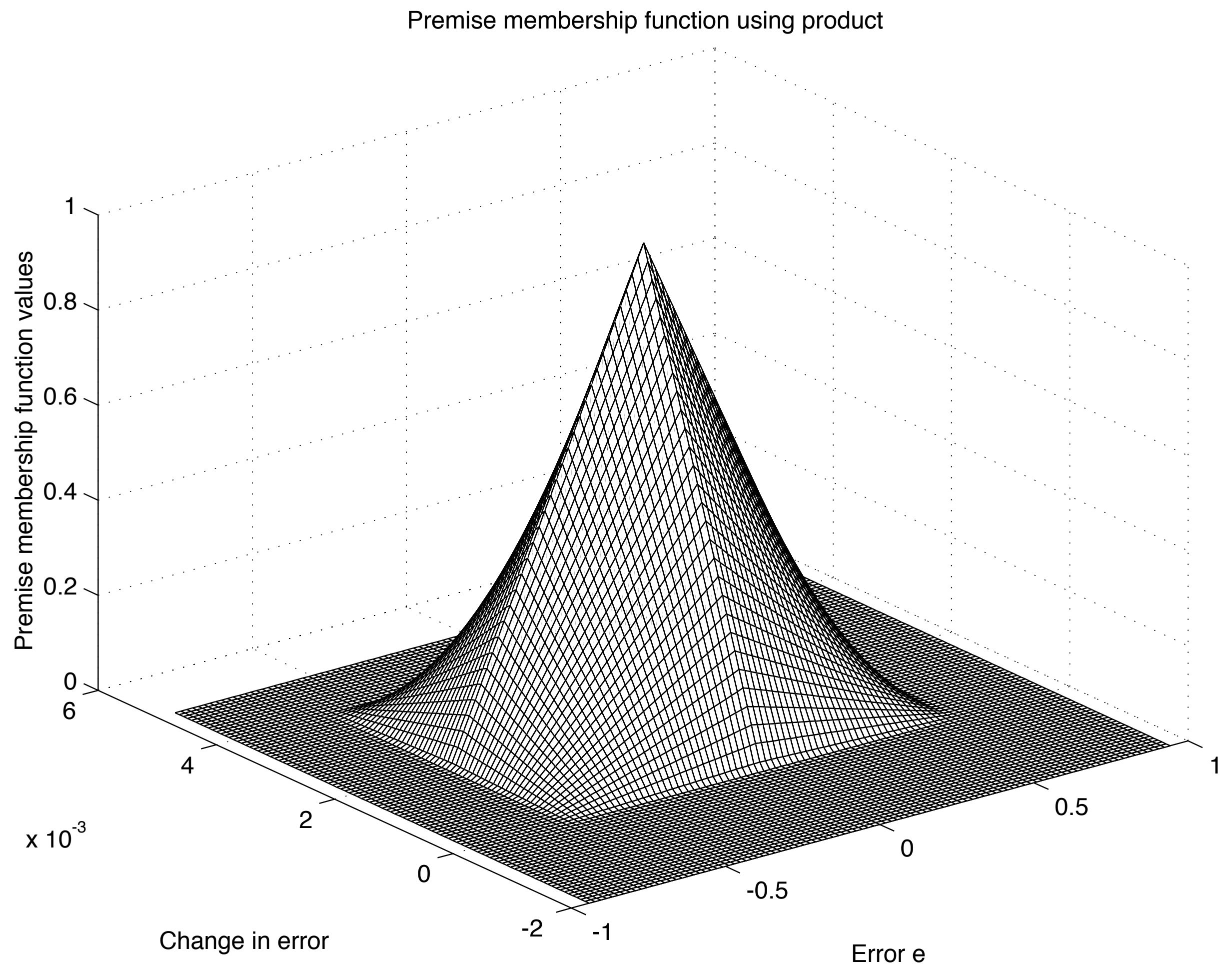

Figure 66: Membership function of the premise for a single rule using product to represent the conjunction.

It is important you picture in your mind the situation where $e(t)$ and $\dot{e}(t)$ change dynamically over time and the values of $\mu_{premise}(e(t), \dot{e}(t))$ for each rule change, and hence the applicability of each rule in the rule base for specifying the rudder input to the ship, changes with time.

Determining Which Rules Are On

- Determining the applicability of each rule is called "matching."
- We say that a rule is "on at time t" if its premise membership function $\mu_{premise}(e(t), \dot{e}(t)) > 0$.
- For the ship example, suppose that

$$e(t) = 0$$

and

$$\dot{e}(t) = 0.0015$$

- Figure 67 indicates with thick black vertical lines the values above for $e(t)$ and $\dot{e}(t)$.

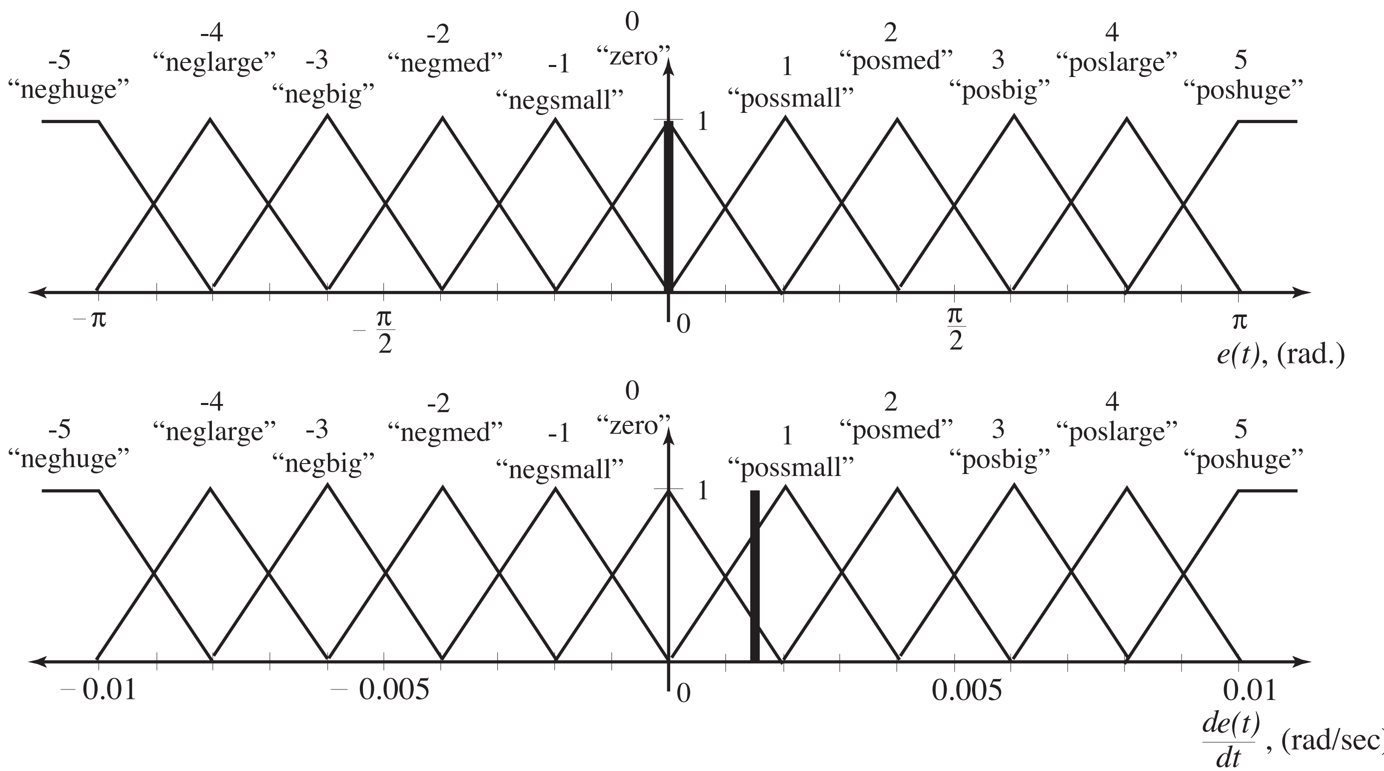

Figure 67: Input membership functions with input values.

- Notice that $\mu_{zero}(e(t)) = 1$ but that the other membership functions for the $e(t)$ input are all "off" (i.e., their values are zero).

- For the $\dot{e}(t)$ input we see that $\mu_{zero}(\dot{e}(t)) = 0.25$ and $\mu_{possmall}(\dot{e}(t)) = 0.75$ and that all the other membership functions are off.

- This implies that rules that have the premise terms

  "error is zero"

  "change-in-error is zero"

  "change-in-error is possmall"

  are on (all other rules have $\mu_{premise}(e(t), \dot{e}(t)) = 0$).

★ So, which rules are these?

- Using Table 1, the following rules are on:

  1. **If** error is zero **and** change-in-error is zero **Then** rudder-input is zero

2. **If** error is zero **and** change-in-error is possmall **Then** rudder-input is negsmall

Note that since for the ship steering example we have at most two membership functions overlapping, we will never have more than four rules on at one time (this concept generalizes to many inputs and is very useful to reduce computational complexity in applications).

★ Actually, for this system we will either have one, two, or four rules on at any one time. Why?

➙ It is useful to consider pictorially which rules are on. Consider Table 3, which is a copy of Table 2 on page 121 with boxes drawn around the consequents of the rules that are on (notice that these are the *same* two rules listed above).

Table 3: Rule Table for the Tanker Ship with Rules That Are "On" Highlighted (body of table holds the output membership function centers where each element should be multiplied by $8\pi/18$).

		$\dot{e}$										
		-5	-4	-3	-2	-1	0	1	2	3	4	5
e	-5	1	1	1	1	1	1	.8	.6	.3	.1	0
	-4	1	1	1	1	1	.8	.6	.3	.1	0	$-.1$
	-3	1	1	1	1	.8	.6	.3	.1	0	$-.1$	$-.3$
	-2	1	1	1	.8	.6	.3	.1	0	$-.1$	$-.3$	$-.6$
	-1	1	1	.8	.6	.3	.1	0	$-.1$	$-.3$	$-.6$	$-.8$
	0	1	.8	.6	.3	.1	$\boxed{0}$	$\boxed{-.1}$	$-.3$	$-.6$	$-.8$	-1
	1	.8	.6	.3	.1	0	$-.1$	$-.3$	$-.6$	$-.8$	-1	-1
	2	.6	.3	.1	0	$-.1$	$-.3$	$-.6$	$-.8$	-1	-1	-1
	3	.3	.1	0	$-.1$	$-.3$	$-.6$	$-.8$	-1	-1	-1	-1
	4	.1	0	$-.1$	$-.3$	$-.6$	$-.8$	-1	-1	-1	-1	-1
	5	0	$-.1$	$-.3$	$-.6$	$-.8$	-1	-1	-1	-1	-1	-1

With this, you should picture in your mind how a region of rules that are on (that involves no more than four cells in the body of Table 3 due to how we define the input membership functions) will dynamically move around in the table as the values of $e(t)$ and $\dot{e}(t)$ change.

Inference Step: Determining Conclusions

Recommendation from One Rule

➙ Consider the conclusion reached by the rule

If error is zero **and** change-in-error is zero **Then** rudder-input is zero

which for convenience we will refer to as "rule (1)."

- Using the minimum to represent the premise, we have

$$\mu_{premise_{(1)}} = \min\{1, 0.25\} = 0.25$$

(the notation $\mu_{premise_{(1)}}$ represents $\mu_{premise}$ for rule (1)) so that we are 0.25 certain that this rule applies to the current situation.

- The rule indicates that if its premise is true then the action indicated by its consequent should be taken.
- For rule (1) the consequent is "rudder-input is zero" (this makes sense, for here the ship is headed in the proper direction, so we should not apply a rudder input that is different from zero since this would tend to move the ship heading away from the desired heading).
- The membership function for this consequent is shown in Figure 68(a).

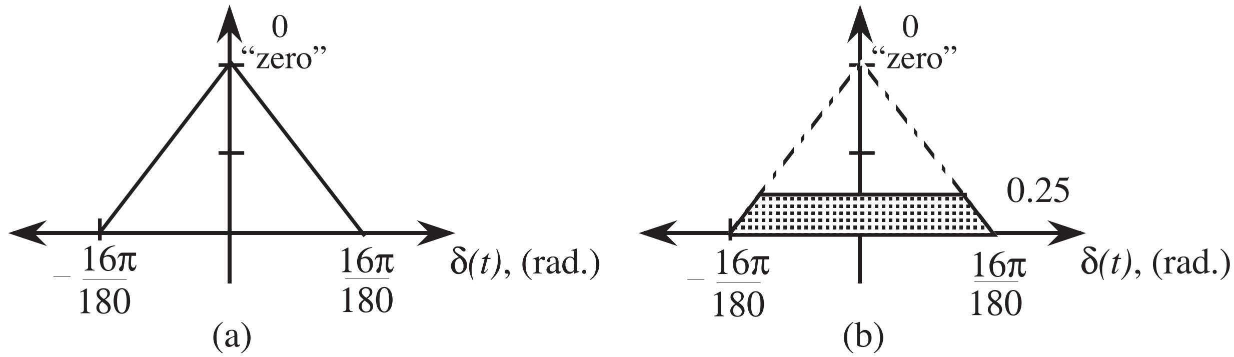

Figure 68: (a) Consequent membership function and (b) implied fuzzy set with membership function $\mu_{(1)}(\delta)$ for rule (1).

- The membership function for the conclusion reached by rule (1), which we denote by $\mu_{(1)}$, is shown in Figure 68(b) and is given by

$$\mu_{(1)}(\delta) = \min\{\mu_{premise_{(1)}}, \mu_{zero}(\delta)\}$$

(where $\mu_{premise_{(1)}} = 0.25$ as determined above).

- This membership function defines the "implied fuzzy set" for rule (1) (i.e., it is the conclusion that is implied by rule (1)).

★ The justification for the use of the minimum operator to represent the implication is that we can be no more certain about our consequent than our premise (Zadeh's compositional rule of inference).

- Notice that $\mu_{(1)}(\delta)$ is a *function* of δ and that the minimum operation will generally "chop off the top" of $\mu_{zero}(\delta)$ to produce $\mu_{(1)}(\delta)$
- $\mu_{(1)}(\delta)$ is in general a time-varying function that quantifies how certain rule (1) is that the force input δ should take on certain values.
- It is most certain that the force input should lie in a region around zero (see Figure 68(b)), and it indicates that it is certain that the force input should not be too large in either the positive or negative direction
- $\mu_{(1)}(\delta)$ quantifies the conclusion reached by only rule (1) and

only for the current $e(t)$ and $\dot{e}(t)$.

Recommendation from Another Rule

- Consider the conclusion reached by the other rule that is on,

  If error is zero **and** change-in-error is possmall **Then** rudder-input is negsmall

  which for convenience we will refer to as "rule (2)."

- Using the minimum to represent the premise, we have

$$\mu_{premise_{(2)}} = \min\{1, 0.75\} = 0.75$$

  so that we are 0.75 certain that this rule applies to the current situation.

- Notice that we are much more certain that rule (2) applies to the current situation than rule (1).

- For rule (2) the consequent is "rudder-input is negsmall" (this makes sense, for here the ship is heading in the proper direction but is moving in the counterclockwise direction with a small velocity).

- The membership function for this consequent is shown in Figure 69(a).

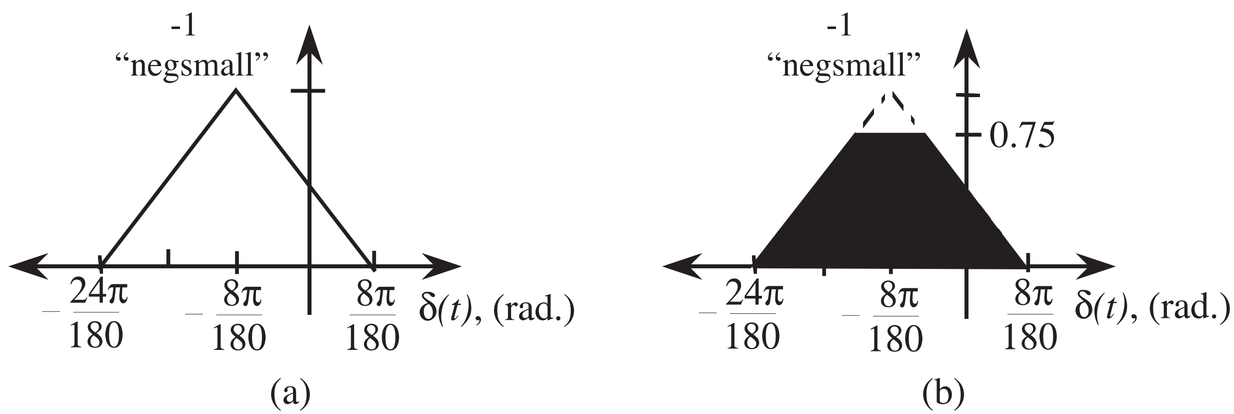

Figure 69: (a) Consequent membership function and (b) implied fuzzy set with membership function $\mu_{(2)}(\delta)$ for rule (2).

- The membership function for the conclusion reached by rule

(2), which we denote by $\mu_{(2)}$, is shown in Figure 69(b) (the shaded region) and is given by

$$\mu_{(2)}(\delta) = \min\{\mu_{premise_{(2)}}, \mu_{negsmall}(\delta)\}$$

(where $\mu_{premise_{(2)}} = 0.75$ as determined above).

- This membership function defines the implied fuzzy set for rule (2) (i.e., it is the conclusion that is reached by rule (2)).
- As rule (2) has a premise membership function that has higher certainty than for rule (1), we see that we are more certain of the conclusion reached by rule (2).

Converting Decisions into Actions

- Done via "defuzzification"
- First, we draw all the implied fuzzy sets on one axis as shown in Figure 70.

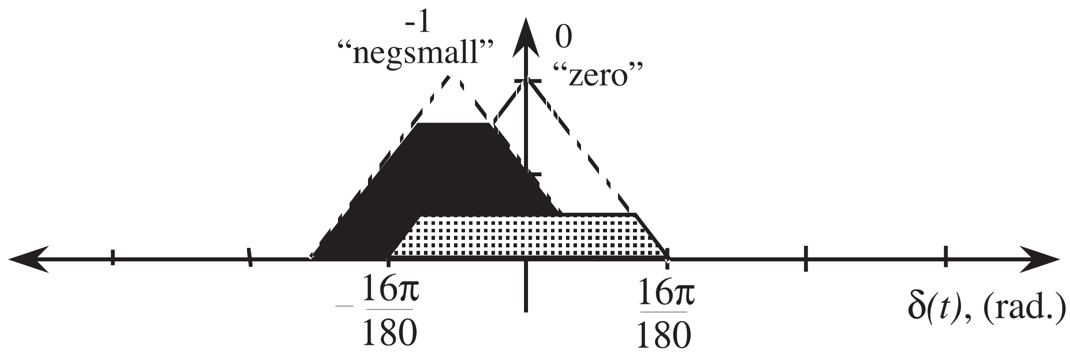

Figure 70: Implied fuzzy sets.

- We want to find the one output, which we denote by "δ^{crisp}," that best represents the conclusions of the fuzzy controller that are represented with the implied fuzzy sets.

➙ There are actually many approaches to defuzzification.

Combining Recommendations

- First consider the "center of gravity" (COG) defuzzification method for combining the recommendations represented by the implied fuzzy sets from all the rules.

- Let b_i denote the center of the membership function for the implied fuzzy set for the i^{th} rule (i.e., where the membership function for the i^{th} rule reaches its peak for our example since the output fuzzy sets are all symmetric about their peaks).
- For our ship example we have

$$b_1 = 0.0$$

and

$$b_2 = -0.1\left(\frac{8\pi}{18}\right)$$

as shown in Figure 70.

- Let

$$\int \mu_{(i)}$$

denote the area under the membership function $\mu_{(i)}$.

- The COG method computes δ^{crisp} to be

$$\delta^{crisp} = \frac{\sum_i b_i \int \mu_{(i)}}{\sum_i \int \mu_{(i)}} \tag{5}$$

- Three items about Equation (5) are important to note:
  1. We cannot have output membership functions that have infinite area (this is why we could not saturate the output membership functions).
  2. You must be careful to define the input and output membership functions so that the sum in the denominator of Equation (5) is not equal to zero no matter what the inputs to the fuzzy controller are.
  3. While at first glance it may not appear so, $\int \mu_{(i)}$ is easy to compute for our example. For the case where we have symmetric triangular output membership functions that

peak at one and have a base width of w, simple geometry can be used to show that the area under a triangle "chopped off" at a height of h (such as the ones in Figures 68 and 69) is equal to

$$w\left(h-\frac{h^2}{2}\right)$$

(note that if w is the same for every output membership function then it cancels in Equation (5)).

- Using Equation (5) with Figure 70 we have

$$\delta^{crisp}=\frac{(0)\left(0.25-\frac{(0.25)^2}{2}\right)+\left(-0.1\frac{8\pi}{18}\right)\left(0.75-\frac{(0.75)^2}{2}\right)}{\left(0.25-\frac{(0.25)^2}{2}\right)+\left(0.75-\frac{(0.75)^2}{2}\right)}=-0.0952$$

as the input to the ship for the given $e(t)$ and $\dot{e}(t)$.

➙ Does this value for a force input (i.e., -5.4545 degrees) make sense?

- Consider Figure 71, where we have taken the implied fuzzy sets from Figure 70 and simply added an indication of what number COG defuzzification says is the best representation of the conclusions reached by the rules that are on.

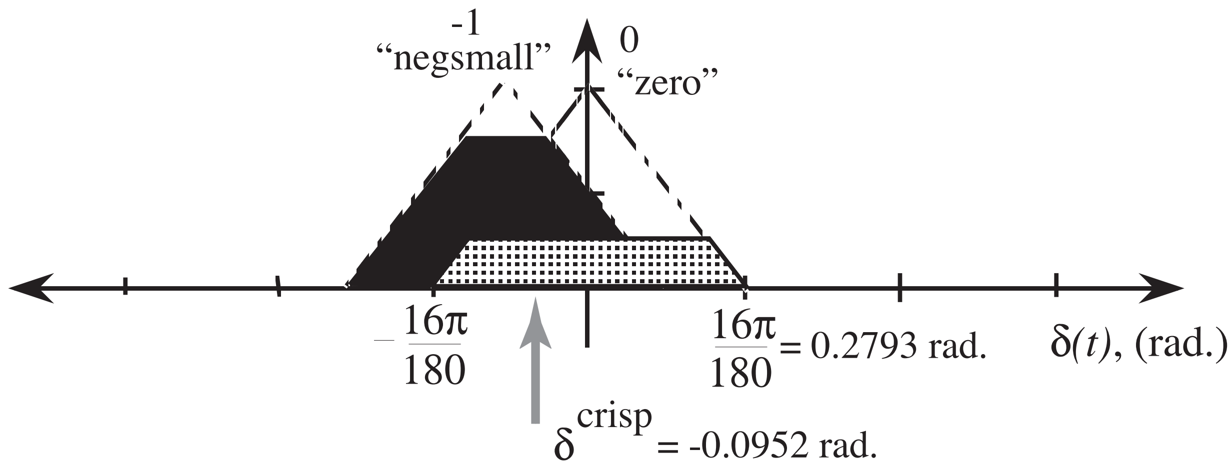

Figure 71: Implied fuzzy sets.

- Notice that the value of δ^{crisp} is roughly in the middle of where the implied fuzzy sets say they are most certain about the value for the force input.

- In fact, recall that we had

$$e(t) = 0$$

and

$$\dot{e}(t) = 0.0015$$

so the ship is at the desired heading at this time instant but is moving counterclockwise with a small velocity; hence it makes sense to apply a small negative rudder input, and the fuzzy controller does this.

- It is interesting to note that for our example it will be the case that

$$-\frac{8\pi}{18} \leq \delta^{crisp} \leq \frac{8\pi}{18}$$

- To see this, consider Figure 72, where we have drawn the output membership functions.

- Notice that even though we have extended the membership functions at the outermost edges past $-8\pi/18$ and $+8\pi/18$ (see the shaded regions), the COG method will never compute a value outside this range.

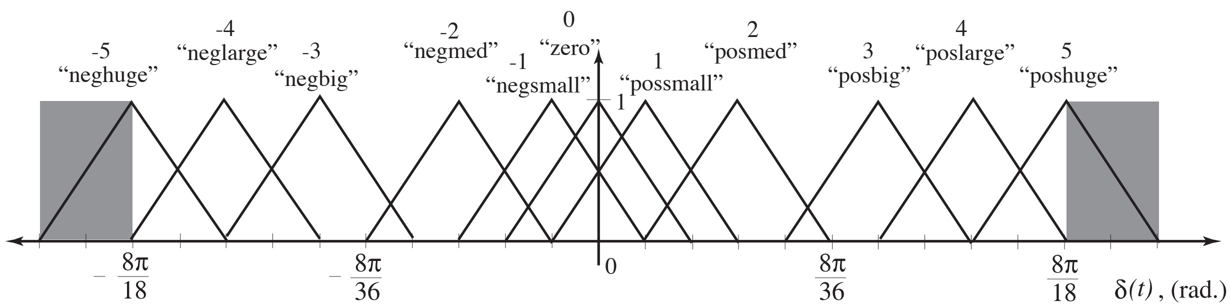

Figure 72: Output membership functions.

Other Ways to Compute and Combine Recommendations

- Consider the use of the product for representing the premise.
- Consider Figure 73, where we have drawn the output membership functions for "negsmall" and "zero" as dotted lines.

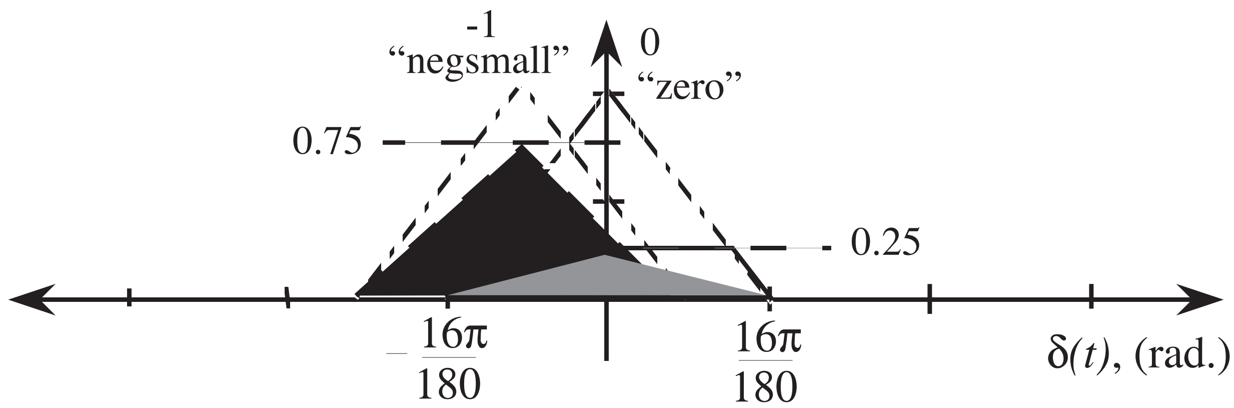

Figure 73: Implied fuzzy sets when the product is used to represent the implication.

- The implied fuzzy set from rule (1) is given by the membership function

$$\mu_{(1)}(\delta) = 0.25\mu_{zero}(\delta)$$

shown in Figure 73 as the shaded triangle; and the implied fuzzy set for rule (2) is given by the membership function

$$\mu_{(2)}(\delta) = 0.75\mu_{negsmall}(\delta)$$

shown in Figure 73 as the dark triangle.

- Notice that computation of the COG is easy since we can use $\frac{1}{2}wh$ as the area for a triangle with base width w and height h (and the factor $\frac{1}{2}w$ cancels in Equation 5).

- When we use product to represent the implication, we obtain

$$\delta^{crisp} = \frac{(0)(0.25) + \left(-0.1\frac{2\pi}{18}\right)(0.75)}{0.25 + 0.75} = -0.1047$$

which also makes sense.

➙ Next, as another example of how to combine recommendations, we will introduce the "center-average" method for defuzzification.

- For this method we let

$$\delta^{crisp} = \frac{\sum_i b_i \mu_{premise_{(i)}}}{\sum_i \mu_{premise_{(i)}}} \tag{6}$$

where b_i once again denotes the center of the membership function for the implied fuzzy set for the i^{th} rule (i.e., where the membership function for the i^{th} rule reaches its peak for our example since the output fuzzy sets are all symmetric about their peaks).

- To compute the $\mu_{premise_{(i)}}$ we use, for example, minimum.

➙ Basically, the center-average method replaces the areas of the implied fuzzy sets that are used in COG with the values of $\mu_{premise_{(i)}}$.

- For the above example, we have

$$\delta^{crisp} = \frac{(0)(0.25) + \left(-0.1\frac{8\pi}{18}\right)(0.75)}{0.25 + 0.75} = -0.1047$$

which is the same value as above (for this special case).

- Some like the center-average defuzzification method because the computations needed are generally simpler than for COG because when the output membership functions are symmetric (the usual case) they are easy to store since the only relevant information they provide is their center values (b_i) Moreover, the areas of the implied fuzzy sets do not have to be computed.

Graphical Depiction of Fuzzy Decision Making

★ For convenience, we summarize the procedure that the fuzzy controller uses to compute its outputs given its inputs in Figure 74.

- Here, we use the minimum operator to represent the "and" in the premise and the implication and COG defuzzification.

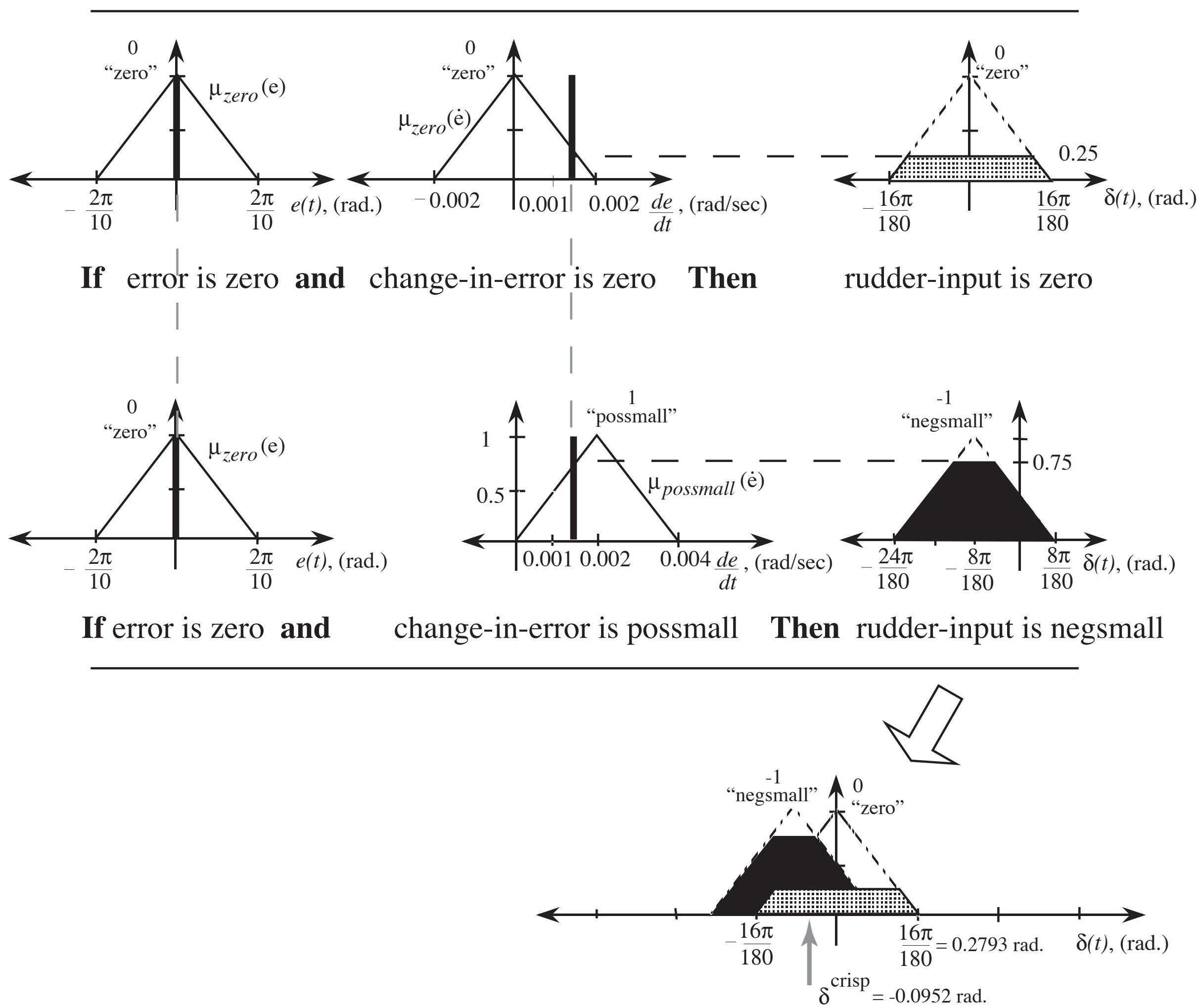

Figure 74: Graphical representation of fuzzy controller operations.

Multiple Input Multiple Output Fuzzy Systems

- A general multiple input multiple output (MIMO) fuzzy system with inputs u_i, $i = 1, 2, \ldots, n$ and outputs y_j, $j = 1, 2, \ldots, m$ is shown in Figure 75.

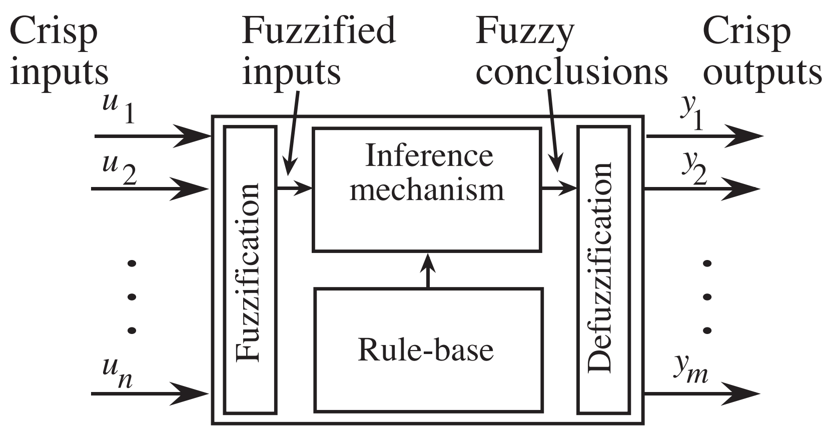

Figure 75: Fuzzy system (controller).

➙ To define a MIMO fuzzy system you simply specify m multiple input single output (MISO) fuzzy systems, where the output of the j^{th} fuzzy system is y_j, $j = 1, 2, \ldots, m$.

- All we need to do is explain how to define a MISO fuzzy system with $n > 2$ inputs (then the case for $n = 1$ will be clear).
- For each input we define membership functions as we did for e and $\dot{e}$ for the ship example.
- You form rules using the n inputs in n premise terms. Fuzzification is the same as earlier.
- Compute the fuzzy logic quantification of the conjunction between n premise terms rather than just two.
- Use the same approach as before, but take the minimum (or product) of n membership function values to represent the conjunction of n premise terms.
- Provides $\mu_{premise_{(i)}}$ for the i^{th} rule.
- From this point on the process is exactly the same as the two-input case (the inference mechanism computations of the

implied fuzzy sets and the defuzzification computations only depend on $\mu_{premise_{(i)}}$).

Takagi-Sugeno Fuzzy Systems

- The fuzzy systems discussed in the previous sections will be referred to as a "standard fuzzy system" ("Mamdani-type").
- Here, we will define a "functional fuzzy system," of which the Takagi-Sugeno fuzzy system is a special case.
- For the functional fuzzy system, we use singleton fuzzification and the premise is defined the same as for the standard fuzzy system.
- In the consequent we use a function $b_i = g_i(\cdot)$ that does not have an associated membership function.
- Notice that often the argument of g_i contains the fuzzy system inputs that are used in the premise of the rule, but other

variables may also be used.

- You may want to choose

$$b_i = g_i(\cdot) = a_{i,0} + a_{i,1}(u_1)^2 + \cdots + a_{i,n}(u_n)^2$$

or

$$b_i = g_i(\cdot) = \exp\left[a_{i,1}sin(u_1) + \cdots + a_{i,n}sin(u_n)\right]$$

- The output is

$$y = \frac{\sum_{i=1}^{R} b_i \mu_i(z)}{\sum_{i=1}^{R} \mu_i(z)} \tag{7}$$

where $\mu_i(z)$ is the premise membership function and θ is the vector of parameters that define the system (z is the premise input).

- In the special case where $z = [u_1, \ldots, u_n]^\top$

$$b_i = g_i(\cdot) = a_{i,0} + a_{i,1}u_1 + \cdots + a_{i,n}u_n$$

(where the $a_{i,j}$ are fixed real numbers) the functional fuzzy system is referred to as a "Takagi-Sugeno fuzzy system."

➙ It performs a nonlinear interpolation between linear mappings.

- The inputs to the premise and consequent terms can be different.

Mathematical Representations of Fuzzy Systems

Rules and Membership Functions

- To represent linguistic rules, let $\tilde{u}_i$, $i = 1, 2, \ldots, n$, and $\tilde{y}$ denote the linguistic variables that describe u_i, $i = 1, 2, \ldots, n$, and y, respectively.
- Let $\tilde{A}_i^j$ denote the j^{th} linguistic value for the i^{th} input universe of discourse (here, suppose that $i = 1, 2, \ldots, n$, but that j can, for instance, take on values that are equal to the linguistic-numeric values).

- Similarly, let $\tilde{B}^p$ denote the p^{th} linguistic value on the output universe of discourse that has linguistic variable $\tilde{y}$.
- With this, a linguistic rule may be described mathematically by

$$\begin{gathered}\textbf{If } \tilde{u}_1 \text{ is } \tilde{A}_1^j \\ \textbf{and } \tilde{u}_2 \text{ is } \tilde{A}_2^k \\ \textbf{and } \cdots \\ \textbf{and } \tilde{u}_n \text{ is } \tilde{A}_n^l \\ \textbf{Then } \tilde{y} \text{ is } \tilde{B}^p\end{gathered}$$

- Suppose that there are R such rules.
- Next, consider the mathematical quantification of membership functions.
- See Tables 4 and 5 for a mathematical characterization of the triangular and Gaussian membership functions (left, right, and center).

Table 4: Mathematical Characterization of Triangular Membership Functions

	Triangular, and related membership functions
Left	$\mu^L(u) = \begin{cases} 1 & \text{if } u \le c^L \\ \max\left\{0, 1 + \frac{c^L - u}{0.5w^L}\right\} & \text{otherwise} \end{cases}$
Centers	$\mu^C(u) = \begin{cases} \max\left\{0, 1 + \frac{u-c}{0.5w}\right\} & \text{if } u \le c \\ \max\left\{0, 1 + \frac{c-u}{0.5w}\right\} & \text{otherwise} \end{cases}$
Right	$\mu^R(u) = \begin{cases} \max\left\{0, 1 + \frac{u-c^R}{0.5w^R}\right\} & \text{if } u \le c^R \\ 1 & \text{otherwise} \end{cases}$

Table 5: Mathematical Characterization of Gaussian Membership Functions

	Gaussian, and related membership functions
Left	$\mu^L(u) = \begin{cases} 1 & \text{if } u \leq c^L \\ \exp\left(-\frac{1}{2}\left(\frac{u-c^L}{\sigma^L}\right)^2\right) & \text{otherwise} \end{cases}$
Centers	$\mu(u) = \exp\left(-\frac{1}{2}\left(\frac{u-c}{\sigma}\right)^2\right)$
Right	$\mu^R(u) = \begin{cases} \exp\left(-\frac{1}{2}\left(\frac{u-c^R}{\sigma^R}\right)^2\right) & \text{if } u \leq c^R \\ 1 & \text{otherwise} \end{cases}$

Parameterization in Terms of Rules

- For the sake of illustration suppose we only use membership functions of the "center" Gaussian form in Table 5 (complicates representation otherwise).

- For the i^{th} rule, suppose that the input membership function is

$$\exp\left(-\frac{1}{2}\left(\frac{u_j - c_j^i}{\sigma_j^i}\right)^2\right)$$

- If we let b_i, $i = 1, 2, \ldots, R$, denote the center of the output membership function for the i^{th} rule, use center-average defuzzification, and product to represent the conjunctions in the premise, then

$$y = \frac{\sum_{i=1}^{R} b_i \prod_{j=1}^{n} \exp\left(-\frac{1}{2}\left(\frac{u_j - c_j^i}{\sigma_j^i}\right)^2\right)}{\sum_{i=1}^{R} \prod_{j=1}^{n} \exp\left(-\frac{1}{2}\left(\frac{u_j - c_j^i}{\sigma_j^i}\right)^2\right)} \tag{8}$$

is an explicit representation of a fuzzy system.

Relationships Between Neural Networks and Fuzzy Systems

★ There are two ways in which there are relationships between fuzzy systems and neural networks.

- Techniques from one area can be used in the other (e.g., training nonlinear maps).
- In some cases the functionality (i.e., the nonlinear function that they implement) is identical (note that for the RBF with a weighted average output computation if $R_i(x) = \mu_i(x)$ then we get one of the standard forms for a fuzzy system).

• Some label the intersection between fuzzy systems and neural networks with the term "fuzzy-neural" or "neuro-fuzzy" to highlight that techniques from both fields are being used.

★ Here, we avoid this terminology.

- Some have claimed that neural networks offer an advantage since they can be trained with data and fuzzy systems cannot—but this is false—fuzzy systems can also be trained with data as we will see.

- Fuzzy systems do facilitate the incorporation of a priori knowledge.

★ Neural networks and fuzzy systems have many differences (e.g., recurrent neural networks, fuzzy dynamical systems, etc.).

Neural vs. fuzzy (or wavelets, polynomials, etc.) is the wrong focus—the focus should be on "structure choice" as we will see later.

Design Example: Tanker Ship Steering

★ We seek to automate the decision-making process of a ship captain in regulating the heading of the ship in Figure 56.

- We have

$$\dddot{\psi}(t) + \left(\frac{1}{\tau_1} + \frac{1}{\tau_2}\right)\ddot{\psi}(t) + \left(\frac{1}{\tau_1\ \tau_2}\right)\dot{\psi}(t) = \frac{K}{\tau_1\ \tau_2}\left(\tau_3\ \dot{\delta}(t) + \delta(t)\right), \tag{9}$$

where ψ is the heading of the ship and δ is the rudder angle.

- Assuming zero initial conditions:

$$\frac{\psi(s)}{\delta(s)} = \frac{K(s\tau_3 + 1)}{s(s\tau_1 + 1)(s\tau_2 + 1)}, \tag{10}$$

where K, τ_1, τ_2, and τ_3 are parameters which are a function of the ship's constant forward velocity u and its length l.

- In particular,

$$\begin{aligned} K &= K_0 \left(\frac{u}{l}\right), \\ \tau_i &= \tau_{i0} \left(\frac{l}{u}\right) \qquad\qquad i = 1, 2, 3 \end{aligned}$$

- Under "ballast" conditions, $K_0 = 5.88$, $\tau_{10} = -16.91$, $\tau_{20} = 0.45$, $\tau_{30} = 1.43$, and $l = 350$ meters.
- The tanker dynamics change due to weight changes. Under "full" conditions we have $K_0 = 0.83$, $\tau_{10} = -2.88$, $\tau_{20} = 0.38$, $\tau_{30} = 1.07$,
- We have $l = 350$ meters and we will assume that nominally the ship is traveling in the x direction at a velocity of $5m/s$.
- For the above model the rudder angle should not exceed approximately 5 degrees otherwise it will be inaccurate.
- We need a model which is suited for rudder angles which are

larger than 5 degrees; one such model is given by

$$\dddot{\psi}(t)+\left(\frac{1}{\tau_1}+\frac{1}{\tau_2}\right)\ddot{\psi}(t)+\left(\frac{1}{\tau_1\ \tau_2}\right)H(\dot{\psi}(t))=\frac{K}{\tau_1\ \tau_2}\left(\tau_3\ \dot{\delta}(t)+\delta(t)\right), \tag{11}$$

where $H(\dot{\psi})$ is a nonlinear function of $\dot{\psi}(t)$.

- An experiment known as the "spiral test" has shown that $H(\dot{\psi})$ can be approximated by

$$H(\dot{\psi})=\bar{a}\dot{\psi}^3+\bar{b}\dot{\psi}$$

where $\bar{a}$ and $\bar{b}$ are real valued constants such that $\bar{a}$ is always positive (we pick $\bar{a}=1$ and $\bar{b}=1$).

- Also, the rudder cannot move past ± 80 degrees (a saturation nonlinearity).

- There is a wind disturbance $w(t)$ that is possible which is modeled by letting the input to the ship be the output generated by the controller (δ) plus the wind disturbance,

$$\bar{\delta} = \delta + w(t)$$

$$w(t) = \frac{0.5\pi}{180}\sin(2\pi(0.001)t)$$

- Simulate ship as a continuous time system with an integration step size of 1. Simulate controller as a digital system with a sampling period of $T = 10$.
- Consider control system shown in in Figure 76.

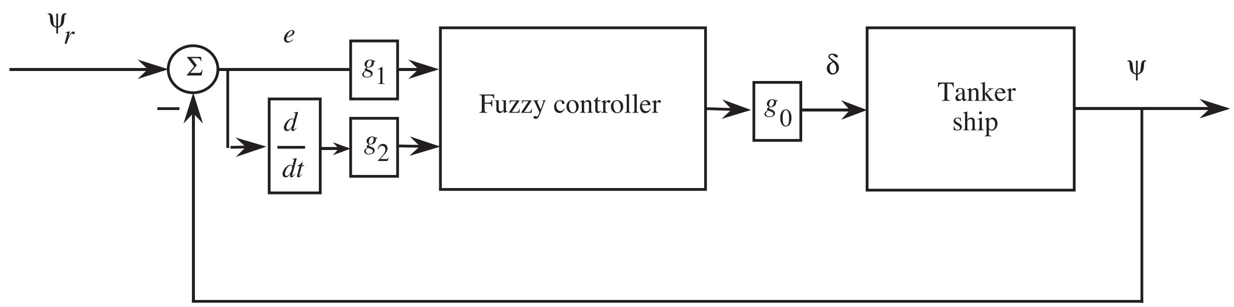

Figure 76: Fuzzy controller for tanker ship with scaling gains g_0, g_1, and g_2.

➙ First, "normalize" the input and output universes of discourse.

- Here, we simply change the membership functions to those shown in Figure 77 (i.e., normalize to an interval $[-1, 1]$).
- With the scaling gains in Figure 77, implemented as in Figure 76, we implement the membership functions in Figure 63.

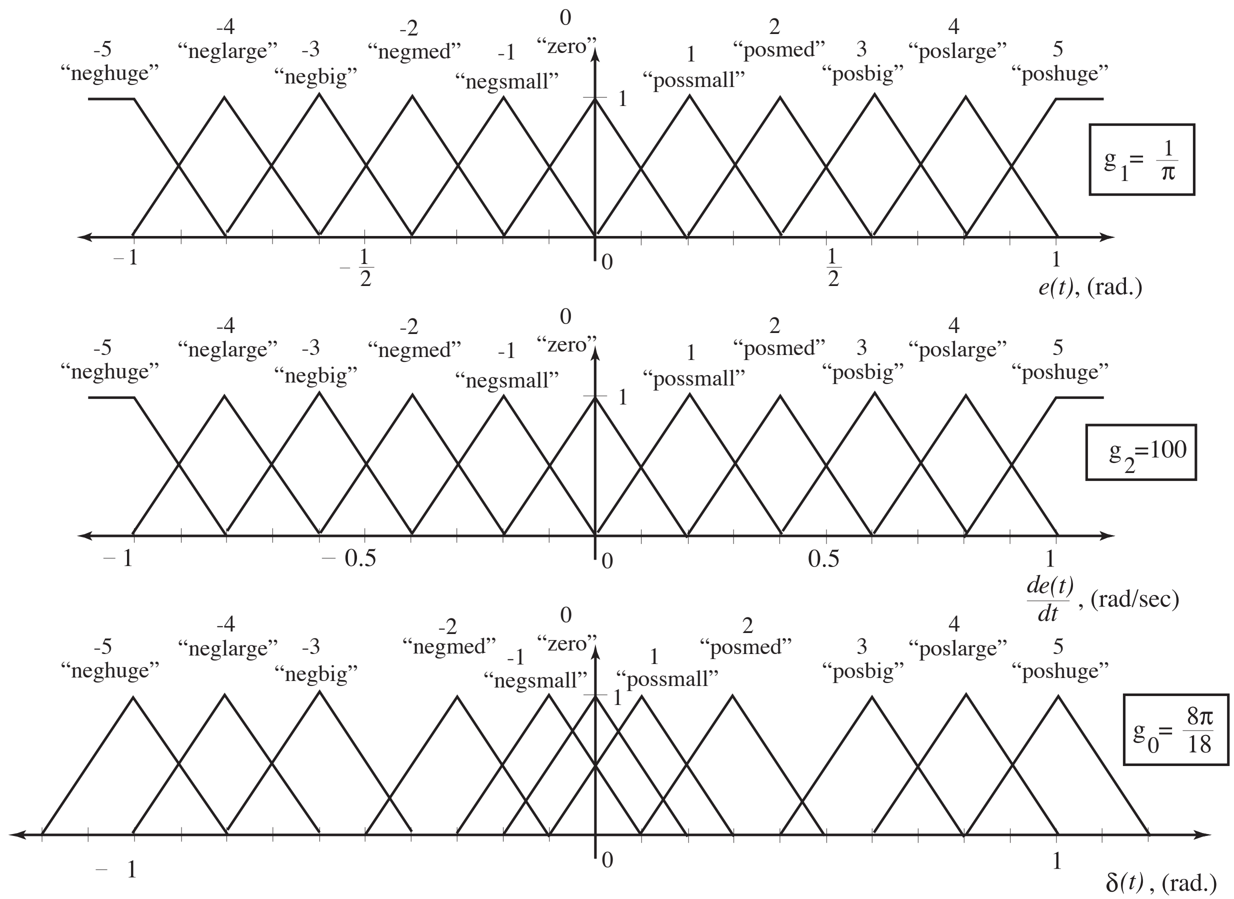

Figure 77: Normalized universes of discourse for fuzzy controller for tanker ship (and boxed values of the scaling gains give the original membership functions shown in Figure 63.)

Design Example: Tanker Ship Steering

★ There is no general systematic procedure for the design of fuzzy controllers that will definitely produce a high-performance fuzzy control system for a wide variety of applications

$\boxed{\text{Learn design via applications}}$

Performance for the First Guess:

- The closed-loop response, using the design we developed, is shown in Figure 78

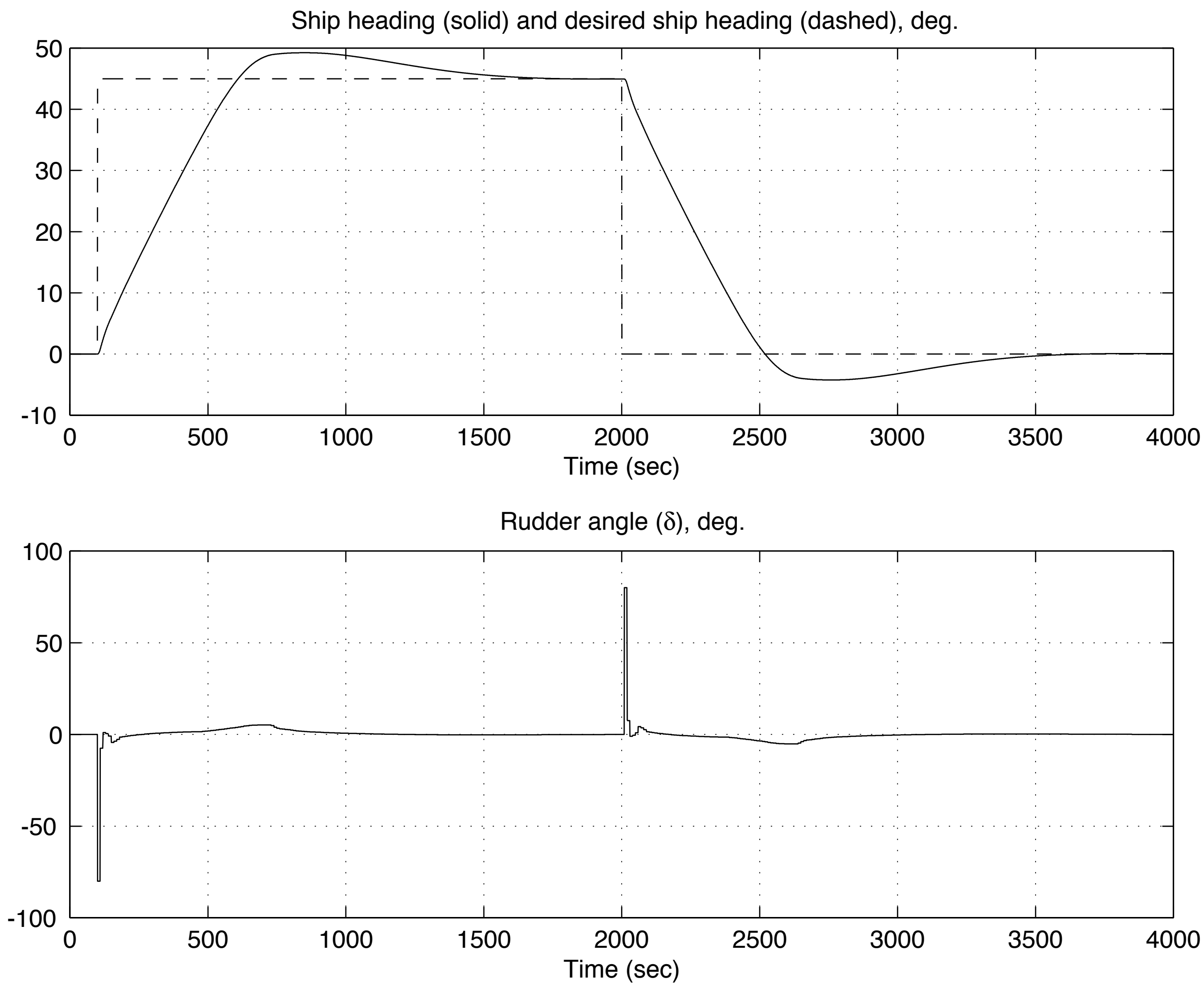

Figure 78: Response of fuzzy controller for tanker ship steering, $g_0 = \frac{8\pi}{18}$, $g_1 = \frac{1}{\pi}$, and $g_2 = 100$.

★ Note that while the response is at least tracking the step changes eventually, there is a significant amount of overshoot.

Tuning the Derivative Gain to Reduce Overshoot:

- To reduce the overshoot, we should increase the gain on the derivative term (so that the controller gets more capability to "predict where the response is going").
- To do this we choose $g_0 = \frac{8\pi}{18}$, $g_1 = \frac{1}{\pi}$, and $g_2 = 200$ and get the response in Figure 79, where we see that we have indeed speeded up the response.

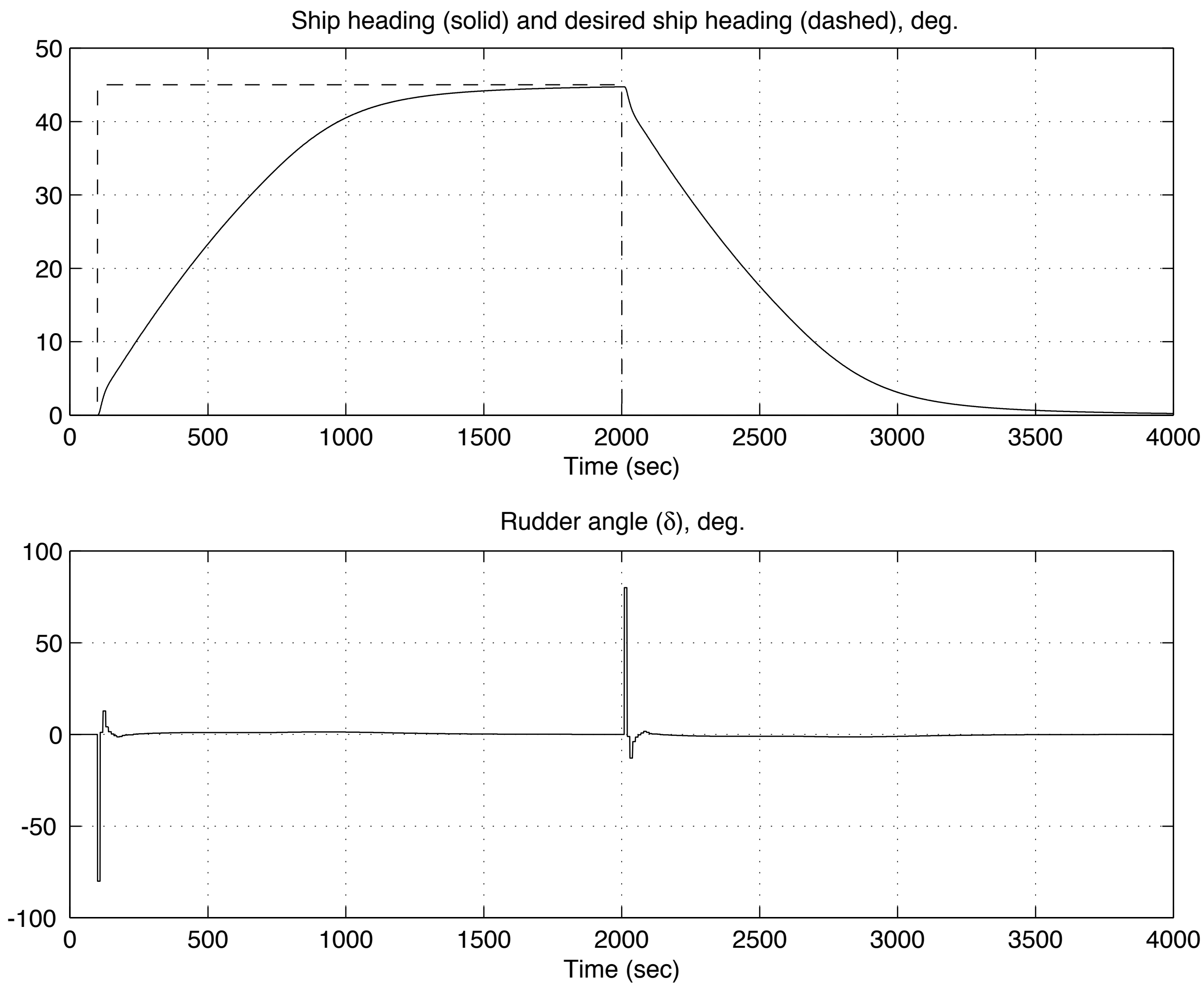

Figure 79: Response of fuzzy controller for tanker ship steering, $g_0 = \frac{8\pi}{18}$, $g_1 = \frac{1}{\pi}$, and $g_2 = 200$.

★ Unfortunately, however, this also reduced the response time of the system (i.e., it "slowed" the system).

Tuning the Proportional Gain to Decrease the Response Time: Finding "Good" Scaling Gains:

- Next, we seek to choose a good set of scaling gains by speeding up the response from the previous case.
- To do this we increase the gain on the proportional term so that we increase the speed of the response and hence reduce the response time.
- When we do this, however, this can cause some overshoot, so we also increase the gain on the derivative term to avoid that.
- In particular, choose $g_0 = \frac{8\pi}{18}$, $g_1 = \frac{2}{\pi}$, and $g_2 = 250$ to get a faster response with very little overshoot as seen in Figure 80.

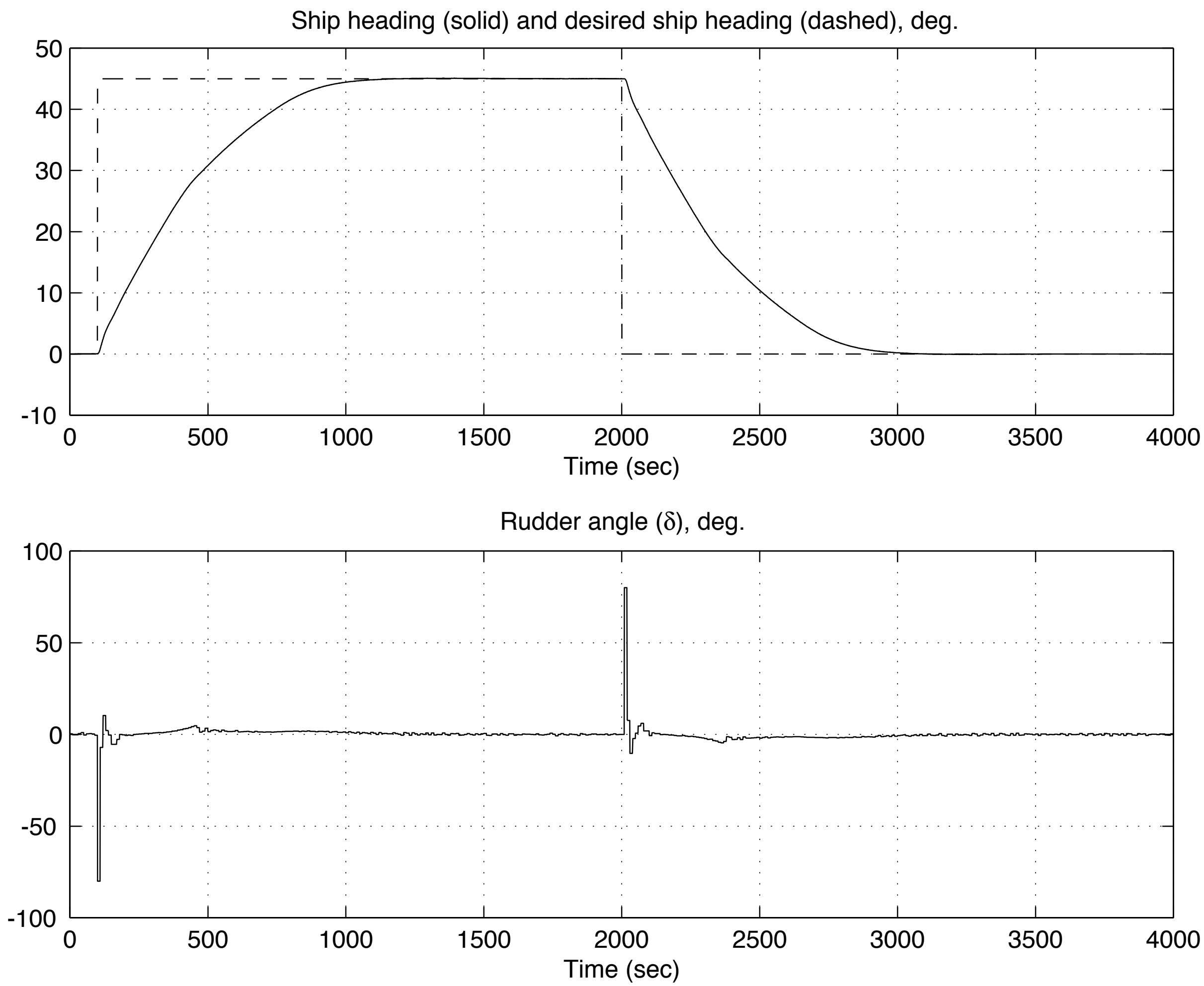

Figure 80: Response of fuzzy controller for tanker ship steering, $g_0 = \frac{8\pi}{18}$, $g_1 = \frac{2}{\pi}$, and $g_2 = 250$.

- ★ We take this set of gains as “good” values
- ★ Notice that we achieved all our tuning via the scaling gains, although this is certainly not possible in all applications

The Resulting Nonlinear Control Surface:

- ★ The fuzzy controller implements a nonlinearity that is shown in Figure 81.
  - Notice that this surface is another way to view the captain’s expertise in ship steering

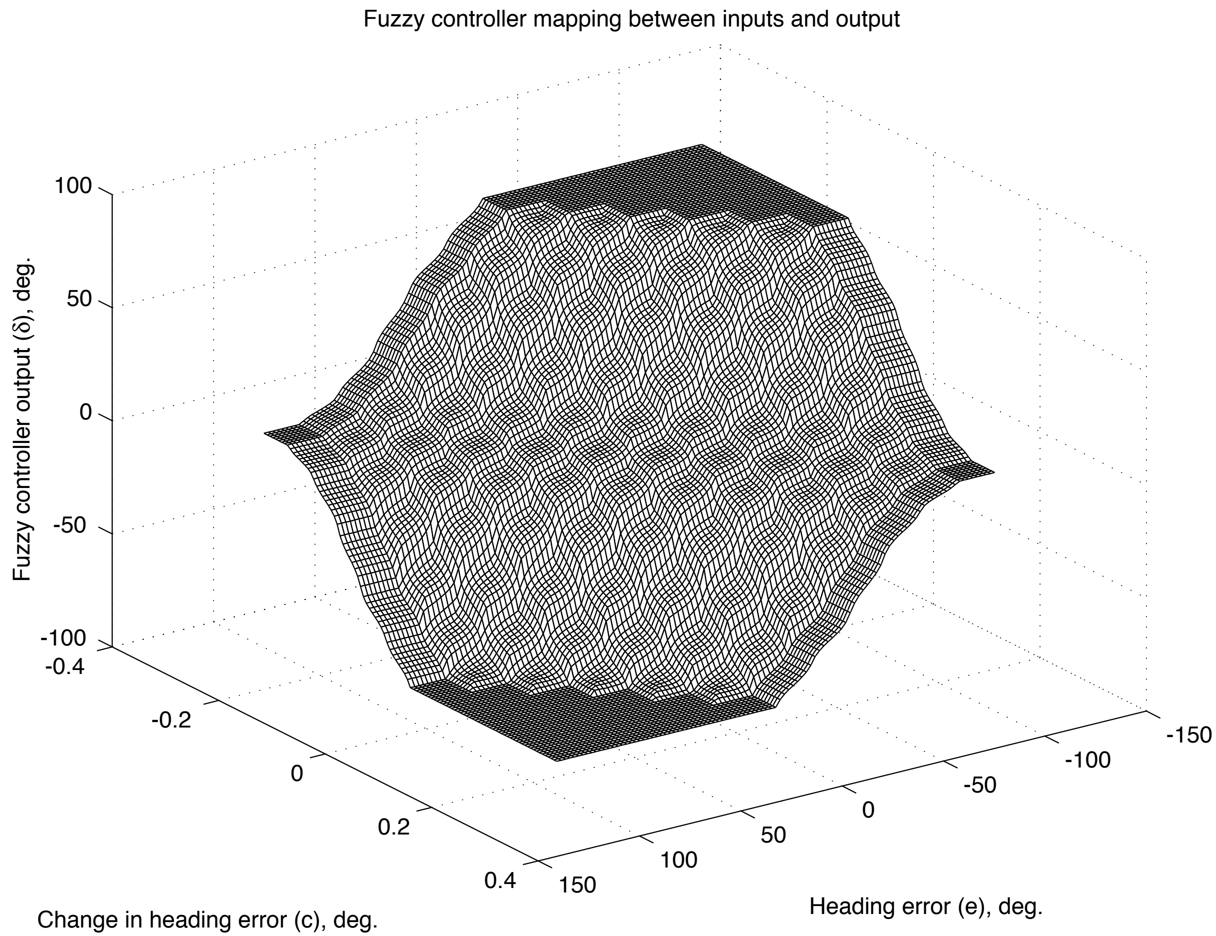

Figure 81: Nonlinear control surface implemented by the fuzzy controller, $g_0 = \frac{8\pi}{18}$, $g_1 = \frac{2}{\pi}$, and $g_2 = 250$.

- A control surface for a simple proportional-derivative (PD) controller is a plane in three dimensions.

  - There is a type of interpolation that is performed by the fuzzy controller

Design Concerns

Understand the Control Problem:

- Obtain a good understanding of the plant (understand the physics and models can help)
- Pay attention to plant constraints (actuator saturation, sensor noise, nonlinearities, disturbances, etc.)
- Develop appropriate specifications (rise time, overshoot, steady state tracking error, stability, performance robustness, etc.)
- Consider if it is possible to redesign the plant!
- Try the simplest thing (e.g., PID control).

Proper Rule Base Construction:

★ The main sources of information for rule base construction are the following:

- Interviews of human plant operators (or learning how to operate the plant yourself).
- A good understanding of the plant, the constraints imposed by it, and the closed-loop specifications that you are trying to achieve.
- Modeling and simulation studies.
- Past development of controllers for the same plant (or similar ones).
- Controller implementation studies for controllers that ultimately do not adequately achieve the specifications (e.g., the controller that you are trying to replace in updating a control system to achieve higher performance).

- There are several issues to pay attention to in rule base construction, including conflicting rules, rule base "completeness," etc.

  Reducing Controller Complexity:

- Memory and "throughput"
- There are two fundamental reasons why complexity arises in fuzzy and expert controllers:
  - Complex nonlinear maps often take many computations to implement.
  - Exponential increase in number of rules with a linear increase in the number of inputs. For example, for our ship steering problem with two inputs and eleven membership functions on each input universe of discourse there are $11^2 = 121$ possible rules.
- Methods to reduce complexity are as numerous as there are

applications

➙ General approaches to reducing complexity:

1. Sometimes some regions of the input space are not visited so the corresponding rules can be removed.
2. Sometimes you simply can get adequate performance with fewer rules. The key is to determine the minimum number of rules that still allows for the implementation of a control surface that can achieve adequate performance.

Effects of Disturbances, Noise, and Plant Changes:

★ Plant parameter variations, disturbances, and sensor noise all affect our ability to achieve good control—here, we study these for the ship.

- Note that on different journeys ships will weigh different amounts and the amount a ship weighs affects your ability to steer it.

- For the simulations up till now we have studied the case for "ballast" conditions (a very heavy ship).
- Next, we will consider the case of how the ship steers when it is under "full" conditions.
- Figure 82 shows how the fuzzy control system, that was tuned for ballast conditions, performs for full conditions.

★ We see that there now is overshoot in the ship heading since a lighter boat steers easier.

★ We see that plant parameter variations can impact performance.

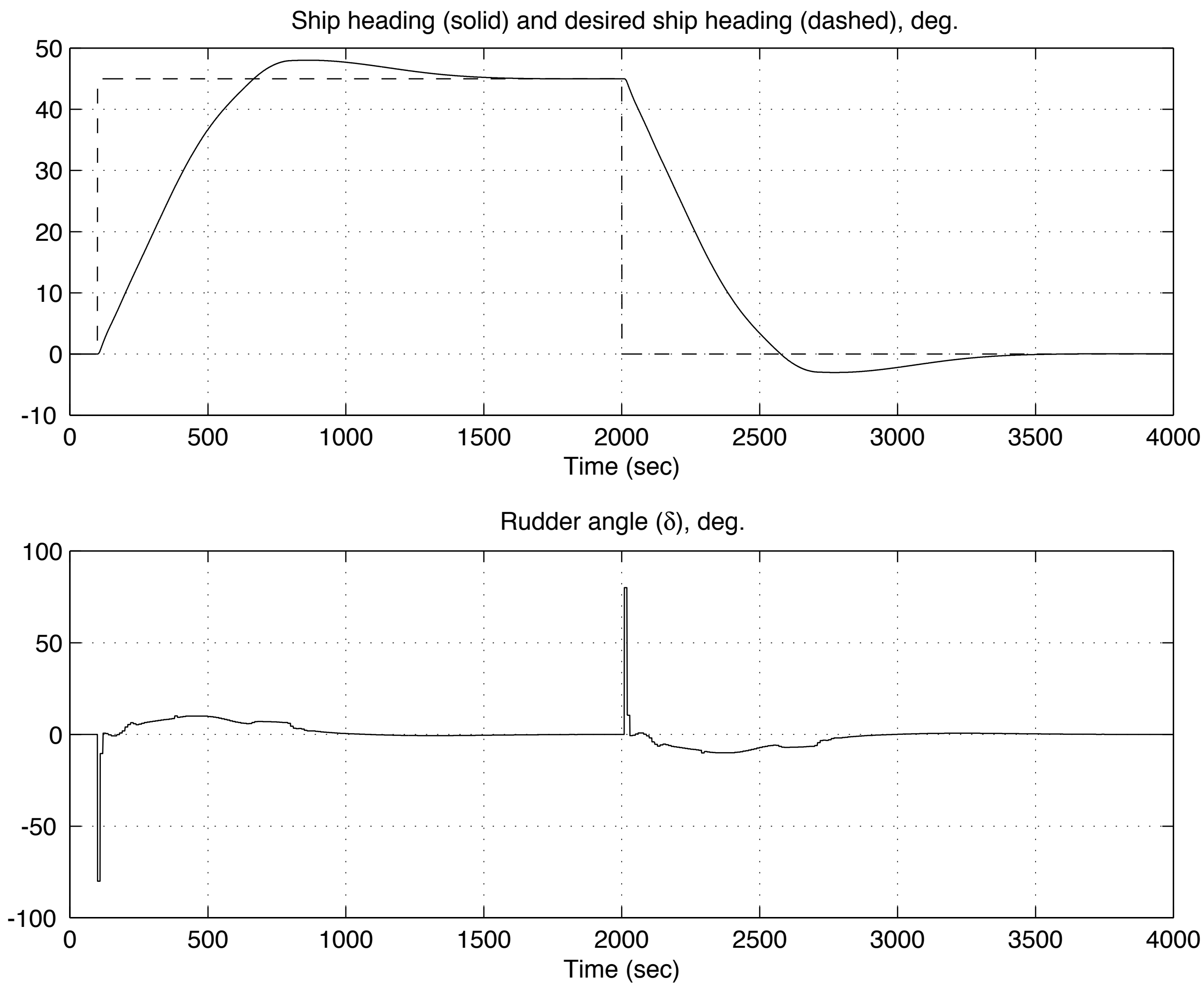

Figure 82: Response of fuzzy controller for tanker ship steering, "full" conditions, $g_0 = \frac{8\pi}{18}$, $g_1 = \frac{2}{\pi}$, and $g_2 = 250$.

★ Next, consider the effect of a wind disturbance on the ship.

- Suppose that the wind is gusting.
- It hits the side of the ship and moves the ship a bit ,which then pushes the rudder against the water which induces a torque to move the rudder.
- To model this we add a disturbance onto the rudder angle input by adding

$$0.5\left(\frac{\pi}{180}\right)\sin\left(2\pi(0.001)t\right)$$

to what the fuzzy controller commands as an input.

- In this case if we use $g_0 = \frac{8\pi}{18}$, $g_1 = \frac{2}{\pi}$, and $g_2 = 250$ (i.e., the good tuned values) we get the response in Figure 82.

★ We see that the wind affects our ability to achieve very good regulation of the ship heading.

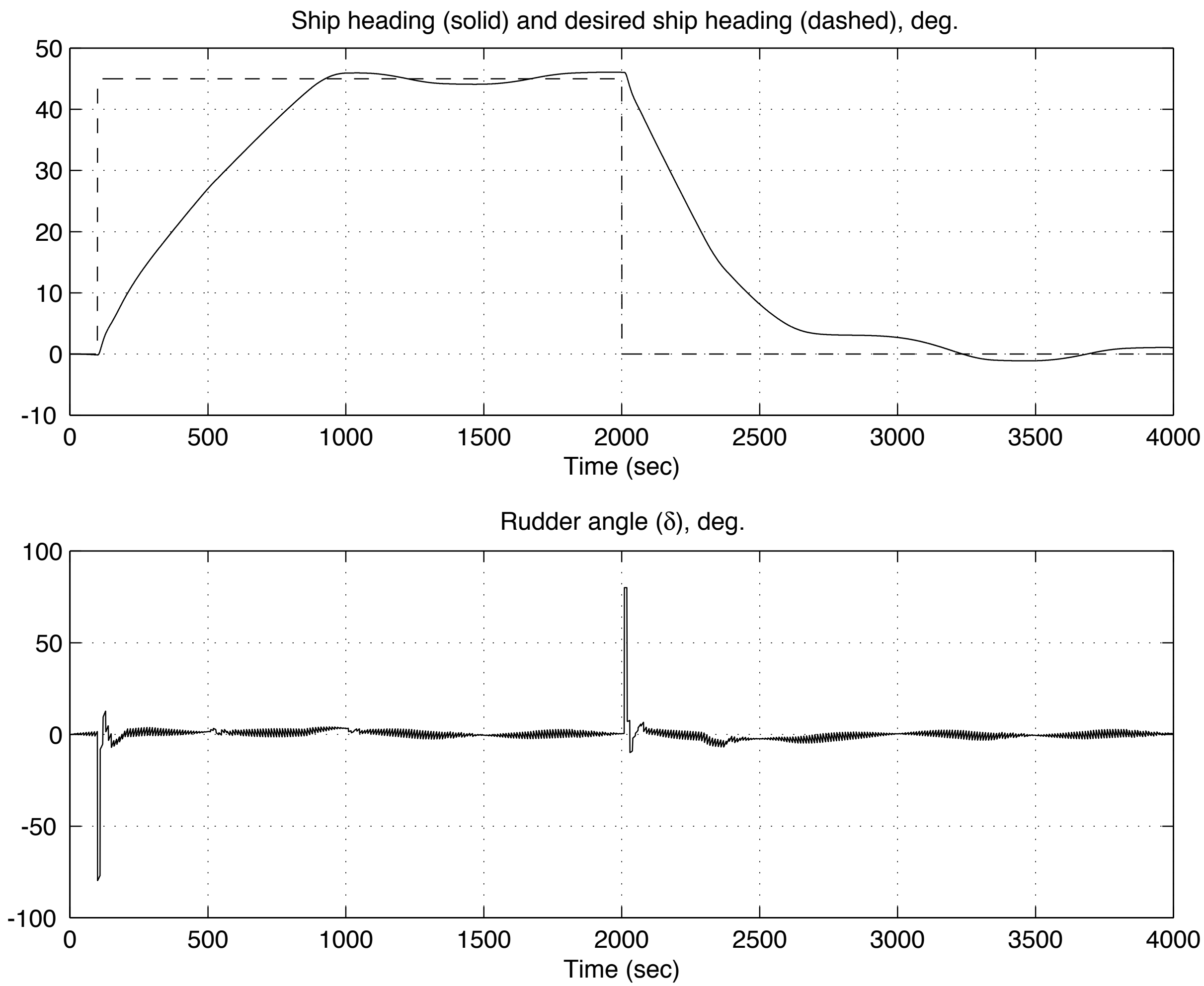

Figure 83: Response of fuzzy controller for tanker ship steering, wind disturbance, $g_0 = \frac{8\pi}{18}$, $g_1 = \frac{2}{\pi}$, and $g_2 = 250$.

★ Effects of speed on steering performance

- If we use $g_0 = \frac{8\pi}{18}$, $g_1 = \frac{2}{\pi}$, and $g_2 = 250$(i.e., the good tuned values) we get the response in Figure 84.

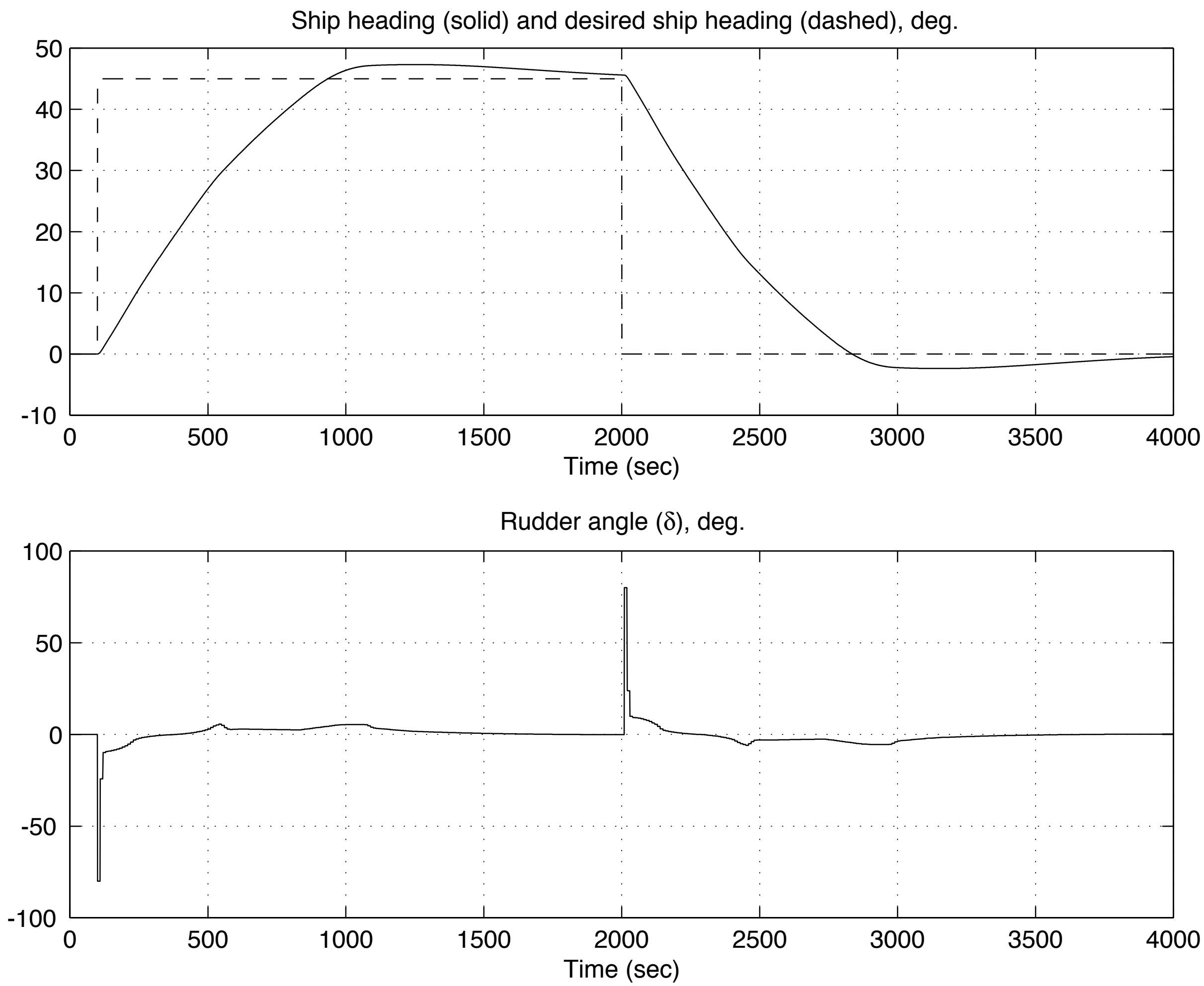

Figure 84: Response of fuzzy controller for tanker ship steering, speed decrease, $g_0 = \frac{8\pi}{18}$, $g_1 = \frac{2}{\pi}$, and $g_2 = 250$.

★ We see that the speed decrease causes a significant overshoot in the response since the rudder is not as effective in tracking.

Stability and Limit Cycles:

- For the ship if you choose $g_0 = \frac{-8\pi}{18}$, $g_1 = \frac{2}{\pi}$, and $g_2 = 250$ (notice minus sign) you can get an unstable response (in this case the controller moves the rudder in the wrong direction to try to reduce a heading error and each time it does this it creates a bigger error).

★ There are many ways to mathematically study stability properties of fuzzy control systems.

★ If you pick the wrong values of the scaling gains you can get such oscillatory behavior.

- For example, if you pick $g_0 = \frac{2000\pi}{18}$, $g_1 = \frac{2}{\pi}$, and $g_2 = 0.000001$ for the ship you get the response shown in Figure 85.

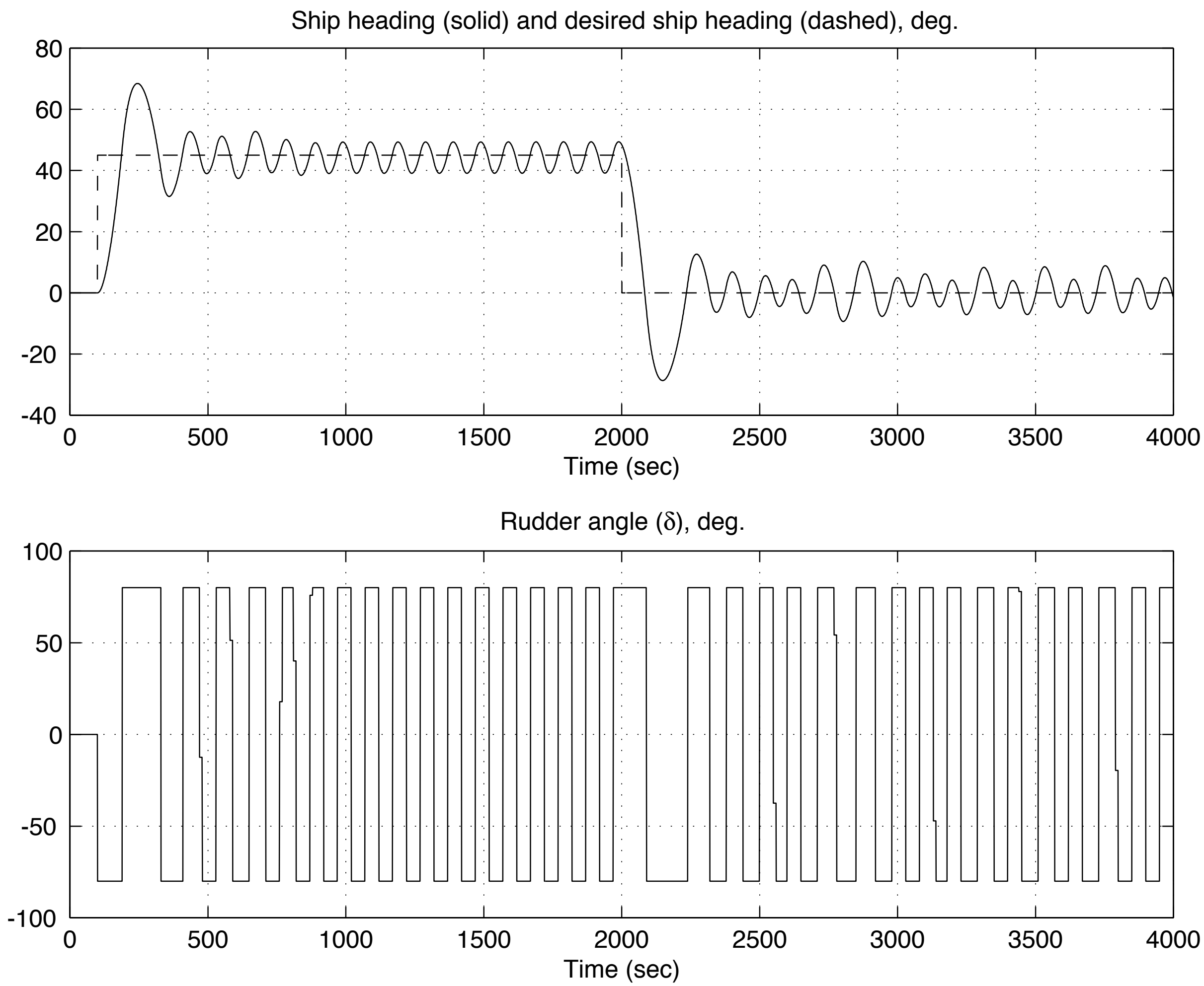

Figure 85: Response of fuzzy controller for tanker ship steering, $g_0 = \frac{2000\pi}{18}$, $g_1 = \frac{2}{\pi}$, and $g_2 = 0.000001$.

Expert Control

- An expert system that is used as a controller for a dynamic system is shown in Figure 86.
- It uses the information in its knowledge-base and its inference mechanism to decide what command input u to generate for the plant.

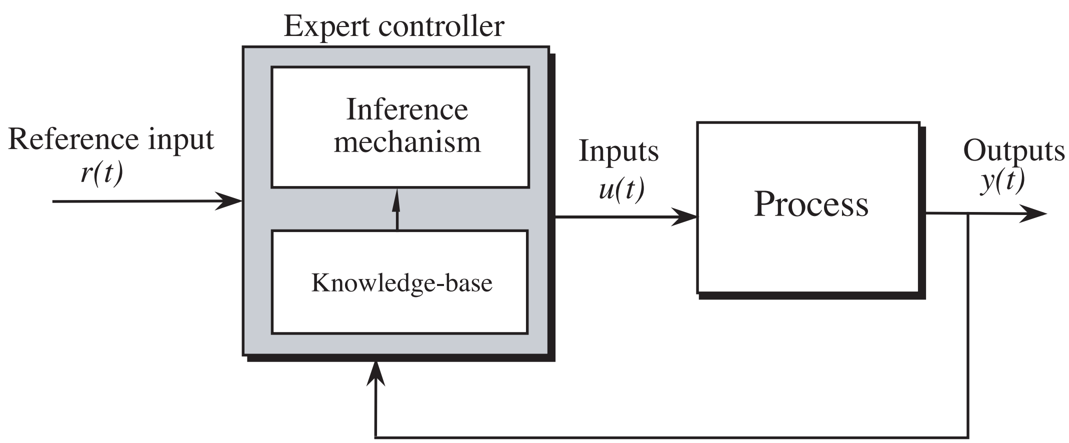

Figure 86: Expert control system.

- ★ Conceptually, the expert controller is closely related to the fuzzy controller.
- ★ Design philosophy is similar to that of the fuzzy controller.
- ★ The expert controller can use more general knowledge representation and inference strategies.
  - It can be used as a controller similar to how we used a fuzzy controller, or it can be used as a "supervisory controller" (i.e., a controller that has as outputs parameters that dicate the application of control strategies).

Planning Systems

Psychology of Planning

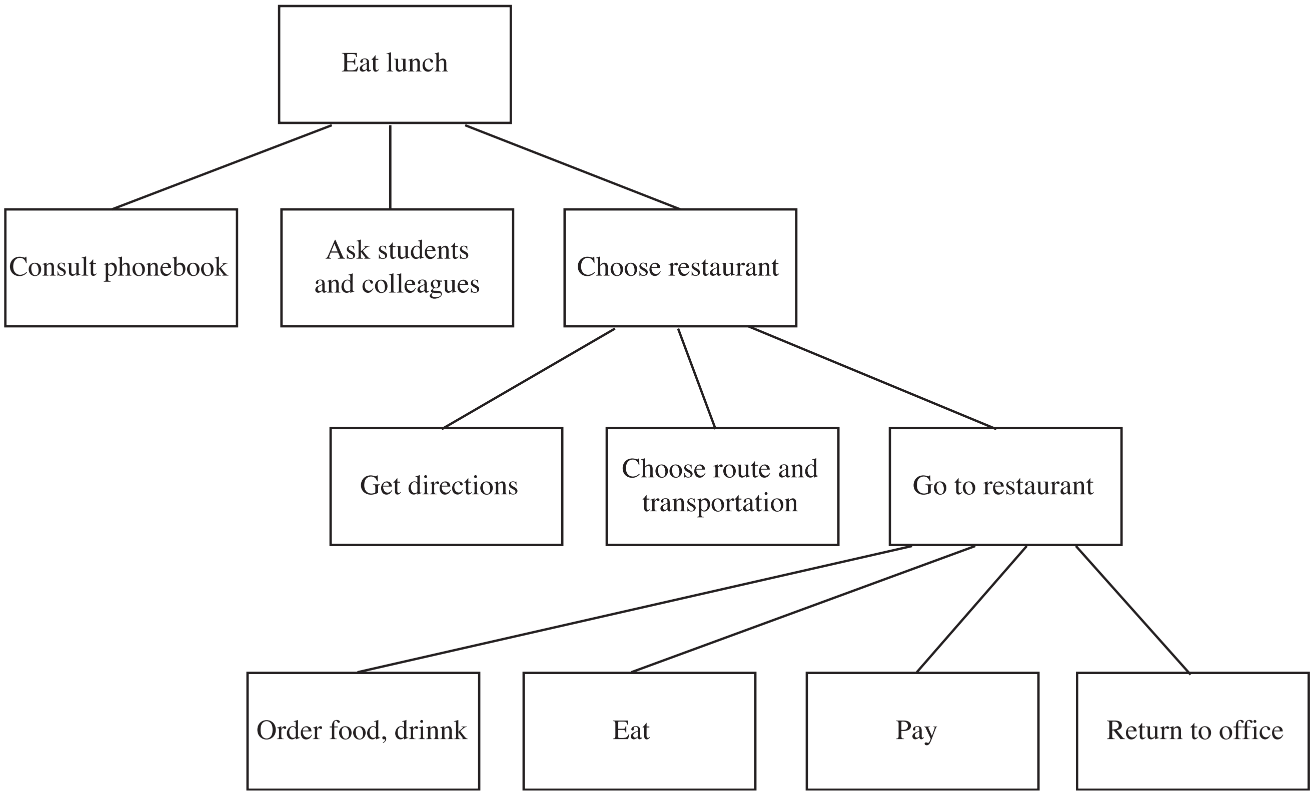

Figure 87: Action plan as an action hierarchy, an example.

- Learning and use of models for prediction is central to the activity of planning.
  - Plan over concepts, physical space ("cognitive map"), etc.
- Optimization is essential to choose which plan is "best."

Steps to Planning

- Generic planning steps:
  1. Represent the problem ("planning domain"): Many different forms, "hard-wired" knowledge or "learned" knowledge. Quality of model affects planning performance. Level of model accuracy depends on the environment and the organism.
  2. Set goal: Without goals there is no purposeful behavior. Goals determined by learning, evolution, values, ideals. Goals are often hierarchical.
  3. Decide to plan: React or plan?

4. Build a plan (select a strategy): Project using a model the alternatives and pick the "best" plan.

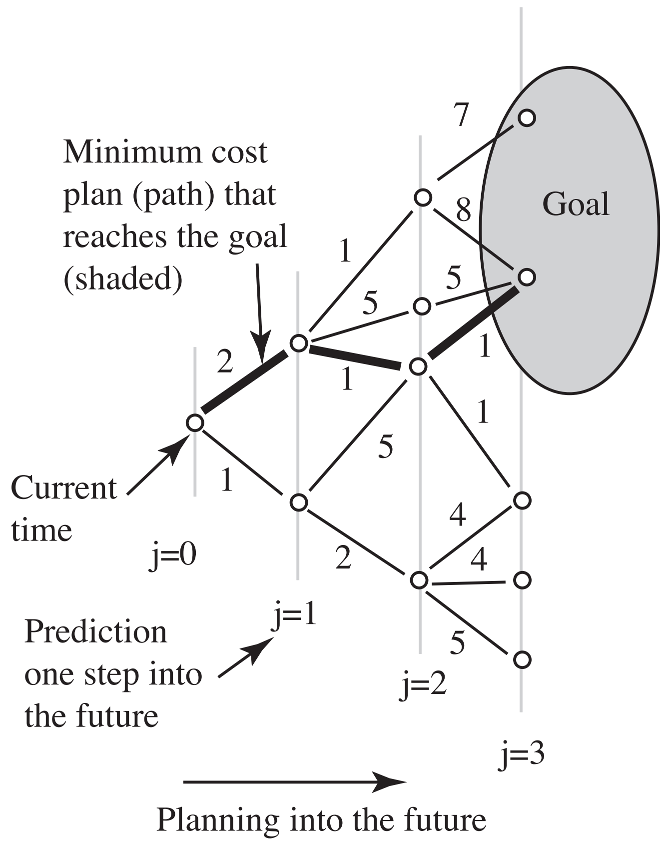

Figure 88: Tree representation of the alternative plans that can be considered at some point in time, along with their costs.

5. Execute plan, monitor, and repair/re-plan: Frequency of replanning? Monitoring depends on plant observability. "Plan failure" is possible. "Tweak" current plan or develop a new one?

Design Example: Planning for Vehicle Guidance

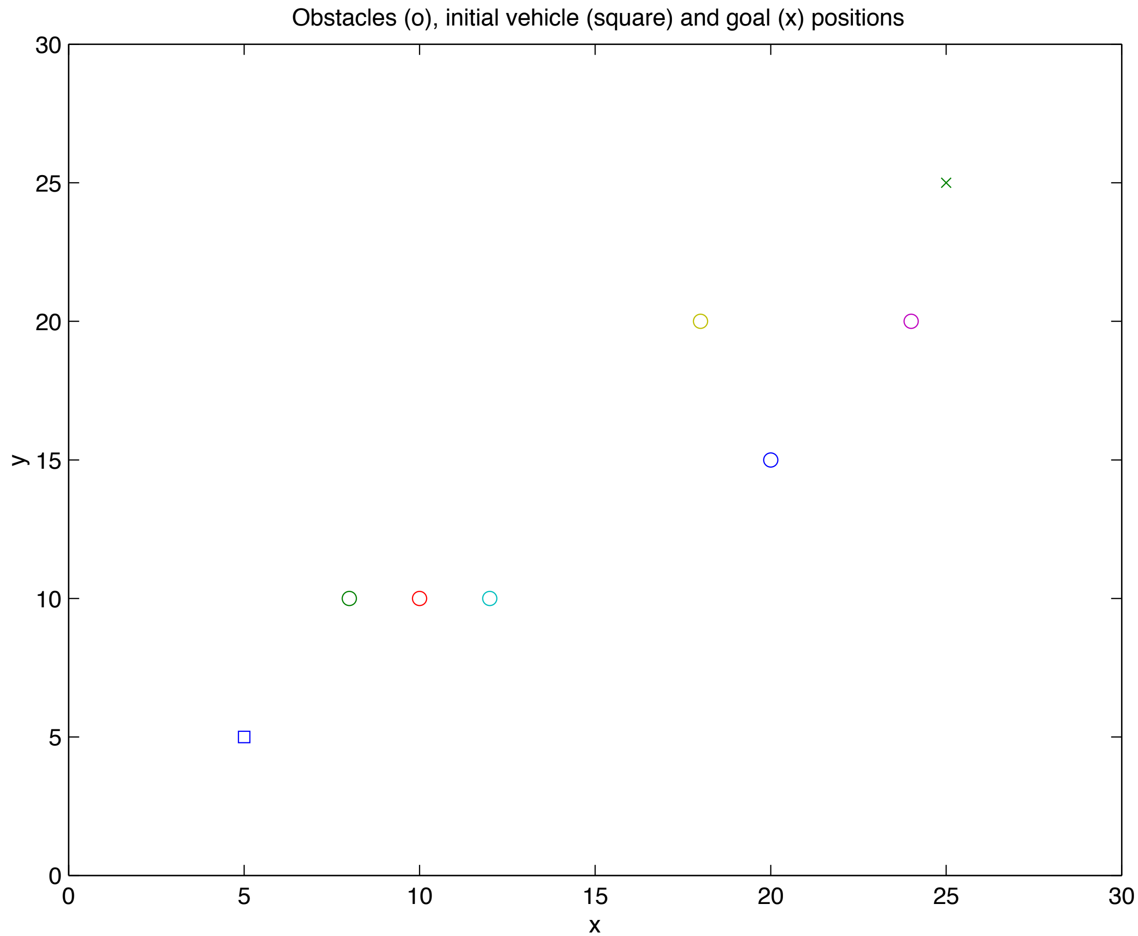

Figure 89: Initial vehicle position, goal position, and obstacles.

- Have perfect information about where obstacles (poles) are
- Vehicle knows its own position (GPS) and the goal position

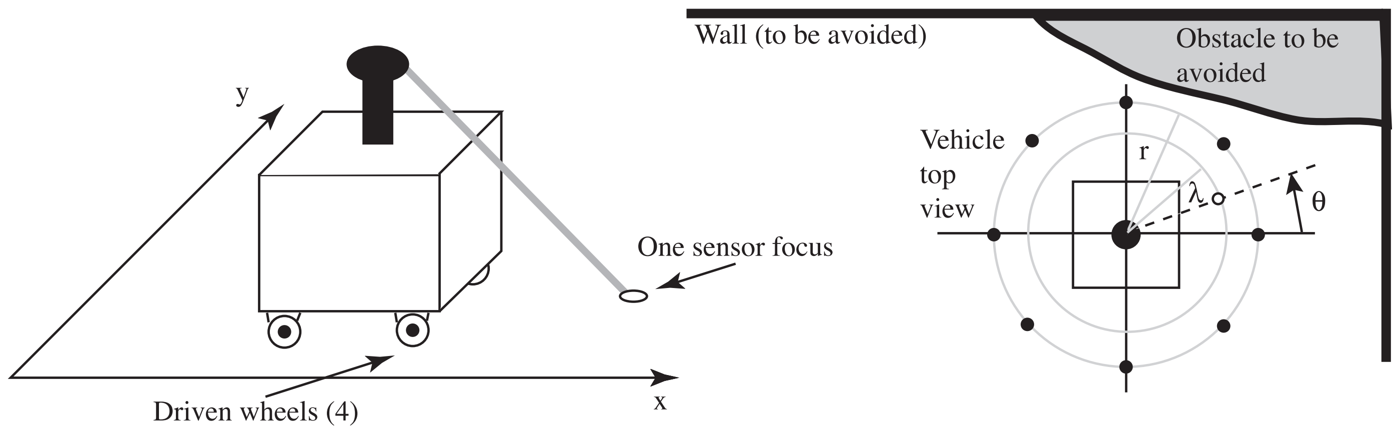

Figure 90: Autonomous vehicle guidance problem (cubical, 2.5 units per side).

➙ Vehicle's current position is $(x(k), y(k))$ and command it to move at an angle θ a distance of $\lambda = 0.1$,

$$\left[\begin{array}{c} x(k+1) \\ y(k+1) \end{array}\right] = \left[\begin{array}{c} x(k) \\ y(k) \end{array}\right] + \lambda \left[\begin{array}{c} \cos(\theta) \\ \sin(\theta) \end{array}\right] + \Delta\lambda \left[\begin{array}{c} \cos(\Delta\theta) \\ \sin(\Delta\theta) \end{array}\right]$$

- Do not perfectly achieve desired position.
- Choose $\Delta\lambda$ to be a random number chosen at each time step uniformly on $[-0.1\lambda, 0.1\lambda]$
- Assume that $\Delta\theta$ is uniformly distributed on $[-\pi, \pi]$.
- We have a perfect model of a *part* of our environment, but not all of it.

➙ Feedback control is used for guidance: the current position is sensed, and the command is made to move the vehicle to the new position—the vehicle may not end up where it was commanded to go, but at the next time instant we will sense the vehicle's position and make adjustments from that point.

Path Planning Strategy

Obstacle and Goal Functions

- ➙ Planning $\rightarrow$ we need to formulate the path-finding problem as an optimization problem.
  - Obstacles: To represent the obstacles in Figure 89 we use an "obstacle function" $J_o(x, y)$ in Figure 91
- ➙ Key idea: Move to minimize $J_o(x, y)$. Would this work?

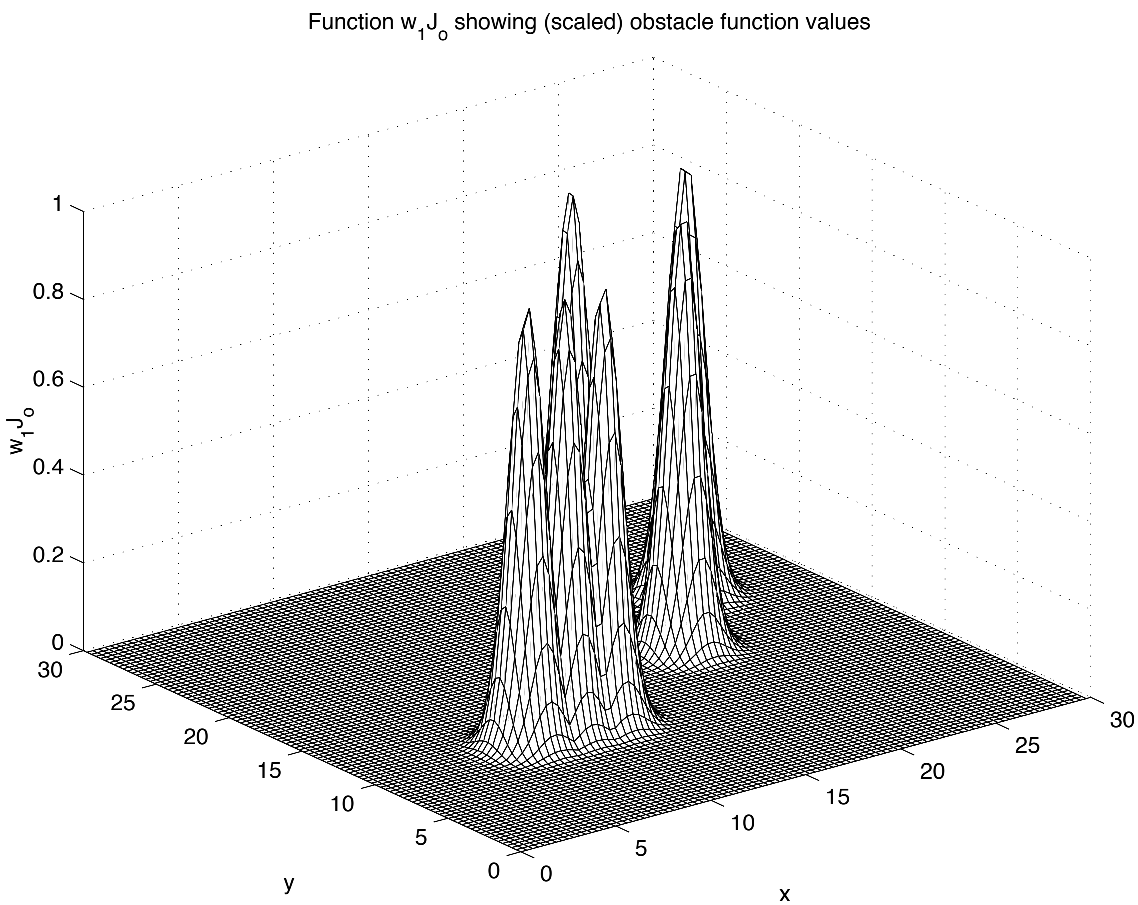

Figure 91: Obstacle function $J_o(x, y)$ (scaled by w_1).

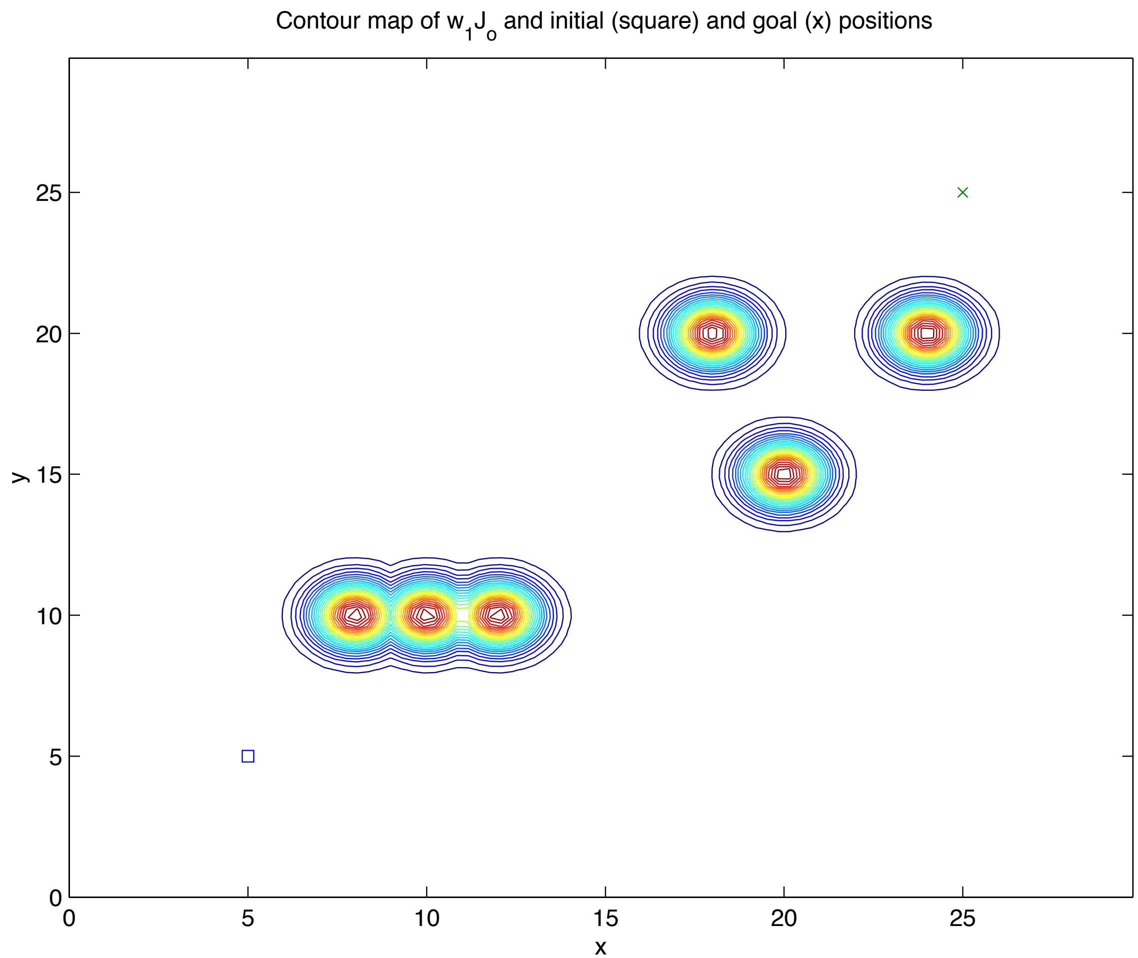

Figure 92: Obstacle function $J_o(x, y)$ (scaled), contour form, with initial vehicle position and goal position.

➙ Goal: Penalize not being at the goal position $(25, 25)$ via

$$w_2 J_g(x, y) = w_2 [[x, y]^\top - [25, 25]^\top]^\top [[x, y]^\top - [25, 25]^\top]$$

where $w_2 = 0.00010$.

- What if the vehicle moved to try to minimize this function?

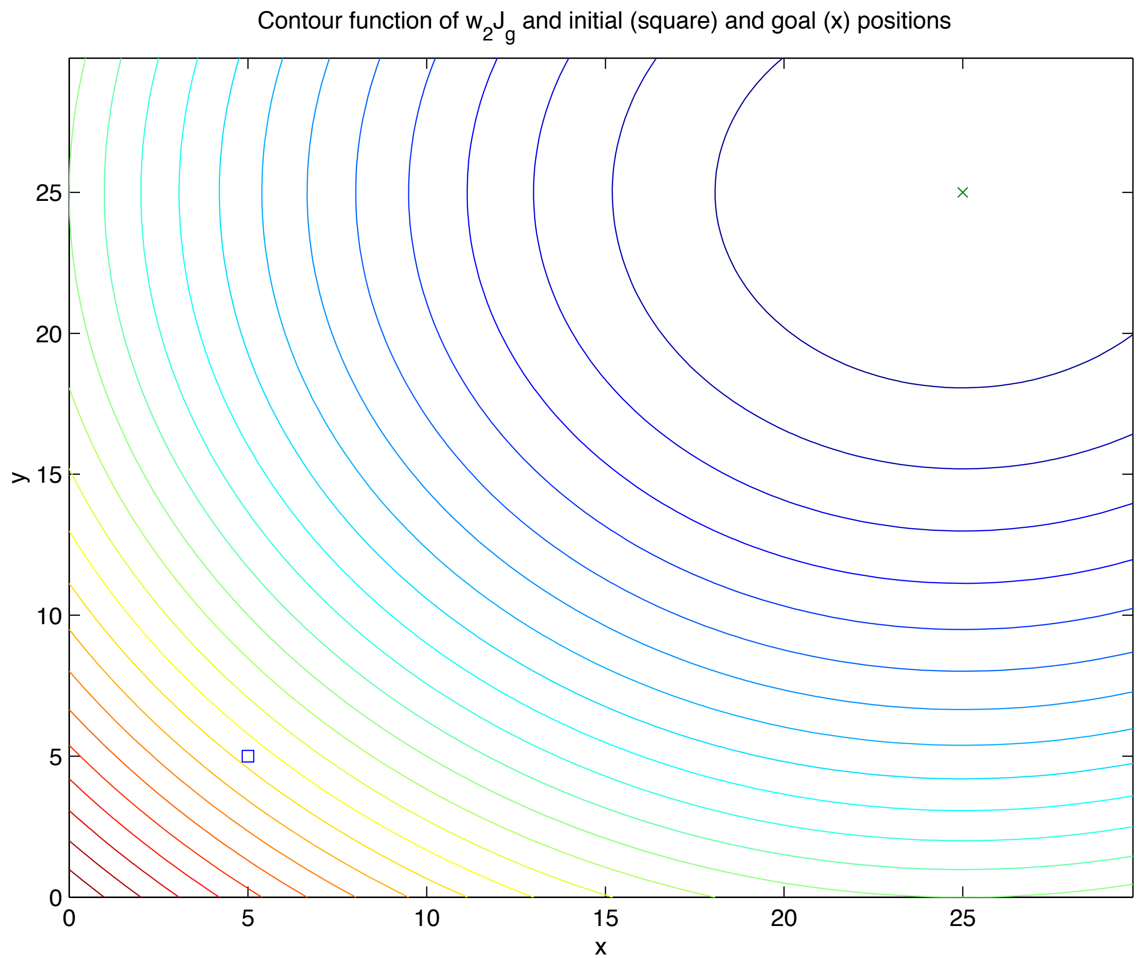

Figure 93: Goal function $w_2 J_g(x, y)$, contour form, with initial vehicle position and goal position.

- Multiple goals can be represented by a multiobjective cost function:

$$J(x, y) = w_1 J_o(x, y) + w_2 J_g(x, y)$$

  shown in Figure 94

- What if it moved to minimize this function?

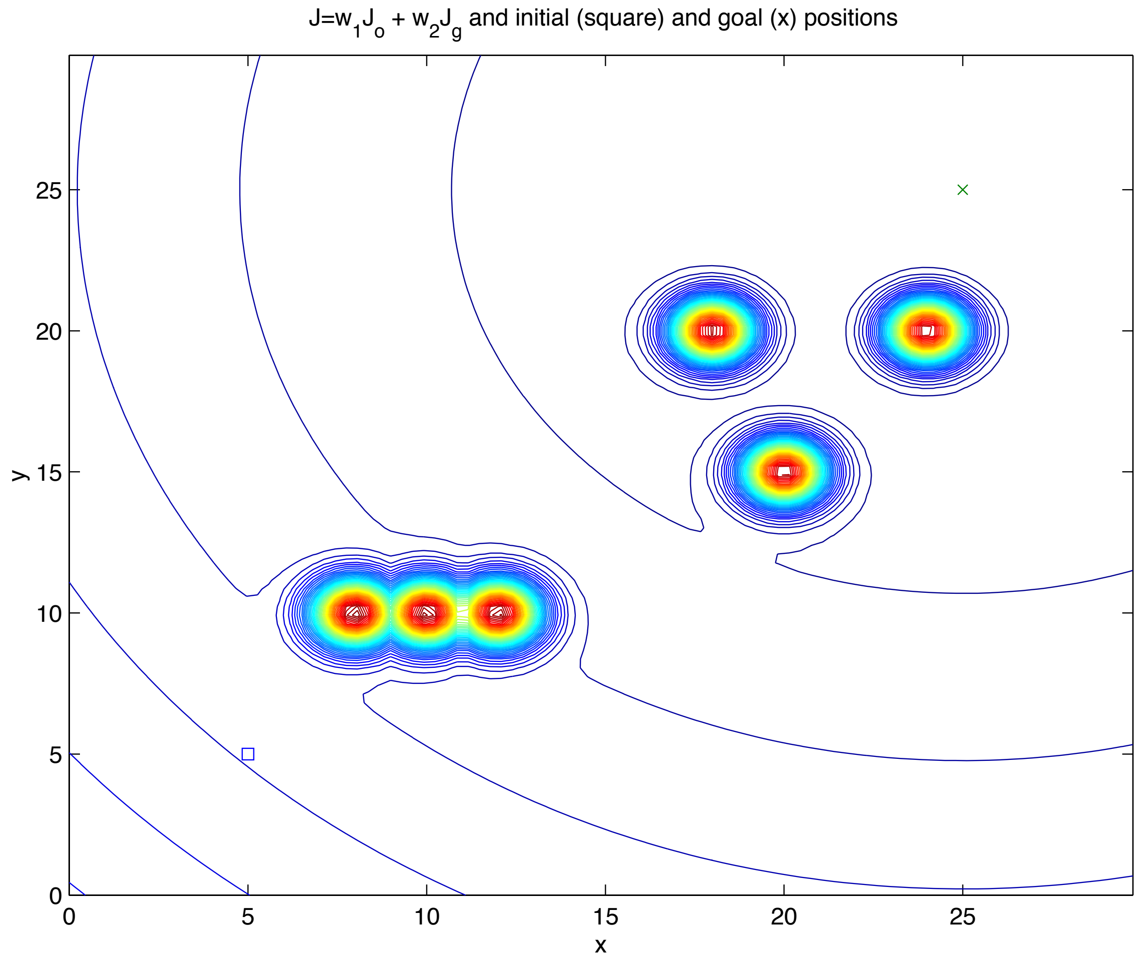

Figure 94: Multiobjective cost function $J(x, y)$ for evaluating plans.

Plan Generation and Selection

- Simple approach: At (x, y) compute the value of J at N_s values (x_i, y_i), $i = 1, 2, \ldots, N_s$, regularly spaced on a circle of radius r around the vehicle position (see Figure 90 where we have $N_s = 8$).
  - Use $r = 1$ and $N_s = 16$.
  - Get 16 plans, where we "predict" ahead one step (plan: "the vehicle is at (x, y), move it to (x_i, y_i)")
- Plan selection: Find i^*,

$$J((x_{i^*}, y_{i^*}) \leq J((x_i, y_i),\ i = 1, 2, \ldots, N_s$$

  - Call this direction $\theta(k)$ and command the vehicle to take a step of length λ in the direction $\theta(k)$.
- Approximates the "steepest descent approach" (but do not need analytical gradient information).

Simulation of the Guidance Strategy

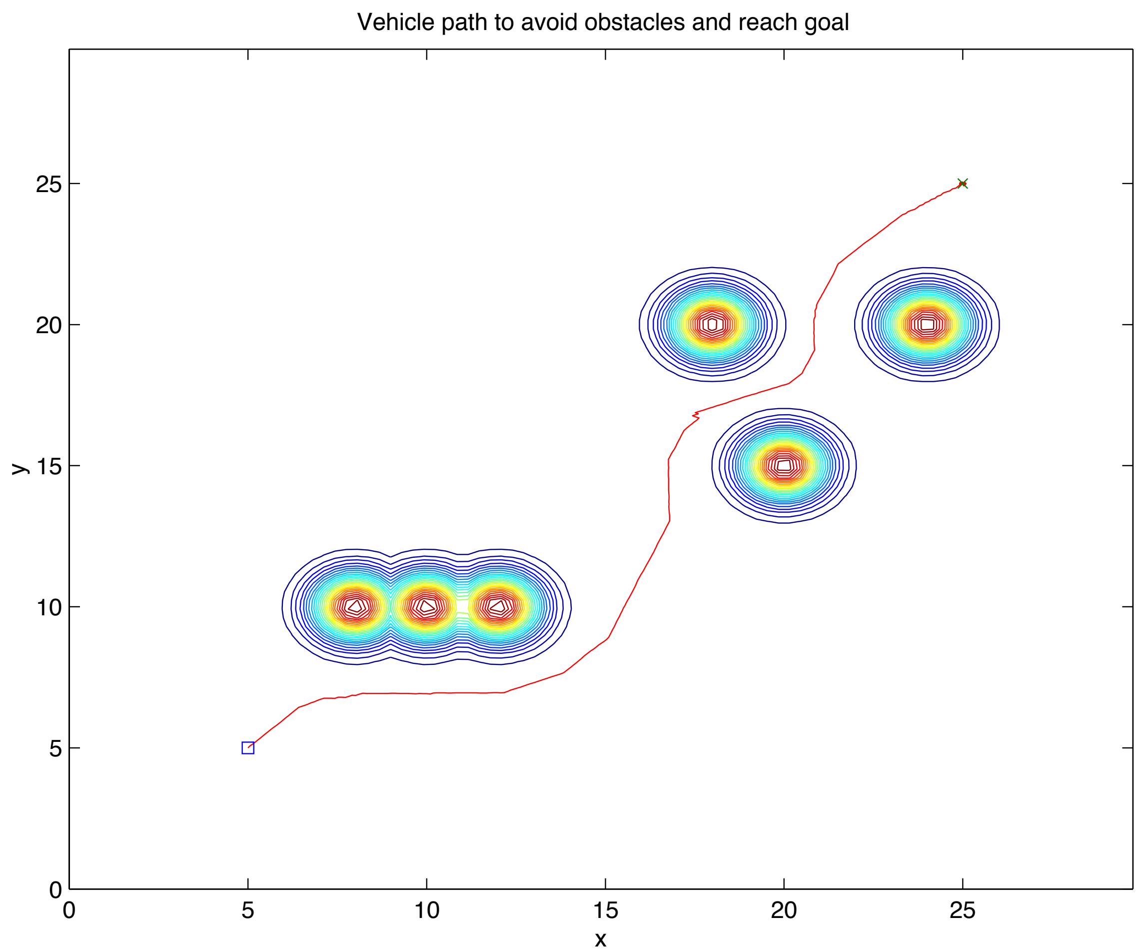

Figure 95: Vehicle path for obstacle avoidance and goal seeking.

More Challenges: Complex Maizes, Mobile Obstacles, and Uncertainty

- Dead ends and circular loops (how to avoid?)
- Mobile obstacles and uncertainty also possible.

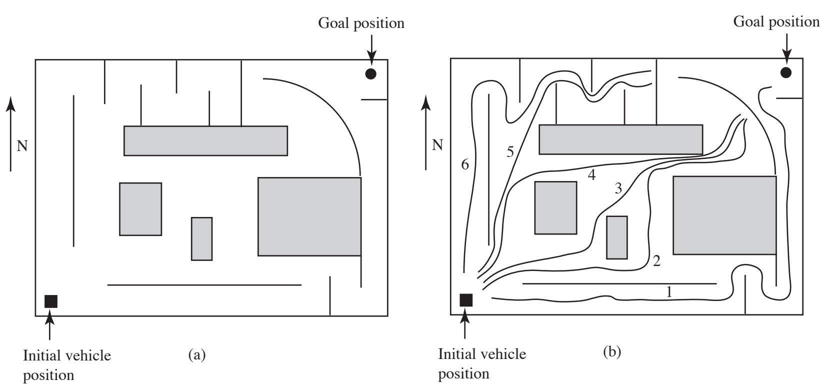

Figure 96: (a) Obstacle path viewed as a maize (notice dead ends), (b) Possible paths through the maize as computed by prediction in a planning strategy (numbered 1–6).

Planning Strategy Design

Closed-Loop Planning Configuration

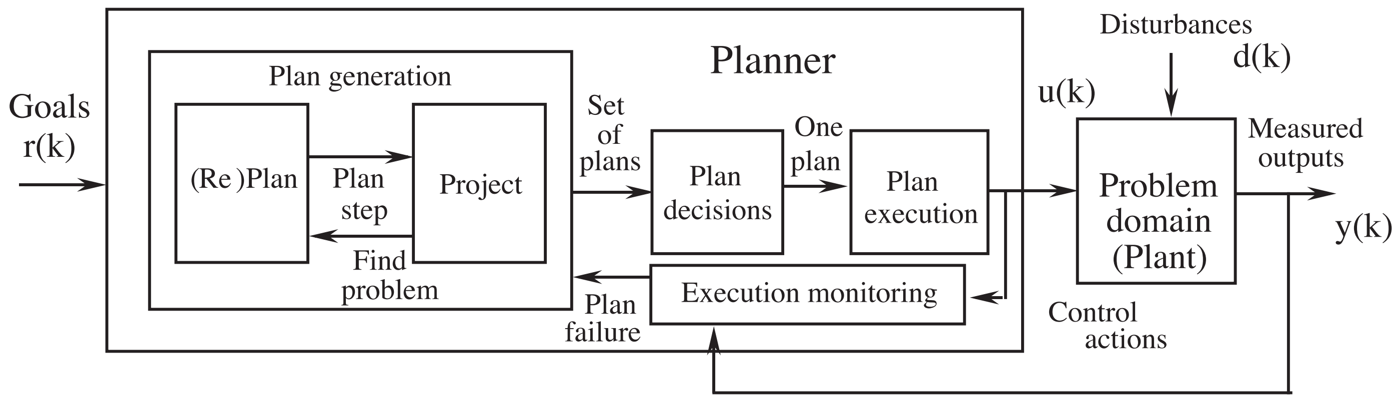

Figure 97: Closed-loop planning system.

- **Plant:**

$$y(k+1) = f(x(k), u(k), d(k)) \qquad (12)$$

where $y(k)$ is the measured output and f is a generally unknown smooth function of the state $u(k)$ and measurable state $x(k)$,

$$x(k) = [y(k), y(k-1), \ldots, y(k-n), u(k-1), u(k-2), \ldots, u(k-m)]^\top \tag{13}$$

- Let

$$e(k+1) = r(k+1) - y(k+1)$$

be the tracking error (want it small, at least asymptotically).

➙ The i^{th} plan of length N at time k is

$$u^i[k,N] = u^i(k,0), u^i(k,1), \ldots, u^i(k,N-1)$$

➙ Objective: Develop a controller that is based on a planning strategy $\rightarrow$ need a model and optimization method to evaluate the quality of each plan.

➙ Best plan is i^*), let the control input be

$$u(k) = u^{i^*}(k, 0)$$

- Different frequency replanning is possible.

Models and Projecting into the Future

➙ Good models lead to good plans, bad models can lead to unstable behavior and poor performance.

- Linear MPC—linear models are used.

➙ No model is perfectly accurate; hence, predictions based on it are always in error.

➙ Here use:

$$y_m(j+1) = f_m(x_m(j), u(j))$$

with output $y_m(j)$, state $x_m(j)$ and input $u(j)$ for

$j = 0, 1, 2, \ldots, N-1$.

- Let $y_m^i(k, j)$ denote the j^{th} value generated at time k using the i^{th} plan $u^i[k, N]$; similarly for $x_m(k, j)$.

➙ To predict the effects of plan i (project into the future) at each time k you compute for $j = 0, 1, 2, \ldots, N-1$,

$$y_m^i(k, j+1) = f_m(x_m(k, j), u^i(k, j))$$

- At time k to simulate ahead in time, for $j = 0$ you initialize with $x_m(k, 0) = x(k)$.

- Then, generate $y_m(k, j+1)$, $j = 0, 1, 2, \ldots, N-1$, using the model and generate values of $u^i(k, j)$, $j = 1, 2, \ldots, N-1$, for each i.

Criterion and Optimization Method for Plan Selection

- Set of plans (strategies) is "pruned" to one plan that is the best one to apply at the current time (where "best" can be determined based on, e.g., consumption of resources).

- A "shortest path problem"??

Criteria for Selecting Plans

➙ Use $J(u^i[k,N])$ for plan $u^i[k,N]$ using the f_m model.

- Assume that the reference input $r(k)$ is either known for all time, or at least that at time k it is known up till time $k+N$.

➙ For example:

$$J(u^i[k,N]) = w_1 \sum_{j=1}^{N} \left(r(k+j) - y_m^i(k,j)\right)^2 + w_2 \sum_{j=0}^{N-1} \left(u^i(k,j)\right)^2 \tag{14}$$

➙ Options: Use the output of a "reference model," an error measure on the other past values of the inputs and outputs, or an error measure on some other system variable.

➙ This is "model predictive control" (or "receding horizon control").

Nonlinear Optimization for Plan Selection

➙ Which optimization method and model to use?

➙ Problem: Complexity (infinite number of plans?)

➙ Problem: Optimization algorithm convergence?

➙ Linear MPC solution: linear model, linear least squares

- What if a linear model is not accurate enough?

➙ Infinite number of plans → nonlinear optimization (e.g., gradient methods). Convergence?

➙ Finite-branching tree of plans → combinatorial optimization. Accuracy?

Brute-Force Approach to Plan Selection

➙ Discretize the model, quantize the inputs

- Creation and evaluation of all possible plans is often computationally prohibitive.
- Suppose that there are N_u possible input values for a deterministic model
- Number of input sequences (plans)?

$$(N_u)^N$$

- The curse of dimensionality!
- Generate all plans, rank, and choose best? Perhaps

➙ There are ways to trade-off computational complexity for the quality of plan selection and ultimately performance.

Planning Using Preset Controllers and Model Learning

- ➙ Try to prune the tree of plans, without reducing prediction accuracy.

Planning Using Multiple Controllers

- ➙ Consider a specific controller (a "preset" controller) applied to the current state and reference input to be a type of "plan template"

  - Suppose that there are S such plan templates, which have the form of functions F_u^i

$$u^i(k,j) = F_u^i(x(k,j), r(k+1)) \; i = 1, 2, \ldots, S$$

  where we assume we can measure $r(k+1)$.

- ➙ At each step we take each of these S plans and project into the future how each will perform, pick the best one, then let the control input be $u^{i^*}(k,0)$ where i^* is the best plan as measured by some cost function.

- ➔ In some applications S need not be too large, and hence if we take the "brute-force" approach of the last section we overcome the problems discussed there in complexity and optimization.
- ➔ Could also take an approach where multiple models are used in planning, or where tuned models are used (adaptive model predictive control).
- ➔ In practical applications often use "hierarchical planning systems." Multiple model granularities, horizons, sequences of tasks/subtasks/goals.

Discussion: Concepts for Stable Planning

- Most work for linear MPC
- ➔ Stability anlaysis of closed-loop planning strategies depends critically on model accuracy, plant uncertaity, and plant nonlinearities.

➙ Key issues:

- Model accuracy: Assume that model is perfect, the same as the physical plant?
- Navigating through uncertainty: Planning your way through a “noisy tree” of paths to the goal state(s). Plant is an adversary! May get trapped in deadends! How far to predict ahead?
- Avoiding traps: Circular traps may make it impossible to reach the goal.

Design Example: Planning for a Process Control Problem

Level Control in a Surge Tank

- "Surge tank" modeled by

$$\frac{dh(t)}{dt} = \frac{-\bar{d}\sqrt{2gh(t)}}{A(h(t))} + \frac{\bar{c}}{A(h(t))}u(t)$$

where $u(t)$ is the input flow (control input), which can be positive or negative (it can both pull liquid out of the tank and put it in); $h(t)$ is the liquid level (the output of the plant); $A(h(t)) = |\bar{a}h(t) + \bar{b}|$ is the cross-sectional area of the tank and $\bar{a} > 0$ and $\bar{b} > 0$ (their nominal values are $\bar{a} = 0.01$ and $\bar{b} = 0.2$); $g = 9.8$; $\bar{c} \in [0.9, 1]$ is a "clogging factor" for a filter in the pump actuator where if $\bar{c} = 0.9$ there is some clogging of the filter and if $\bar{c} = 1$ the filter is clean so there is no clogging (we will take $\bar{c} = 1$ as its nominal value); and $\bar{d} > 0$ is a parameter related to the diameter of the output pipe (and its nominal value is $\bar{d} = 1$).

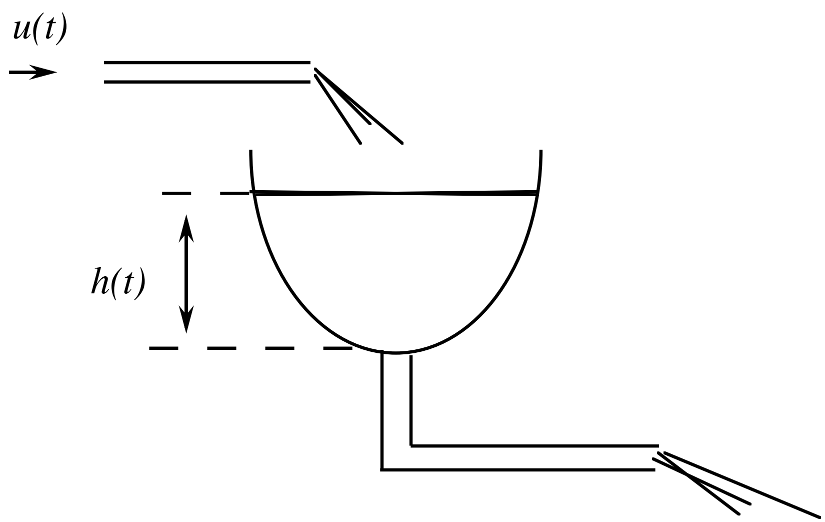

Figure 98: Surge tank.

- Let $r(t)$ be the desired level of the liquid in the tank (the reference input) and $e(t) = r(t) - h(t)$ be the tracking error.
- Assume that you know the reference trajectory a priori and assume that $r(t) \in [0.1, 8]$ and that we will not have $h(t) > 10$.

- Assume that $h(0) = 1$.

➙ Use discretization:

$$h(k+1) = h(k) + T\left(\frac{-\bar{d}\sqrt{19.6h(k)}}{|\bar{a}h(k)+\bar{b}|} + \frac{\bar{c}}{|\bar{a}h(k)+\bar{b}|}u(k)\right)$$

where $T = 0.1$.

- Saturate the input:

$$u(k) = \begin{cases} 50 & \text{if} \quad \bar{u}(k) > 50 \\ \bar{u}(k) & \text{if} \quad -50 \le \bar{u}(k) \le 50 \\ -50 & \text{if} \quad \bar{u}(k) < -50 \end{cases}$$

- Avoid negative levels:

$$h(k{+}1) = \max\left\{0.001, h(k) + T\left(\frac{-\bar{d}\sqrt{19.6h(k)}}{|\bar{a}h(k)+\bar{b}|} + \frac{\bar{c}}{|\bar{a}h(k)+\bar{b}|}u(k)\right)\right\}$$

Planner Design

- Use the discrete model as our model for planning, but with

$$A(h(t)) = \bar{a}_m (h(t))^2 + \bar{b}_m$$

with $\bar{a}_m = 0.002$ and $\bar{b}_m = 0.2$.

- We do not assume that we know the values of $\bar{c}$ and $\bar{d}$, so for these we use $\bar{c}_m = 0.9$ and $\bar{d}_m = 0.8$.

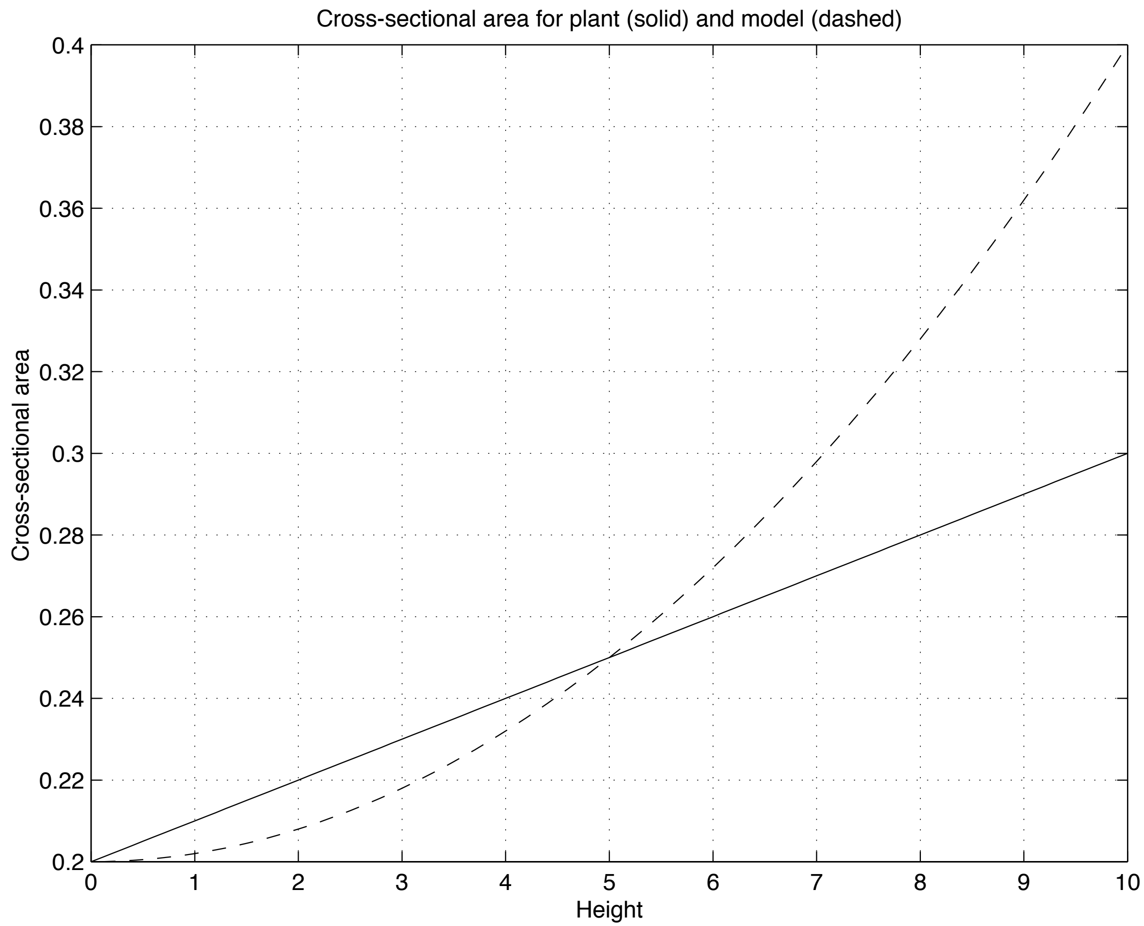

Figure 99: Cross-sectional area $A(h)$ for the plant (solid) and model to be used for projection (dashed).

- So, is the model accurate enough to be used in projection?
- Consider a simple controller to evaluate this.

➙ Use a PI controller as the "plan template."

- If $e(k) = r(k) - h(k)$,

$$u(k) = K_p e(k) + K_i \sum_{j=0}^{k} e(j) \tag{15}$$

➙ Goal: Reasonably fast response, with no overshoot in the tracking error $e(k)$.

- Suppose that via experience in designing PI controllers for surge tanks with various cross-sectional areas you know that typically

$$K_p \in [0, 0.2]$$

and

$$K_i \in [0.15, 0.4]$$

- With $K_p = 0.01$ and $K_i = 0.3$ you get the response in Figure 100.

★ Notice that while the response is relatively fast, there is some overshoot and that is undesirable.

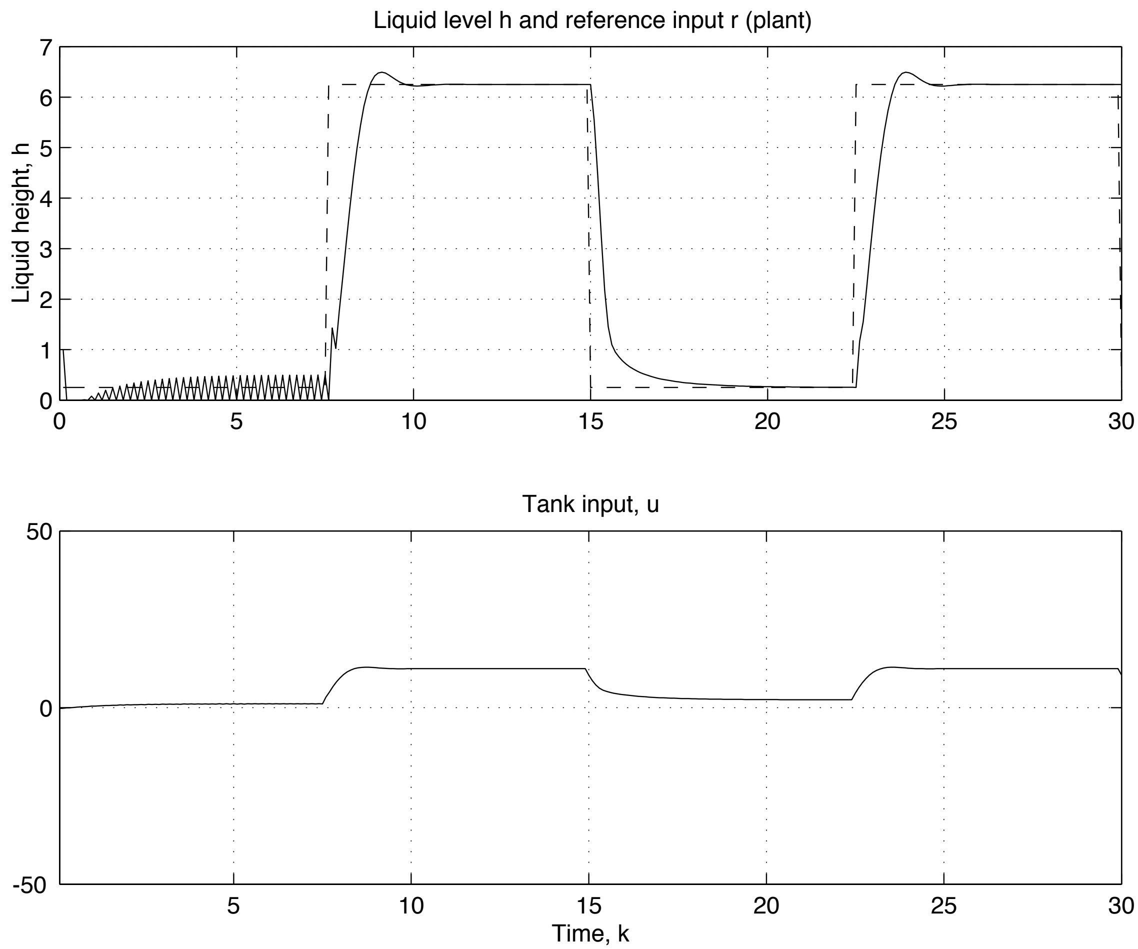

Figure 100: Closed-loop behavior of the surge tank using a PI controller.

- ★ You actually get a similar response if you use the same gains for the above model that will be used for projection in our planning strategy.
- ★ To see this consider Figure 101
- • But the true test is whether it works in a planner.

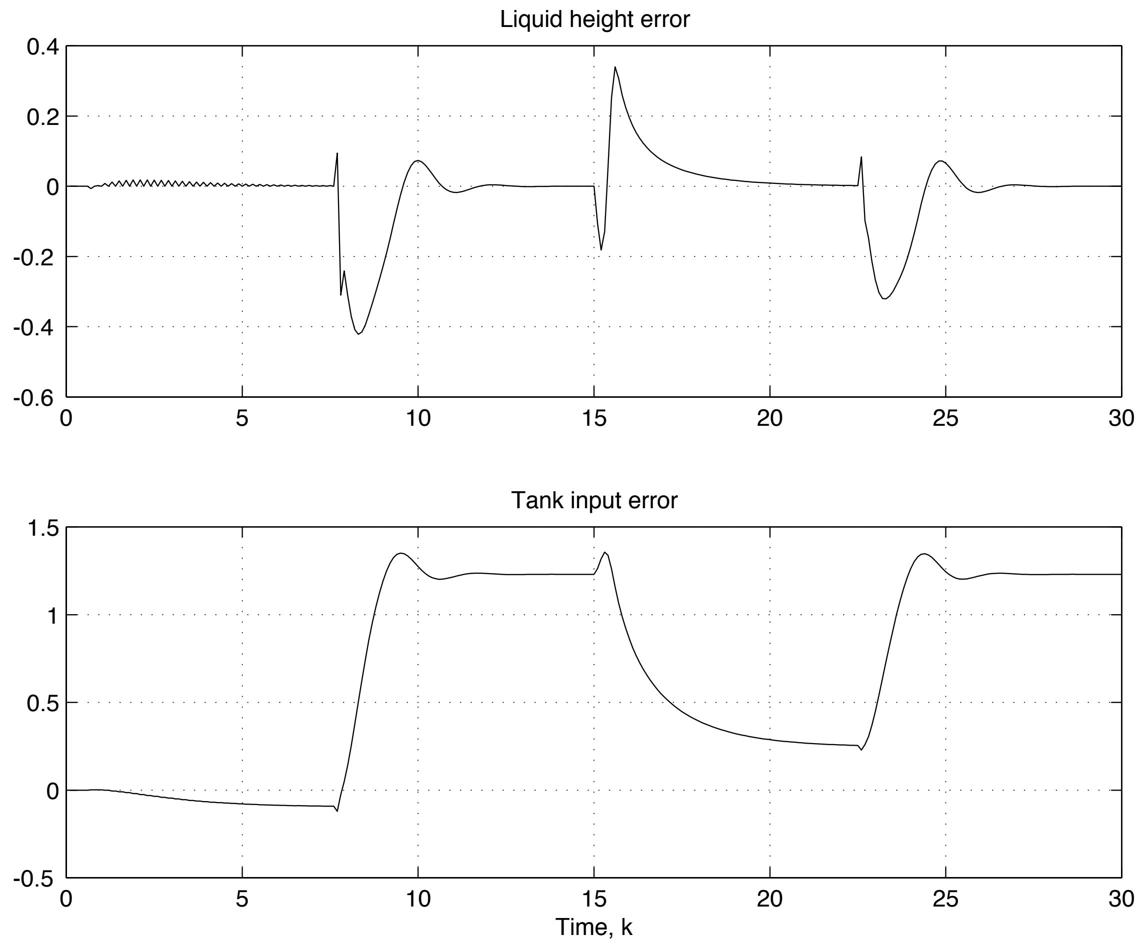

Figure 101: Error between cases where the truth model and projection model are used as the plant.

- We use the cost function in Equation (14) with $N = 20$ (for two seconds projection into the future), $w_1 = 1$, and $w_2 = 1$
  - We assume at each time instant that the reference input remains constant while we project into the future.
  - Our "plan templates" are the PI controllers, with different values of K_p and K_i.
- Create a grid on the above ranges by considering all possible combinations of

$$K_p \in \{0, 0.05, 0.1, \ldots, 0.2\}$$

and

$$K_i \in \{0.15, 0.2, \ldots, 0.4\}$$

  - There are $5 \times 6 = 30$ different plans (controllers) that are evaluated at each time step.
  - Predict 2 s into future for each PI controller.

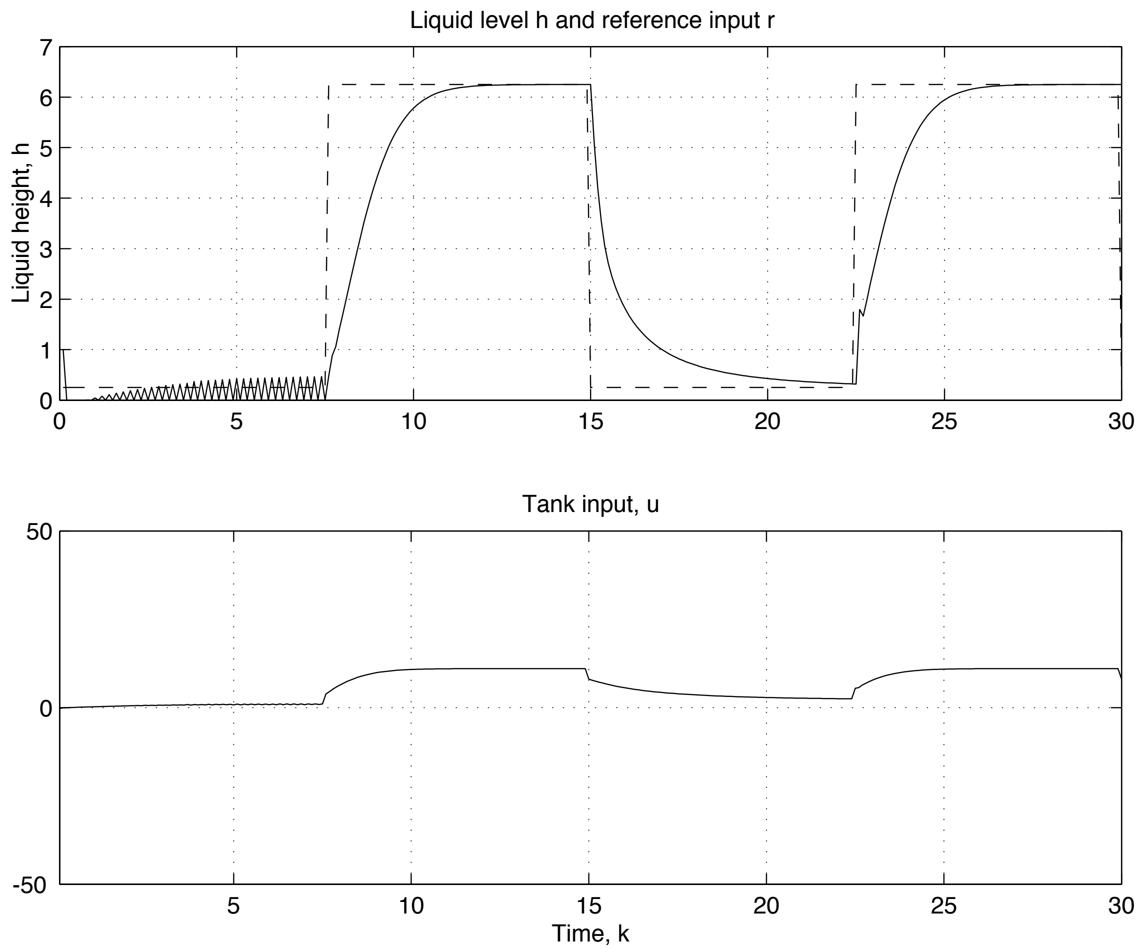

Figure 102: Closed-loop behavior of the surge tank using a planning strategy.

- ★ Get a slower rise-time than in Figure 100 when we used the PI controller, but that we were able to tune the planning strategy (by adjusting w_1, w_2, and the grid on the PI gains) so that there is no overshoot and that was our main objective.

➙ How does it achieve this performance?

- ★ It switches controllers on-line and to see this consider Figure 103.

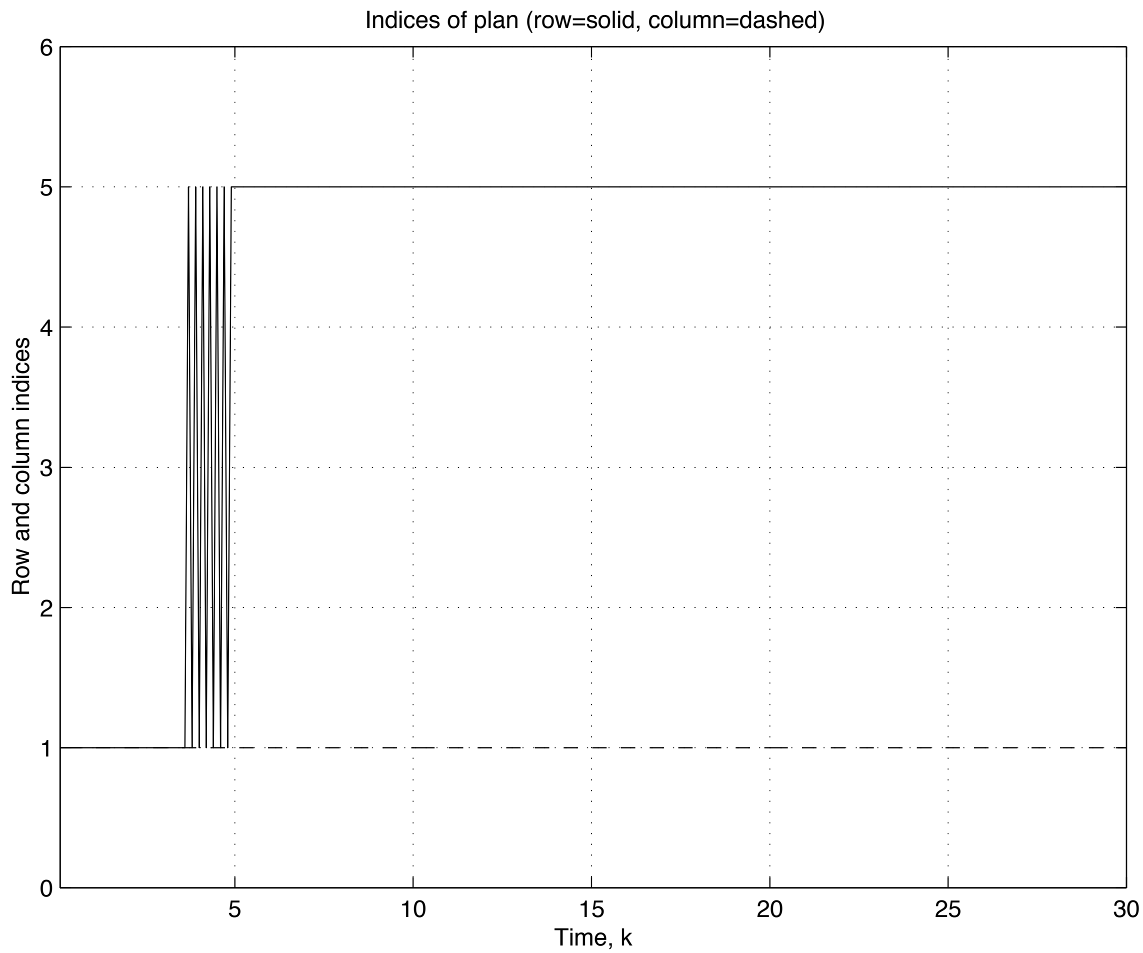

Figure 103: Indices of PI controllers that are used at each time step for the tank.

★ We used indices that are proportional in size to the K_p and K_i values; hence it seeks to increase the K_p value to reduce tracking error and get a good rise-time, and lower the K_i value to try to reduce overshoot.

- What about plant parameter variations?
- Let $\bar{c} = 0.8$ (representing more clogging) you get similar results to above.

★ If you use the nominal value for $\bar{c}$ and use $\bar{a} = 0.05$ you get the the cross-sectional area shown in Figure 104 and we get the closed-loop response in Figure 105.

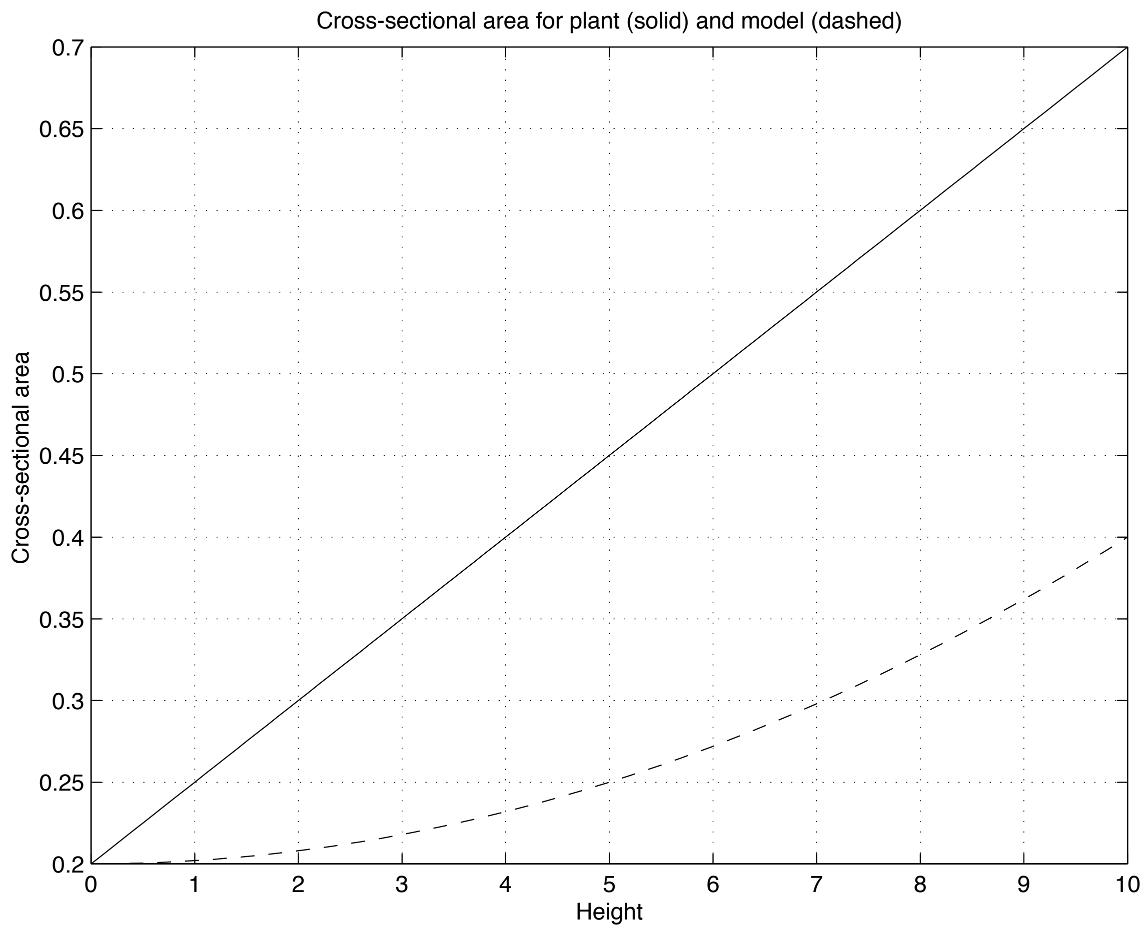

Figure 104: Cross-sectional area $A(h)$ for the plant (solid) and model to be used for projection (dashed).

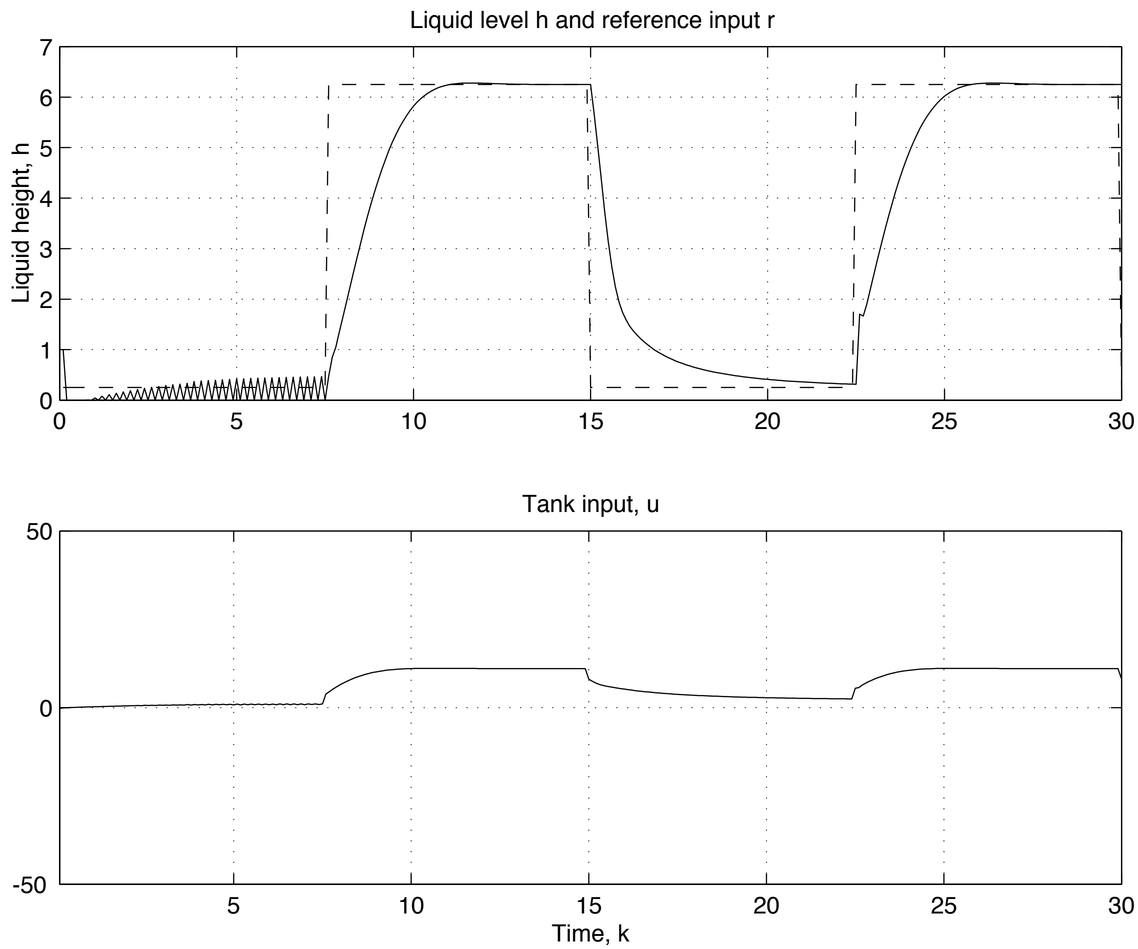

Figure 105: Closed-loop behavior of the surge tank using a planning strategy (different cross-sectional area).

- ★ Get an adequate rise-time, but a small amount of overshoot—may need to tune.
  - How robust is the controller to plant perturbations?

Effects of Planning Horizon Length

- ➙ Use nominal plant and study the effect of changing the projection length N.
- ★ In particular we plot the tracking energy

$$\frac{1}{2}\sum_{k}(e(k))^2$$

and control energy

$$\frac{1}{2}\sum_{k}(u(k))^2$$

vs.

$$N \in \{1, 5, 10, 15, 17, 20, 25, 30, 33, 35, 36, 37, 38, 39, 40, 45, 50\}$$

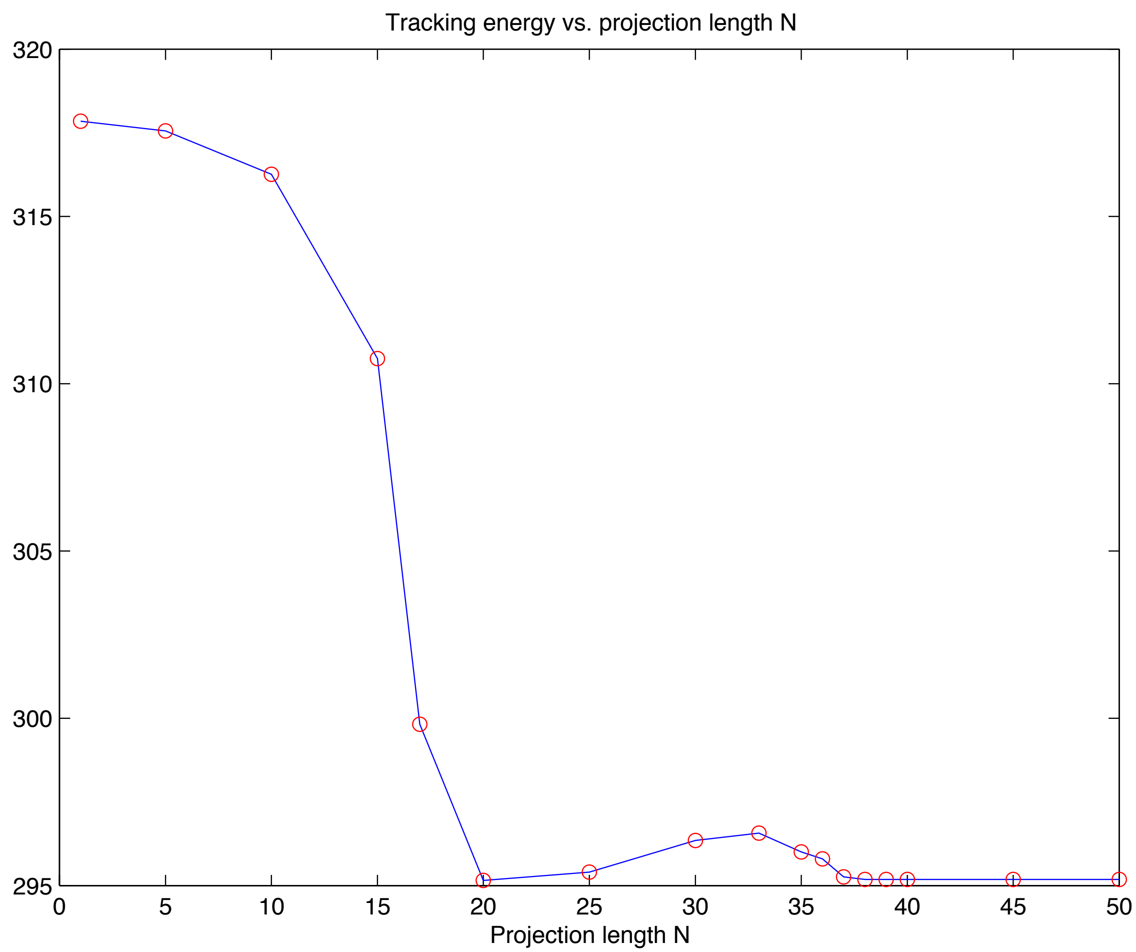

Figure 106: Tracking energy vs. projection length N.

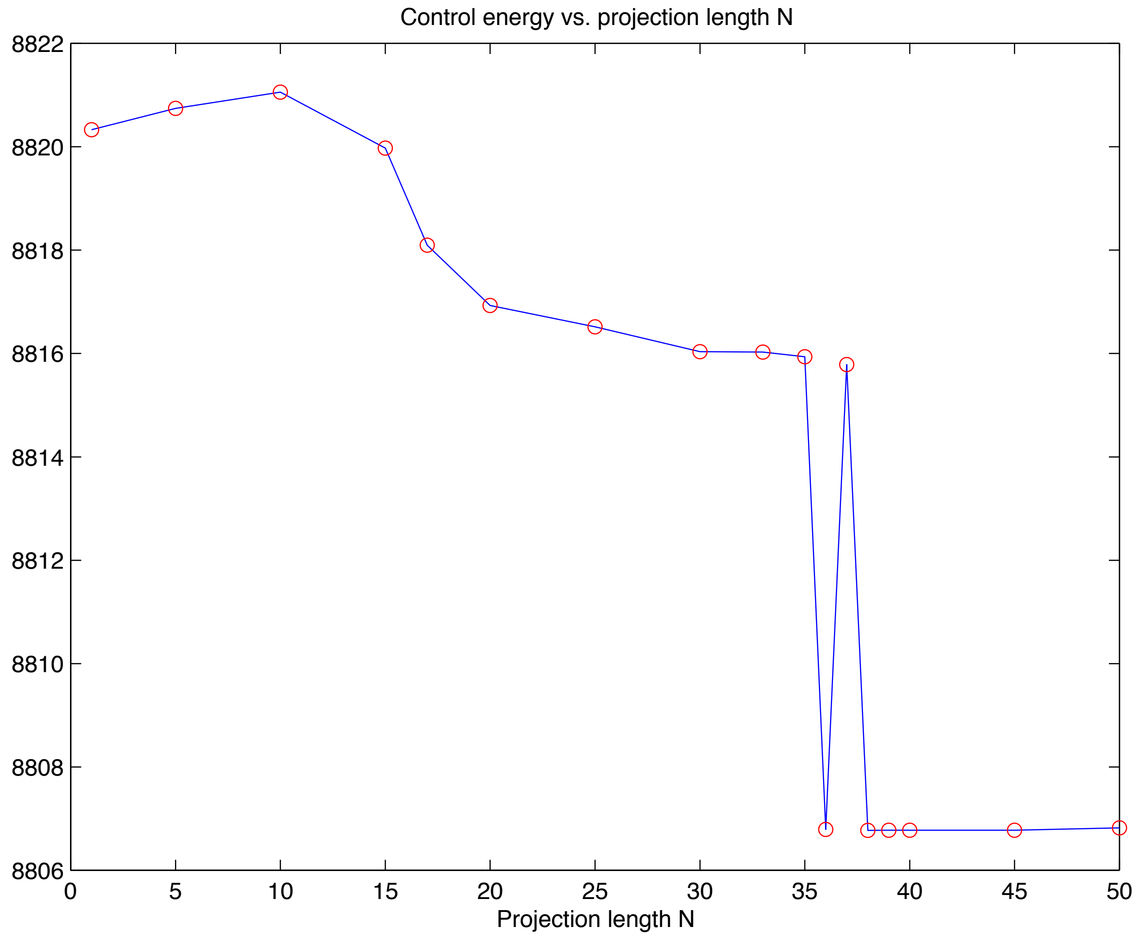

Figure 107: Tracking energy vs. projection length N.

- Prediction horizon choice is difficult.
- Prediction too far into the future is often not useful due to plant uncertainty, it costs many computations, and could result in performance degradation.

Attentional Systems

Neuroscience and Psychology of Attention

- Attention is the process of focusing or concentrating.
- Hierarchy: Consciousness, sleeping, awareness, and attentiveness
- Disengage from one focus, move, and then engage on another focus ("vigilance").
- Have attention for all senses!
- Attention allows us to amplify some sensory signals and attenuate others (e.g., "cocktail party effect")

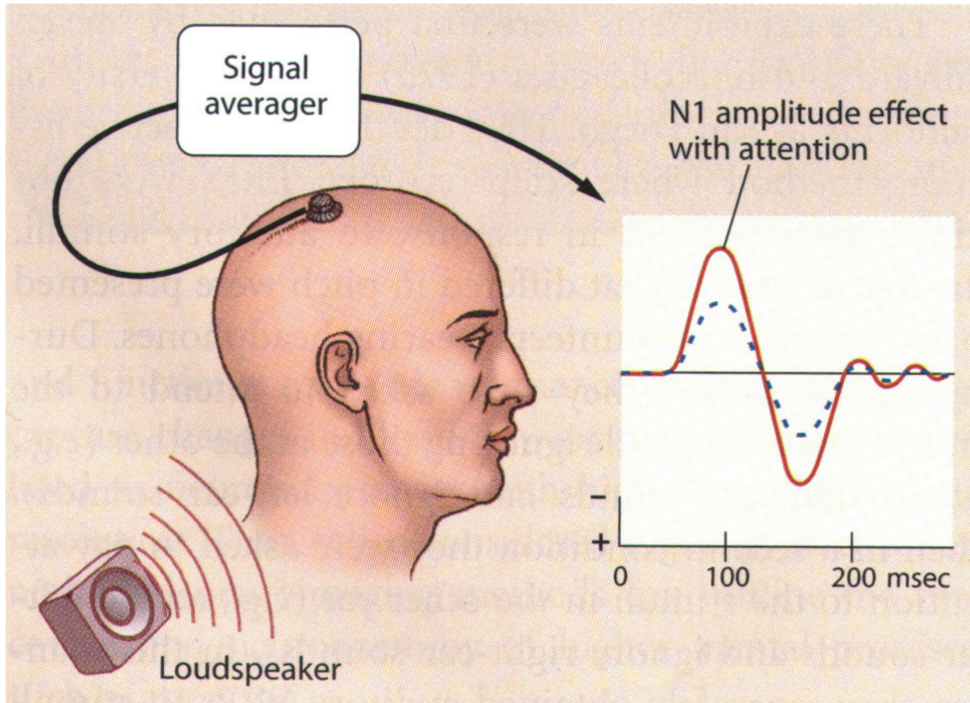

Figure 108: Attentional experiment with event-related potentionals illustrating amplification of signals that are attended to (figure taken from [3]).

Dynamically Changing Focus

- For vision, focus of attention is a type of "spot light" (where signals are amplified)
  - Spot light may coincide with where our eyes are focused—"overt" attention
  - Or, it may not—"covert" attention
- Control of attentional focus:
  - Goal-driven (often "voluntary") attention reorientation: Executive functions reorient the focus. "Top-down" refocusing based on our problem-solving strategy and goals. Typically, slower less "potent" refocusing than...
  - Stimulus-driven (often "involuntary") attention reorientation: Sensory signals control the focus in a "bottom-up" fashion. For example, an object that is moving on a trajectory toward us, a bright flash of light (e.g., a

fire), or blood (with evolutionary forces likely at work). Sensory inputs can achieve an *automatic* reorienting of attention, faster and more potent

- Learning can play a key role in attention (habituation, sensitization).

Multistage Processing: Filtering, Selection, and Resource Allocation

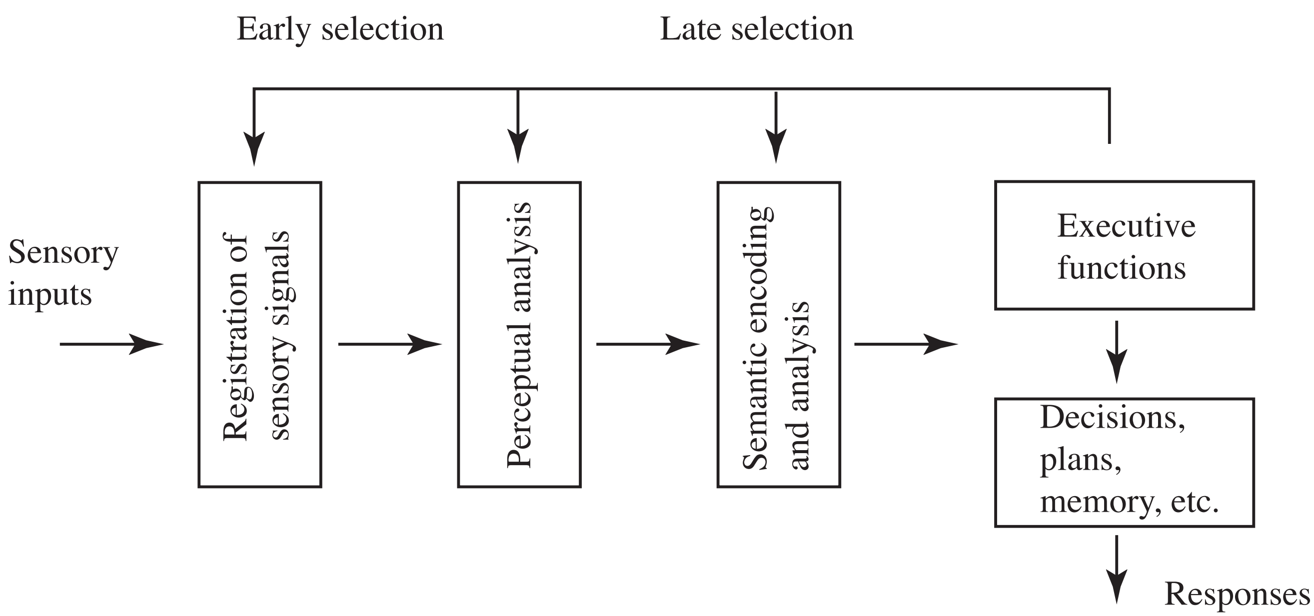

Figure 109: Multistage attention process (adapted from [3]).

- Attention involves filtering out (discarding) some information.
- "Early selection" before higher-level processing, and "late selection" via abstract anlaysis and processing of sensory

signals

- A multistage process with feedback control paths.
- A cascaded filtering process, the most important information is focused on ("selected"), and less important information is ignored.
- Attention helps cope with "bottlenecks" in information processing.
- Attention allows us to allocate our cognitive resources to help us meet our goals.

➙ Key aspect: Strategies used to allocate cognitive resources, especially in an "optimal" manner.

➙ Attention is for complexity management.

Attentional Strategies for Multiple Predators and Prey

- Organism in environment with multiple predators (to avoid) and prey (to pursue)
- Want an accurate "picture" of environment with limited resources to obtain it.
- Organism dynamically focuses its attention.

Cognitive Resource Allocation Model

- Assume there is a recognizer for predators and prey
- Need to decide what to focus on (cognitively process).
- Selection process that could be occurring in either early or late selection
- There is a "limited channel" or one resource that must be shared, and the attention strategy must decide how it is shared.

- Resource allocation
  - Ignore overt/covert attention

    Quantifying Length of Time Predators/Prey are Ignored

- Set of predators and prey:

$$P = \{1, 2, \ldots, N\}$$

  - Let

$$T_i(t), i \in P, t \geq 0$$

    denote the last time at which predator/prey i was detected.

- "Detected"—the organism has focused its attention on the predator/prey, has identified it, and its characteristics (e.g., its position).
- Attentional strategies = "controllers"

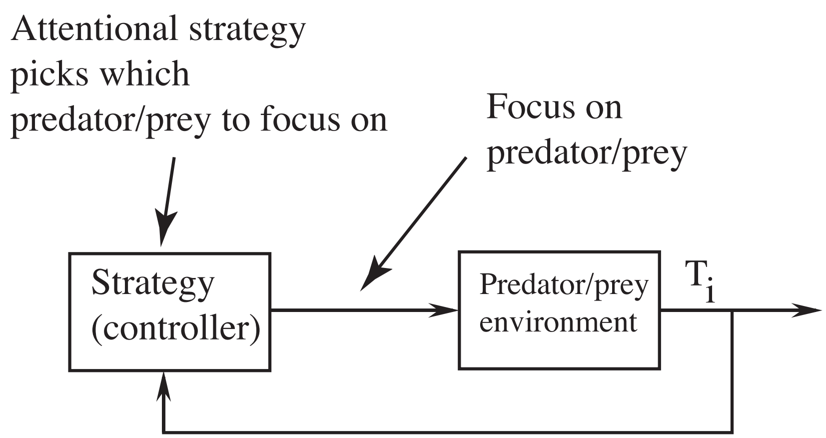

Figure 110: Attentional strategy viewed as a controller.

- Assume there is a cognitive tracking mechanism that is trying to estimate where predators/prey are moving.
- Not perfect tracking of multiple predators/prey—allow them to be "lost" for a period of time (due to hiding, limited field of view, etc.).
- Initially

$$T_i(0) = 0, i \in P$$

- Meaning? A good initialization?
- If attention not active, then

$$T_i(t) \to \infty, i \in P, t \to \infty$$

- Attention strategy tries to avoid $T_i(t) \to \infty$ for any $i \in P$ and indeed try to keep the $T_i(t)$ values as small as possible
- Why?
- Assume each predator/prey will *persistently* periodically "appear" (in fact, a finite amount of time between predator/prey appearances)

Environmental and Cognitive Delays Affecting Attentional Switching

- Let $\delta(t) > 0$ denote a "processing delay" that may represent the delay from the environment (e.g., due to a predator being occluded for a period of time bounded by δ^i) and a "cognitive processing delay" (known $\delta_{i,j}$ to switch from focus on i to j)

➡ Assume: $\delta_s = \delta_{i,j}$ for all $i, j \in P$.

- Let $\delta_e(t)$ be the delay incurred by the organism in first getting an indication of the presence of a predator/prey, from the time that it gets switched to focus on that predator/prey.
- If we let

$$\bar{\delta} = \max_i \left\{ \delta^i \right\}$$

then $\delta_e(t) \leq \bar{\delta}$. Let

$$\delta(t) = \delta_s + \delta_e(t)$$

- Let δ denote a constant that is the least upper bound on $\delta(t)$

Rate of Cognitive Processing

- The organism may take additional time to detect an predator/prey that it has not detected for a long period of time.
- Let

$$a_i, i \in P$$

and $1/a_i$ represents a "rate" at which the organism cognitively processes information about predators/prey in order to detect them.

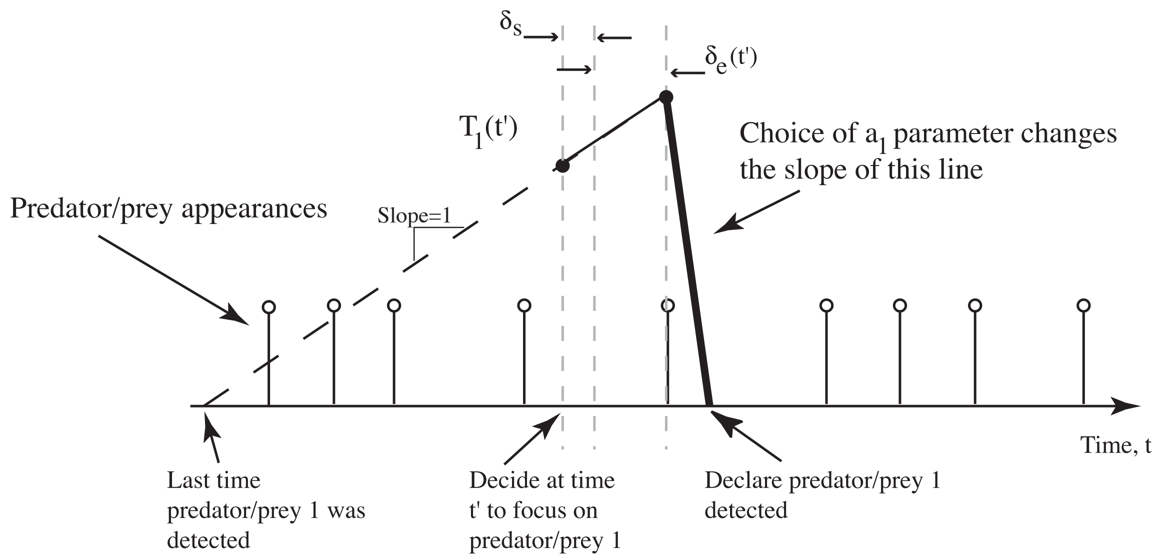

Figure 111: Illustration of timing of organism decision-making and predator/prey appearances (note that pulses represent the first times that predators/prey appear).

- At time $t' + \delta_s$ the organism has switched its focus to predator/prey i.
- Starting at $t' + \delta_s$ the organism is looking for predator/prey i

and before $t' + \delta_s + \delta^1$ we know that an predator/prey appearance will occur.

- At time $t' + \delta_s + \delta_e(t')$ the organism initiates the completion of the "detection" of predator/prey 1 and the amount of time that it takes to do that is dictated by the a_1 parameter.
- Declare predator/prey 1 "detected" at the time at which T_1 is decreased to zero.

➙ Consider an attentional strategy...

Focus on a Predator/Prey Ignored for the Longest Time

- Let D_{k_r} denote the time at which the attentional strategy chooses a predator/prey to focus on (i.e., it is the decision time), and suppose that $D_1 = 0$.

➙ An attentional strategy that focuses on the predator/prey that was ignored for the longest time makes choices of which predator/prey to focus on such that at D_{k_r} the attentional strategy chooses to focus on predator/prey $i^*(k_r)$ such that

$$T_{i^*(k_r)}(D_{k_r}) \geq T_i(D_{k_r}), \forall i \in P \tag{16}$$

and focuses on it until it detects it.

- Ties broken randomly.

Decision-Timing for Attentional Switches

➙ The times when the attentional strategy makes decisions are given by

$$D_{k_r+1} = D_{k_r} + \delta(D_{k_r}) + a_{i^*(k_r)} T_{i^*(k_r)}(D_{k_r}) + (D_{k_r+1} - D_{k_r})\, a_{i^*(k_r)} \tag{17}$$

1. The first term is simply the last decision point D_{k_r}.
2. The second term is the delay $\delta(D_{k_r})$ where

$$\delta(D_{k_r}) = \delta_s + \delta_e(D_{k_r})$$

3. Third, the term $a_{i^*(k_r)}T_{i^*(k_r)}(D_{k_r})$ is the amount of time it takes to detect predator/prey $i^*(k_r)$ that arises due to the fact that we have not detected it for some time.
4. Finally, the fourth term quantifies that there is additional time needed to detect the predator/prey simply because during the time that the cognitive processing for the predator/prey is occuring, even when it is focused on, the length of time since the last detection continues to increase

- Rearrange Equation (17) to get

$$D_{k_r+1} = D_{k_r} + \frac{\delta(D_{k_r}) + a_{i^*(k_r)}T_{i^*(k_r)}(D_{k_r})}{1 - a_{i^*(k_r)}} \tag{18}$$

- The delay δ directly influences the rate at which we can switch attentional focus.
- The length of time between decisions can be lengthened if a particular predator/prey has been ignored for too long due to

the effects of the a_i parameters.

The Cogntive Capacity Constraint

➙ What is the effect of the a_i parameters on how fast a predator/prey is detected?

- What is the slope of the bold line in Figure 111?
- Notice that the peak value

$$T_{i^*(k_r)}(D_{k_r} + \delta_s + \delta_e(D_{k_r})) = T_{i^*(k_r)}(D_{k_r}) + \delta_s + \delta_e(D_{k_r})$$

since the slope of the dashed line in Figure 111 is unity.

- Notice that Equation (18) gives the amount of time between the decision time D_{k_r} and time of detection D_{k_r+1} so that the slope of the bold line in Figure 111 is

$$-\left\{ \frac{T_{i^*(k_r)}(D_{k_r}) + \delta_s + \delta_e(D_{k_r})}{\frac{\delta_s+\delta_e(D_{k_r})+a_{i^*(k_r)}T_{i^*(k_r)}(D_{k_r})}{1-a_{i^*(k_r)}} - (\delta_s + \delta_e(D_{k_r}))} \right\}$$

which with some simple algebra reduces to

$$-\frac{(1 - a_{i^*(k_r)})}{a_{i^*(k_r)}} \tag{19}$$

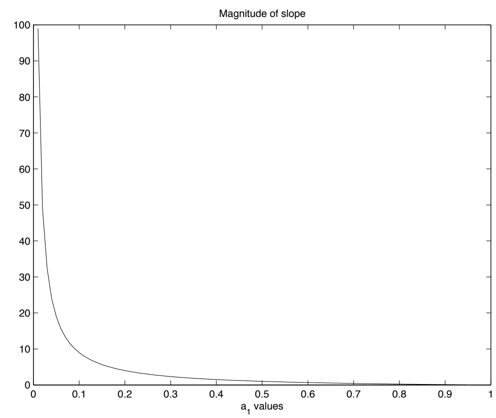

Figure 112: Magnitude of the slope of the bold line in Figure 112 for various values of a_1.

- You will see that it is necessary that $a_{i^*(k_r)} < 1$.
- Using this fact, Equation (19) indicates how fast detection

occurs as shown in Figure 112 (i.e., how fast cognitive processing occurs).

- Small values of a_i (high values of $1/a_i$, the rate of processing by the organism in trying to detect) we get fast detection, and with larger ones we get slower detection.
- It is necessary that the "capacity condition"

$$\rho = \sum_{i=1}^{N} a_i < 1 \tag{20}$$

be satisfied in order for *any* attentional strategy to ensure that the values of $T_i(t)$, $i \in P$, remain bounded.

- If the predators/prey can get more difficult to detect if they have not been detected for a long time, the organism must be able to operate "fast enough" to be able to find them.
- Cognitive capacity quantifies when an environment presents

too large of an attentional load for an organism.

Additional Attentional Strategies

➙ Other strategies...

➙ Choose the predator/prey to focus on that has been ignored more than the average time that all the predators/prey have been ignored.

- At D_{k_r} the attentional strategy chooses to focus on predator/prey $i^*(k_r)$ such that

$$T_{i^*(k_r)}(D_{k_r}) \geq \frac{1}{N}\sum_{i=1}^{N} T_i(D_{k_r}) \tag{21}$$

and focus on it until it detects it.

- Note that Equation (18) also holds for this strategy, and that of course the capacity condition Equation (20) must hold.

- How does the strategy choose which particular predator/prey to focus on?
  - Randomly or "predator/prey priorities"
- An attentional strategy that focus on the predator/prey that may be the one that is most difficult to find makes choices of which predator/prey to focus on such that at D_{k_r} the attentional strategy chooses to focus on predator/prey $i^*(k_r)$ such that

$$a_{i^*(k_r)}T_{i^*(k_r)}(D_{k_r}) \geq a_i T_i(D_{k_r}), \forall i \in P \quad (22)$$

  and focuses on it until it detects it.
  - Since a_i is the amount of "load" you can think of this attentional strategy as choosing the predator/prey to focus on that may be the most difficult one to find.
- Consider one more strategy...

➙ If you pick predator/prey $i^*(k_r)$ to focus on

$$T_{i^*(k_r)}(D_{k_r}) + \delta_s + \delta_e(D_{k_r})$$

is the peak that $T_{i^*(k_r)}(D_{k_r})$ reaches before the predator/prey is detected.

- We do not know $\delta_e(D_{k_r})$
- However, a known bound on the peak value is given by

$$T_{i^*(k_r)}(D_{k_r}) + \delta_s + \delta_e(D_{k_r}) \leq T_{i^*(k_r)}(D_{k_r}) + \delta_s + \delta^{i^*(k_r)}$$

➙ Consider choosing predator/prey $i^*(k_r)$ to focus on at time D_{k_r} if

$$i^*(k_r) = \arg\max_i \left\{ w_i \left(\frac{T_i(D_{k_r}) + \delta_s + \delta^i}{\frac{(1-a_i)}{a_i}} \right) \right\} \qquad (23)$$

where $w_i > 0$, $i \in P$ are weighting factors.

➙ The strategy picks the predator/prey to focus on that is expected to have the highest peak, and hence in a sense the one that is most difficult to detect.

➙ The weighting factors w_i can be chosen to force the predator/prey to focus on some predators/prey.

➙ Can also be tuned to improve performance measures.

➙ Strategies based on priority parameters $p_i > 0$, $p_i \in \Re$, $i \in P$, that scale T_i.

- Simply scale T_i by p_i, $i \in P$ in each of the cases and then make all decisions based on the same formulas as above, but with T_i replaced by $p_i T_i$, $i \in P$.

- Assume:

$$\rho_p = \sum_{i=1}^{N} p_i a_i < 1 \tag{24}$$

➙ Can view attention scheduling as on-line optimization. How?

Design and Simulation of Attentional Strategies

- Use a sampling period of $T_s = 0.01$ and $N = 4$ predators/prey.
- Predator/prey appearance sequences (constant frequency).

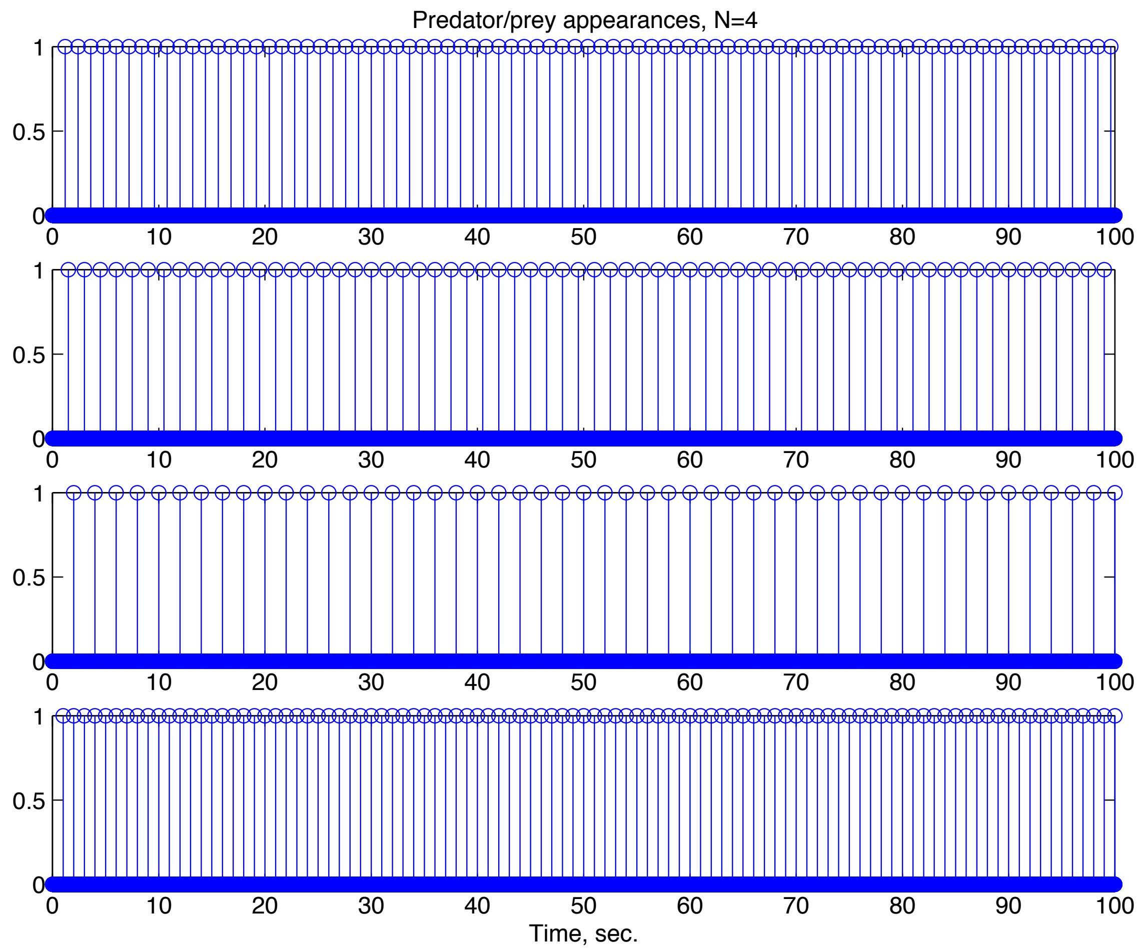

Figure 113: Predator/prey appearance sequences, for $N = 4$ predators/prey (predator/prey $i = 1$ is the top plot, $i = 2$ is the next one down, $i = 3$ is below that, and $i = 4$ is the bottom plot).

➙ Let:

$$\delta^1 = 1.05,\ \delta^2 = 1.15,\ \delta^3 = 1.25,\ \delta^4 = 1.35$$

- Let $\delta_s = 0.03$.

➙ Choose:

$$a_1 = 0.1,\ a_2 = 0.2,\ a_3 = 0.3,\ a_4 = 0.1$$

➙ Performance measures:

$$\frac{1}{N}\sum_{i=1}^{N} T_i(k)$$

and the time average of this quantity.

$$\max_{i=1}\{T_i(k)\}$$

and the time average of this quantity.

- To measure priority focusing:

$$\frac{1}{N}\sum_{k} i^*(k)$$

Attentional Strategy Behavior

- Strategy: Chooses the predator/prey to focus on that has not been detected for the longest period of time.

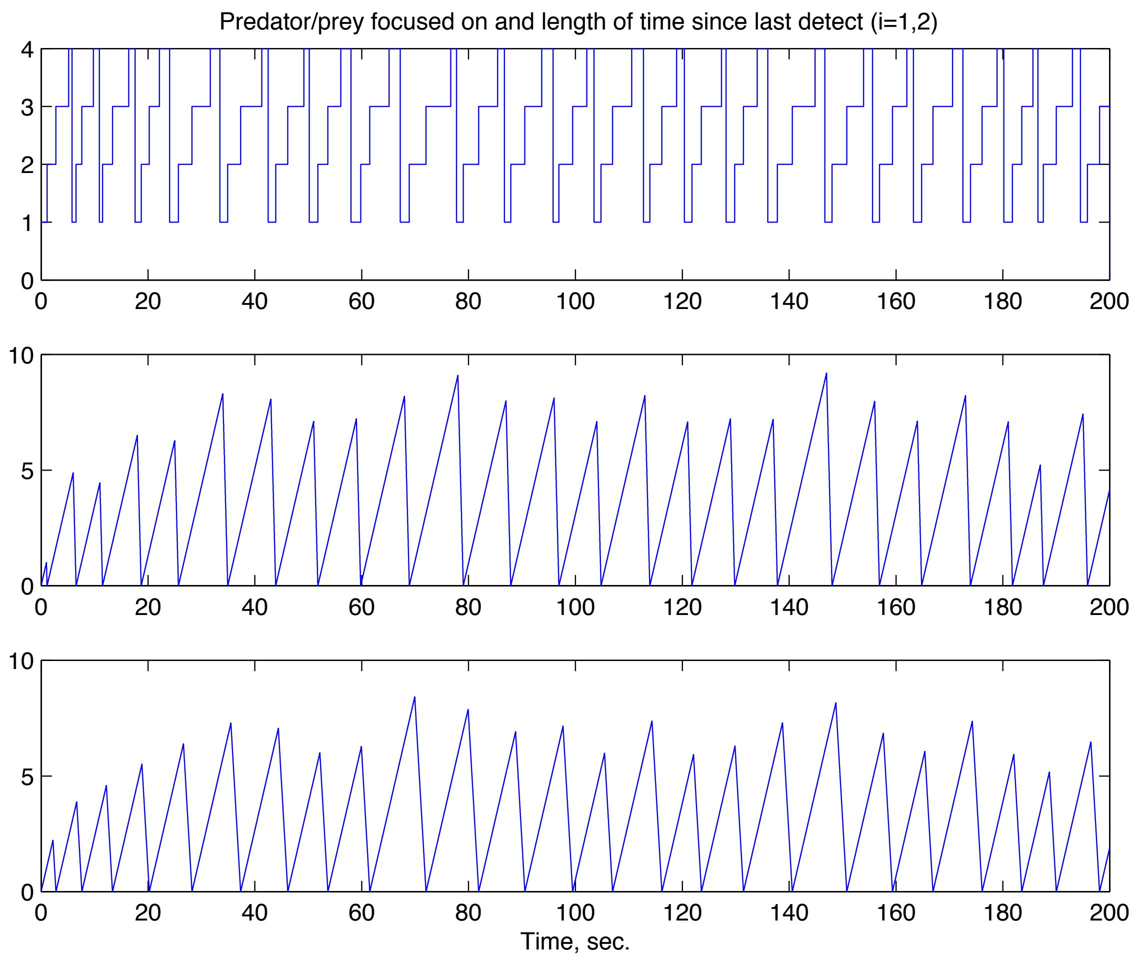

Figure 114: Attention scheduler decisions, and $T_i(t)$ for predators/prey 1 and 2. Notice focusing sequences.

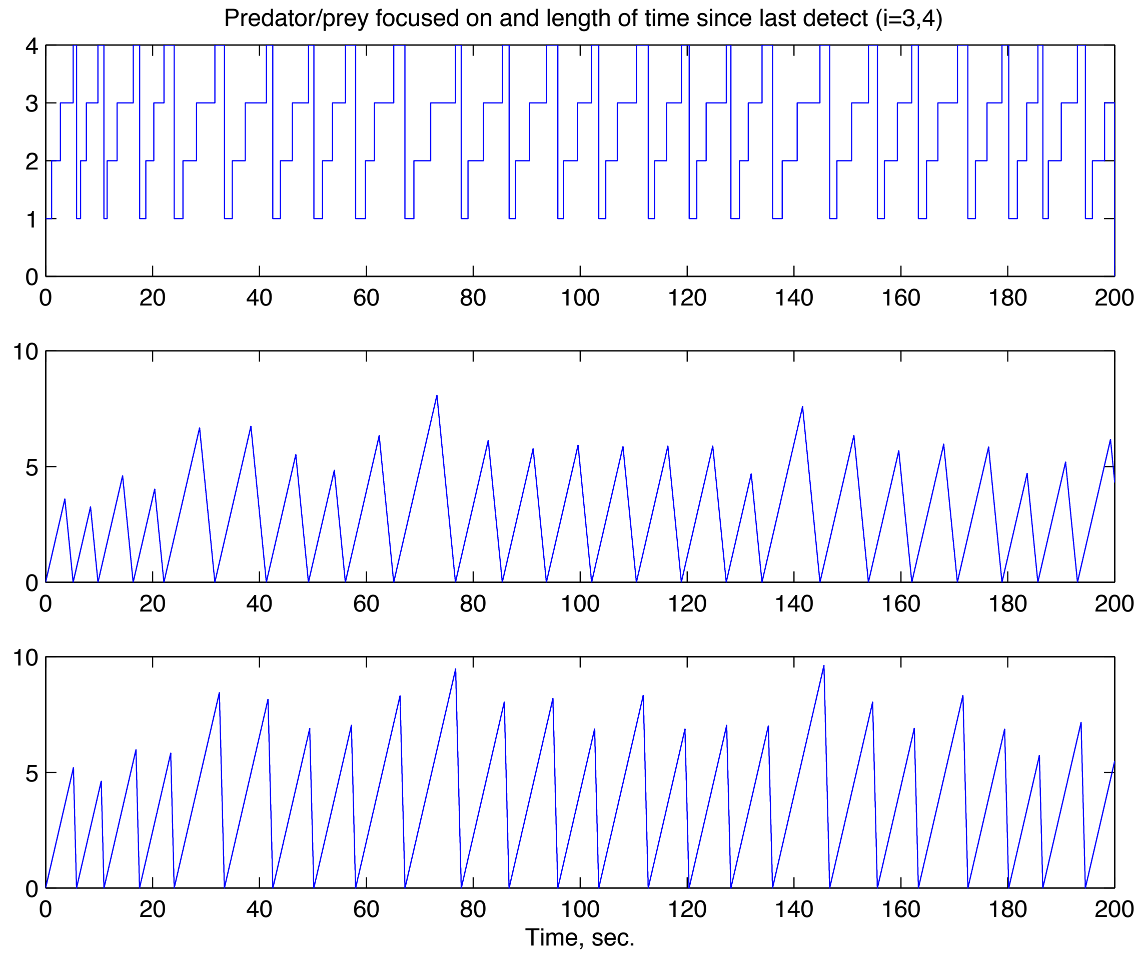

Figure 115: Scheduler decisions, and $T_i(t)$ for predators/prey 3 and 4.

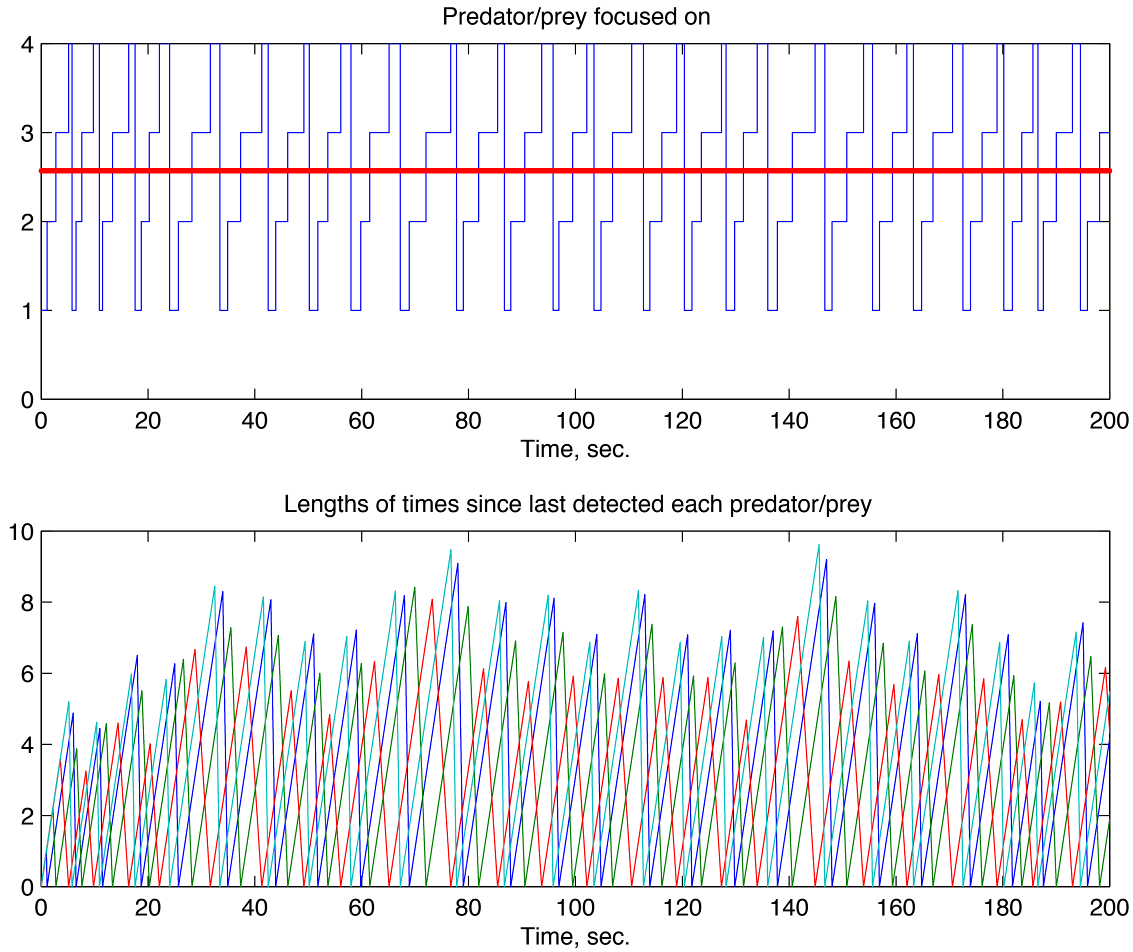

Figure 116: Attention scheduler decisions, and $T_i(t)$ for predators/prey $i = 1, 2, 3, 4$.

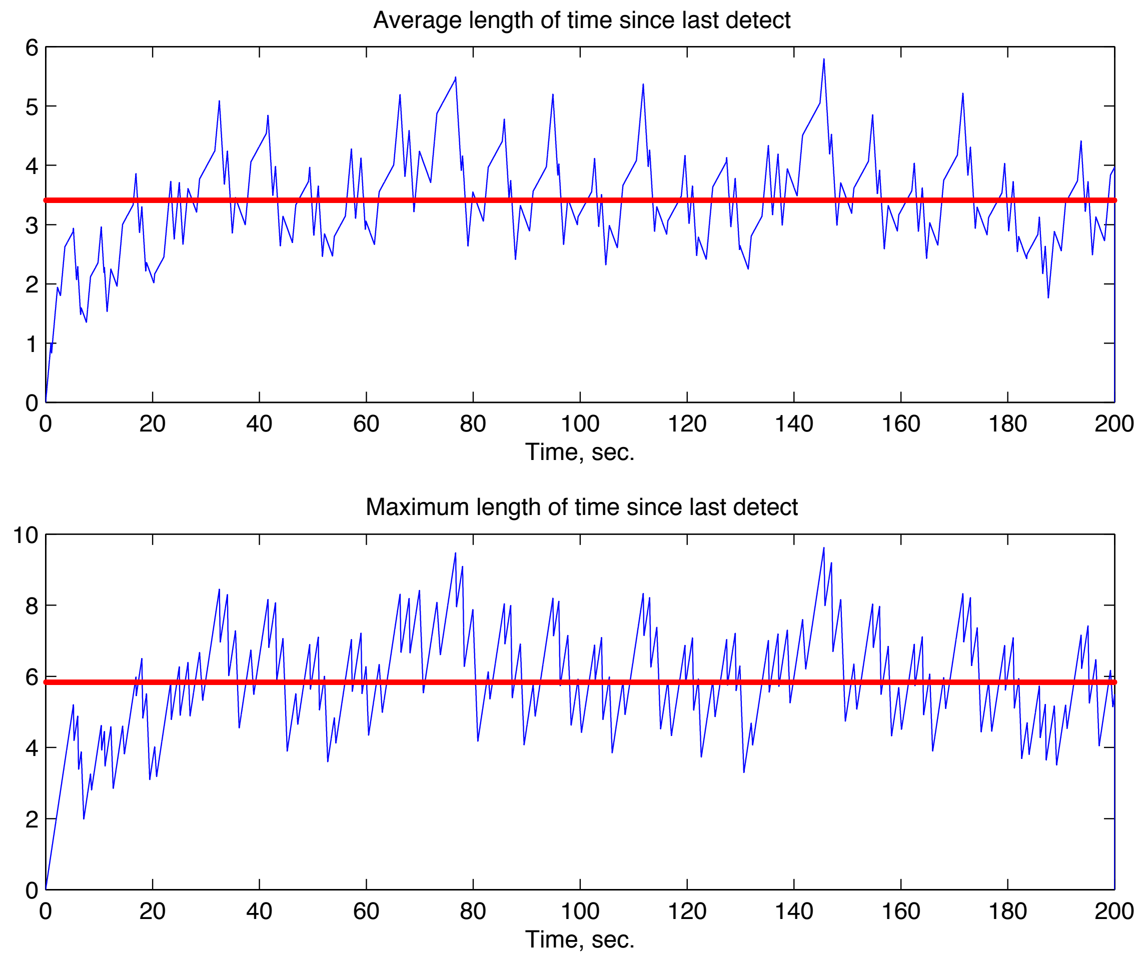

Figure 117: Performance measures (average and maximum times since last detection) and the time averages of their values (3.4066).

- ★ The time average of the average values is 3.4066, and this provides a good measure of scheduler performance. Good performance?

Focusing on High Priority Predators/Prey

- ➙ Strategy: Choose the predator/prey that has been ignored longer than the average one, but with priorities ($i > j$, i higher priority)

- ★ Get different focusing sequences.

- ★ The time average of the average values of the lengths of times waited is 3.8204 (worse)

- ★ Frequent focusing on high priority predators/prey generally requires you to ignore others for longer periods of time.

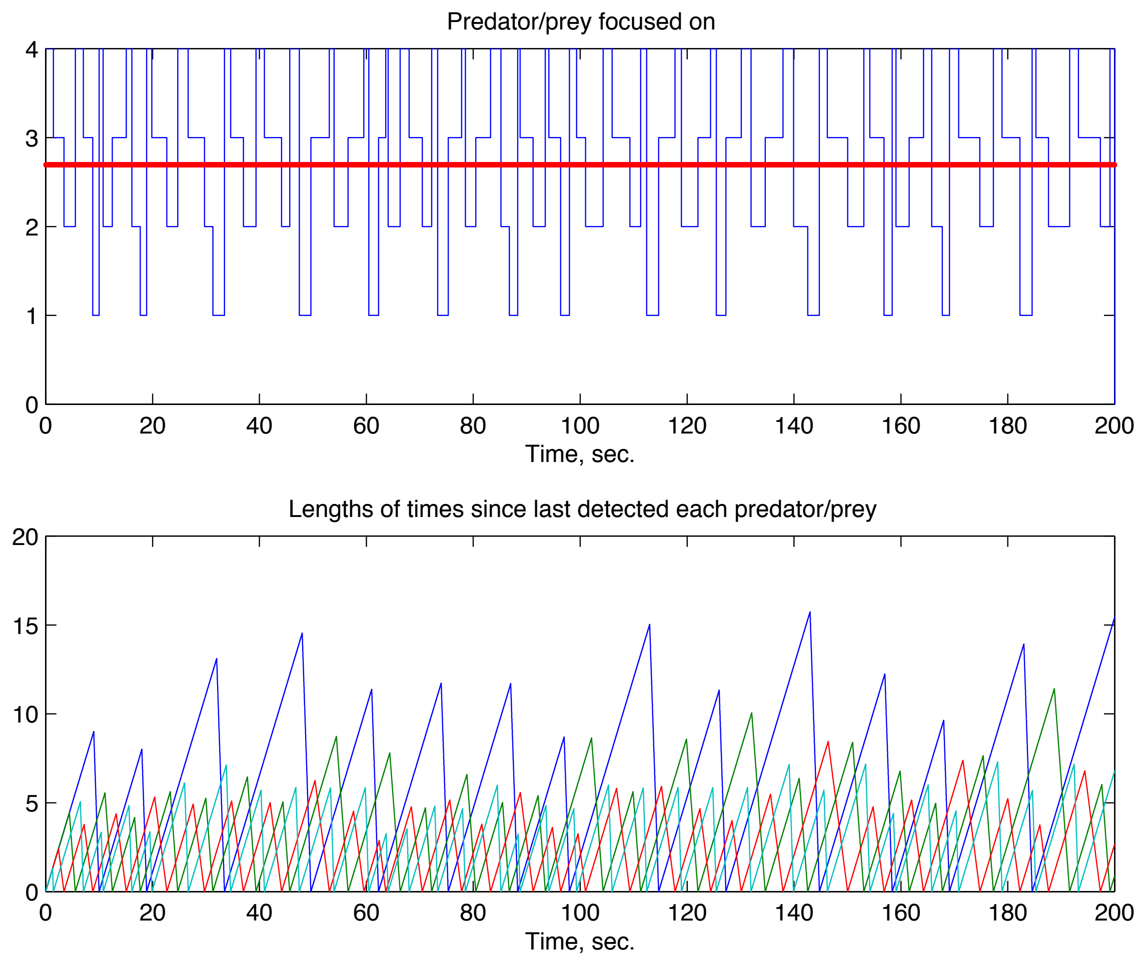

Figure 118: Attention scheduler decisions, and $T_i(t)$ for predators/prey $i = 1, 2, 3, 4$.

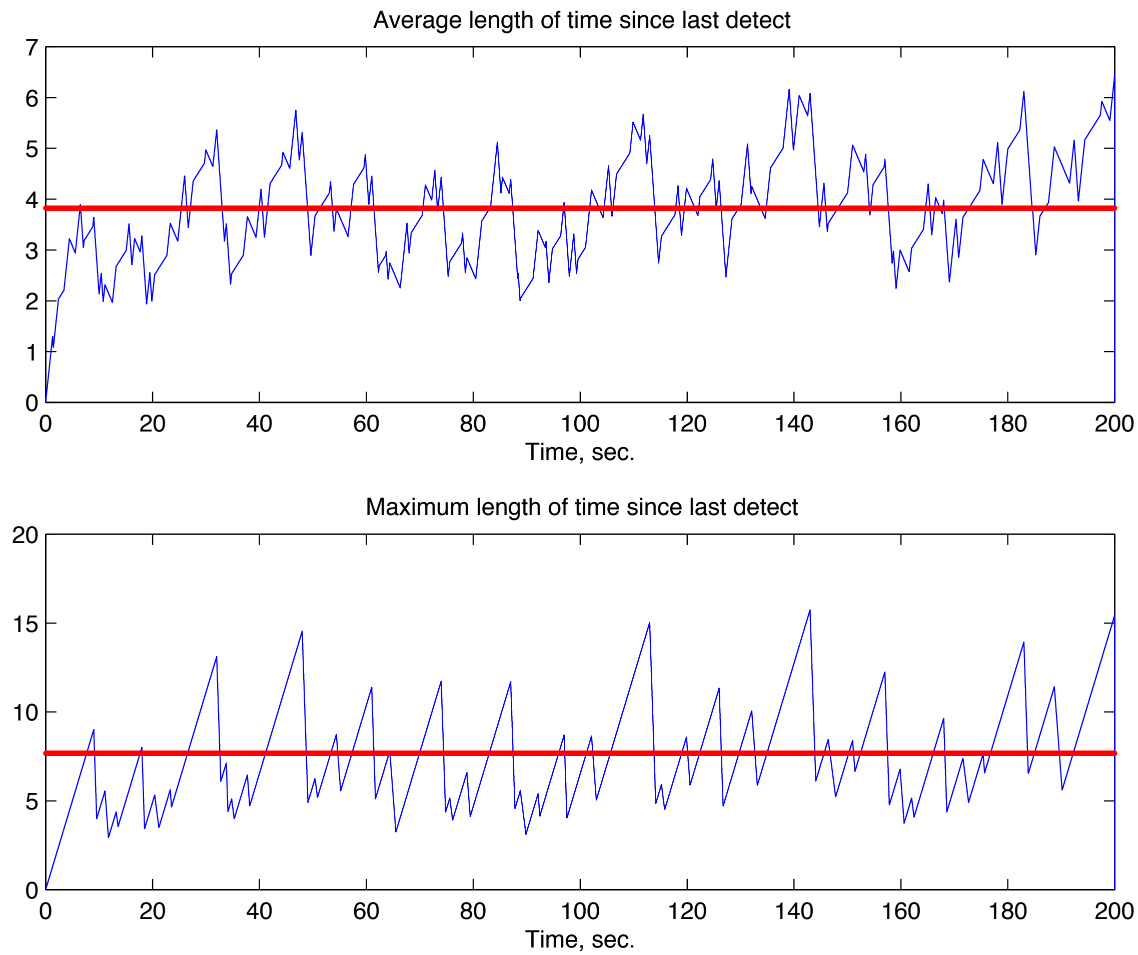

Figure 119: Performance measures (average and maximum times since last detection) and the time averages of their values.

Tuning Attentional Strategy Parameters

- Can we tune the w_i values to improve the performance measures?
- Choose $w_1 = 4$, $w_2 = 2$, $w_3 = 1$, and $w_4 = 4$ (tune via raising weight on predator/prey with higher peaks on $T_i(t)$ values)
- Get the time average of the average values of the lengths of times waited as 3.2755 - better

Stablity Analysis of Attentional Strategies

Stability Properties of Attentional Strategies

Theorem 1: Assume that Equation (20) holds. The attentional strategies where the predator/prey that was ignored the longest time, or one that has been ignored longer than the average one, as defined in Equations (16) and (21), have the following properties:

- They are *stable* in that

  $$\sup_{t \geq 0}\{T_i(t)\} < B_i, i \in P$$

  for some $B_i > 0$, $i \in P$ so that they will not ignore any predator/prey for too long.

- A specific bound on the ultimate longest time that the organism will ignore any predator/prey is given by

$$\lim_{t\to\infty} \sup \sum_{i=1}^{N} T_i(t) \leq \delta \left[\frac{\sum_{i=1}^{N} a_i}{\underline{a}} + \frac{\bar{a}N}{\underline{a}\left(1 - \sum_{i=1}^{N} a_i\right)} \max_i \left\{ \frac{-a_i + \sum_{i=1}^{N} a_i}{a_i} \right\} \right]$$

where $\underline{a} = \min_i\{a_i\}$ and $\bar{a}\max_i\{a_i\}$.

Attention and Control: Synergies

- Neural networks models of attention exist
- Attention for rule-based control, rules for attention
- Attention for planning, planning for attention
- Attention for learning, learning for attention,

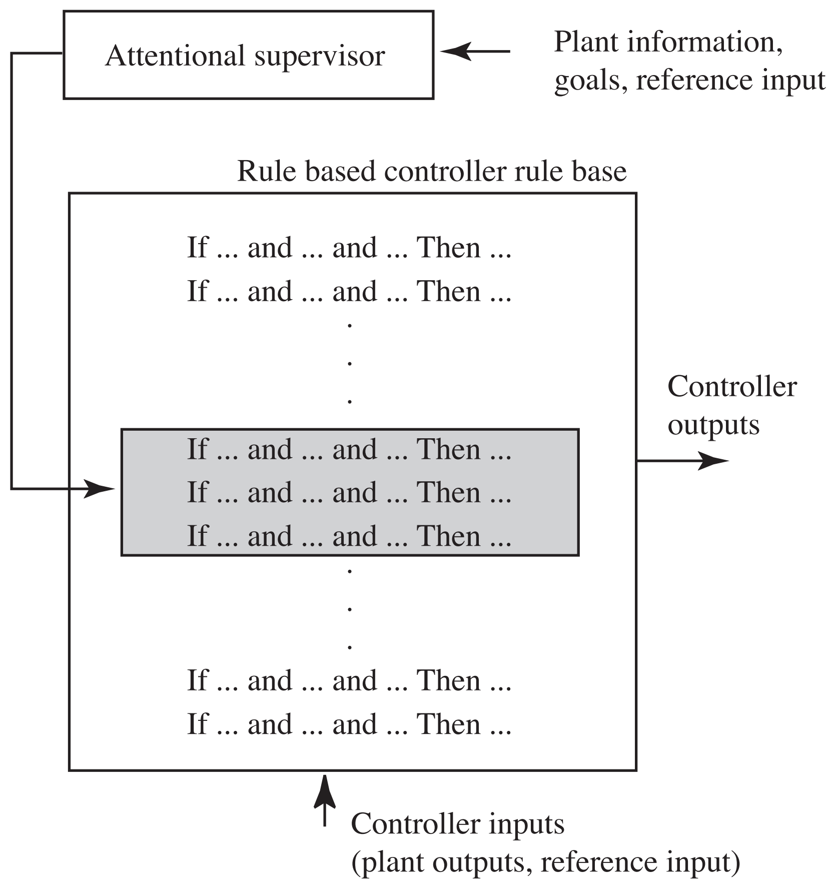

Figure 120: Attentional strategy for rule pruning for rule based control.

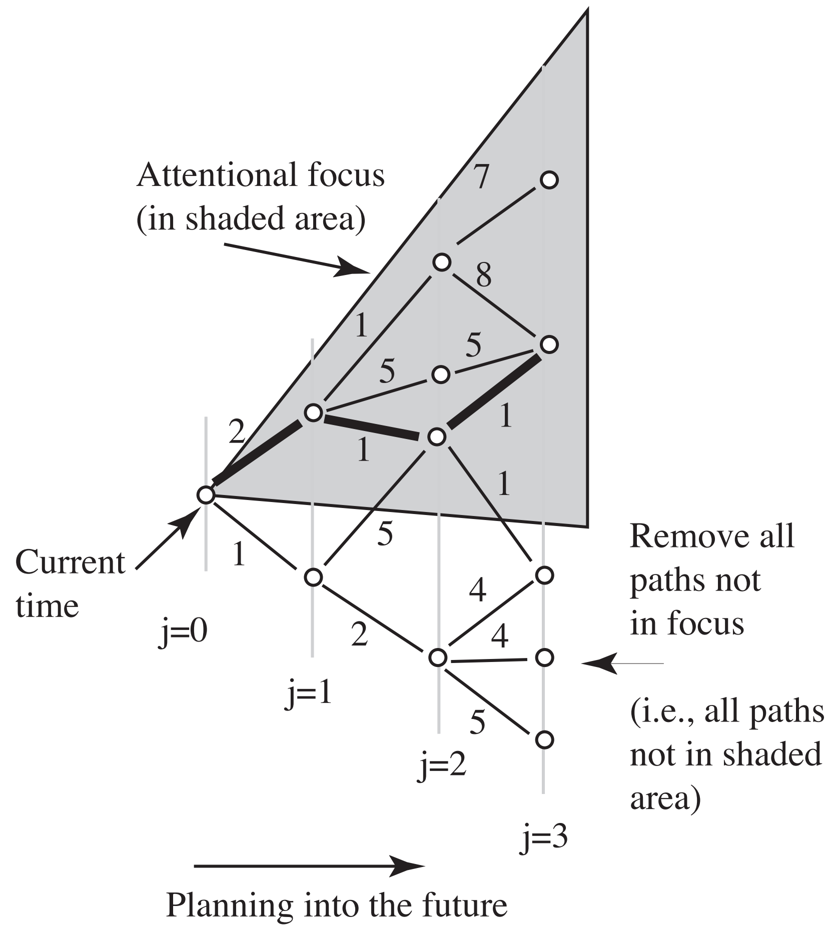

Figure 121: Attentional strategy for plan pruning.

References

[1] Neil A. Campbell. *Biology*. Benjamin Cummings Pub. Co., Menlo Park, CA, 1996.

[2] D. Chrisiansen. Pigeon-guided missles that bombed out. *IEEE Spectrum*, page 46, August 1987.

[3] Michael S. Gazzaniga, Richard B. Ivry, and George R. Mangun. *Cognitive Neuroscience: The Biology of the Mind*. W. W. Norton and Co., NY, 1998.

[4] P. Gray. *Psychology*. Worth Pub., NY, 3 edition, 1999.

[5] M. Groening. *Simpsons Comics: Simpsorama*. HarperPerennial, Kansas City, 1995.

[6] G. Larson. *The Far Side Gallery*. Andrews and McMeel, Kansas City, 1984.

[7] Irwin B. Levitan and Leonard K. Kaczmarek. *The Neuron: Cell and Molecular Biology*. Oxford University Press, Oxford, 2nd edition, 1997.

[8] M.T. Madigan, J.M. Martinko, and J. Parker. *Biology of Microorganisms*. Prentice Hall, NJ, 8 edition, 1997.

[9] K.M. Passino. *Biomimicry for Optimization, Control, and Automation*. ???, ???, ???

[10] P. Whitfield. *From So Simple a Beginning*. Macmillan Pub. Co. NY, 1993.