# Image Cover Sheet



**TITLE**

SOME NEURAL COMPUTING APPROACHES TO TARGET TRACKING

# Some Neural Computing Approaches to Target Tracking

by

Joseph N. Maksym

Defence Research Establishment Atlantic

**Abstract**

Multiple targets, false detections, and manoeuvres present difficulties in the application of tracking and data association algorithms. The discussion examines potential applications of neural networks in target tracking and data association problems. While neural computing has proved useful for many pattern recognition and control applications, it is not yet clear how and where neural networks should be used in systems for tracking targets. Preliminary work on the application of neural networks as adjuncts to more conventional Kalman filter tracking systems will be discussed. This paper examines the possible use of neural networks applied to model selection in a multiple model approach to tracking manoeuvring targets, as well as a neural network approach to measurement-to-track data association.

# Overview

- Kalman Filter Target Tracking
  - Signal model
  - Kalman equations
  - Manoeuvres
  - Data Association
- Neural Networks
  - Optimization networks (Hopfield)
  - Rapid learning (Gaussian Sum)
  - Multisensor track initiation

2 Defence Research Establishment Atlantic

NOTES

The talk is in two parts.

The first part is a kind of introduction to some of the issues involved in target tracking. It reviews a traditional approach – target tracking with Kalman filters.

Some of the issues involved are signal models, the Kalman equations, the effect of manoeuvres, and the problem of data association.

The second part looks at several neural computing paradigms that may have application to target tracking – especially for multiple target multiple sensor situations.

In particular we will examine Hopfield networks, and rapid learning networks applied to a multisensor track initiation problem.

## Multiple Target Stochastic Model

$$\mathbf{x}^t(k+1) = \mathbf{F}^t\,\mathbf{x}^t(k) + \mathbf{G}^t\,\mathbf{u}^t(k) + \mathbf{f}^t(k+1|k)$$

$$\mathbf{z}_j(k) = \mathbf{H}_j\mathbf{x}^t(k) + \mathbf{w}_j(k)$$

Time index: k = 0, 1, ... , N

Target index: t = 1, 2, ... ,S

Measurement index: j = 1, 2, ... , m(k)

3 Defence Research Establishment Atlantic

NOTES

First a review of the Kalman filter approach to multiple target tracking.

This is basically a parametric method. We model the target state (e.g. its position, velocity etc.) by the top equation. In general there is one such stochastic model for each target.

**u**(k) is a zero-mean white Gaussian noise vector with covariance **Q**, and **f**(k+1|k) represents a deterministic input such as a target manoeuvre, and **w**(k) is a zero-mean white Gaussian noise vector with covariance **R**.

At each time step k a number m(k) of measurements are made. This number will often differ from the number of targets because of missed detections and false alarms.

# Kalman Filter State Estimation

Measurement Update* :

$$\hat{x}(k|k) = \hat{x}(k|k-1) + K(k)\,[z(k) - H\,\hat{x}(k|k-1)]$$

$$S(k) = H\,P(k|k-1)\,H^T + R$$

$$K(k) = P(k|k-1)\,H^T\,S(k)^{-1}$$

$$P(k|k)^{-1} = P(k|k-1)^{-1} + H^T R^{-1} H$$

Time Update* :

$$\hat{x}(k+1|k) = F\,\hat{x}(k|k)$$

$$P(k+1|k) = F\,P(k|k)\,F^T + G\,Q\,G^T$$

* Target index t has been dropped for clarity.

4 Defence Research Establishment Atlantic

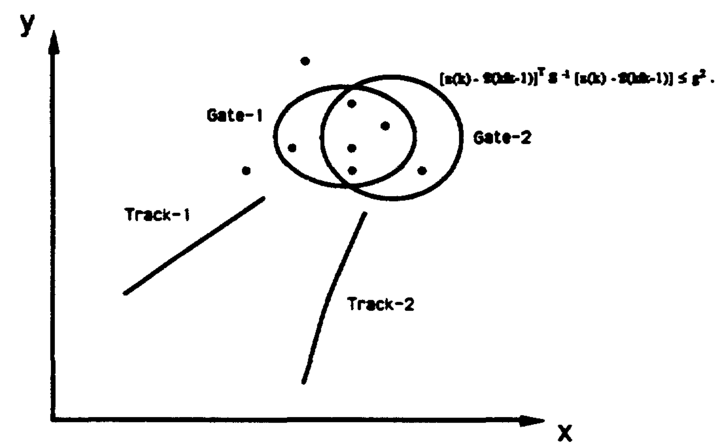
NOTES

The basic Kalman filter equations are summarized here (the target index t has been dropped).

Basically, Kalman tracking consists in picking an initial state for each target, and an initial value for the filter covariance matrix **P**, and then updating these expressions using measurements made at each time step.

There are two steps: a measurement update where the target state and covariance estimates are updated to account for the new measurement, and a time update, which uses the assumed target model to extrapolate to the next time step.

# Data Association Problem




 Defence Research Establishment Atlantic

NOTES

Multiple targets complicate the problem of data association, as illustrated here for a simple situation with two converging tracks.

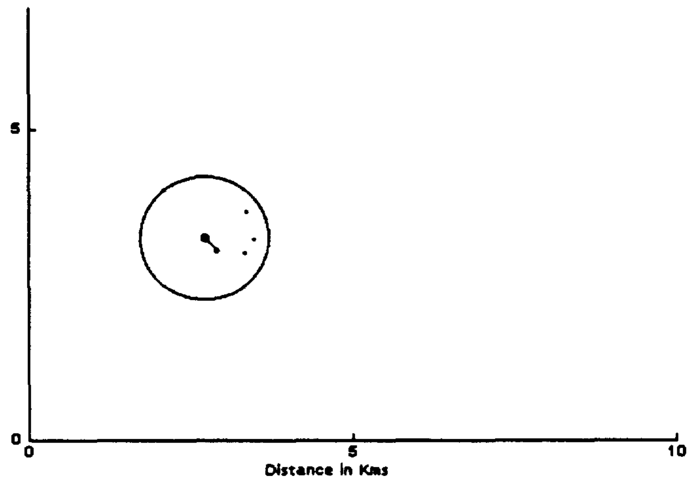
The problem is to decide which observation is to be associated with each track.

Each Kalman filter computes a gate region centred on the expected measurement (Size of the gates depends upon the target stochastic model as well as measurement noise).

Normally the gate size is chosen such that the true target return will be inside the gate, say 0.999 of the time.

A very simple data association techniques is the nearest neighbour method, which minimizes the distance of the chosen measurement from the prediction made by the filter. The distance measure illustrated in the viewgraph can be used.

# Nearest Neighbour: k = 3




6 Defence Research Establishment Atlantic

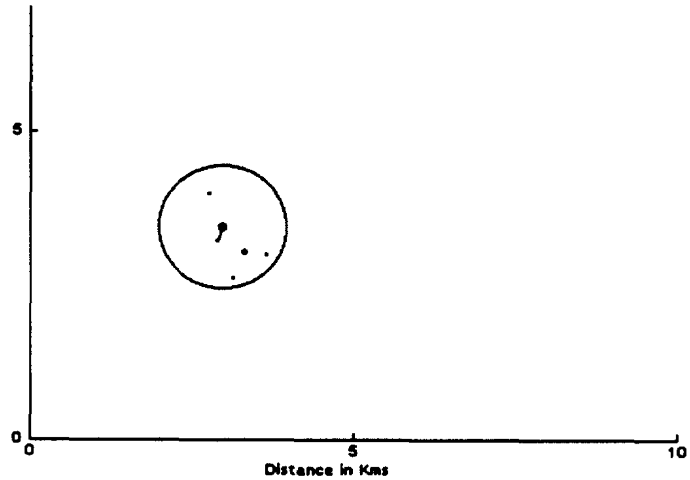
NOTES

This viewgraph illustrates the nearest neighbour method in a simulation.

At time = 3 we have three false alarms (small dots) and the true return inside the gate.

The nearest-neighbour associated the true return with the track.

## Nearest Neighbour: k = 4

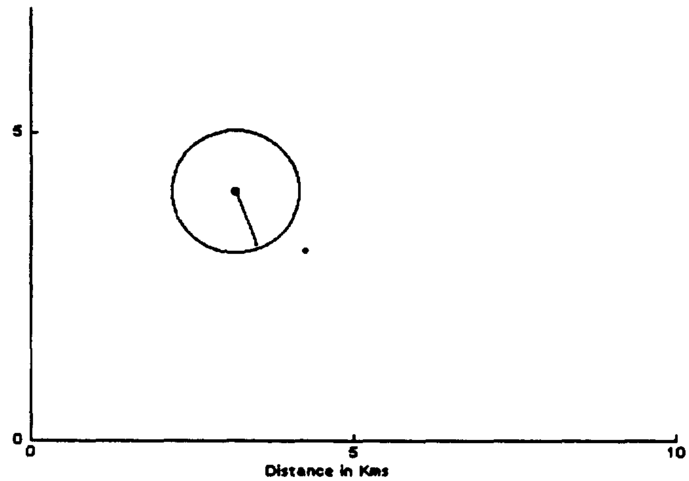



7 Defence Research Establishment Atlantic

NOTES

At time = 4 the nearest measurement is a false alarm.

# Nearest Neighbour: k = 6



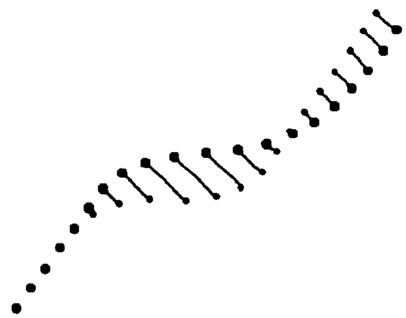

Defence Research Establishment Atlantic

8

NOTES

At time = 6 the filter loses the true return – it falls outside the gate.

# Tracking a Manoeuvring Target



9

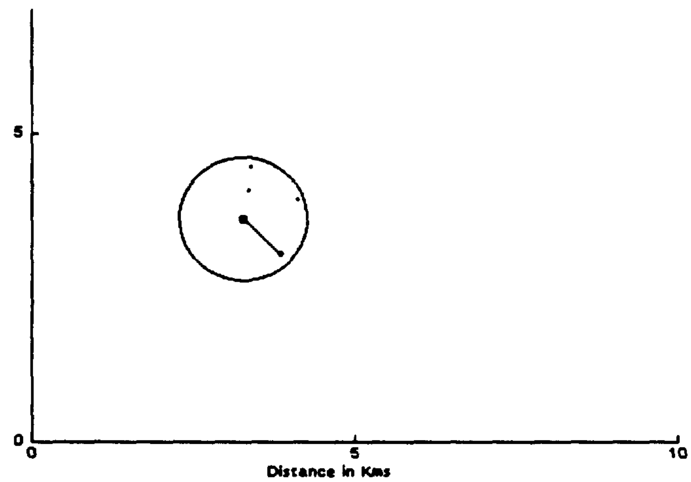


NOTES

Clearly, a Kalman filter based on assumed dynamics (constant velocity target, for example) is mismatched when a strong manoeuvre occurs.

This is shown here. A good deal of work on adaptive filters is reported in the literature on Kalman filter tracking. Most of these adaptive studies propose some form of system identification, which models the non-white residual error occurring during a manoeuvre.

## Modified Nearest: k = 5




10 Defence Research Establishment Atlantic

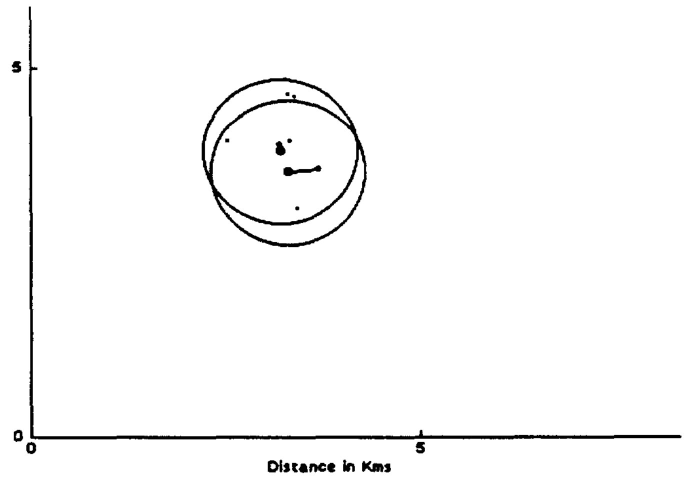
NOTES

Basically, when a residual associated with a measurement is highly correlated with past residuals the measurement is a good choice for association.

A simple technique for improving nearest neighbour data association can be based on this concept.

The viewgraph illustrates how the modified nearest neighbour tends to associate the true return rather than a false alarm, which happens to be closer to the expected measurement.

# Two Targets: k = 5



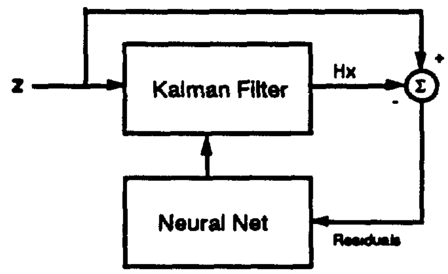

11 Defence Research Establishment Atlantic

NOTES

In this viewgraph an example with two crossing targets is shown.

The modified nearest neighbour association can assist in preventing target returns from being associated with the wrong track.

# Manoeuvre Identification






## NOTES

The examples we have just seen indicate the importance of having a correct model for a target – during manoeuvres the Kalman filter becomes mismatched.

The use of neural networks for system identification is now well established.

The viewgraph indicates, conceptually, how it might work in a Kalman filter tracker. Here the neural net would be used as an adjunct to a Kalman filter.

There are several problems, however:

1. Standard backpropagation networks have long training times and would need to be trained off-line.
2. A large variety of manoeuvres are possible.

# Neural Networks

## Some Network Types

- Multi-layer Backpropagation
- Hopfield Networks (energy minimisation)
- Self-Organizing Nets (Kohonen, ART)
- Probabilistic Networks

13 Defence Research Establishment Atlantic

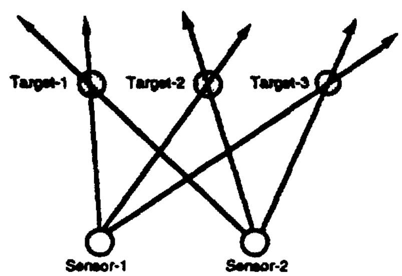
NOTES

The remainder of the talk will look at some neural network concepts for multiple sensor target tracking.

Some of the neural architectures considered were the multilayer backpropagation network, energy minimisation networks such as the Hopfield, self-organizing nets, and probabilistic networks.

The discussion will focus on Hopfield and Probabilistic networks.

# Multi-sensor Multi-target Tracking




14 Defence Research Establishment Atlantic

NOTES

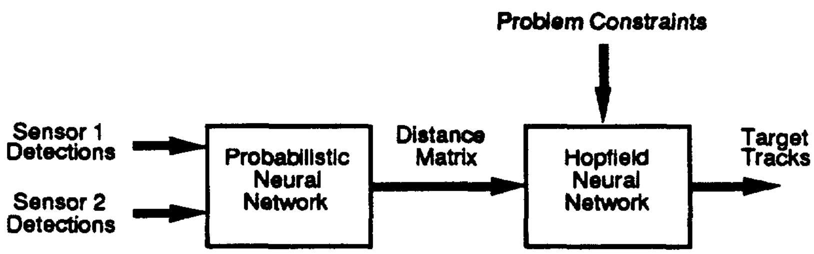
Consider a scenario consisting of three targets and two sensors that produce only a bearing indication for each target. In practice, noise will introduce bearing errors, false detections, and missed detections, but these issues will be ignored for the moment.

Given the set of detected bearings at each sensor the tracking problem is to determine the location of the three targets.

There are two aspects of the problem: data fusion, and constrained optimization.

We will examine a system that combines two neural architectures, a probabilistic network and a Hopfield network.

# Multi-sensor Data Fusion and Tracking System




15 Defence Research Establishment Atlantic

## NOTES

The inputs to the probabilistic network are the sets of detections from the sensors.

If a detection at one sensor and a detection at another sensor are similar the probabilistic network produces a high output – that is the distance measure between feature vectors is small.

The resulting distance matrix is passed to a Hopfield network where it determines, in part, the interconnection weights.

The Hopfield interconnection weights are also influenced by problem constraints.

Problem constraints might include a limit of one target present on each detected bearing, an impossibility of targets in a given region, etc.

# Probabilistic Neural Network

$$\Omega = \{\omega_1, \omega_2, \dots \omega_s\}$$

$$\hat{p}(\mathbf{x} \mid \omega) = \frac{1}{(2\pi)^{\frac{n}{2}} \sigma^n m} \sum_{k=1}^{m} \exp\left[-\frac{\|\mathbf{x}-\mathbf{x}_{k\omega}\|^2}{2\sigma^2}\right],$$

$$g_i(\mathbf{x}) = p(\mathbf{x} \mid \omega_i)\, p(\omega_i), \qquad i = 1, 2, \dots, S$$



NOTES

This briefly summarizes a probabilistic network (It may also be referred to as a Gaussian sum or Parzen window approach).

Suppose we have S classes of pattern, and that within each class there is some variation, although smaller than variation between classes.

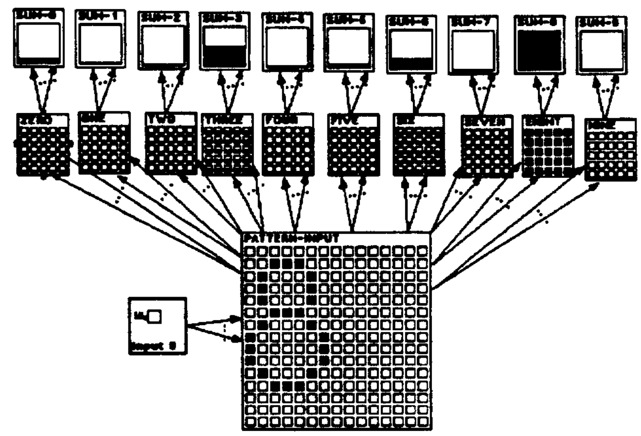
We can develop a density function for the feature vector **x** for a class as the summation of some number, say m, of individual Gaussian densities, each centred on an example feature vector.

The resulting "Gaussian sum" thus reflects the (variations) within the class.

Training the network is rapid – a set of network weights are stored for each training exemplar.

After training, the network computes discriminants $g_i(\mathbf{x})$ for each class.

# Character-Recognition Network



17 Defence Research Establishment Atlantic

## NOTES

A character-recognition example illustrates the implementation.

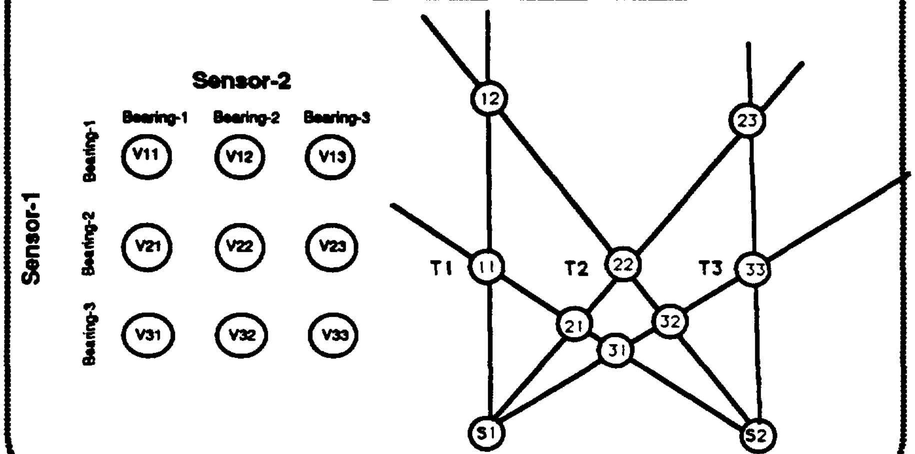
An square array of neural units receives input characters.

There is a second layer of units for each class. The links from each unit in the second layer store example patterns during the learning phase. For this network there are twenty-five units in the second layer for each class. Consequently, up to twenty-five exemplars can be stored.

The top layer is a set of single units. Each unit forms the Gaussian sum. The output values (height of black bar) indicate probability of the input belonging to the various classes.

Here, the trained network is presented the character "eight". The largest output is that corresponding to the class "eight", although some output is observed for the classes "three" and "six" due to pattern similarities.

# Hopfield Net for Two Sensor Track Initiation






## NOTES

The viewgraph illustrates a mapping between potential target locations and neurons in a two-dimensional Hopfield net.

The target location T1, for, example maps into the neuron labelled v11.

With three detections on each sensor there will be nine potential target locations consistent with the observed detections.

## Network Design

Cost function:

$$E = \frac{A}{2}\sum_{i=1}^{3}\sum_{j=1}^{3}\sum_{\substack{k=1\\k\neq j}}^{3} v_{ij}v_{ik} + \frac{B}{2}\sum_{i=1}^{3}\sum_{j=1}^{3}\sum_{\substack{l=1\\l\neq i}}^{3} v_{ij}v_{lj} + \frac{C}{2}\sum_{i=1}^{3}\sum_{j=1}^{3} d_{ij}v_{ij}$$

Minimisation favours the following:

- Network states with at most one "on" neuron per row.
- Network states with at most one "on" neuron per column.
- Network states where "on" neurons correspond to a low distance measure $d_{ij}$ between detection-i on sensor-1 and detection-j on sensor-2.

19 Defence Research Establishment Atlantic

**NOTES**

Hopfield network design is usually based on definition of an energy or cost function that the network is expected to minimize.

Shown here is a possible cost function for the two-sensor three-target problem.

Basically, the network is to select those target locations that have maximum distance of target features subject to satisfying constraints of the problem.

In most applications of the Hopfield net seen in the literature, constants A, B, C, etc. were empirically adjusted for good performance. This is seen here as potential source of difficulty.

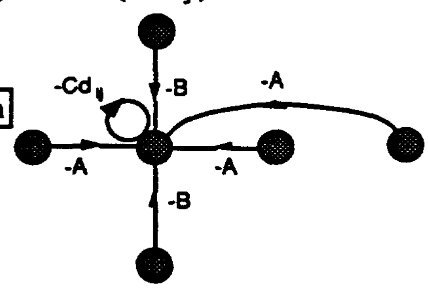
## Interconnections

$$E = \frac{A}{2}\sum_{i=1}^{3}\sum_{j=1}^{3}\sum_{\substack{k=1\\k\neq j}}^{3} v_{ij}v_{ik} + \frac{B}{2}\sum_{i=1}^{3}\sum_{j=1}^{3}\sum_{\substack{l=1\\l\neq i}}^{3} v_{ij}v_{lj} + \frac{C}{2}\sum_{i=1}^{3}\sum_{j=1}^{3} d_{ij}v_{ij}$$

Calculation of $T_{ijkl}$

$$T_{ijkl} = -A\,\delta_{jl}\left(1-\delta_{ik}\right) - B\,\delta_{ik}\left(1-\delta_{jl}\right) - C\delta_{ik}$$

Connections into the $ij^{th}$ neuron




20 Defence Research Establishment Atlantic

## NOTES

The form of the energy function translates into link weights.

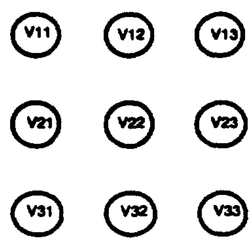
This is shown in the sketch, which indicates links going into the ij-th neuron.

First, there are inhibitory links of strength A acting along each row.

Next, there are inhibitory links of strength B acting along each column.

Finally, each neuron has a cyclic link (from its output back to its input). The strength of this link is dependent on distance – a high distance.

# Difference Equations




$$v_{ij} = g(u_{ij}) = \frac{1}{2}\left(1 + \tanh(\lambda u_{ij})\right)$$

$$u_{ij}(t+1) = u_{ij}(t) + \Delta u_{ij}$$

$$\Delta u_{ij} = \left(-\frac{u_{ij}}{\tau} + \sum_{k=1}^{3} \sum_{l=1}^{3} T_{ijkl} v_{kl} + I_{ij}\right)$$

21 Defence Research Establishment Atlantic

## NOTES

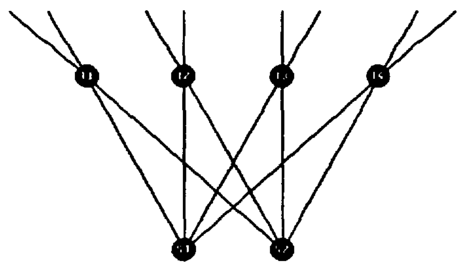
This viewgraph shows the difference equations for updating the network.

Each neuron produces an output v(i,j) that is a nonlinear function of its input u(i,j). The nonlinearity is usually a soft limiter, such as the expression at the top of the viewgraph. Lambda is a gain constant, which determines the shape of the nonlinearity.

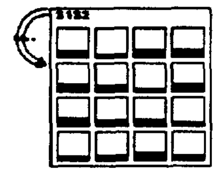
At each iteration the input to a neuron is the sum of: the previous input decayed with a time constant, a weighted sum of the outputs of all neurons in the network, and an input current I.

Actually, Tank and Hopfield's original papers described an analogue implementation involving operational amplifiers, resistors, and capacitors.

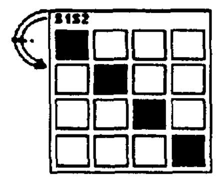
# Two Sensor Four Target Example



Example Scenario



Initial network state



Final network state

22 Defence Research Establishment Atlantic

## NOTES

The action of a small Hopfield network for a two-sensor problem is shown here:

Operation is as follows:

Detections are received from each sensor – these determine the size of the Hopfield net.

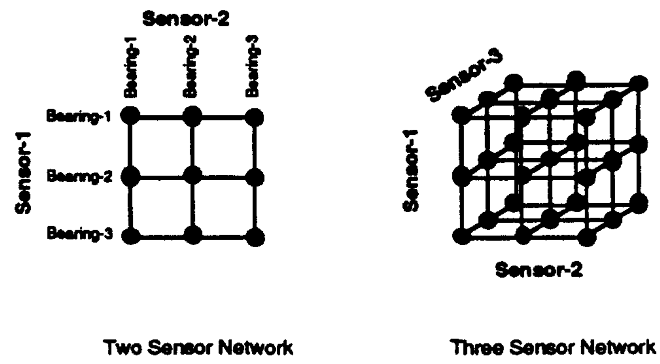
A probabilistic network is used to learn probability distributions for detections at each sensor. It also computes distance, i.e. how likely is it that a detection on sensor 1 represents the same target as a detection on sensor 2.

The distance matrix, together with auxiliary constraints determines the interconnections of the Hopfield net.

The Hopfield net is then initialized with small random values.

The network then iterates, reaching a stable state, which should indicate the most likely target locations.

# Two Sensor and Three Sensor Hopfield Networks




23 Defence Research Establishment Atlantic

NOTES

The approach used for the two-sensor problem requires a Hopfield network that is a two-dimensional array of neurons. The size of the array will, however, depend on the number of detections at each sensor.

For a three-sensor problem, a three-dimensional array of neurons is required – one dimension per sensor.

# Further Work

- Problem of convergence to local minima
- Combining track initiation with track maintenance
- Feature selection at sensors
- Comparison with more traditional methods of constrained optimization

24 Defence Research Establishment Atlantic

NOTES

Some directions for further work are indicated here.

One potential problem is the possibility of convergence of Hopfield networks to a local minimum. Some techniques that might be used to avoid this have been reported in the literature.

Another potential direction is to extend the neural network track initiation system to that of track maintenance. Extension of the track initiation system to multiple times as well as multiple sensors is one possible approach.

An important issue is feature selection. Obviously a Gaussian sum network that stores a complete Lofargram for each detection is not practical.

Another question is whether traditional techniques such as linear programming are in fact better and more reliable. Certainly, for the simple examples discussed in this talk a Hopfield net, or other optimizer, is hardly necessary – simple search will do.