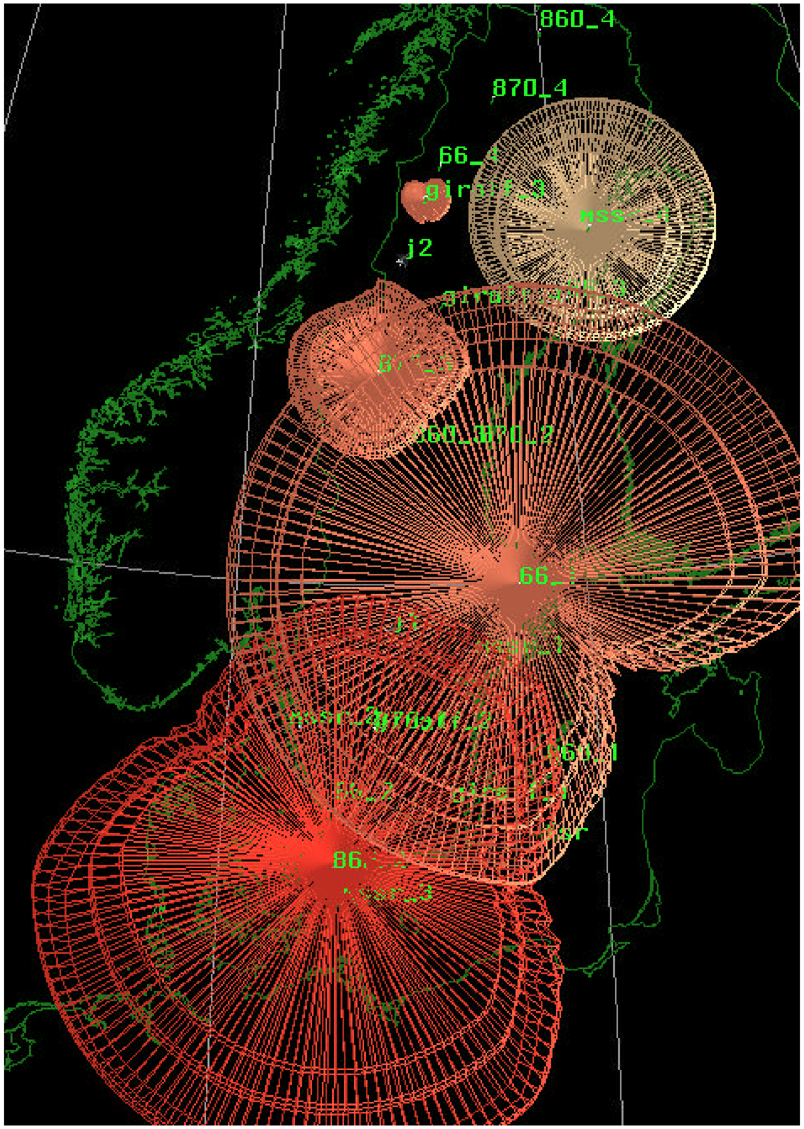



# Principer för multisensor-målföljning

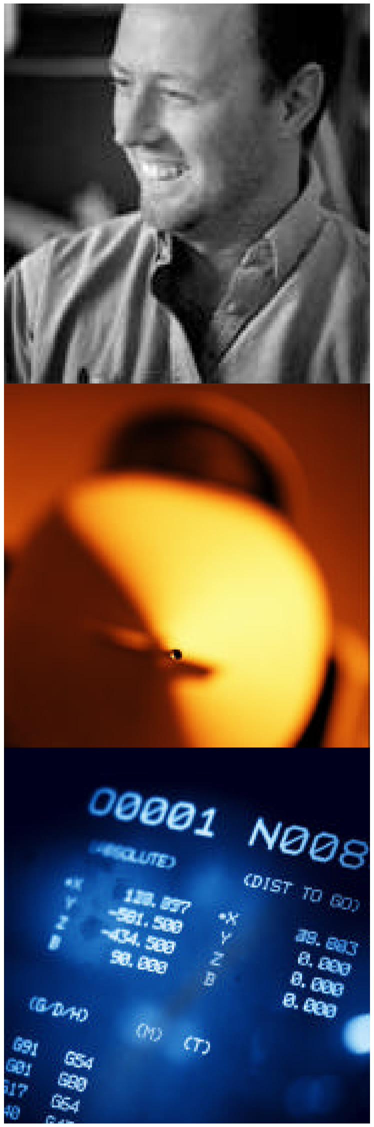
Niclas Bergman

Avd. för datafusion

SaabTech Systems, Järfälla

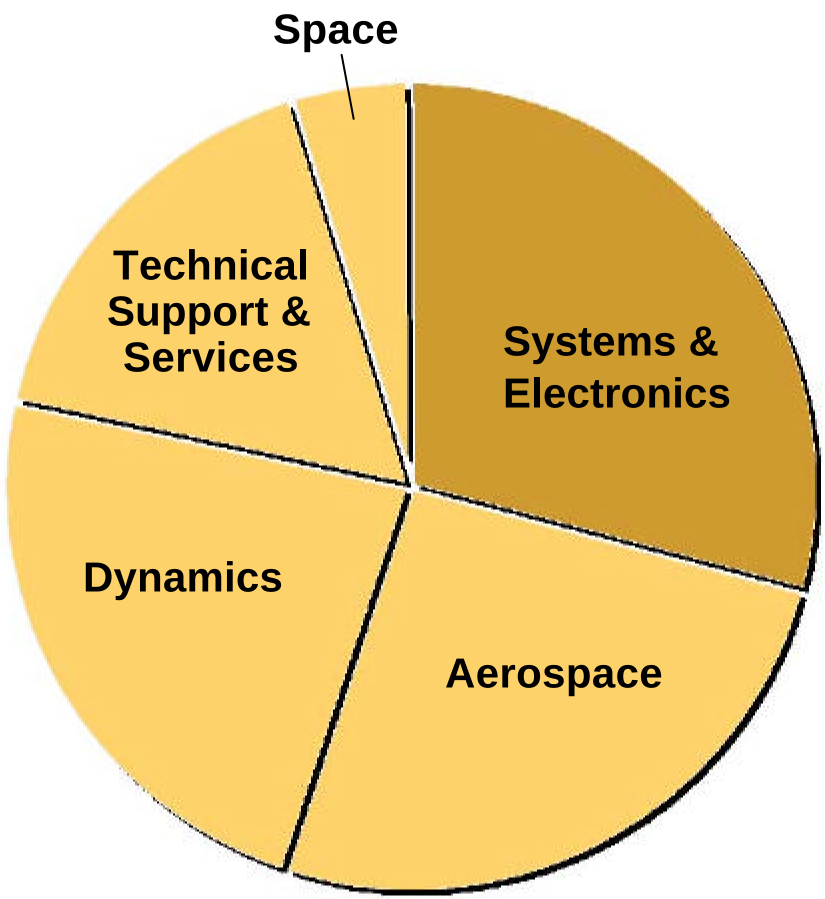
# Agenda

- SaabTech Systems
  - Företaget, produkter, teknikområde
- Målföljning
  - Algoritmer och tekniker som används inom målföljning
  - Exempel
- Sammanfattning




**SAAB**

KTH 2001-03-29




**Saab's core business areas**

Saab AB Pro forma 1999:

| | |
|---|---|
| Sales, MSEK | 19,264 |
| No. of employees | 17,213 |

# Saab Systems & Electronics

**SYSTEMS & ELECTRONICS**

KTH 2001-03-29

# Saab Systems & Electronics Areas

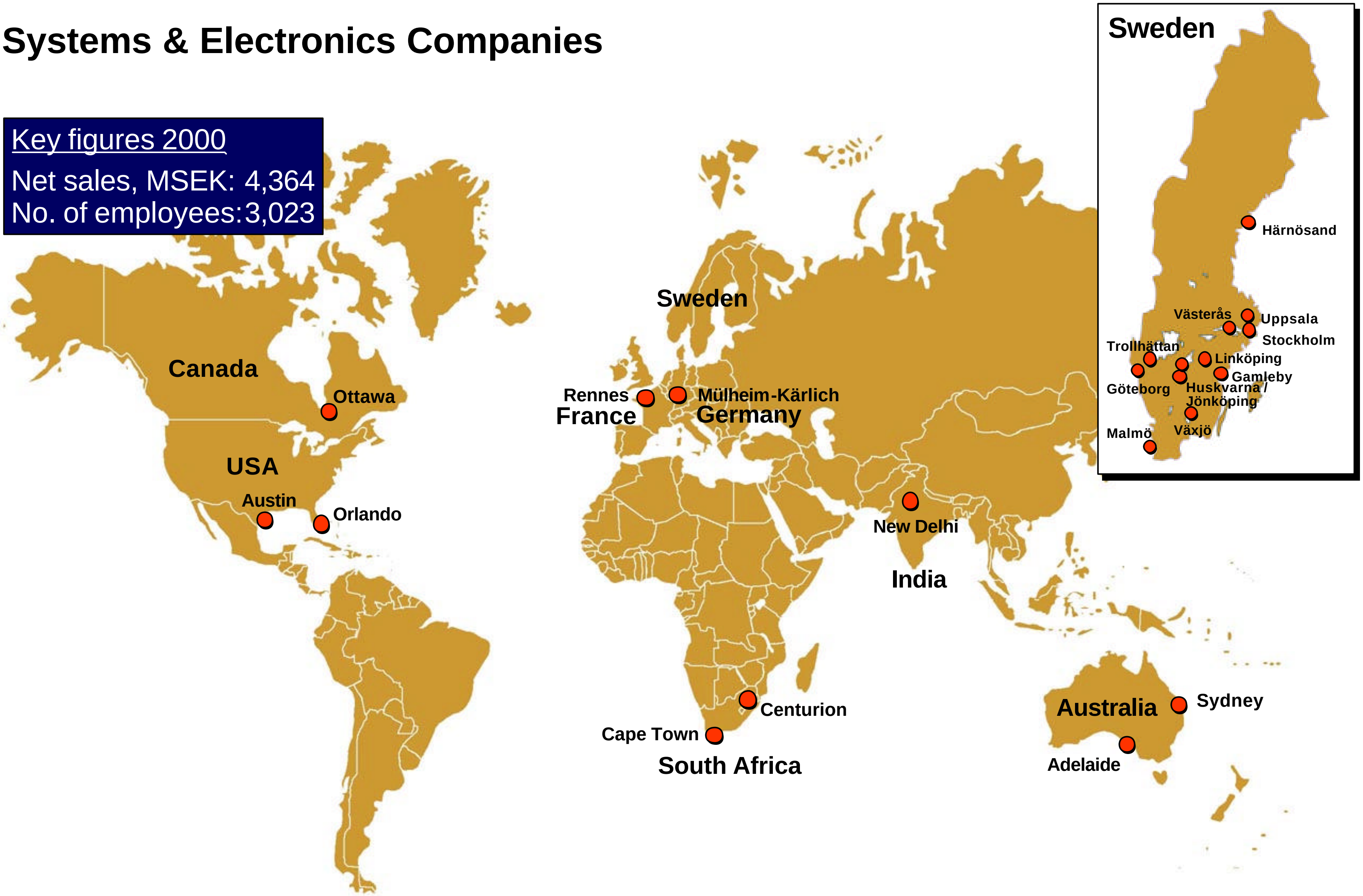
- Command and Control Systems
- Avionics
- Electronic Warfare
- Simulation and Training
- Signature Management
- Spin-Off Applications

# Systems & Electronics Companies

Key figures 2000

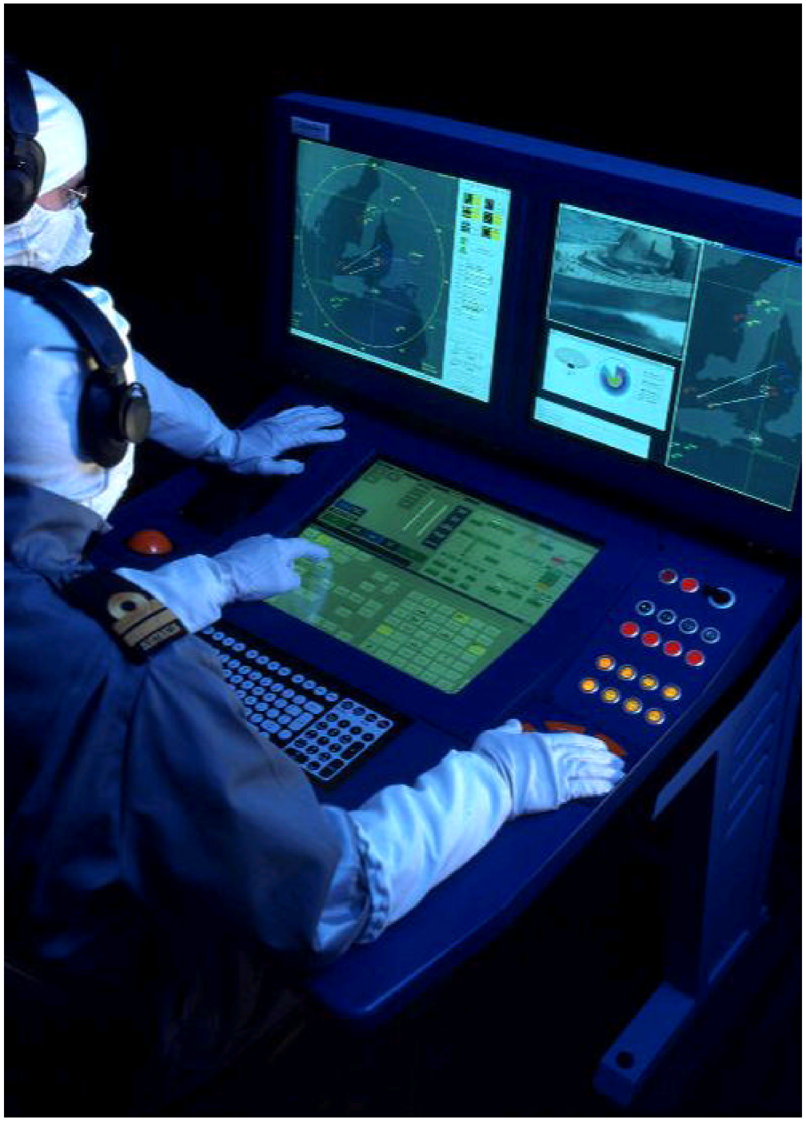
Net sales, MSEK: 4,364

No. of employees: 3,023




SYSTEMS & ELECTRONICS

KTH 2001-03-29



# Command and Control ($C^2$) Systems

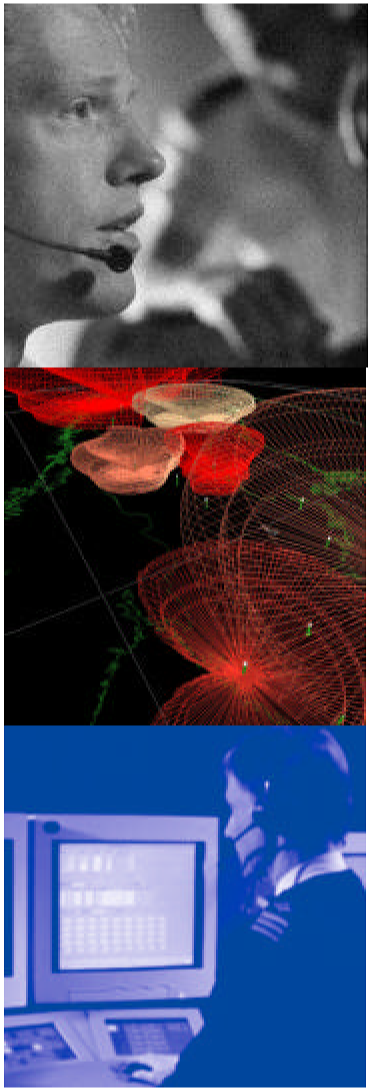
Command superiority for armed forces enabled by world class functionality such as:

- Battle Force Planning
- Decision Support
- Automatic Air and Area Defence
- Wide Area Situation Picture
- Data Fusion
- Weapon Control

# Saab Netdefense

**SYSTEMS & ELECTRONICS**

KTH 2001-03-29



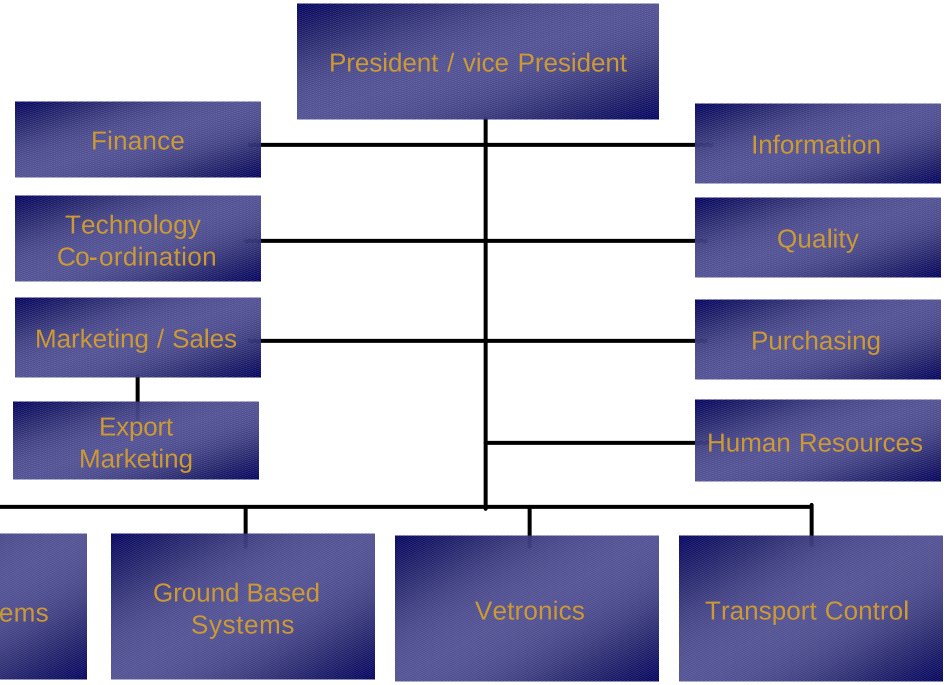
# SaabTech Systems

**SAABTECH SYSTEMS**

KTH 2001-03-29

# Organisation

*The Järfälla Head Office*




| | Saab Systems | Naval Systems | Ground Based Systems | Vetronics | Transport Control |
|---|---|---|---|---|---|
| Technical Manager | X | X | X | X | X |
| Data Fusion | | | X | | |
| Anti Air Warfare | | X | | | |
| Combat Vehicle Warfare | | | | X | |
| Modelling & Simulation | | | X | | |
| Data Links | X | | | | |
| Network Communication | | | | | X |

# Product Line

- Naval Combat Management Systems
- Weapon Control Systems
- Coastal Defence Systems
- Mine Hunting and Mine Clearance Systems

---

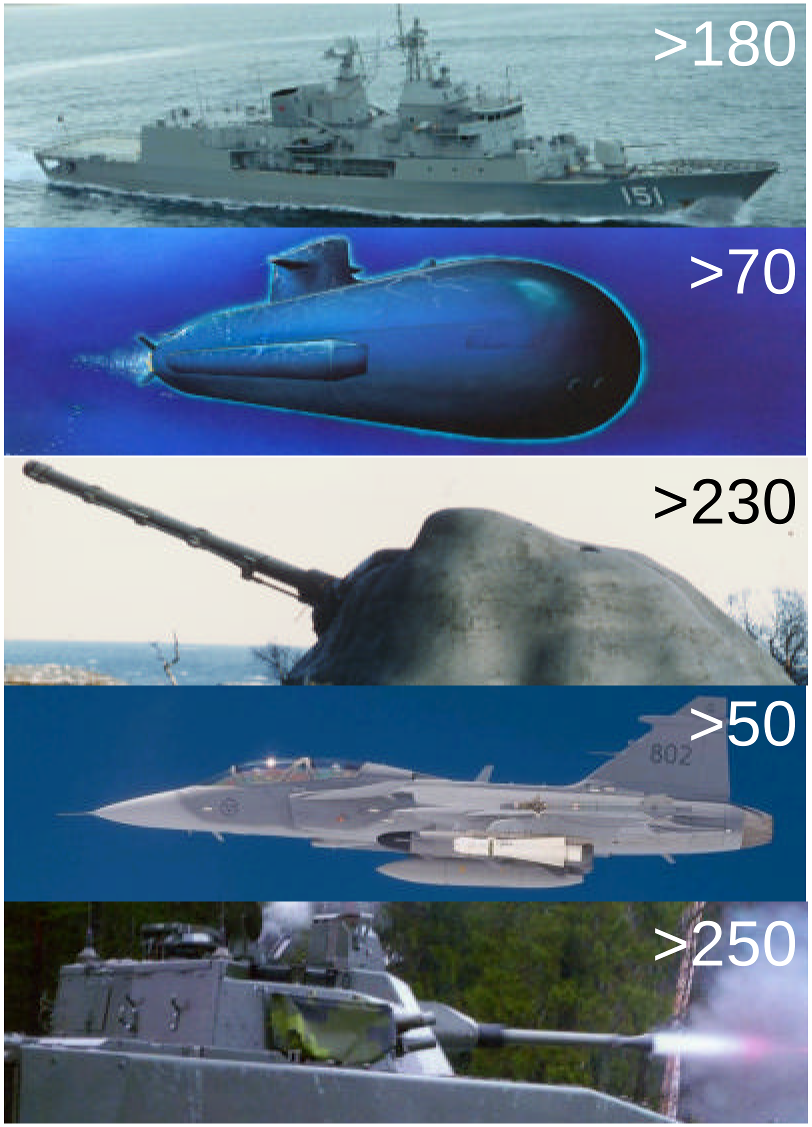
- Air Defence Ground Environment Systems
- Simulators
- Data Fusion
- Short Term Conflict Alert

---

- $C^3$ Systems
- Fire Control Systems
- Sensor & Countermeasure Systems

---

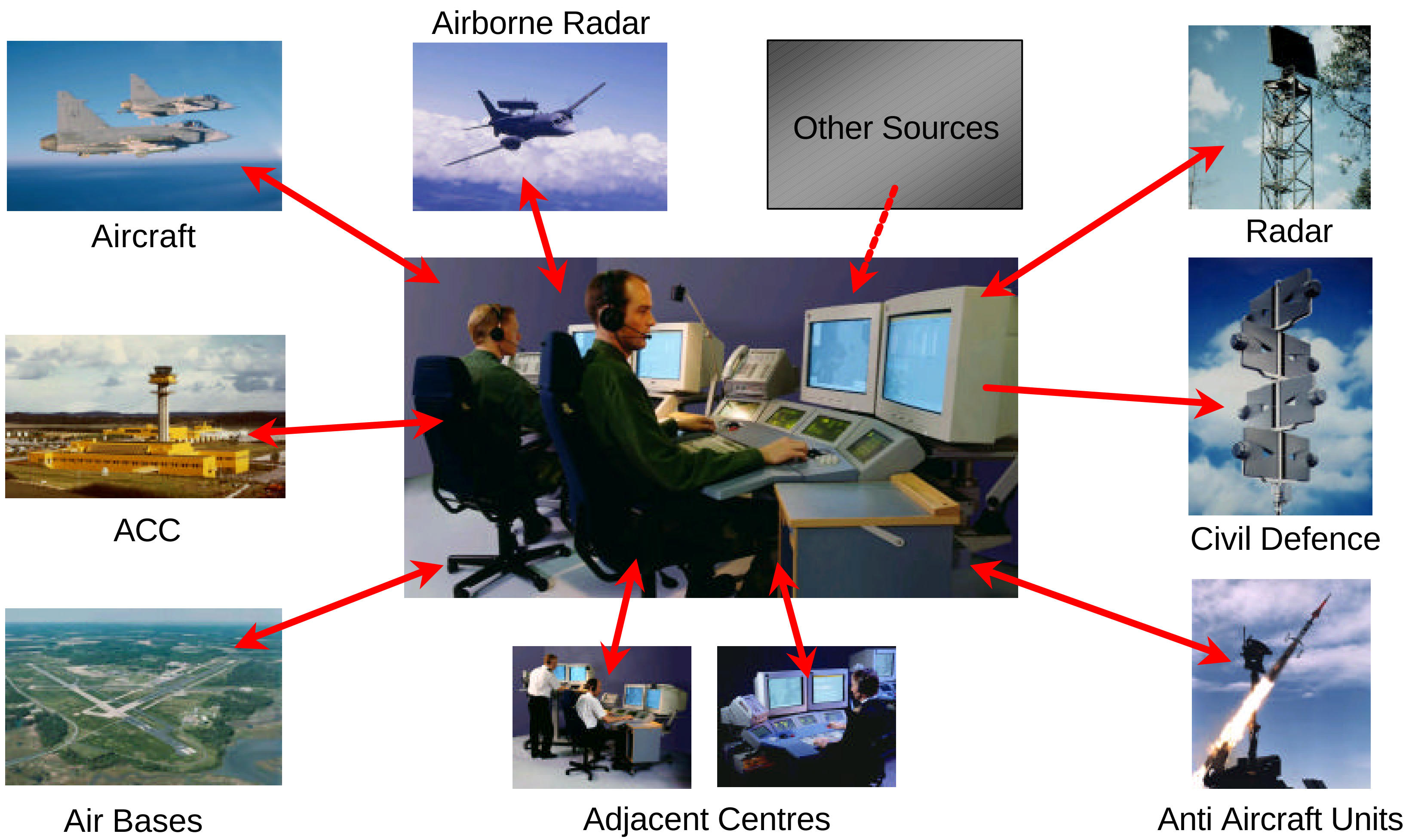
- Vehicle Tracking Systems



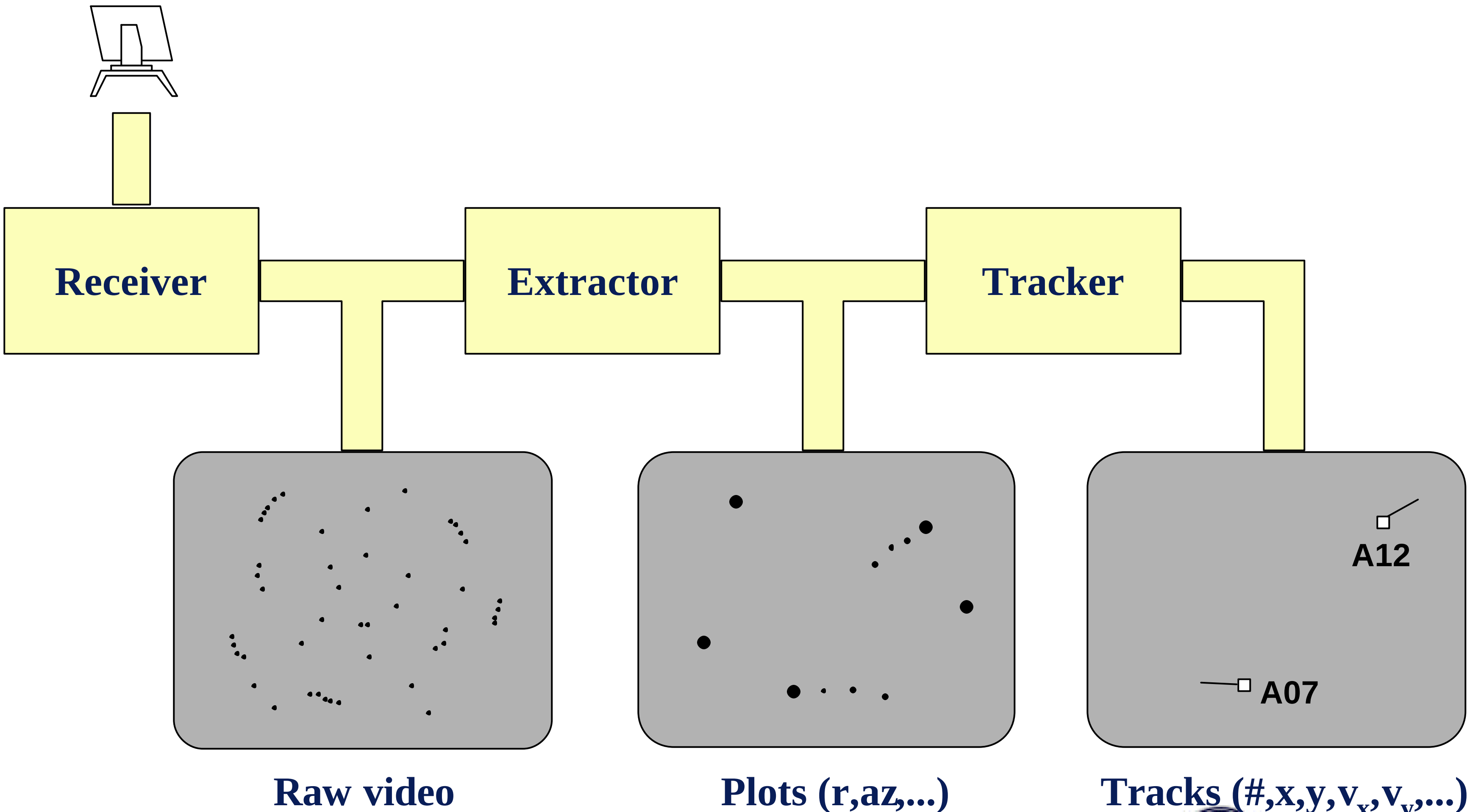

# System Track Record

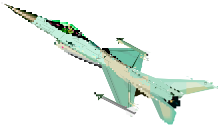
Total: > 750

# Air Defence Command and Control

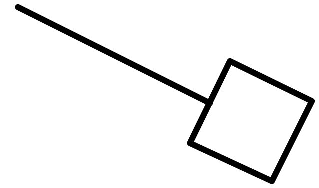


# Vad är målföljning (Tracking)?





# The Track
**(Målspåret)**



- The Track should contain as much information as possible about a target
  - A filter that estimates the kinematics of the target
  - A target classification state
- The filter is updated with the sensor measurement, i.e. a subset of: azimuth, range, elevation
- The estimated target classification state is updated by the amplification data. It can also utilize the kinematic information

# Important Performance Measures

- Accuracy of estimated state (position, speed, target type etc.)
- Fast track initiation/Low false track density
- Track continuity in situations with
  - high clutter (jammed environment)
  - low Pd targets (stealth targets)
  - fast maneuvers

# Sensor types

| Sensor | Azimuth | Range | Elevation | Amplification Data |
| --- | --- | --- | --- | --- |
| 3D-Radar | x | x | x | * |
| 2D-Radar | x | x | | * |
| 2D-Passive | x | | x | * |
| 1D- Passive | x | | | * |

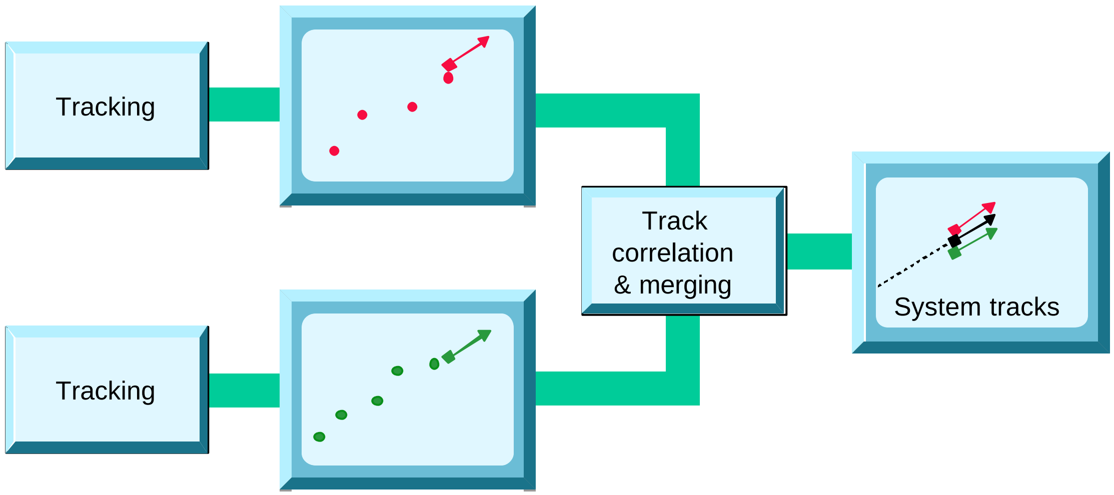
# Passive Sensors

- ECM strobes from radars
- IR/EO sensors
- ESM/CESM sensors

# Amplification Data

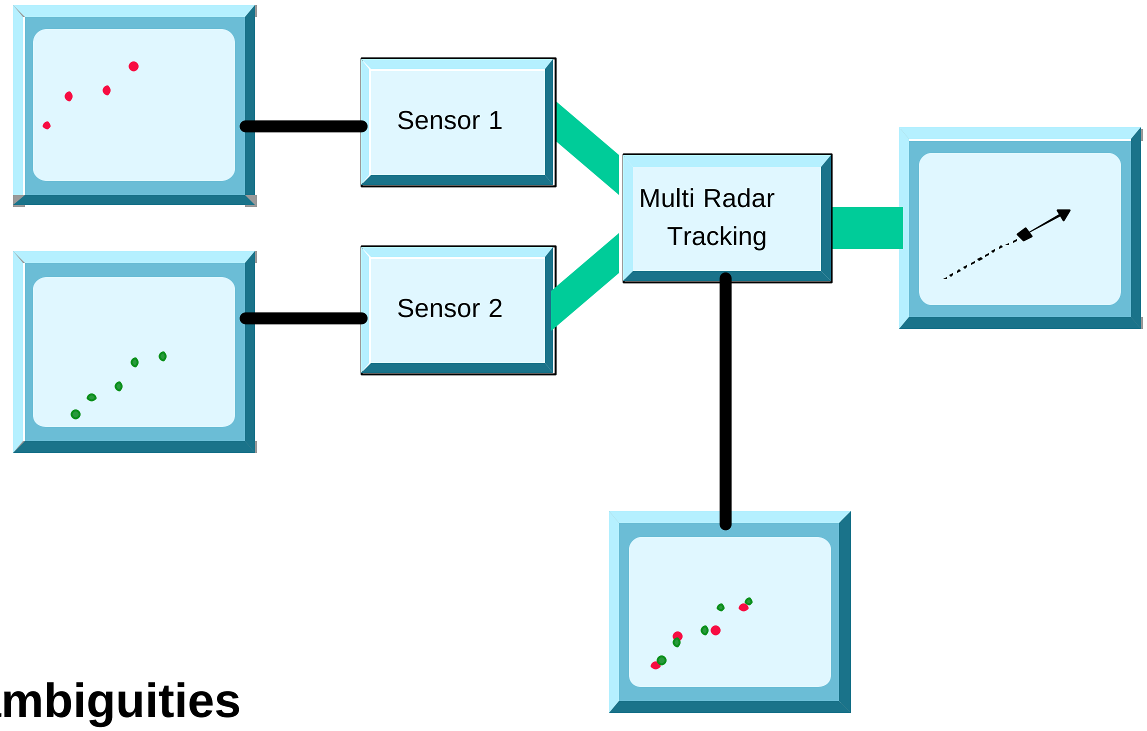
- Target Type (ESM)
- Emitter Type/Mode (ESM)
- Parameters Generated by the Signal Analysis (ESM)
- JEM data (Radar)
- Transponder Returns (IFF)

# Decentralised Data Fusion




- **Tracking performed for each radar**
- **Tracks are compared and correlated**
  - **Manually or automatic**
  - **Weighing or selective**
- **Detection probability dependent on performance of each radar**
- **Accuracy dependent on update frequency from each radar**

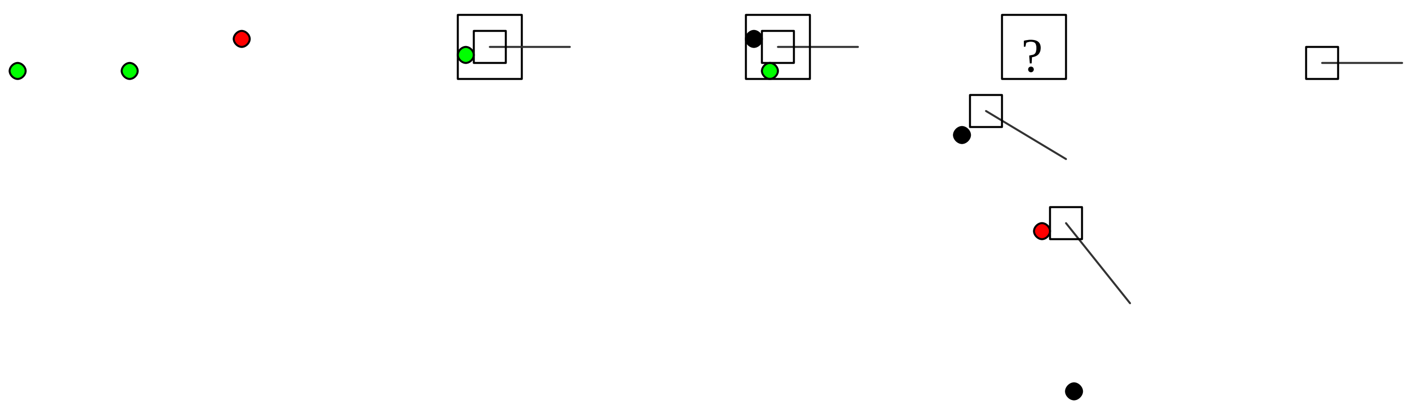
# Centralised Data Fusion




- **Less ambiguities**
- **Frequent updates - better performance**
- **Best use of overlapping sensors**
- **The natural way to integrate radars with passive sources and satellite navigation data**

# Centralised Fusion:

# Decentralised Fusion:



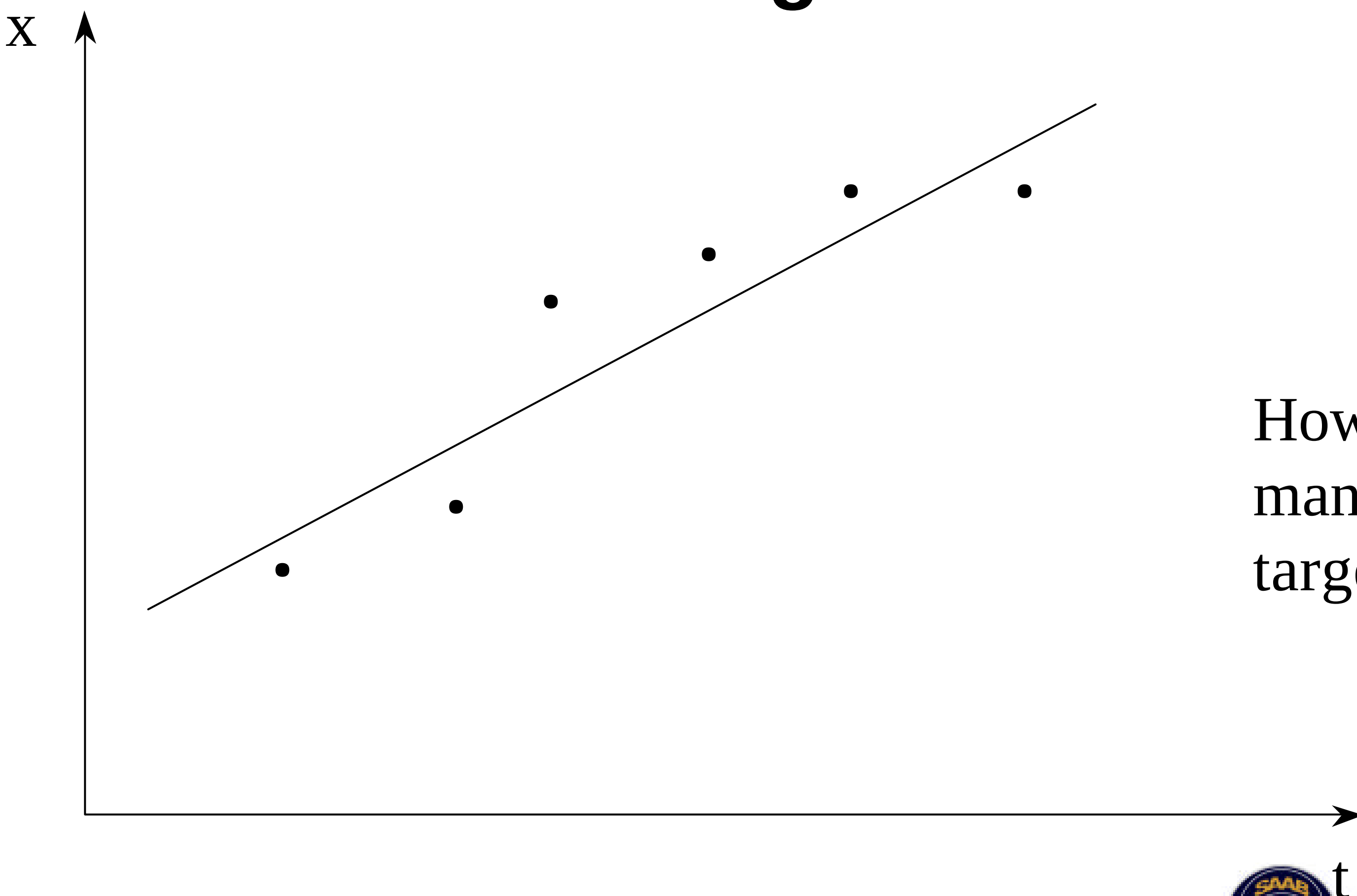
# Filtering techniques

**(Estimating position, speed & heading ...)**

- Linear regression (least squares batch processing)
  (hardly used in this context)
- (70's) Alpha-Beta
- (80's) Adaptive Kalman
- (90's) Interactive Multiple Model (IMM)
- (2000's ?) Non-linear filtering?

SAAB

# Linear regression




How to handle maneuvering targets???

# Alpha-Beta filtering

$\tilde{\mathbf{x}}, \tilde{\dot{\mathbf{x}}}$ *predicted position, speed*

$\hat{\mathbf{x}}, \hat{\dot{\mathbf{x}}}$ *updated position, speed*

$\mathbf{x}_m$ *measured position*

α and β are tuning constants between 0 and 1

**Prediction step**

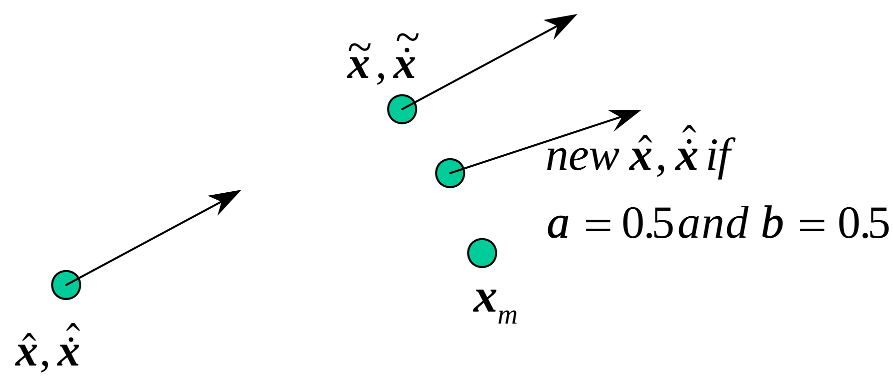
$$\tilde{\mathbf{x}} = \hat{\mathbf{x}} + \hat{\dot{\mathbf{x}}}\, T$$

$$\tilde{\dot{\mathbf{x}}} = \hat{\dot{\mathbf{x}}}$$

**Updating step**

$$\hat{\mathbf{x}} = \tilde{\mathbf{x}} + a\left(\mathbf{x}_m - \tilde{\mathbf{x}}\right)$$

$$\hat{\dot{\mathbf{x}}} = \tilde{\dot{\mathbf{x}}} + b\,\frac{\mathbf{x}_m - \tilde{\mathbf{x}}}{T}$$




α=β=0: Measurement has no effect

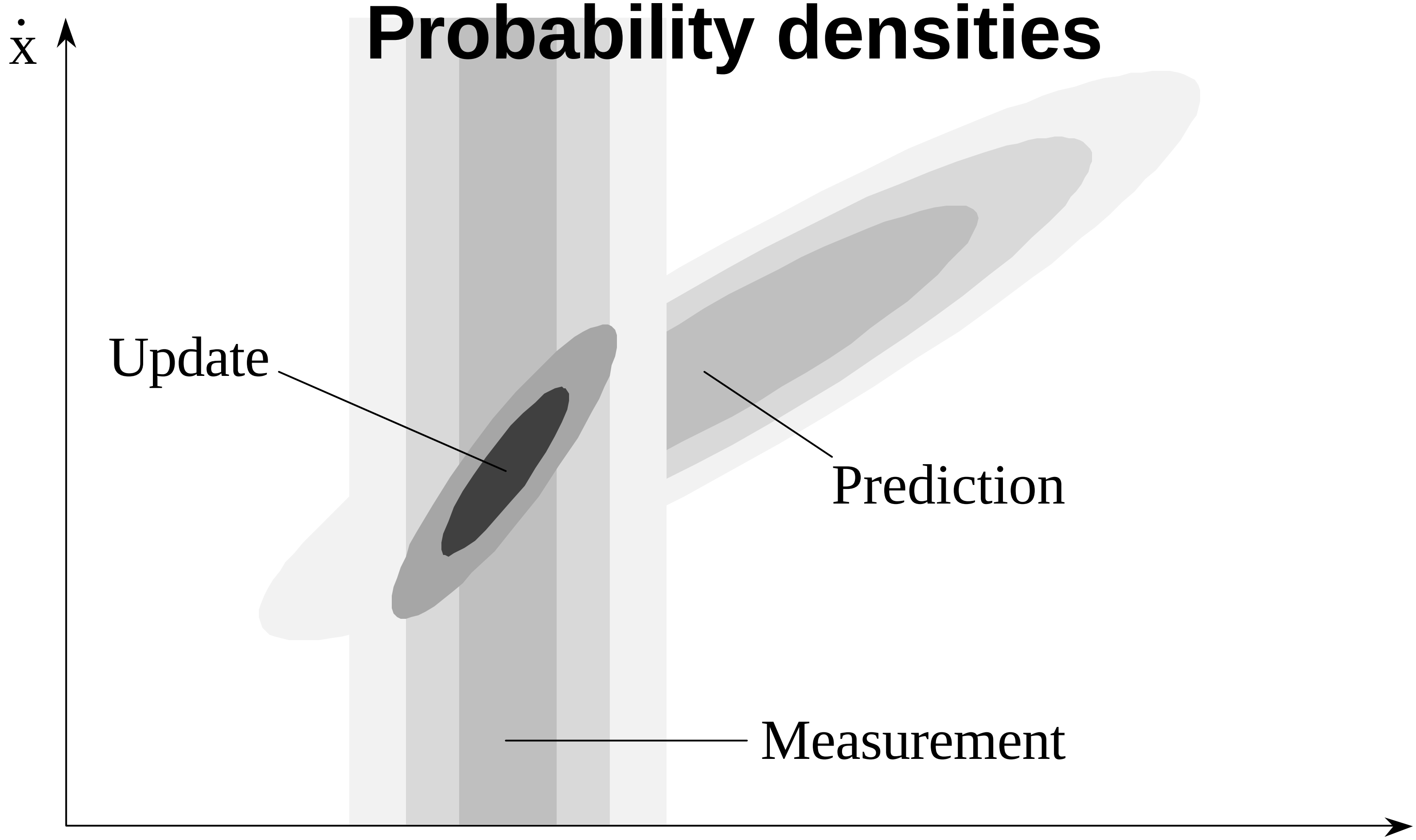
α=β=1: History has no effect

# Kalman filtering

Current state & uncertainties
+
Measurement & uncertainties
=
New state & uncertainties

Like a-b-filter, but:
Automatically optimizes a and b
Best weighting between history
and measurement
Output includes estimated accuracy

# Probability densities



# Kalman filter maths (I)

$\boldsymbol{x}, \dot{\boldsymbol{x}}$

replaced by a state vector, often:

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{x} \\ \dot{\boldsymbol{x}} \end{pmatrix}$$

and a covariance matrix, describing the uncertainties:

$$\boldsymbol{P} = \begin{pmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{pmatrix}$$

$$P_{11} = \overline{(X_1 - \overline{X}_1)^2} = s_1^{\,2}$$

$$P_{12} = \overline{(X_1 - \overline{X}_1)(X_2 - \overline{X}_2)},\ etc.$$

# Kalman filter maths (II)

Prediction:

$$\tilde{\boldsymbol{X}} = \Phi \hat{\boldsymbol{X}}$$

where $\Phi$ is a transition matrix, often like $\Phi = \begin{pmatrix} 1 & T \\ 0 & 1 \end{pmatrix}$ (or higher dimension).

Update:

$$\hat{\boldsymbol{X}} = \tilde{\boldsymbol{X}} + \boldsymbol{K} \cdot \left( \boldsymbol{x}_m - \boldsymbol{H}\tilde{\boldsymbol{X}} \right)$$

where $\boldsymbol{H}$ is a projection matrix, like $\begin{pmatrix} 1 & 0 \end{pmatrix}$

and $\boldsymbol{K}$ is the Kalman gain (replacing $\alpha$ and $\beta$)

# Kalman filter maths (III) - for reference

The optimal choice for $\boldsymbol{K}$ is

$$\boldsymbol{K} = \tilde{\boldsymbol{P}}\boldsymbol{H}^T\left(\boldsymbol{R} + \boldsymbol{H}\tilde{\boldsymbol{P}}\boldsymbol{H}^T\right)^{-1}$$

where R is the covariance matrix for the measurement noise (in Cartesian coordinates)

Covariance matrix for predicted $\boldsymbol{X}$:

$$\tilde{\boldsymbol{P}} = \Phi\hat{\boldsymbol{P}}\Phi^T + \boldsymbol{Q}$$

$\Phi$ transfers speed uncertainties to position. $\boldsymbol{Q}$ is the *process noise*, describing unpredictable maneuvers.

Covariance matrix for updated $\boldsymbol{X}$:

$$\hat{\boldsymbol{P}} = \tilde{\boldsymbol{P}} - \boldsymbol{K}\boldsymbol{H}\tilde{\boldsymbol{P}}$$

# Filter Ekvationer-teori

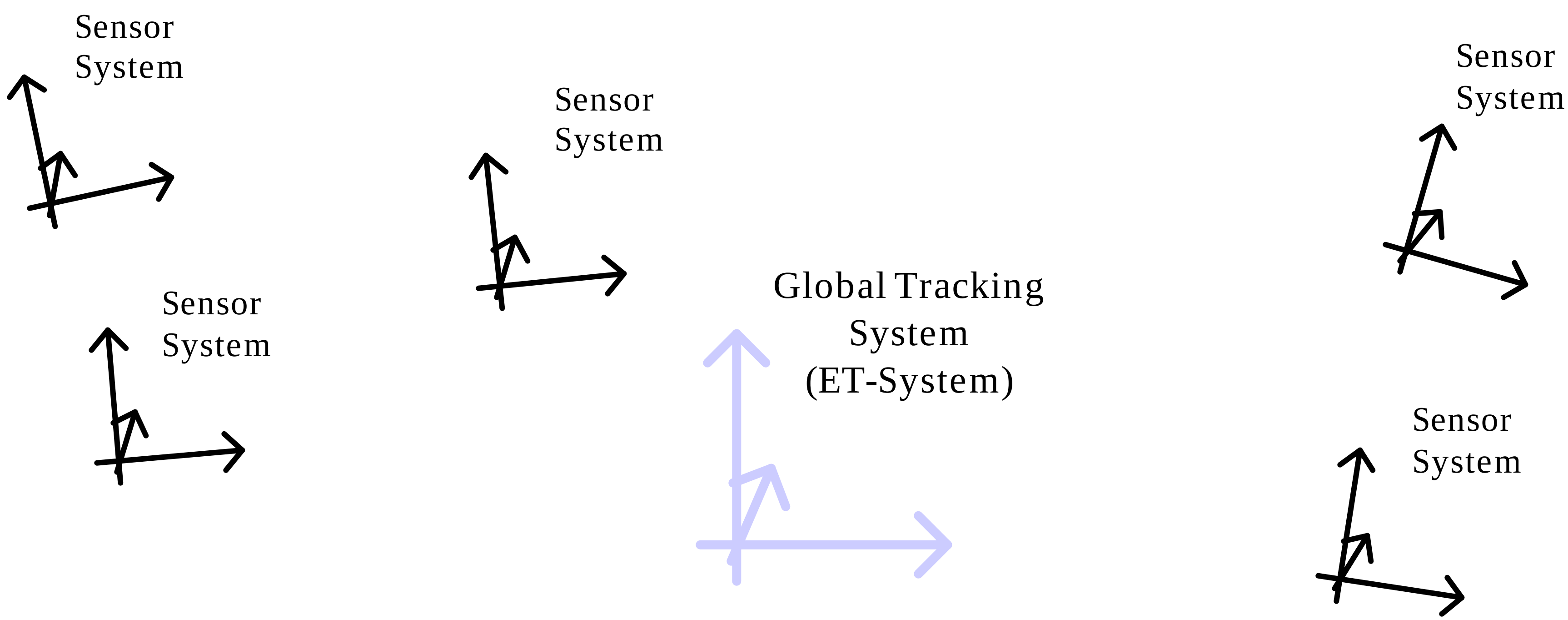
$$\begin{cases} dX_t = f(X_t, t)\,dt + db & \text{Dynamik och processbrus} \\ Z = h(X_t, t) + g & \text{Mätekvation och mätbrus} \end{cases}$$

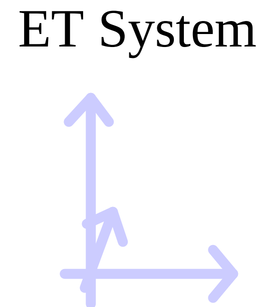
- ***X*** en stokastisk variabel som beskriver tillståndet som estimeras. I Kalman filtret antas den vara normalfördelad, dvs ***X*** kan beskrivas med ett ”medelvärde”+”kovariansmatris”.

- Mätrummet ***Z*** oftast en delmängd av ***(r,az,el).*** Det krävs en mätekvation för varje sensor.

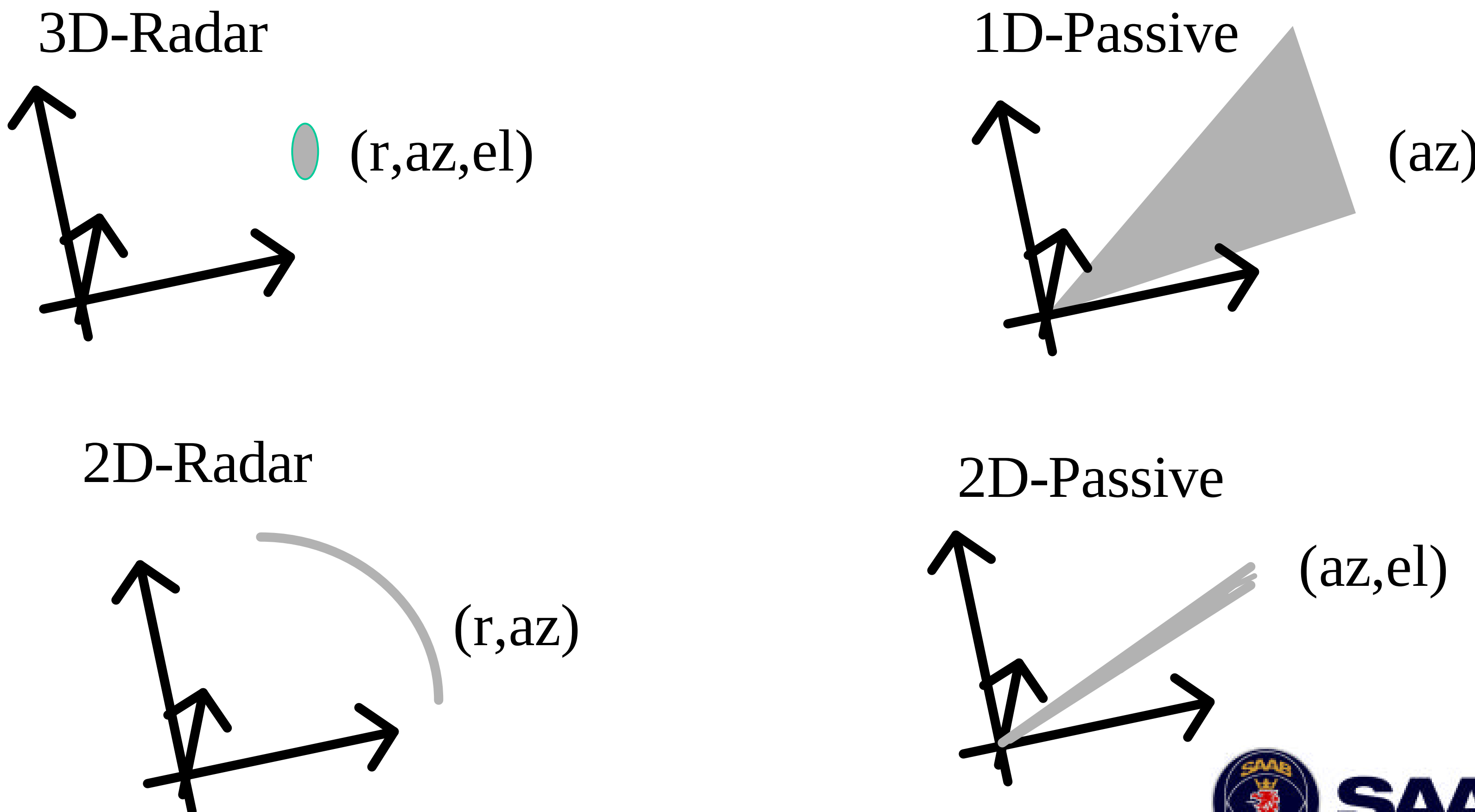
- För att egenskapen att vara normalfördelad skall bevaras måste ***f*** och ***h*** vara linjära. I annat fall görs approximationer (lineariseringar osv).

# The Coordinate Systems

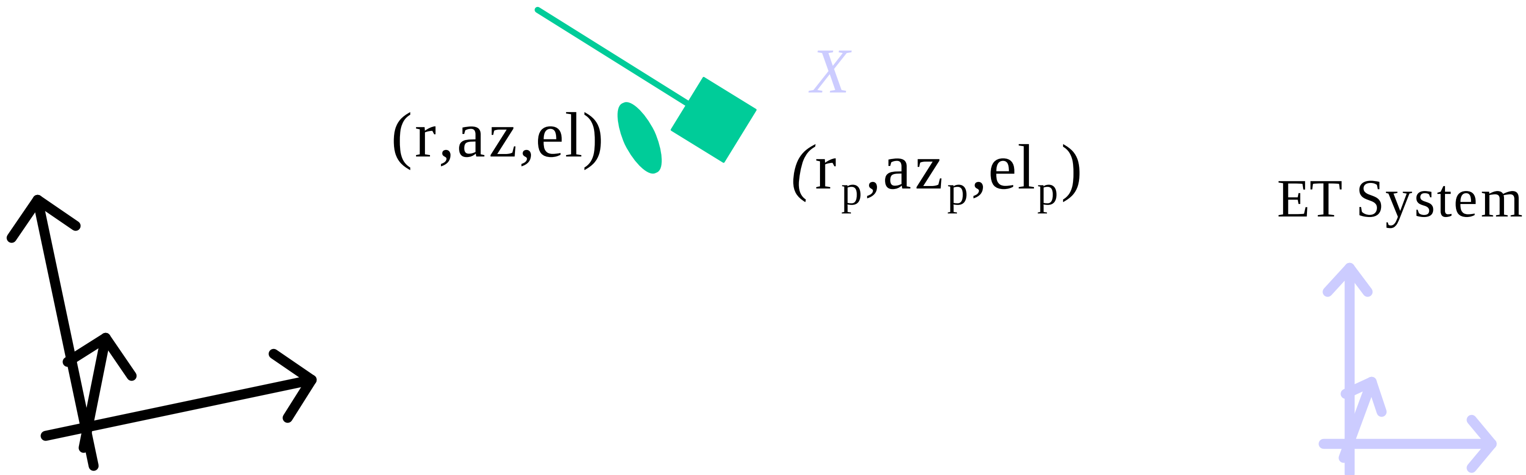





# Updating in the Measurement Space



# Updating the Filter




Transform track position to measurement space:

$$\tilde{X} \rightarrow \left(r_p, az_p, el_p\right)$$

Update the track state in the ET system:

$$\hat{X} = \tilde{X} + K_{X,r}\left(r - r_P\right) + K_{X,az}\left(az - az_P\right) + K_{X,el}\left(el - el_P\right)$$

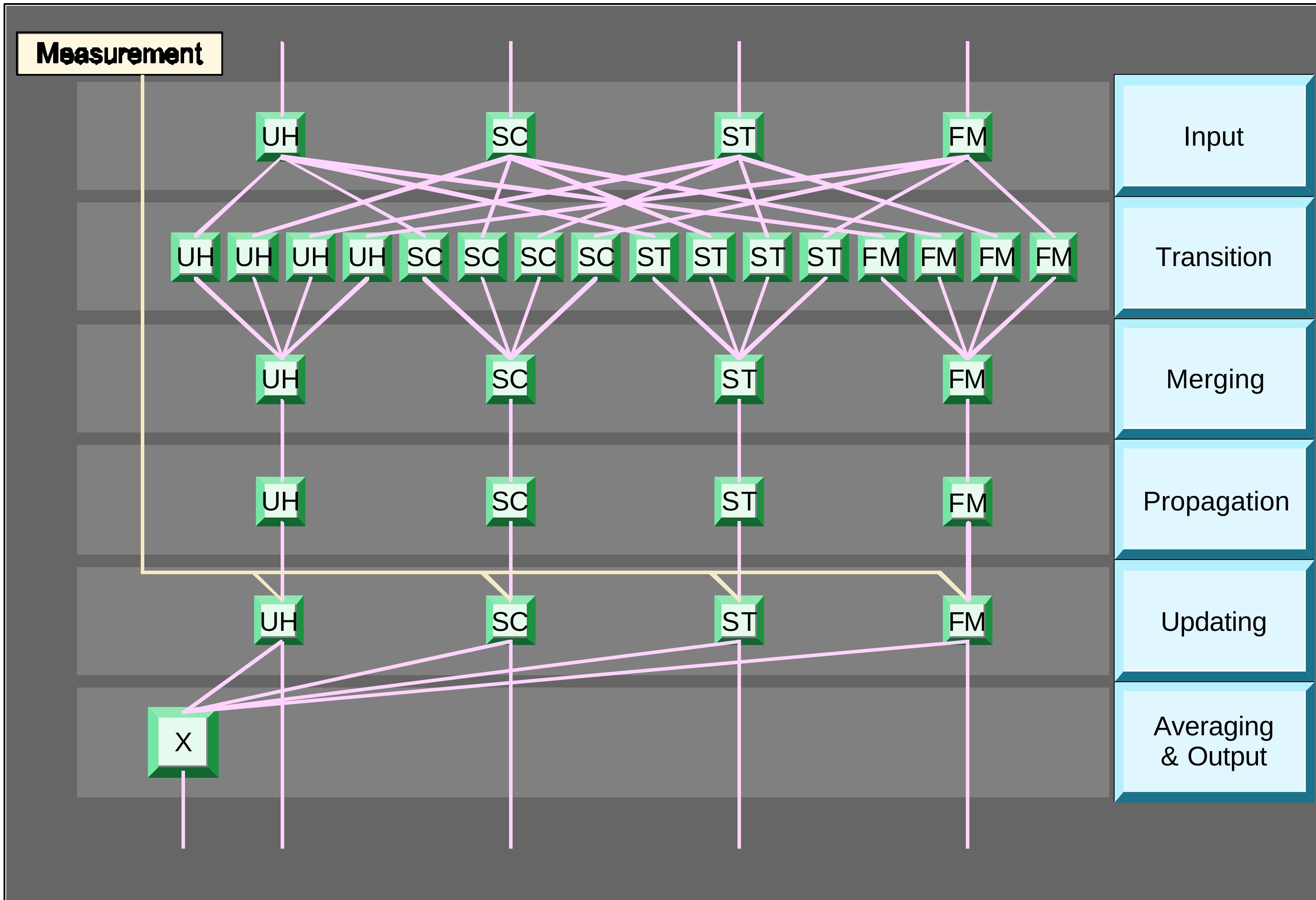
# IMM-Filters

- Single Kalman filter never sufficient
- Different methods have been developed
  using multiple Kalman filters, or different maneuver detection schemes
- Our early filters used two simple Kalman filters with different bandwidth
- New filter use the IMM technique
  (TBMS, ATC, AIR-DEFENCE)

# IMM States

| | State Vector and Filter Type | Dynamics |
|---|---|---|
| Uniform Horizontal Motion | $(x_1, x_2, x_3, \dot{x}_1, \dot{x}_2, \dot{x}_3)$ Linear Kalman filter | $\ddot{\mathbf{x}} = 0; \dot{\mathbf{x}} \cdot \mathbf{u} = 0$<br>$\mathbf{u} =$ vertical unit vector |
| Speed Changes | $(x_1, x_2, x_3, \dot{x}_1, \dot{x}_2, \dot{x}_3)$ Linear Kalman filter | $\ddot{\mathbf{x}} \cdot \mathbf{u}$ and $\ddot{\mathbf{x}} \cdot \mathbf{l} =$ white noise<br>$\mathbf{l} =$ longitudinal unit vector<br>$\ddot{\mathbf{x}} \cdot (\mathbf{u} \times \dot{\mathbf{x}}) = 0$ |
| Slow Turns | $(x_1, x_2, x_3, \dot{x}_1, \dot{x}_2, \dot{x}_3, w)$ $2^{nd}$ order Extended Kalman | $\dot{w} = 0$<br>$w =$ turn rate |
| Fast Maneuvers | $(x_1, x_2, x_3, \dot{x}_1, \dot{x}_2, \dot{x}_3)$ Linear Kalman filter | $\ddot{x} \cdot \boldsymbol{l} =$ white noise<br>$\ddot{x} \cdot \boldsymbol{s}, \ddot{x} \cdot \boldsymbol{t} =$ white noise<br>$\boldsymbol{l} \cdot \boldsymbol{s} = \boldsymbol{l} \cdot \boldsymbol{t} = \boldsymbol{s} \cdot \boldsymbol{t} = 0$ |

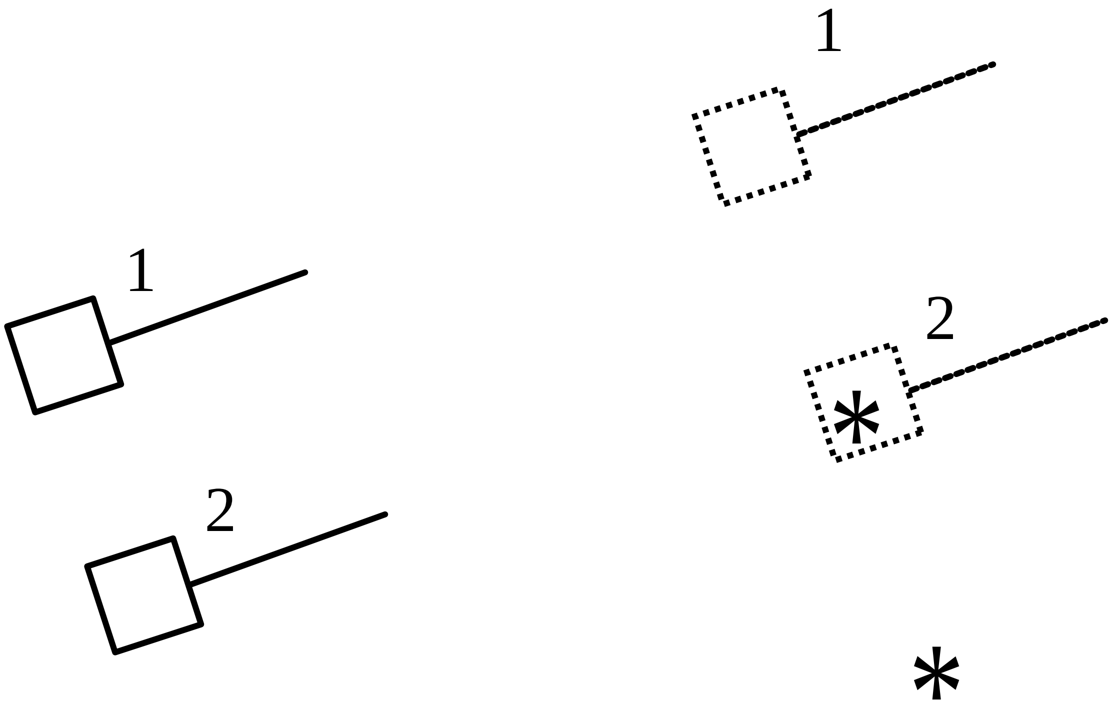
# IMM structure



# Association methods

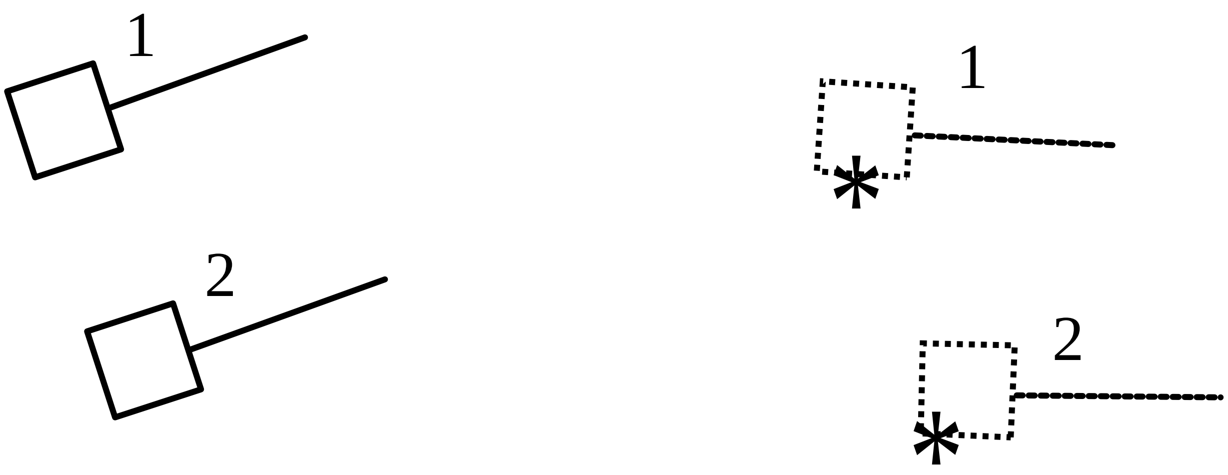
- Nearest neighbour
- Nearest neighbour with statistical distance
- Global minimisation on statistical distances
- Global maximisation of probability
- Probabilistic Data Association (PDA)
- Joint Probabilistic Data Association (JPDA)
- Multiple Hypothesis Tracking (MHT)

# Example: NN



# Example: LPQ-method

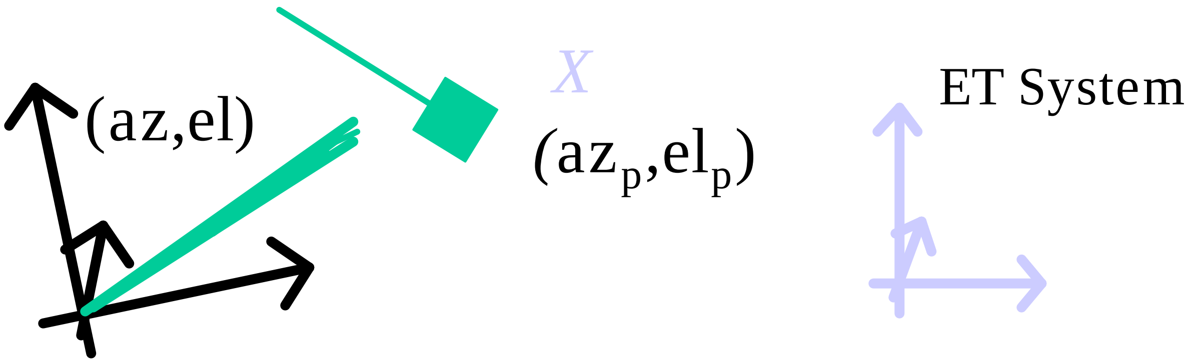
**(Global maximisation of probability)**






SAAB

# Measurement-to-Track Association



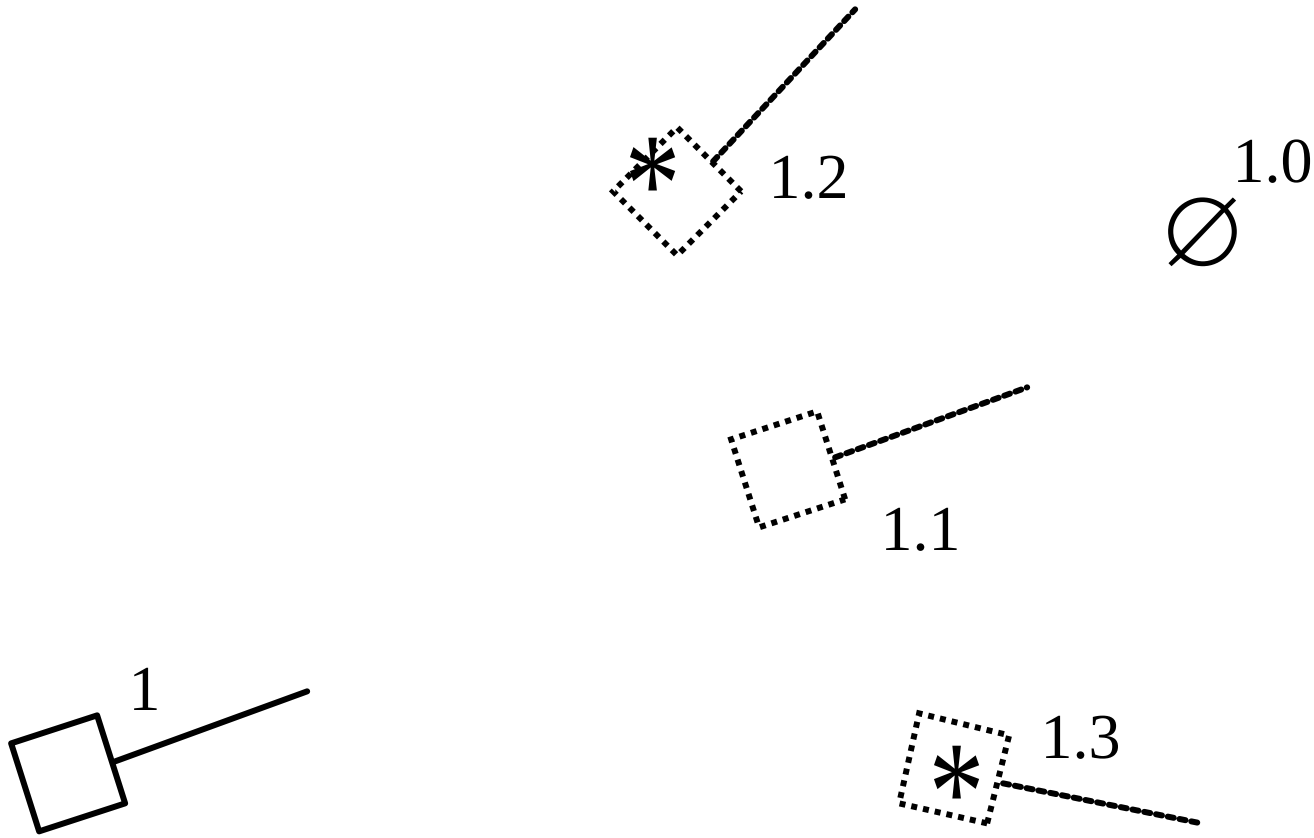

Transform track position and its covariance to measurement space:

$$X \rightarrow \left(az_p, el_p\right), \quad \left(P_{az\,az}, P_{az\,el}, P_{el\,el}\right)$$

$$d^2(p,t) = \Delta S^{-1}\,\Delta^t\,, \quad \Delta = \left(az - az_p \quad el - el_p\right), \quad S = \begin{pmatrix} P_{az\,az} + s_{az}^2 & P_{az\,el} \\ P_{az\,el} & P_{el\,el} + s_{el}^2 \end{pmatrix}$$

$$f(p,t) = \exp\left(-\frac{d^2(p,t)}{2}\right) / \left(2p\sqrt{\det(S)}\right)$$

# Track Oriented Multi Hypothesis Tracking (MHT)



# MHT: nötter

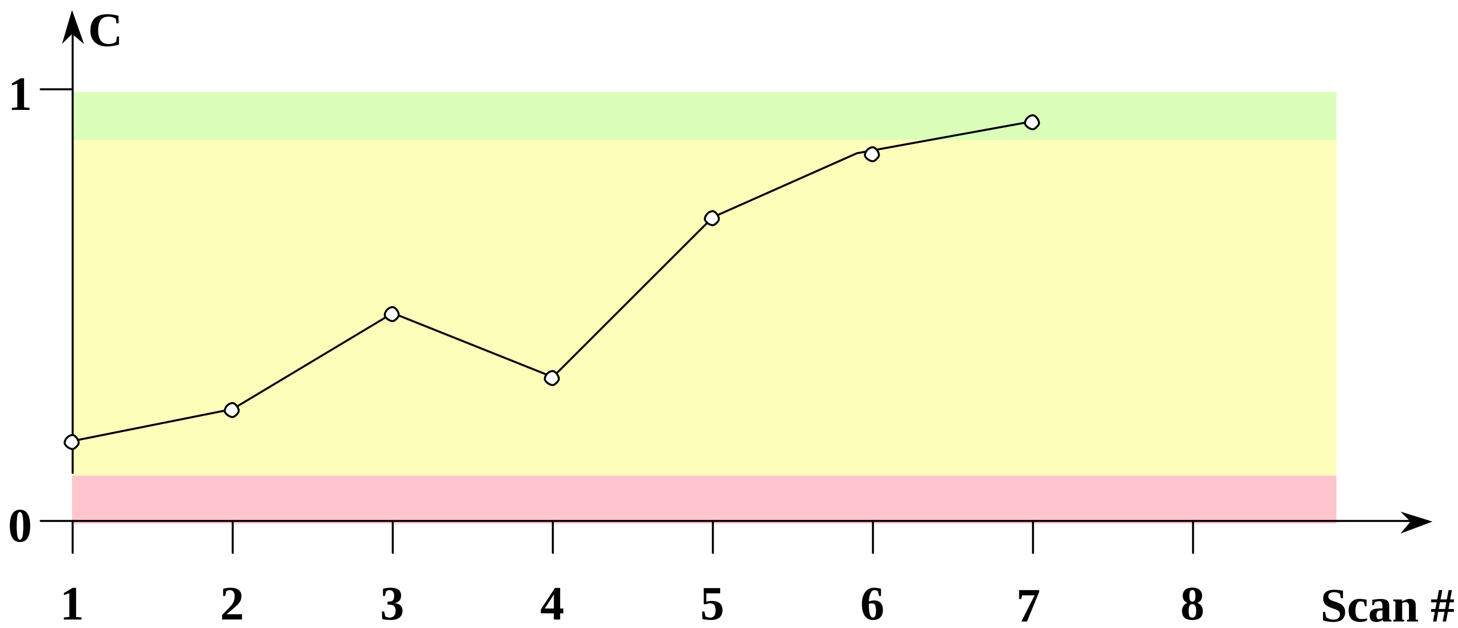
- Den hypotes med högsta sannolikhet visas (viss hysteresis).
- Hypoteser med låg sannolikheter tas bort (pruning).
- Antagandet att våra hypoteser är uttömmande stämmer inte alltid. T.ex. om två plottar eller fler, fås p.g.a. motmedel, eller att flera mål ej var upplösta från början.
- Metoden hanterar ej initiering av målspår.

# Databaserad MHT

- Har fördelen att den inkluderar initiering av målspår.
- Nackdelen är att den har hög beräkningskomplexitet.
- Den behöver mycket handpåläggning för att fungera i praktiken (pruning ...).
- Kan användas för att hantera ofullständigheter hos sensorer.

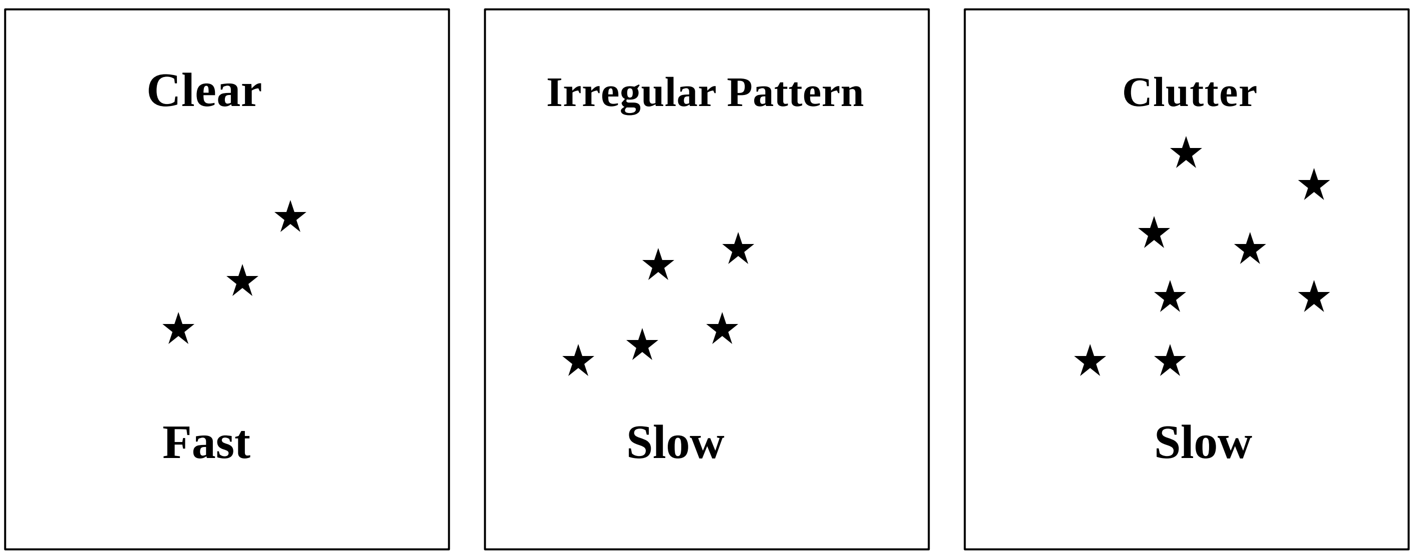
# Track initiation

- Required: Fast initiation and low false track rate
- Sequential hypothesis testing
- Credibility C » likelihood that a potential track is genuine



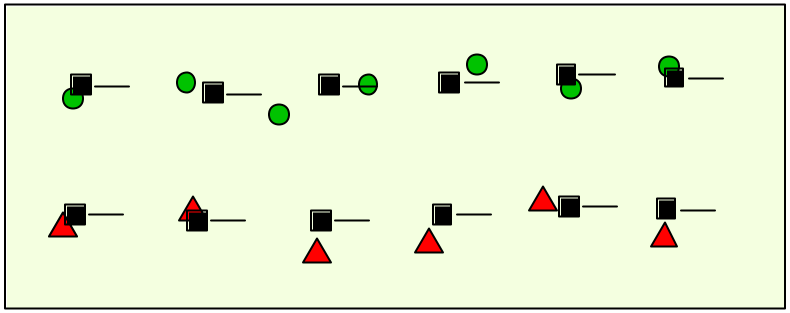
# Track initiation - cont’d

- **Credibility update depends on**
  - Spurious plot density
  - Hit proximity

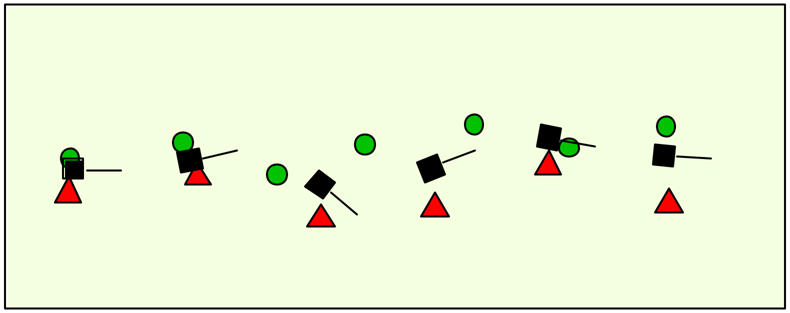


# Bias Estimation and Compensation

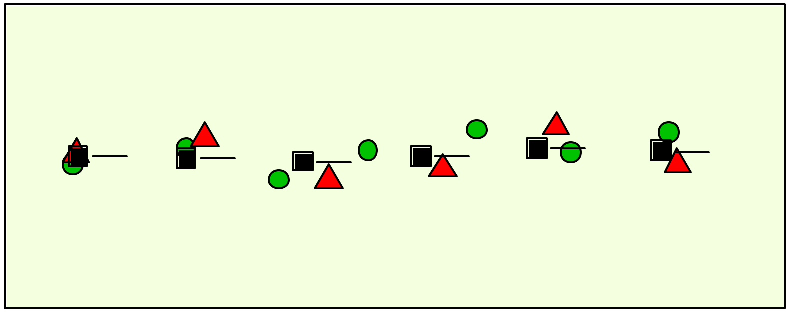
# Bias Errors



Large bias errors may cause multiple tracks for one target



Small bias errors may impair accuracy



Efficient bias compensation is crucial

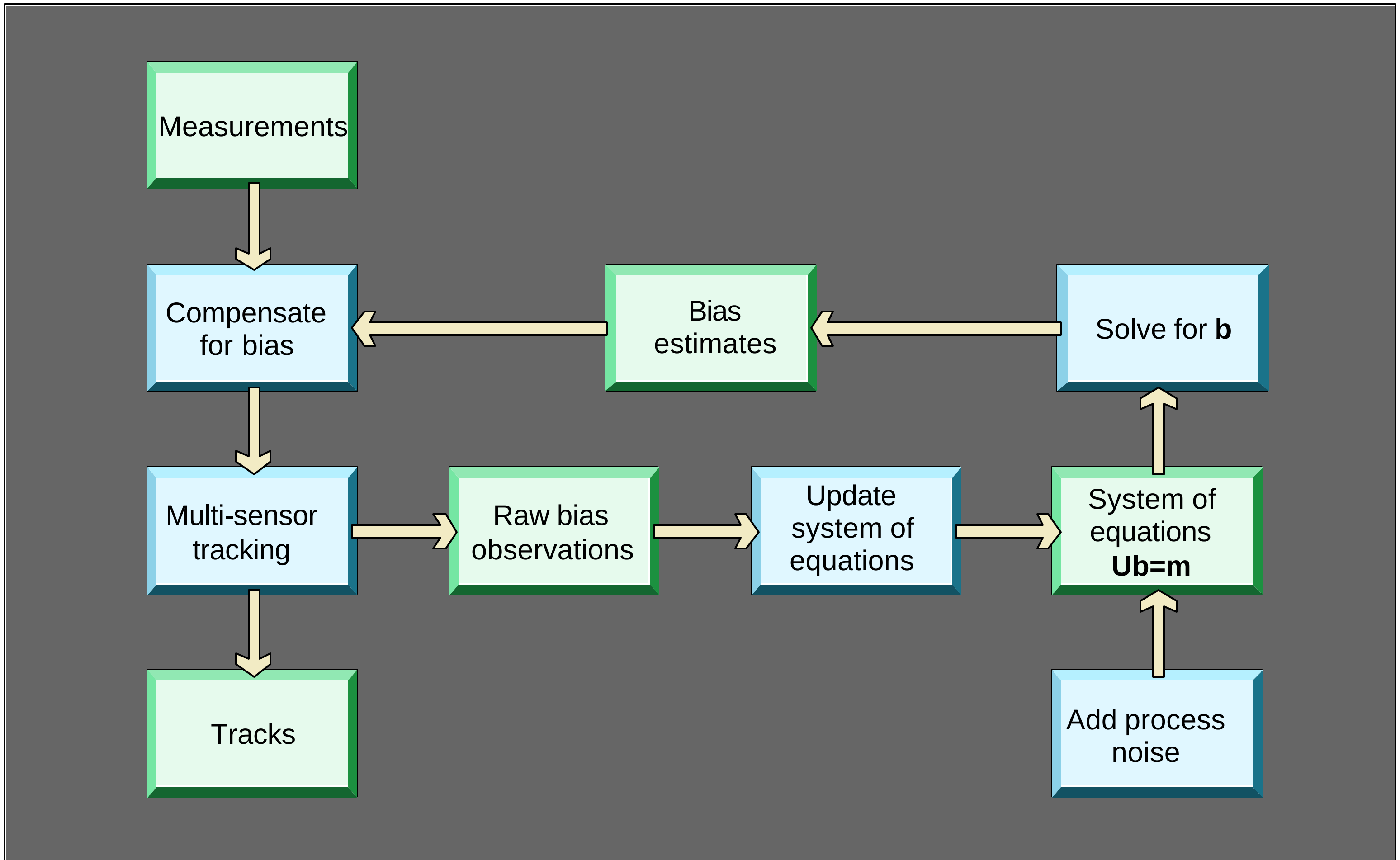
# Bias Parameters

- For each sensor a subset of the following can be selected:
  - Timing error
  - Sensor location up/east/north
  - Axis tilting east/north
  - North alignment
  - Range offset
  - Range gain
  - Antenna roll/pitch

# Estimation process

- Works continuously in background
- Measurements are collected from different sensors
- Bias parameters are calculated to minimize discrepancies
- Quasi-recursive: Both sides of a system of equations updated continually; solved periodically
- Process noise injected to account for drifting of bias errors

# Bias estimator



# Strobe Handler

Filtering of Strobes

and

Computation of Crosses

KTH 2001-03-29

SAAB

# Terminology (SH)

- Strobe Track: An estimate of angle and angular velocity of a possible target.
- Crossing: Computed crossing between two or more strobe tracks. It contains an estimate of position, speed and heading.
- Ghosts: Tracks or crossings formed using strobes originating from different targets

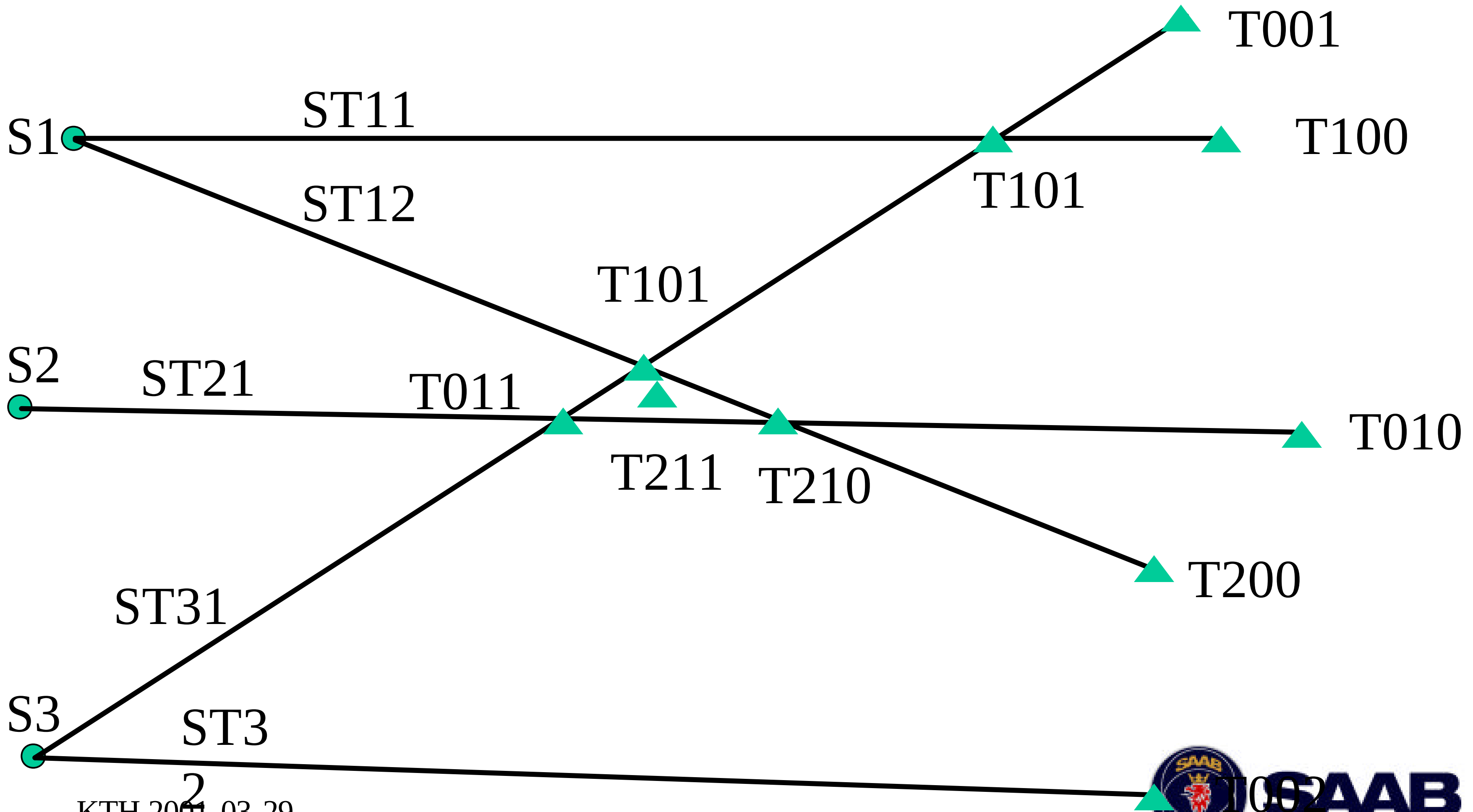
# Main functionality of SH

- Maintain Strobe Tracks per Sensor
- Use Strobe Tracks to Compute possible Target Position (Crosses)
- If enabled, Initiate Tracks using High Quality Crosses

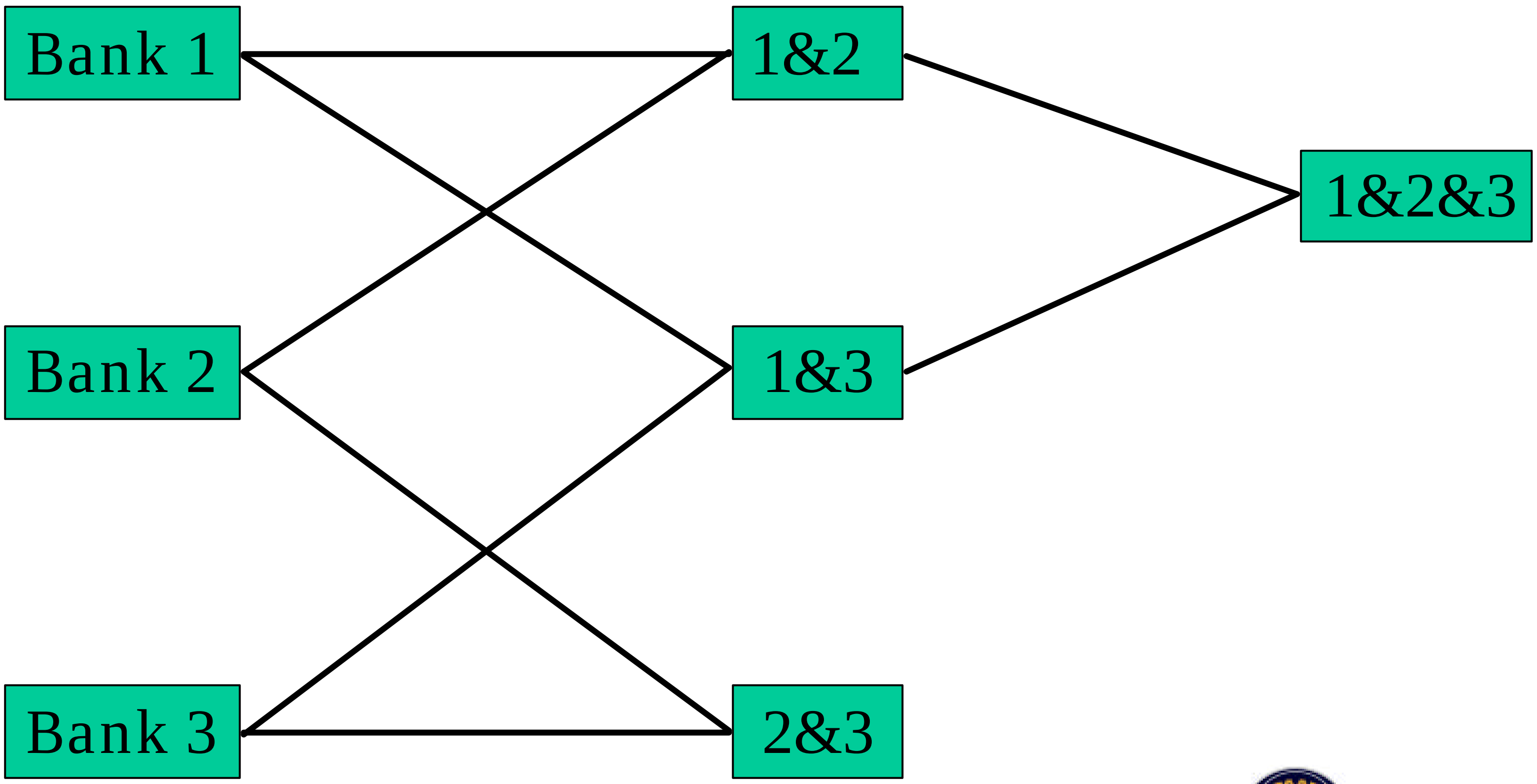
# Computation of Crosses

- All Strobe Tracks are Propagated to Common Time
- Compute all Possible Crosses
- Compute the Quality of the Crosses
- Select the Crosses with Highest Quality under the Constraint that a Strobe Track is used only once

# Formation of Crosses

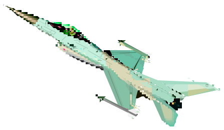


# Formation of Hypothetical Crosses

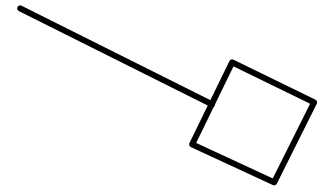


## MST with integrated
## Non-cooperative Target Type Recognition

- Algorithms and a prototype has been developed
- Tested with simulated data
- Next step is to test the algorithms further with live sensor data, radars, IRST and ESM sensors.

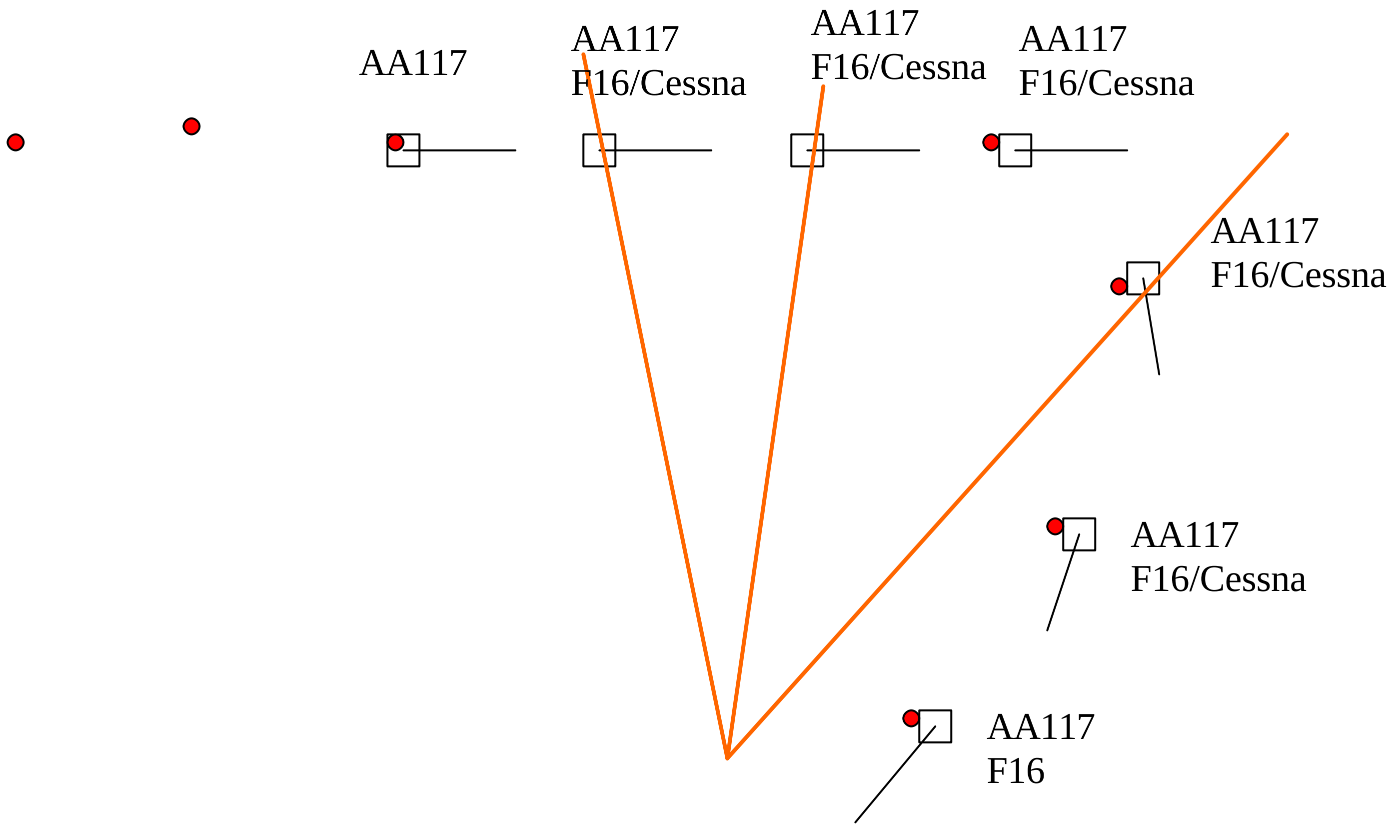


# The MST-Track

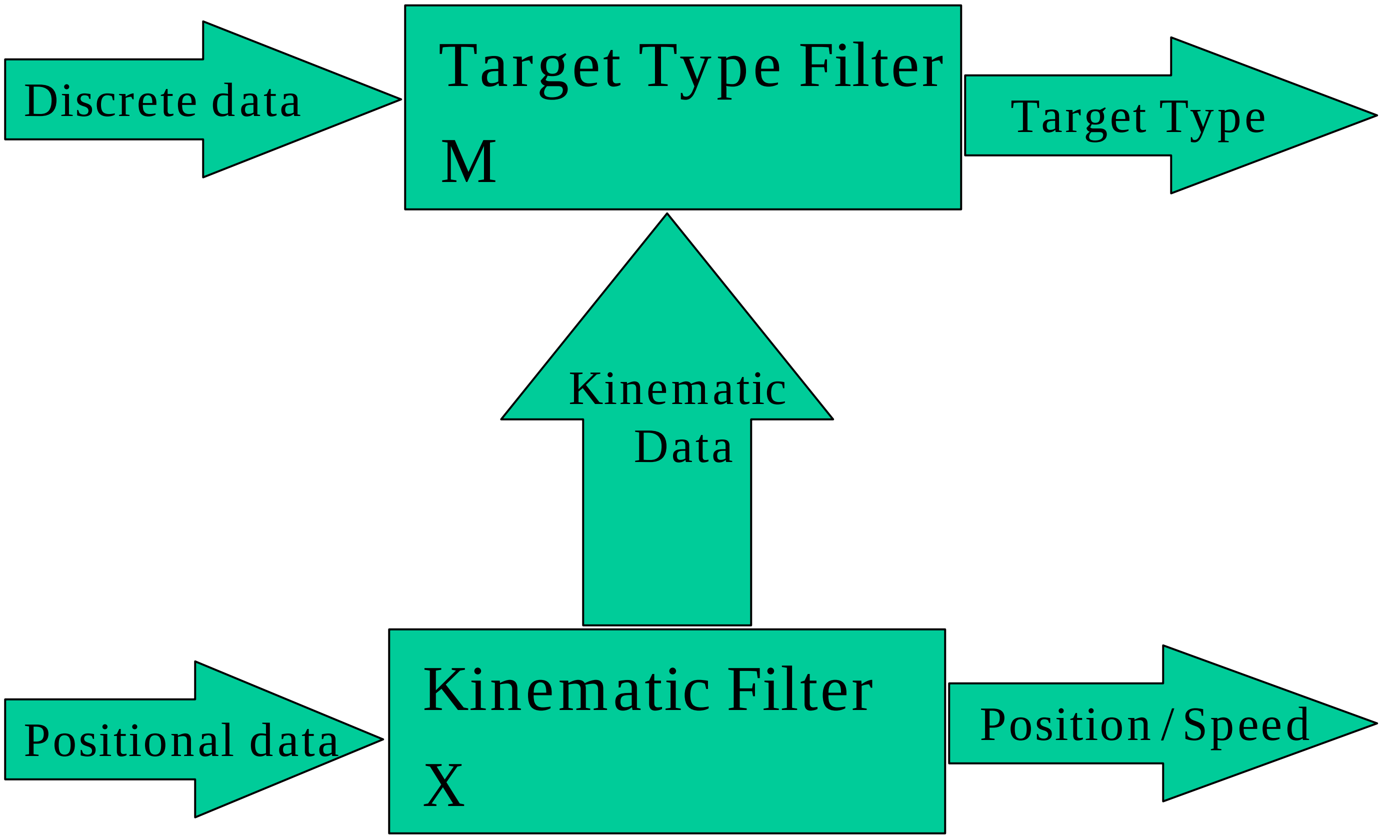


- The Track should contain as much information as possible about a target
  - A filter that estimates the kinematics of the target
  - *A target classification state*
- The filter is updated with the sensor measurement, i.e. a subset of: Azimuth, Range, Elevation
- *The estimated target classification state is updated by the amplification data. It can also utilize the kinematic information from the filter or direct information.*

# Integration of Non-cooperative Target Type Recognition



# Tracking/Typing Filters



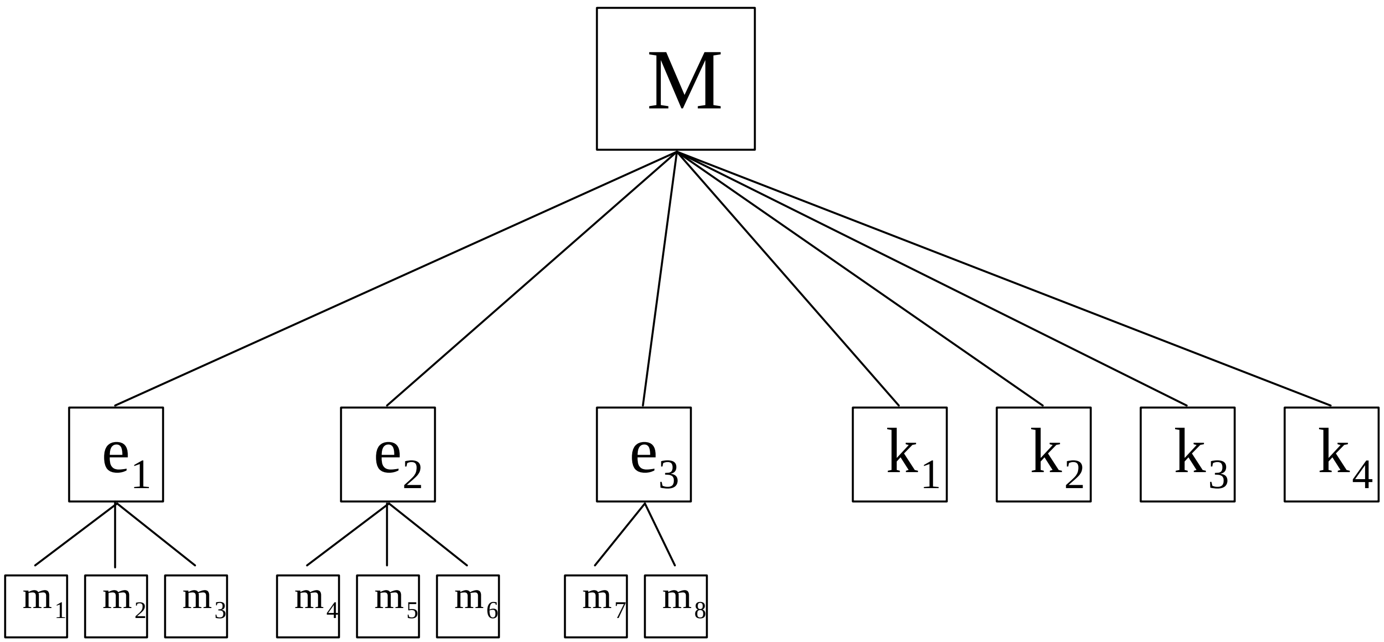
# Common methods for Typing

- Dempster-Shafer (and variants of it)
- Bayesian (probabilistic)
- Neural networks (training)

# We use the Bayesian method

- A recursive algorithm with a state vector that grows linearly the number of target types.
- Process noise models ”forgetting”
- Easy to implement in data association, cross computation etc, since it is probabilistic
- Need some engineering to make it work in real life

# Målinformations model



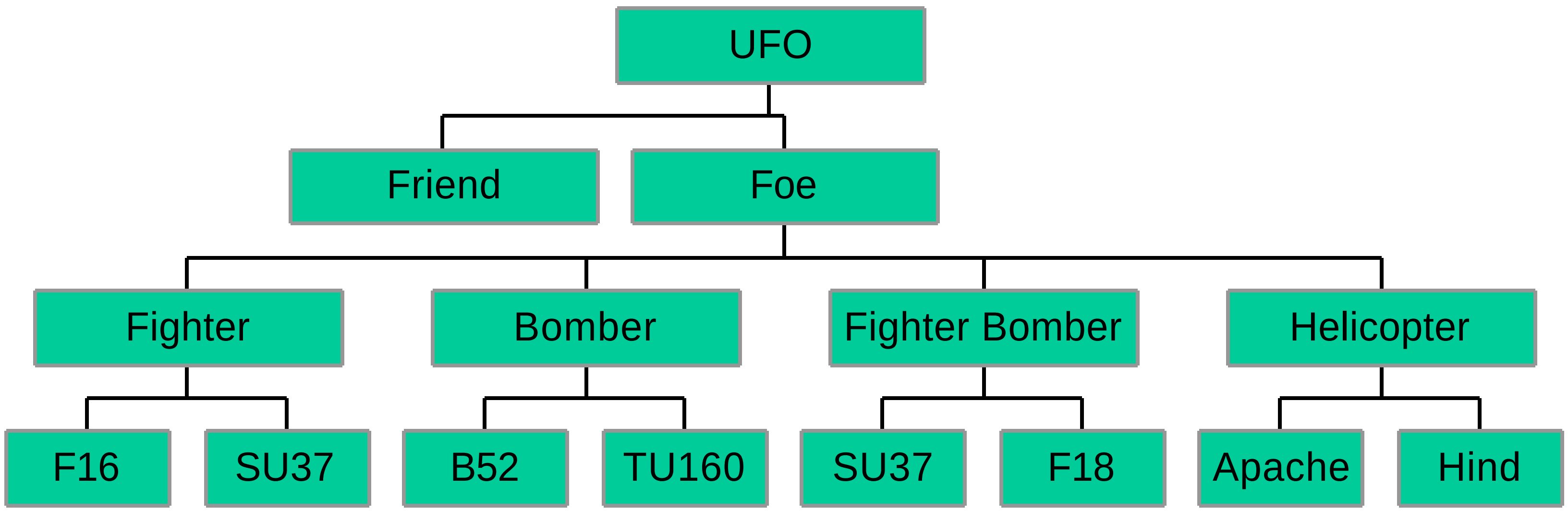

M=Mål Typ (Fine grid: Cessna, F16, Mig29, Concorde etc)

$e_i$=Emitter Typ (radar, after burner,height finder, nav. radar)

$m_i$=Emitter Mod (attack / search, on / off)

$k_i$=Flygenveloper (altitude, min / max speed, climb rate etc.)

Target Type Refinement



# Filter ekvationer för attribut

Kinematisk filtrering:

$$\begin{cases} \dot{X} = f(X,t) + w & \text{Dynamik och processbrus} \\ Z = h(X,t) + g & \text{Mätekvation och mätbrus} \end{cases}$$

Attribut filtrering?:

$$\begin{cases} \dot{M} = f(M,t) + w & \text{Vad är M, dynamik och mätbrus?} \\ A = h(M,t) + g & \text{Mätekvation, mätbrus och mätrummet?} \end{cases}$$

M = måltyp (F16, SU30, ...), en diskret sannolikhetsfördelning

A = emitter moder, kinematiska data, ... . Osäkerheter ges explicit.

Varför fungerar inte Kalmanfilter?

# Filter ekvationer ...

- I princip behövs ingen dynamisk ekvation. Man vill dock ”glömma gammal information, p.g.a. risk för ”track swap” eller felaktigt associerat data.
- Mätekvationen mappar ned måltypen på attribut enligt föregående träd.
- Uppdateringen sker m.h.a. Bayes metod. Detta kräver en hel del pyssel för att fungera i praktiken

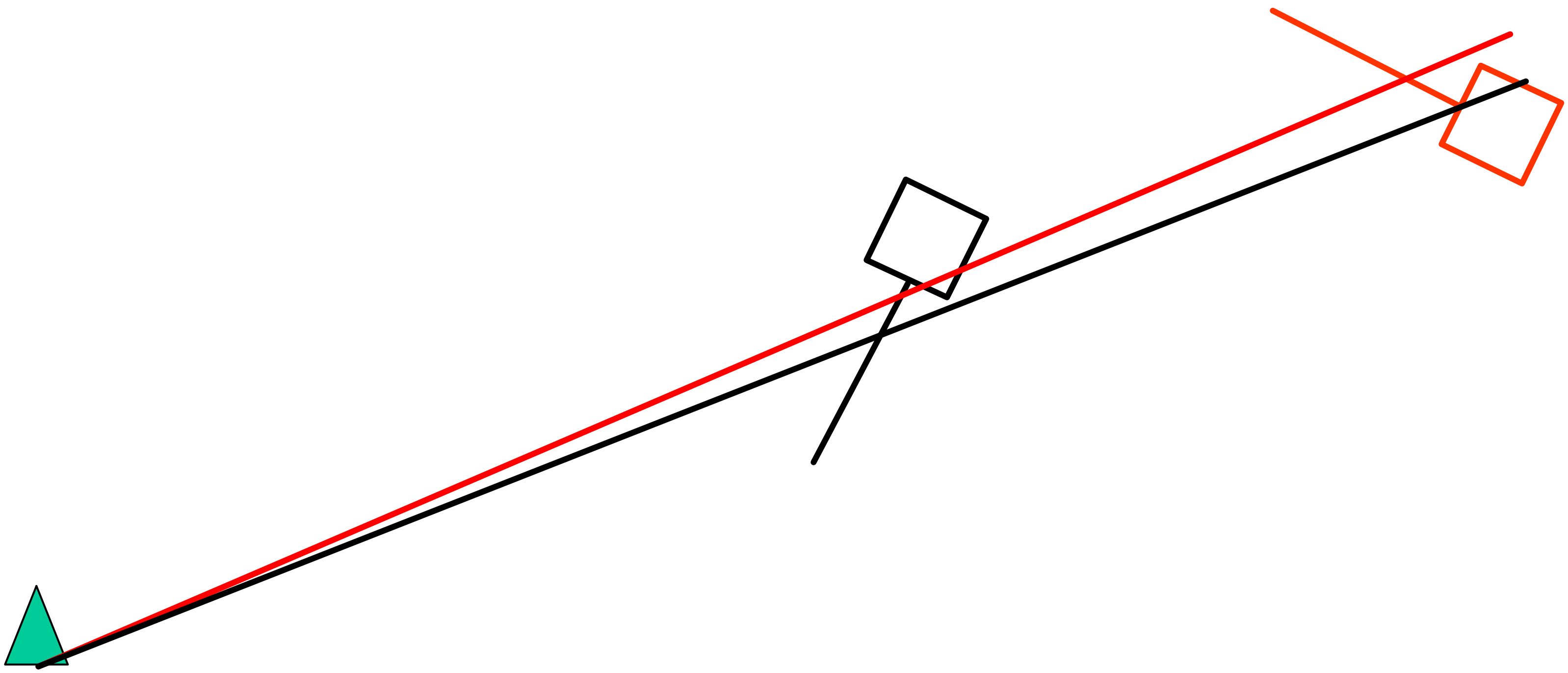
# Basic usage

- Target typing for display to operator
- Improved plot-track association, cross computations, MHT ... .

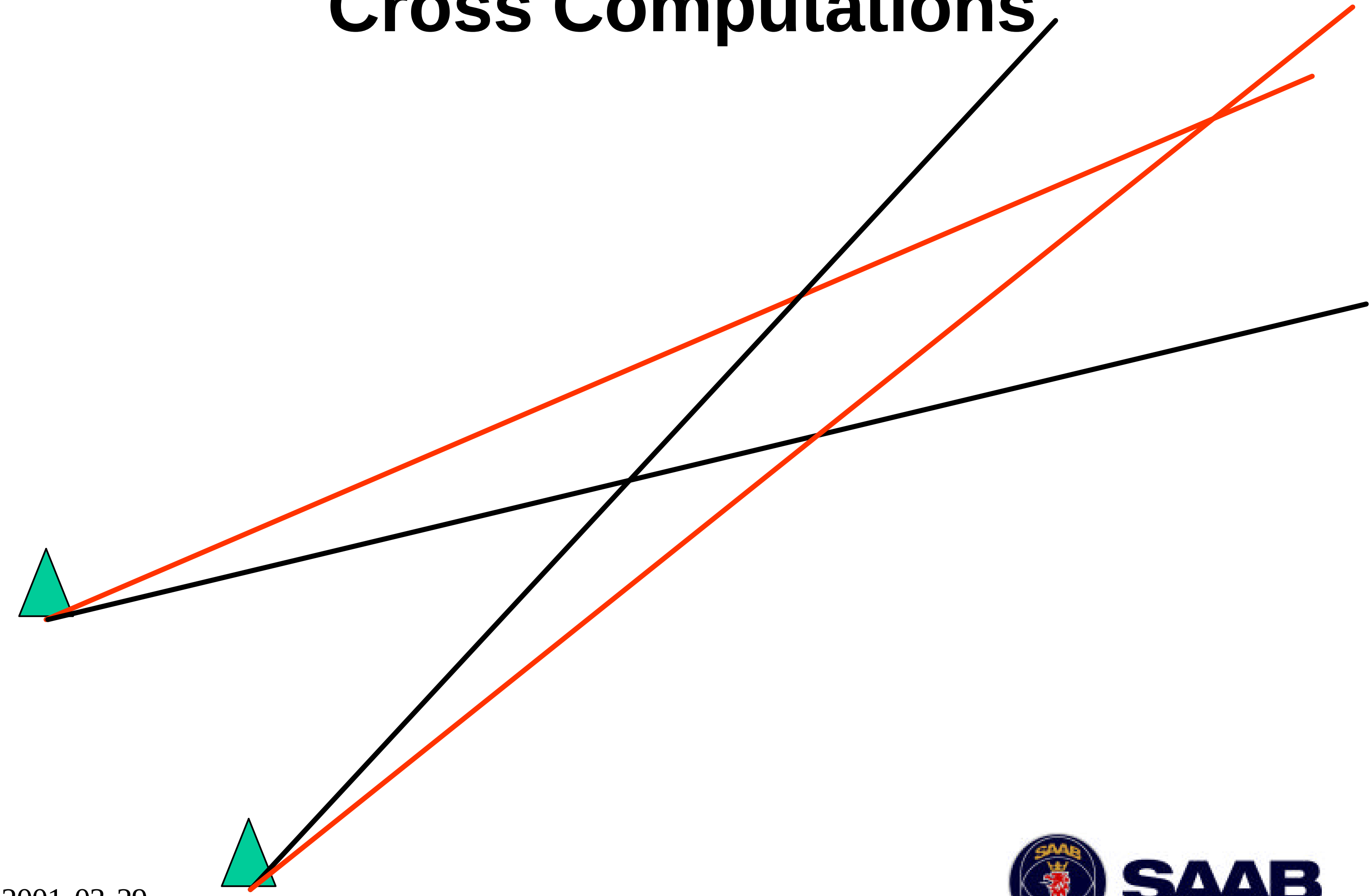
$$\tilde{X} = (X, M), \quad \tilde{Z} = (Z, A)$$

$$f(p, t) = f(p_Z, t_Z)\, f(p_A, t_A)$$

# Improved
# plot/strobe-to-track association



# Improved
# Cross Computations



# Summary - principles for Target Tracking

- Centralised fusion of different sensors
- Multiple model Kalman filtering
- Measurement-to-track association
- Initiation/deletion of tracks
- Passive strobe handling (filtering and association)
- Sensor bias estimation and compensation
- Target typing

Statisitcal (Bayesian) algorithms in real time software