Sensor Health State Estimation for Target Tracking with Binary Sensor Networks

Christos Laoudias, Michalis Michaelides and Christos Panayiotou

KIOS Research Center for Intelligent Systems and Networks
Department of Electrical and Computer Engineering
University of Cyprus, Nicosia, Cyprus

Supported by the Cyprus Research Promotion Foundation under Grant TIIE/OPIZO/0609(BE)/06
Outline

Introduction

Fault Model

Tracking Architecture

Simulation Results

Conclusions
Motivation of our work

- Binary sensor networks
  - Popular for demanding and safety critical applications, e.g. large area monitoring, target tracking
Motivation of our work

- Binary sensor networks
  - Popular for demanding and safety critical applications, e.g. large area monitoring, target tracking

- Living with faults
  - Sensing can be tampered (accidentally or deliberately) and detection/estimation suffers from faulty sensors
  - Tracking accuracy can be severely degraded
Motivation of our work

- Binary sensor networks
  - Popular for demanding and safety critical applications, e.g. large area monitoring, target tracking

- Living with faults
  - Sensing can be tampered (accidentally or deliberately) and detection/estimation suffers from faulty sensors
  - Tracking accuracy can be severely degraded

- Faulty sensors should NOT be used
  - Localization algorithms typically use all sensor readings regardless of the actual sensor’s state
  - Sensor states are usually unavailable or extremely hard to obtain in real WSN applications
  - **Sensor Health State Estimation**: Intelligently select (at least mostly) healthy sensors for target tracking
Binary Sensor Network Model

Assumptions

1. A set of static sensor nodes $\ell_n = (x_n, y_n), \ n = 1, \ldots, N$
2. A source moving at steady speed $\ell_s(t) = (x_s(t), y_s(t))$
3. The source emits a continuous omnidirectional signal

$$z_n(t) = \frac{c}{1 + d_n(t)^\gamma} + w_n(t),$$

where $d_n(t) = ||\ell_n - \ell_s(t)||$.

Sensor Alarm Status

$$A_n(t) = \begin{cases} 0 & \text{if } z_n(t) < T \\ 1 & \text{if } z_n(t) \geq T \end{cases}$$
Stochastic Model
Markov Chain model with two discrete sensor states $s_n(t) \in \{F, H\}$

$$\pi_n(t + 1) = C^T \pi_n(t)$$

- Sensor state probabilities
  $$\pi_n(t) = [\pi^F_n(t) \pi^H_n(t)]^T,$$
  $$\pi^i_n(t) = \mathbb{P}[s_n(t) = i], \ i \in \{F, H\}$$

- $C = \begin{bmatrix} p^{F,F} & p^{F,H} \\ p^{H,F} & p^{H,H} \end{bmatrix}$

- Steady state probabilities
  $$\pi^i_n = \lim_{t \to \infty} \mathbb{P}[s_n(t) = i], \ i \in \{F, H\}$$

Fault Generation
- Diverse fault types
- Different duration, e.g. temporary, permanent
- $p^{H,H} = 0.925$ and $p^{F,F} = 0.7$ gives $[\pi^F_n \pi^H_n]^T = [0.2 \ 0.8]^T$
- $p^{F,F} = 1$ injects permanent faults
Sensor Fault Model

Stochastic Model
Markov Chain model with two discrete sensor states \( s_n(t) \in \{F, H\} \)

\[
\pi_n(t + 1) = C^T \pi_n(t)
\]

- Sensor state probabilities
  \[
  \pi_n(t) = [\pi_n^F(t) \pi_n^H(t)]^T, \\
  \pi_n^i(t) = P[s_n(t) = i], \ i \in \{F, H\}
  \]

- \( C = \begin{bmatrix} p^{F,F} & p^{F,H} \\ p^{H,F} & p^{H,H} \end{bmatrix} \)

- Steady state probabilities \( \pi_n^i = \lim_{t \to \infty} P[s_n(t) = i], \ i \in \{F, H\} \)

Fault Generation
- Diverse fault types
- Different duration, e.g. temporary, permanent
  \[
  p^{H,H} = 0.925 \text{ and } p^{F,F} = 0.7 \text{ gives } [\pi_n^F \pi_n^H]^T = [0.2 \ 0.8]^T
  \]
- \( p^{F,F} = 1 \) injects permanent faults
Types of faults

Reverse Status (RS)

- Sensors report the opposite readings than the expected ones
- Software bugs, compromised sensors, malicious network
Types of faults

Reverse Status (RS)

- Sensors report the opposite readings than the expected ones
- Software bugs, compromised sensors, malicious network

Stuck-At-1 (SA1)

- Sensors constantly report the presence of a source
- Board overheating, low battery, wrongly programmed threshold (i.e., low $T$), deployment of small decoy sources
Types of faults

Reverse Status (RS)

- Sensors report the opposite readings than the expected ones
- Software bugs, compromised sensors, malicious network

Stuck-At-1 (SA1)

- Sensors constantly report the presence of a source
- Board overheating, low battery, wrongly programmed threshold (i.e., low $T$), deployment of small decoy sources

Stuck-At-0 (SA0)

- Sensors fail to detect the source inside their ROC
- Dropped packets, high threshold $T$
Fault Tolerant Target Tracking Architecture

Sensor State Estimation component

- $\hat{s}_n(t)$: Estimated health state of each sensor

Localization component

- $\hat{l}_s(t)$: estimated target location
- $\hat{e}_s(t)$: estimation of the localization error (uncertainty)

Smoothing component

- $\tilde{l}_s(t)$: final location estimate (more accurate)
Overview of SNAP Localization

Subtract on Negative Add on Positive (SNAP) algorithm

- Event detection in binary sensor networks
- Low computational complexity and fault tolerance
Overview of SNAP Localization

Subtract on Negative Add on Positive (SNAP) algorithm

- Event detection in binary sensor networks
- Low computational complexity and fault tolerance

Algorithm Steps

1. **Grid Formation**: The entire area is divided into a grid $G$ with dimensions $R_x \times R_y$ and grid resolution $g$.
2. **Region of Coverage (ROC)**: Given $G$, the $ROC_n$ of a sensor is a neighborhood of grid cells around the sensor node location.
3. **Likelihood Matrix $L$ Construction**: All sensors add $+1$ (alarmed) or $-1$ (non-alarmed) to the cells that correspond to their $ROC$ and contributions are added for each cell.
4. **Maximization**: The maximum value in $L$ matrix, denoted as $L_{max}$, points to the estimated source location.
Example Application of SNAP

- Square $ROC_n$ for alarmed and non-alarmed sensors
- Source is correctly localized in the grid cell with $L_{max} = +3$
Particle Filter Tracking

Target state and measurement model

\begin{align}
X(t) &= \Phi X(t - 1) + \Gamma W(t - 1) \\
Y(t) &= M X(t) + U(t),
\end{align}

where \( X(t) = [x_s(t) \ y_s(t) \ u_x(t) \ u_y(t)]^T \) is the target state

Particle Filter Steps

A set of particles \( \{X^i(t - 1)\}_{i=1}^{N_p} \) with weights \( \{\omega^i(t - 1)\}_{i=1}^{N_p} \)

1. \( X^i(t) = \Phi X^i(t - 1) + \Gamma W(t - 1) \)

2. \( \left( \hat{\ell}_s(t), \hat{e}_s(t) \right) = SNAP(\hat{s}_n(t), A_n(t)) \)

3. \( \omega^i(t) = \omega^i(t - 1)p(t), \quad p(t) = \frac{1}{\sqrt{2\pi \sigma(t)}} \exp\left(-\frac{(\bar{X}^i(t) - \hat{\ell}_s(t))^2}{2\sigma(t)^2}\right) \)

4. \( \omega^i(t) = \frac{\omega^i(t)}{\sum_{i=1}^{N_p} \omega^i(t)} \) and Linear Time Resampling

5. \( \tilde{\ell}_s(t) = \sum_{i=1}^{N_p} \omega^i(t)X^i(t) \)
The estimator is based on a Markov Chain model

\[ \hat{\pi}_n(t + 1) = \hat{C}_n(t)^T \hat{\pi}_n(t), \]  

where \( \hat{\pi}_n(t) = [\hat{\pi}_n^F(t) \hat{\pi}_n^H(t)]^T, \hat{\pi}_n^i(t) = P[\hat{s}_n(t) = i], i \in \{F, H\} \)

\[ \hat{C}_n(t) = \begin{bmatrix} \hat{p}_n^{F,F}(t) & \hat{p}_n^{F,H}(t) \\ \hat{p}_n^{H,F}(t) & \hat{p}_n^{H,H}(t) \end{bmatrix}, \]

where \( \hat{p}_n^{i,j}(t) \neq p^{i,j} i, j \in \{F, H\} \).

Binary error signal \( r_n(t) \)

\[ r_n(t) = \begin{cases} 
1 & \text{if } d_n(t) \leq R_l \ \text{AND} \ A_n(t) = 0 \\
1 & \text{if } d_n(t) > R_l \ \text{AND} \ A_n(t) = 1 \\
0 & \text{if } d_n(t) \leq R_l \ \text{AND} \ A_n(t) = 1 \\
0 & \text{if } d_n(t) > R_l \ \text{AND} \ A_n(t) = 0 
\end{cases} \]
Main Idea
Obtain $\hat{s}_n(t + 1)$ by calculating the probability of a sensor being at a specific state given the current error signal, i.e.
$\hat{\pi}_n^{i|q}(t) = P[s_n(t) = i|r_n(t) = q], \ i \in \{F, H\}, \ q \in \{0, 1\}.$

ML Sensor State Estimate

$\hat{s}_n(t + 1)|r_n(t) = q = \arg \max_{i \in \{F, H\}} \hat{\pi}_n^{i|q}(t), \ q \in \{0, 1\}. \ \ (6)$

Using Bayes’ rule

$\hat{\pi}_n^{i|q}(t) = \frac{P[r_n(t) = q|s_n(t) = i]\hat{\pi}_n^i(t)}{P[r_n(t) = q]} \ \ (7)$

$\hat{\pi}_n^{i|q}(t) = \frac{P[r_n(t) = q|s_n(t) = i]\hat{\pi}_n^i(t)}{\sum_{j \in \{F, H\}} P[r_n(t) = q|s_n(t) = j]\hat{\pi}_n^j(t)} \ \ (8)$
Definitions
Probability of a sensor having a \textit{wrong} output given its state

\begin{itemize}
  \item $p^h_n(t) = P[r_n(t) = 1|s_n(t) = H]$  
  \item $p^f_n(t) = P[r_n(t) = 1|s_n(t) = F]$  
\end{itemize}

\begin{align}
\hat{\pi}^F_{n|1}(t) &= \frac{p^f_n(t) \cdot \hat{\pi}^F_n(t)}{p_n(t) \cdot \hat{\pi}^F_n(t) + p^h_n(t) \cdot \hat{\pi}^H_n(t)} \tag{9} \\
\hat{\pi}^H_{n|1}(t) &= \frac{p^h_n(t) \cdot \hat{\pi}^H_n(t)}{p_n(t) \cdot \hat{\pi}^F_n(t) + p^h_n(t) \cdot \hat{\pi}^H_n(t)} \tag{10} \\
\hat{\pi}^F_{n|0}(t) &= \frac{(1 - p^f_n(t)) \cdot \hat{\pi}^F_n(t)}{(1 - p^f_n(t)) \cdot \hat{\pi}^F_n(t) + (1 - p^h_n(t)) \cdot \hat{\pi}^H_n(t)} \tag{11} \\
\hat{\pi}^H_{n|0}(t) &= \frac{(1 - p^h_n(t)) \cdot \hat{\pi}^H_n(t)}{(1 - p^f_n(t)) \cdot \hat{\pi}^F_n(t) + (1 - p^h_n(t)) \cdot \hat{\pi}^H_n(t)}. \tag{12}
\end{align}
In case the sensor output is wrong, i.e. $r_n(t) = 1$

$$
\hat{s}_n(t + 1)|r_n(t)\neq 1 = \begin{cases} 
H & \text{if } \hat{\pi}^H_n(t) > \frac{p^f_n(t)}{p^f_n(t) + p^h_n(t)} \\
F & \text{otherwise}
\end{cases} \quad (13)
$$

In case the sensor output is correct, i.e. $r_n(t) = 0$

$$
\hat{s}_n(t + 1)|r_n(t)\neq 0 = \begin{cases} 
F & \text{if } \hat{\pi}^H_n(t) < \frac{1-p^f_n(t)}{2-p^f_n(t)-p^h_n(t)} \\
H & \text{otherwise}
\end{cases} \quad (14)
$$

- Only $\hat{\pi}^H_n(t)$, $p^h_n(t)$ and $p^f_n(t)$ need to be computed for estimating the sensor health state, given that $r_n(t)$ is known
- **Problem:** $r_n(t)$ is not available (target location is unknown)
- **Solution:** $\tilde{r}_n(t)$ estimates $r_n(t)$ by substituting $d_n(t)$ with $\tilde{d}_n(t)$, where $d_n(t) = \|\ell_n - \ell_s(t)\|$
Simple Estimator

Assumption
The error signal $\tilde{r}_n(t)$ is always equal to 1 when the sensor is Faulty and always equal to 0 when the sensor is Healthy.

Sensor State Estimate
This means that $p^f_n(t) = 1$ and $p^h_n(t) = 0$, $\forall t$ leading to

$$\hat{s}_n(t + 1) = \begin{cases} H & \text{if } \tilde{r}_n(t) = 0 \\ F & \text{if } \tilde{r}_n(t) = 1 \end{cases}$$  \hspace{1cm} (15)

▶ Intuition: If we fully trust the error signal, then the sensor health state is reliably estimated by $\tilde{r}_n(t)$

▶ Problem: Fully trusting the error signal $\tilde{r}_n(t)$ is not a good strategy

▶ Solution: Incorporate previous estimations that are encapsulated in the estimated sensor state probabilities
**Static Estimator**

**Assumption**

The Markov Chain in the Sensor State Estimator has reached equilibrium.

**Sensor State Estimate**

We may employ an estimate of the unknown steady state probability $\hat{\pi}_n^H$ to determine the sensor health state as

$$
\hat{s}_n(t+1)|_{r_n(t)=1} = \begin{cases} 
H & \text{if } \hat{\pi}_n^H > \frac{p_n^f(t)}{p_n^f(t)+p_n^h(t)} \\
F & \text{otherwise}
\end{cases}
$$

(16)

$$
\hat{s}_n(t+1)|_{r_n(t)=0} = \begin{cases} 
F & \text{if } \hat{\pi}_n^H < \frac{1-p_n^f(t)}{2-p_n^f(t)-p_n^h(t)} \\
H & \text{otherwise}
\end{cases}
$$

(17)
Static Estimator

The steady state probabilities are computed with

\[
\begin{bmatrix}
\hat{\pi}_n^F \\
\hat{\pi}_n^H
\end{bmatrix} = \hat{C}_n^T(t) \begin{bmatrix}
\hat{\pi}_n^F \\
\hat{\pi}_n^H
\end{bmatrix},
\]

where \( \hat{p}^{i,j}_n(t) \) in \( \hat{C}_n(t) \) can be estimated online by

\[
\hat{p}^{i,j}_n(t) = \frac{R^{i,j}_n(t)}{\sum_{k \in \{F, H\}} R^{i,k}_n(t)}, \quad i, j \in \{F, H\},
\]

where \( R^{i,j}_n(t) \) increases by one if \( \hat{s}_n(t - 1) = i \) and \( \hat{s}_n(t) = j \).

**Calculation of \( p^h_n(t) \) and \( p^f_n(t) \)**

\[
p^h_n(t) = (1 - Q_w(t))(1 - Q_d(t)) + Q_w(t)Q_d(t)
\]

\[
p^f_n(t) = (1 - Q_w(t))Q_d(t) + Q_w(t)(1 - Q_d(t))
\]

\[
Q_w(t) = Q \left( \frac{T - \mu_n(t)}{\sigma_w} \right), \quad \mu_n(t) = \frac{c}{1 + \tilde{d}_n(t)\gamma}, \quad Q_d(t) = Q \left( \frac{R_l - \tilde{d}_n(t)}{\sigma_d} \right).
\]
Dynamic Estimator

Main Idea

Consider the error signal not only for estimating the unknown sensor state, but also for updating the estimated sensor state probabilities.

\[
\begin{bmatrix}
\hat{\pi}_n^F(t+1) \\
\hat{\pi}_n^H(t+1)
\end{bmatrix} = \hat{C}_n^T(t) \begin{bmatrix}
\hat{\pi}_n^F|q(t) \\
\hat{\pi}_n^H|q(t)
\end{bmatrix}, \quad q \in \{0, 1\} \quad (22)
\]

- **Intuition:** All previous observations of the error signal are encapsulated in the estimated sensor state probabilities, thus affecting the future estimation steps.
Simulation Setup

Sensor field
100 × 100 field, \( N = 600 \) sensors, single source, staircase path \( M = 180 \)

Fault model
2-state Markov Chain with varying \( p^{i,j} \), \( i, j \in \{ F, H \} \) to generate temporary and permanent faults

Performance Metrics

- Cumulative state estimation error \( E_s = \frac{1}{NM} \sum_{t=1}^{M} \sum_{n=1}^{N} \epsilon_n(t) \)
  
  \[ \epsilon_n(t) = \begin{cases} 
  0 & \text{if } \hat{s}_n(t) = s_n(t) \\
  1 & \text{if } \hat{s}_n(t) \neq s_n(t) 
  \end{cases} \]

- Tracking error \( E_T = \frac{1}{M} \sum_{t=1}^{M} ||\tilde{\ell}_s(t) - \ell_s(t)|| \)
Results (Permanent faults)

\[ z_n(t) = \frac{5000}{1+d_n(t)^2} + w_n(t), \quad w_n \sim \mathcal{N}(0, 1000), \quad T = 50 \text{ and } R_I = 10 \]

- Reverse Status faults
- Adaptive particle filter with \( N_p = 500 \) particles
Results (Permanent and Temporary faults)

- Temporary mixed and permanent Reverse Status faults
Concluding Remarks

- Introduced a Markov Chain fault model to generate different types of real faults documented in the literature
- The proposed architecture addresses the joint target tracking and sensor health state estimation problem in binary WSNs
- Maintain a high level of tracking accuracy, even when a large number of sensors in the field fail
- Next steps
  - Incorporate the correlation of the alarm status $A_n(t)$ for neighboring sensors into the error signal $r_n(t)$
  - Decentralized architecture for multiple target tracking
Thank you for your attention

Contact
Christos Laoudias
KIOS Research Center for Intelligent Systems and Networks
Department of Electrical & Computer Engineering
University of Cyprus
Email: laoudias@ucy.ac.cy
Extra Slides
Region of Influence (ROI)
Area around the source where a sensor is alarmed with $p \geq 0.5$

Region of Coverage ($ROC_n$)
Area around a sensor $n$ where a source (if present) it will be detected with $p \geq 0.5$
False Positive and False Negative sensors
Erroneous Sensor Behaviour

- **False Positive** and **False Negative** sensors

---

IEEE International Conference on Communications, Budapest, Hungary

12 June 2013
Interpretation of the Error Signal

$\text{interpretation}$

- Motivation
- WSN Model
- Fault Model
  - Markov Chain Model
- Tracking Architecture
  - Block Diagram
  - Localization
  - Smoothing
  - Sensor State Estimation
- Simulation Results
  - Simulation Setup
  - Evaluation
- Conclusions
Interpretation of the Error Signal

\[ r_n(t) = \begin{cases} 1 & \text{sensor output is wrong} \\ 0 & \text{sensor output is correct} \end{cases} \]

- sensor \( n \) is inside the ROI and is non-alarmed or
- sensor \( n \) is outside the ROI and is alarmed

- sensor \( n \) is inside the ROI and is alarmed or
- sensor \( n \) is outside the ROI and is non-alarmed
Interpretation of the Error Signal

- $r_n(t) = 1$ (sensor output is wrong)
  - sensor $n$ is inside the ROI and is non-alarmed or
  - sensor $n$ is outside the ROI and is alarmed
Interpretation of the Error Signal

- $r_n(t) = 1$ (sensor output is *wrong*)
  - sensor $n$ is inside the ROI and is non-alarmed or
  - sensor $n$ is outside the ROI and is alarmed

- $r_n(t) = 0$ (sensor output is *correct*)
  - sensor $n$ is inside the ROI and is alarmed or
  - sensor $n$ is outside the ROI and is non-alarmed
In this scenario $\tilde{r}_n(t) \neq r_n(t)$ for 6 sensors.
In this scenario $\tilde{r}_n(t) \neq r_n(t)$ for 6 sensors
Results with SNAP (RS faults)
Results (RS and SA faults)

- **Cumulative Estimation Error ($\varepsilon_s$)**

- **Percentage of faulty sensors ($\alpha$)**

- **SNAP+SPF**
- **TI+SPF**
- **ftTRACK(Simple)**
- **ftTRACK(Static)**
- **ftTRACK(Dynamic)**

**Tracking Error**

- **Percentage of faulty sensors ($\alpha$)**
Results with SNAP (SA faults)

**Figure:** SA1 faults.

**Figure:** SA0 faults.
Introduction
- Motivation
- WSN Model

Fault Model
- Markov Chain Model

Tracking Architecture
- Block Diagram
- Localization
- Smoothing
- Sensor State Estimation

Simulation Results
- Simulation Setup
- Evaluation

Conclusions

Results with Variable Source Energy

Figure: Temporary RS faults \((\alpha = 25\%)\).

Figure: Temporary mixed faults \((\alpha = 38\%)\).
Results with CE (RS faults)

\[ z_n(t) = \frac{3000}{1+d_n(t)^2} + w_n(t), \quad w_n \sim \mathcal{N}(0, 1), \quad T = 5 \text{ and } R_I = 24.5 \]

\[ \hat{\ell}_s(t) = \left( \frac{1}{P} \sum_{p=1}^{P} x_p, \frac{1}{P} \sum_{p=1}^{P} y_p \right) \]

\[ (x_p, y_p), \quad p = 1, \ldots, P \quad (P \leq N) \text{ and } A_p(t) = 1 \]

\[ \text{Standard particle filter with } N_p = 500 \text{ particles} \]
Results with CE (RS and SA faults)