Enhanced Distributed Filtering using a Trusted Core

Kiran K. Somasundaram and John S. Baras

- **Realistic sensor networks**: Normal nodes, faulty or corrupted nodes, malicious nodes
- **Hierarchical scheme** – inspired from social and economic architectures which provides global trust on a particular context without requiring direct trust on the same context between all agents.
- Combines techniques from fusion centric, collaborative filtering, estimation propagation
- **Trusted Core(TC)**
  - Trust Particles, higher security, additional sensing capabilities, broader observation of the system, confidentiality and integrity, multipath communication.
  - Every sensor can communicate with one or more trust particles by paying an communication and crypto cost.

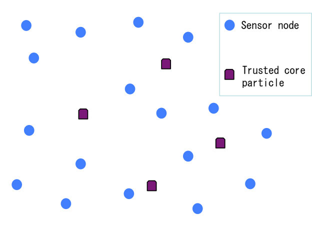


Figure 1: Sensor Network Realization

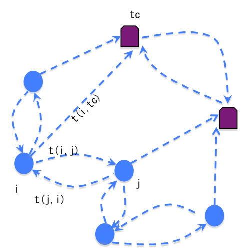


Figure 2: Local Dynamic Trust Relations

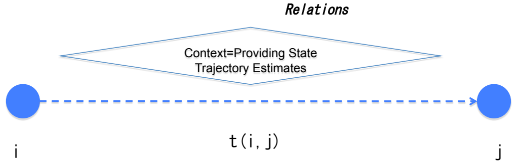


Figure 3: Context in trust relation

Goals of Trusted Core System Design

1. All the sensors which abide by the protocols of sensing and message passing, should be able to track the trajectories.
2. This implies that those nodes which have poor sensing capabilities, *nodes with corrupted sensors, should* be aided by their neighbors in tracking.
3. Those nodes which are *malicious and pass false estimates,* should be quickly detected by the trust mechanism and their estimates should be discarded.

Gauss Markov System Model

$$\underline{x}[n+1] = \mathbf{A}\underline{x}[n] + \mathbf{B}\underline{w}[n]$$

Observations

$$\underline{z}_i[n] = \mathbf{H}_i[n]\underline{x}[n] + \underline{v}_i[n]$$

Distributed Estimation Algorithms

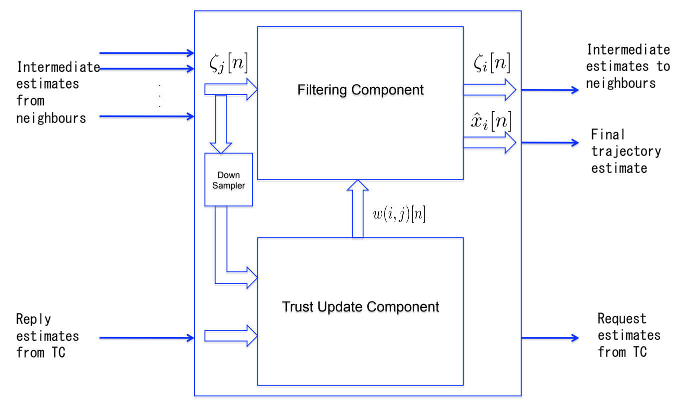
Algorithm 1 Trusted Kalman Filter

Init $M[0], \hat{\underline{x}}_i = \underline{x}(0), n = 0$
repeat
  $n \leftarrow n+1;$
  Prediction MSE
  $\mathrm{P}[n] = \mathrm{A}\mathrm{M}[n-1]\mathrm{A}^T + \mathrm{B}\mathrm{Q}\mathrm{B}^T$
  Kalman Gain
  $\mathrm{K}[n] = \mathrm{P}[n]\mathrm{H_i}^T(\mathrm{R_i} + \mathrm{H_i}\mathrm{P}[n]\mathrm{H_i}^T)^{-1}$
  Local correction
  $\underline{\zeta}_i[n] = \mathrm{A}\hat{\underline{x}}_i[n-1] + \mathrm{K}[n](\underline{z}_i[n] - \mathrm{H_i}\mathrm{A}\hat{\underline{x}}_i[n-1])$
  The nodes exchanges the local estimates $\hat{\underline{x}}_j, \forall j \in \mathcal{N}^+(i)$
  Trust sensitive filtering
  $\hat{\underline{x}}_i[n] = \sum_{j in \mathcal{N}^+(i)} w_{ij} \times \underline{\zeta}_j[n]$
  Estimation MSE
  $\mathrm{M}[n] = (\mathbf{I} - \mathrm{K}[n]\mathrm{H_i}[n])\mathrm{P}[n]$
until Forever

Algorithm 2 Trust Update for the inclusive neighborhood

Init $t(i,j)[0] = \frac{1}{|\mathcal{N}^+(i)|}, \quad \forall j \in \mathcal{N}^+(i)$, and $k = 0$
repeat
  Wait for Exponential time τ
  $k \leftarrow k + \tau$
  Request Estimate update from the TC
  The TC replies with its trustworthy estimate $\hat{\underline{x}}_{tc}$
  for all $j \in \mathcal{N}^+(i)$ **do**
    $dev(j) = ||\underline{\zeta}_j - \underline{\zeta}_{tc}||_2$
    $t(i,j)[k]$
    $= \begin{cases} \min(max_T, t(i,j)[k-1]) + \delta & dev(j) \leq Dev_T \\ t(i,j)[k-1]/2 & dev(j) > Dev_T \end{cases}$
  end for
until Forever

Component Architecture for Filtering Particles

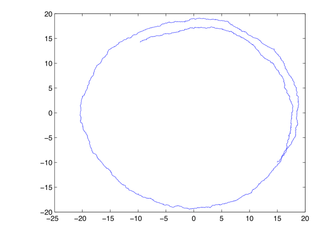


Simulations

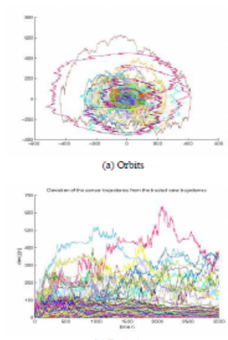
$$\mathbf{A} = 2\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

$$\mathbf{Q} = 25\mathbf{I}_2, \quad \underline{x}(0) = (15, -10)^T$$

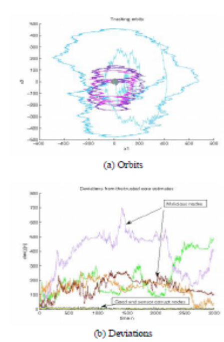
$$\mathbf{H_{tc}} = I_2, \quad \mathbf{R_{tc}} = 30\mathbf{I_2}$$



System Orbit



Noisy Open Loop Tracking Performance



Trust Feedback-Closed Loop Tracking Performance