Distributed and Network-Aware Estimation

Lubomír Baramov, Vladimír Havlena

Honeywell ACS AT Laboratory Prague

Honeywell

Agenda

- Network-aware estimation - lost data
- Network-aware estimation - delayed data
- Distributed Kalman filtering

Objectives

- Explain core ideas
- Handouts
  - further reading
  - selected references

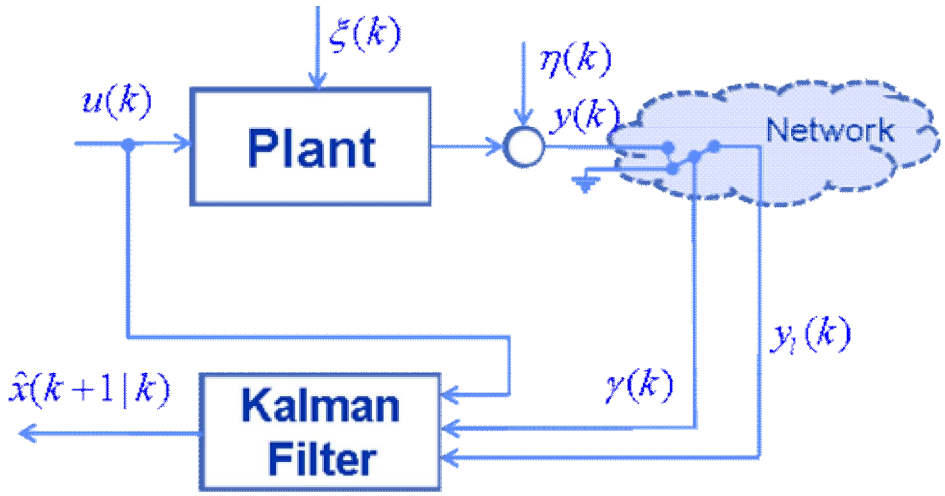
Network-aware estimation 1

LOST DATA

Kalman Filtering with intermittent observations

Honeywell

Problem formulation



- **Plant model**

$$x(k+1) = Ax(k) + Bu(k) + \xi(k)$$
$$y(k) = Cx(k) + \eta(k)$$

  - Initial state

$$x(0) = N\left(\hat{x}(0), P_0\right)$$

- **Noise model**
  - White, Gaussian, zero-mean

$$\mathrm{E}\left(\xi(k)\xi(k)^T\right) = Q, \quad \mathrm{E}\left(\eta(k)\eta(k)^T\right) = R$$
$$\mathrm{E}\left(\xi(k)\eta(k)^T\right) = 0$$

- **Network model**
  - Measurement received/lost
  - Double-valued random variable

$$\gamma(k) \in \{0,1\}$$

Kalman Filtering with intermittent observations

Honeywell

- **Data update step (filtering)**
  - Output error

$$e(k \mid k-1) = \gamma(k)\left(y(k) - C\hat{x}(k \mid k-1)\right)$$

  - Kalman gain

$$K(k) = P(k \mid k-1)C^T\left(CP(k \mid k-1)C^T + R\right)^{-1}$$

  - State-mean (conditioned by data up to time k)

$$\hat{x}(k \mid k) = \hat{x}(k \mid k) + K(k)e(k \mid k-1)$$

  - State covariance

$$P(k \mid k) = P(k \mid k-1) - \gamma(k)K(k)CP(k \mid k-1)$$

- **Time update step (prediction)**
  - State mean

$$\hat{x}(k+1 \mid k) = A\hat{x}(k \mid k) + Bu(k)$$

  - State covariance

$$P(k+1 \mid k) = AP(k \mid k)A^T + Q$$

- **Covariance matrix does not converge to steady-state covariance**
  - Time-varying system due to parameter $\gamma(k)$
  - State covariance matrix $P(k \mid k-1)$ is a stochastic variable

Kalman Filtering with intermittent observations

Honeywell

- **Asymptotic analysis**
  - *Sinopoli et al, 2004*
- **Network stochastic model**
  - Data loss probability: Bernoulli i.i.d.

$$P\big(\gamma(k)=1\big)=\lambda$$

- **Does the mean of the covariance matrix converge?**

$$\bar{P}=\lim_{k\to\infty}\mathrm{E}\big(P(k\,|\,k-1)\big)$$

- **Assumptions on plant model**
  - Detectability of (A,C)
  - Observability of (A,$Q^{1/2}$)

- **Critical probability of data delivery**

$$\lambda\le\lambda_c \;\Rightarrow\; \exists P(0):\ \bar{P}\to\infty$$
$$\lambda>\lambda_c \;\Rightarrow\; \forall P(0):\ \bar{P}<\infty$$

- **Bounds on critical probability**

$$\underline{\lambda}\le\lambda_c\le\bar{\lambda}$$

  - Lower bound obtained from spectral radius
  - Upper bound: LMI feasibility problem

- **Bounds on the state covariance**

$$\lambda>\bar{\lambda}:\quad \bar{S}_\lambda\le\bar{P}\le\bar{V}_\lambda$$

  - LMI-based bounds

Kalman Filtering with intermittent observations

Honeywell

Riccati equation convergence

- Banach fixed point theorem

$$P_{k+1} = f(P_k)$$

when $\lim_{k\to\infty} P_k = P^*; \quad P^* = f(P^*)$?

  - Contraction mapping $f(.)$

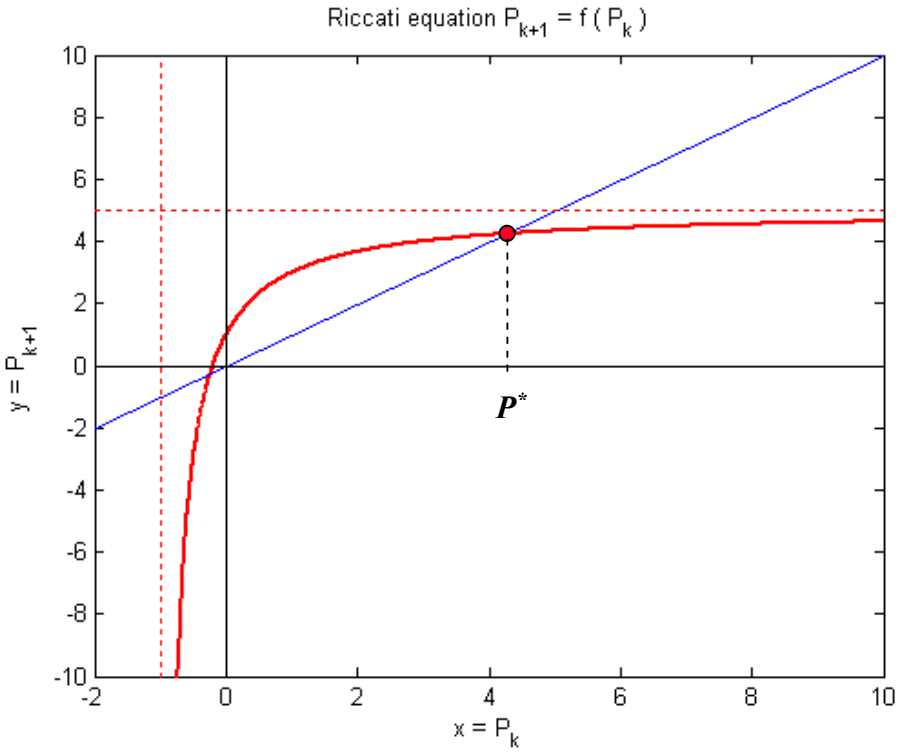
- Scalar Riccati equation

$$f(p_k) = a^2 p_k + q - \frac{a^2 c^2 p_k^2}{c^2 p_k + r}$$

- Convergence to $p^+ > 0$
  - asymptotes

$$a_x: \quad y = q + \frac{a^2}{c^2} r$$

$$a_y: \quad x = -\frac{r}{c^2}$$



Example: a=2, c=1, q=1, r=1

Kalman Filtering with intermittent observations

Honeywell

- modified Riccati equation

$$f(p_k) = a^2 p_k + q - \lambda \frac{a^2 c^2 p_k^2}{c^2 p_k + r}$$

- Convergence to $p^+ > 0$
  - Asymptotes

$$a_x: \quad y \approx (1-\lambda)a^2 x$$

$$a_y: \quad x = -\frac{r}{c^2}$$

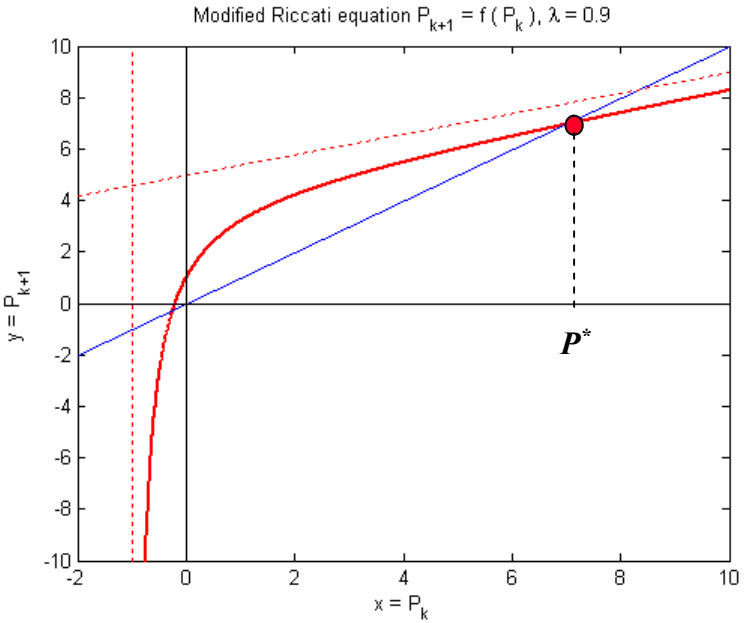
  - Convergence condition

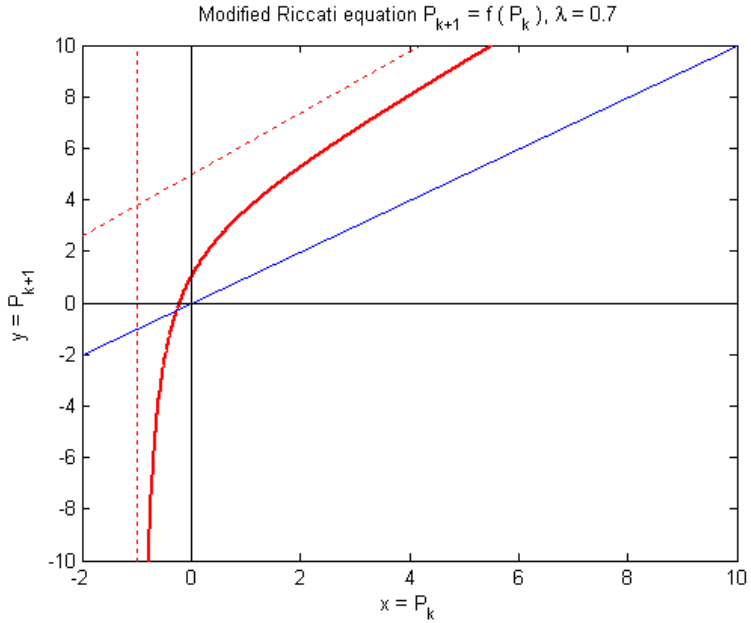
$$(1-\lambda)a^2 < 1 \quad \Rightarrow \quad \lambda > 1 - \frac{1}{a^2}$$

  - Example:

    a=2, c=1, q=1, r=1

$$\lambda > 1 - \frac{1}{2^2} = 0.75$$





Kalman Filtering with intermittent observations

Honeywell

Some extensions

- **Partial packet loss**
  - Multiple observation channels with independent data loss processes

    Liu and Goldsmith, 2004
- **Modified performance measure**
  - *Epstein et al, 2008*
  - Using 'covariance bound with high probability'

$$\Pr\left[P(k \mid k-1) \le M\right] > 1-\varepsilon$$

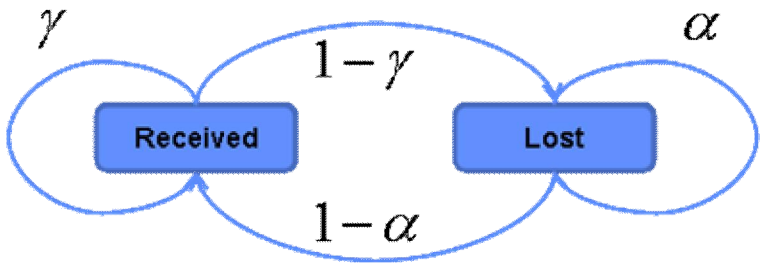
  - instead of

$$\lim_{k\to\infty} \mathrm{E}\left(P(k \mid k-1)\right)$$

- **Network model**
  - 2 state Markov Chain (Gilbert-Elliot model of network loss)
  - 'Bursting character', more realistic than Bernoulli model

    Huang and Dey (2007)

    Wang and Guo (2009)



Estimation with data loss in control channel

Honeywell

- *Schenato et al, 2007*
- **Assumption**
  - Control signal is zero, if no data received
- **Critical issue:**

  passing the info on lost packets to the estimator
  - **TCP – like protocol**

    (KF receives an acknowledgement on lost packets in the following sampling interval)
  - **UDP – like protocol**

    (KF receives no acknowledgement)

- **Network model**
  - i.i.d. Bernoulli probability distributions of successful transmission
  - Sensor data transmission

$$P\left(\gamma(k)=1\right)=\overline{\gamma},$$

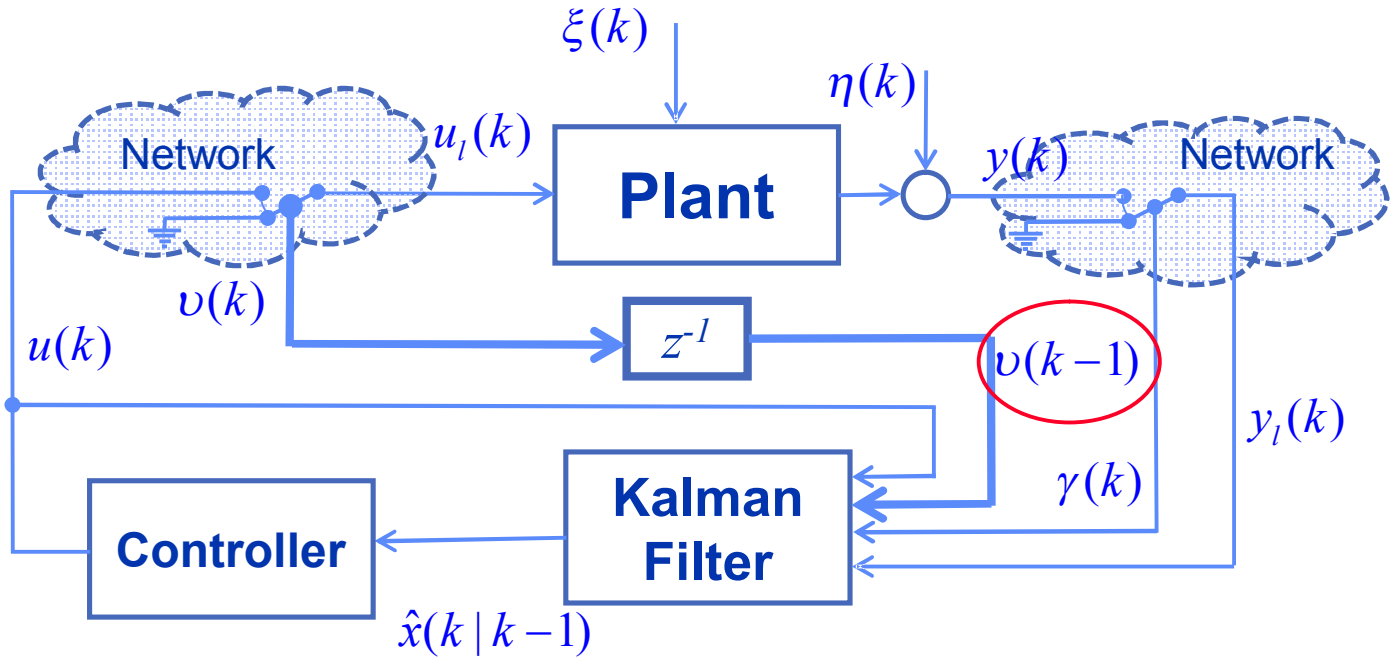
  - Actuator command transmission

$$P\left(\upsilon(k)=1\right)=\overline{\upsilon}$$

Estimation with data loss in control channel

Honeywell

Loop with the TCP-like protocol



Information about u(*k*) delivery available to KF

Estimation with data loss in control channel

Honeywell

Optimal estimation under the TCP-like protocol

- **Data update step (filtering)**
  - Prediction error

$$e(k \mid k-1) = \gamma(k)\left(y(k) - C\hat{x}(k \mid k-1)\right)$$

  - Kalman gain

$$K(k) = P(k \mid k-1)C^T\left(CP(k \mid k-1)C^T + R\right)^{-1}$$

  - State-mean
    (conditioned by data up to time *k*)

$$\hat{x}(k \mid k) = \hat{x}(k \mid k) + K(k)e(k \mid k-1)$$

  - State Covariance

$$P(k \mid k) = P(k \mid k-1) - \gamma(k)K(k)CP(k \mid k-1)$$

- **Time update step (prediction)**
  - State mean

$$\hat{x}(k+1 \mid k) = A\hat{x}(k \mid k) + \upsilon(k)Bu(k)$$

  - State covariance

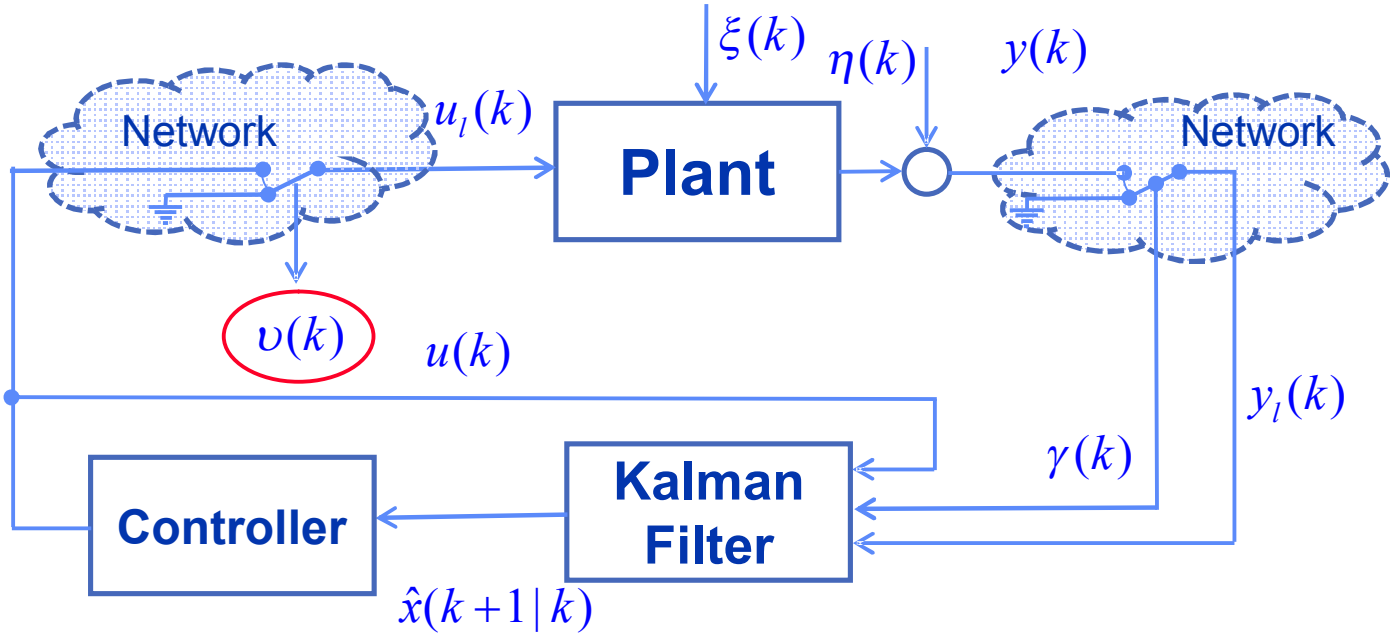
$$P(k+1 \mid k) = AP(k \mid k)A^T + Q$$

- **State covariance**
  - Independent on control
  - Separation principle holds

Estimation with data loss in control channel

Honeywell

Loop with the UDP-like protocol



Information about *u*(*k*) delivery not available

Estimation with data loss in control channel

Honeywell

Optimal estimation under the UDP-like protocol

- **Time update step (prediction)**
  - State mean

$$\hat{x}(k+1|k) = A\hat{x}(k|k) + \bar{\upsilon} Bu(k)$$

  - State covariance

$$P(k+1|k) = AP(k|k)A^T + Q + \bar{\upsilon}\left(1-\bar{\upsilon}\right) Bu(k)u(k)^T B^T$$

  - Input → stochastic disturbance
- **State covariance depends on control**
  - Separation principle does not hold
  - ***Optimal H2 control is no longer linear***
- **Critical transmission probabilities**
  - higher than for TCP-like case

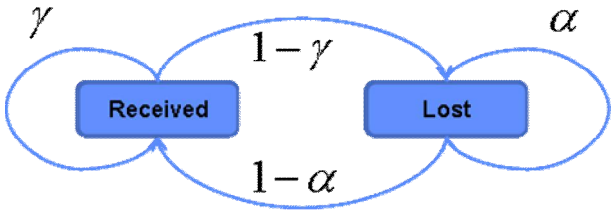
Jump Estimators for Jump Systems

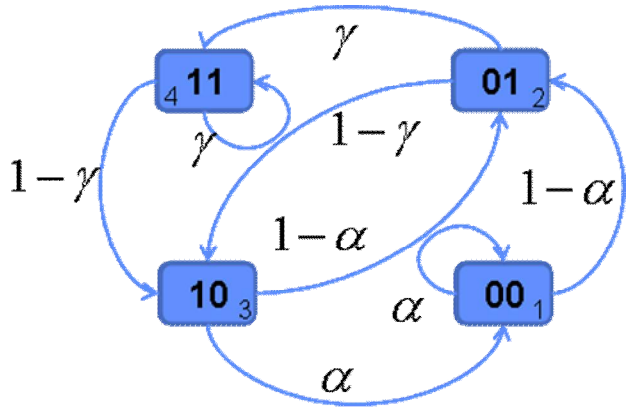
Honeywell

- **More complex loss modeling**
  - *Smith & Seiler, 2003*
  - Modeling data loss as a Markov chain
  - Sub-optimal State estimator as a switched linear system
  - Optimal in the sense of asymptotic mean of the state covariance matrix
- **Loss model as a 2-state Markov chain**

- **Finite loss history**
  - d_{max} – long history
  - 2^{dmax} states
  - Markov chain model /transition matrix

$$P = \left[p_{ij} \right] = \begin{bmatrix} \alpha & 1-\alpha & 0 & 0 \\ 0 & 0 & 1-\gamma & \gamma \\ \alpha & 1-\alpha & 0 & 0 \\ 0 & 0 & 1-\gamma & \gamma \end{bmatrix}$$





Jump Estimators for Jump Systems

Honeywell

Solution

- **Set of 'modal' covariances**

$$\left\{P^{(i)}(k)\right\}_{i=1}^{2^{d\max}}$$

- **Obtained via 'Riccati-like' eqn**
  - for $i = 1,\ldots, 2^{dmax}$

$$\begin{aligned} M^{(i)}(k) &= P^{(i)}(k) - \gamma(k) P^{(i)}(k) C^T \\ &\quad \times \left(CP^{(i)}(k)C^T + R\right)^{-1} CP^{(i)}(k) \end{aligned}$$

$$P^{(i)}(k+1) = \sum_{j=1}^{2^{d\max}} p_{ij}\left(AM^{(j)}(k)A^T + Q\right)$$

- **If converges, MS optimal switched estimator exists, minimizing**

$$J_\infty = \lim_{k\to\infty} \operatorname{trace}\left(\mathrm{E}\left[P(k\mid k)\right]\right)$$

- **Kalman gains**

$$\overline{P}_i = \lim_{k\to\infty} P^{(i)}(k)$$

$$\begin{aligned} K(k) &= K(i(k)) \\ &= \overline{P}^{(i(k))} C^T \left(C\overline{P}^{(i(k))}C^T + R\right)^{-1} \end{aligned}$$

- **Pre-computed off-line**
  - On-line effort very low

Lost data case: References

Honeywell

- Sinopoli, B., Schenato, L., Franceschetti, M., Poola, K., Jordan, M.I. and Sastry, S.: Kalman Filtering with Intermittent observation, *IEEE TAC*, Vol 49, Sept 2004, pp 1453-1464.
- Schenato, L.,. Sinopoli, B., Franceschetti, M., Poola, K., and Sastry, S.: Foundations of Control and Estimation Over Lossy Networks. *Proceedings of the IEEE*, Vol. 95, January 2007, pp163-187.
- Liu, X, and Goldsmith, A.: Kalman Filtering with Partial Observation Losses. *43rd IEEE Conf Decision and Control*, Nassau, Bahamas, 2004.
- Smith, S.C., and Seiler, P.: Estimation with Lossy Measurements: Jump Estimators for Jump Systems*, IEEE TAC*, Vol. 48, pp 2163-2171, 2003
- Huang, M. Dey, S.: Stability of Kalman filtering with Markovian packet losses, *Automatica*, vol 43, 2007, pp 598-607
- Sahebsara, M., Chen, T. and Shah, S. L.: Optimal H2 Filtering in Networked Control Systems With Multiple packet Dropout. IEEE TAC, Vol 52, 2007, 1508-1513
- Wang, B.-F., Guo,G.: Kalman filtering with Partial Markovian packet Losses*, Int J. of Automation and Computing*, Vol 6, 2009, 395-400.
- Epstein, M., Shi, L., Tiwari, A. and Murray, R.M.: probabilistic performance of state estimation acrosss a lossy network. *Automatica*, Vol. 44, 2008, pp 3046-3053.

Network-aware estimation 2

DELAYED DATA

Problem formulation

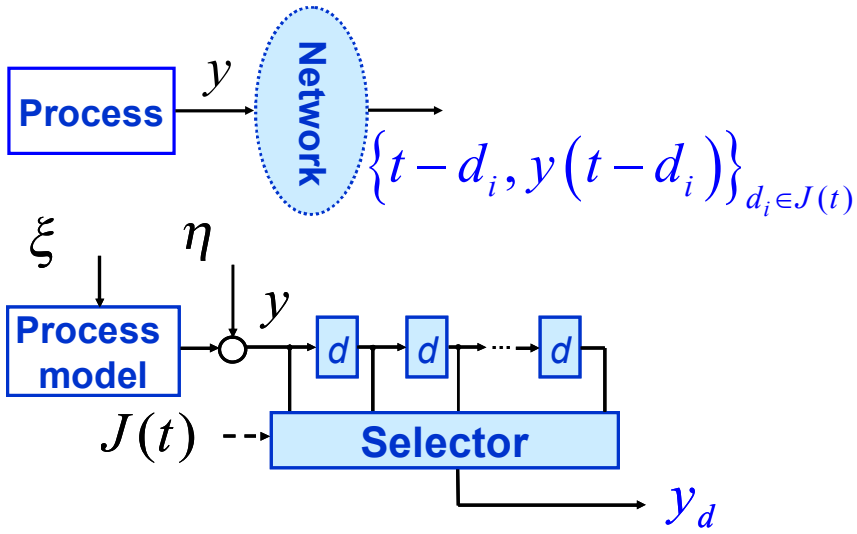
Honeywell

Assumptions

- **Measurement data arriving via wireless networks with random time delays**
- **Data are time-stamped**
  - Time-stamps marking the time of acquiring data by the sensor
  - Clock synchronization assumed
  - Delays known to the filter
  - Data arriving possibly out-of-sequence
  - More than one sample at a time from one sensor of different time stamps.
- **Lost packets – infinite delay**

Practical considerations:

- **finite buffer for delayed data**
  - Maximum delay considered d_{max}.
  - Delays are integer multiples of sampling period
- **Kalman filter framework**
  - Augmenting plant model by a chain of delays



Kalman filter for delayed measurements

Honeywell

- **Augmented model**

$$\begin{bmatrix} \mathbf{x}_0(k+1) \\ \hline \mathbf{x}_d(k+1) \end{bmatrix} = \left[\begin{array}{c|cc} \mathbf{A}_0 & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{C}_0 & \mathbf{0} & 0 \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \end{array}\right] \begin{bmatrix} \mathbf{x}_0(k) \\ \hline \mathbf{x}_d(k) \end{bmatrix} + \begin{bmatrix} \xi_0(k) \\ \hline \eta(k) \\ \mathbf{0} \end{bmatrix}$$

- **Data delivered at time *k***

$$J(k) = \{0, 2\}$$

$$\begin{bmatrix} y(k) \\ y(k-1) \\ y(k-2) \\ \vdots \\ y(k-n_{d\max}) \end{bmatrix} = \left[\begin{array}{c|ccccc} \mathbf{C}_0 & 0 & 0 & \cdots & 0 \\ \mathbf{0} & 1 & 0 & \cdots & 0 \\ \mathbf{0} & 0 & 1 & & 0 \\ & & & \ddots & \\ \mathbf{0} & 0 & 0 & \cdots & 1 \end{array}\right] \begin{bmatrix} \mathbf{x}_0(k) \\ \hline \mathbf{x}_d(k) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \eta(k)$$

$$\mathbf{y}_d(k) = \mathbf{C}_d(k)\mathbf{x}(k) + \mathbf{D}_r(k)\eta(k)$$

- **Time-varying measurement equation**

- **Classical time-varying KF**

$$\hat{\mathbf{x}}_{k+1|k} = \left(\mathbf{A} - \mathbf{L}(k)\mathbf{C}_d(k)\right)\hat{\mathbf{x}}_{k|k-1} + \mathbf{L}(k)\mathbf{y}_d(k)$$

$$\mathbf{P}_{k+1|k} = \mathbf{A}\mathbf{P}_{k|k-1}\mathbf{A}^T - \mathbf{L}(k)\mathbf{W}(k)\mathbf{L}(k)^T + \mathbf{Q}$$

$$\begin{aligned} \mathbf{L}(k) &= \left(\mathbf{A}\mathbf{P}_{k|k-1}\mathbf{C}_d^T(k) + \mathbf{S}(k)\right)\mathbf{W}(k)^{-1}, \\ \mathbf{W}(k) &= \mathbf{C}_d(k)\mathbf{P}_{k|k-1}\mathbf{C}_d(k)^T + \mathbf{D}_r(k)r_0\mathbf{D}_r(k)^T, \\ \mathbf{S}(k) &= E\left(\xi(k)\eta(k)^T\right)\mathbf{D}_r(k)^T. \end{aligned}$$

- Optimal in the linear-Gaussian setting
- Numerically extensive
- No steady-state estimator

Kalman filter for delayed measurement

Honeywell

Addressing computational issues

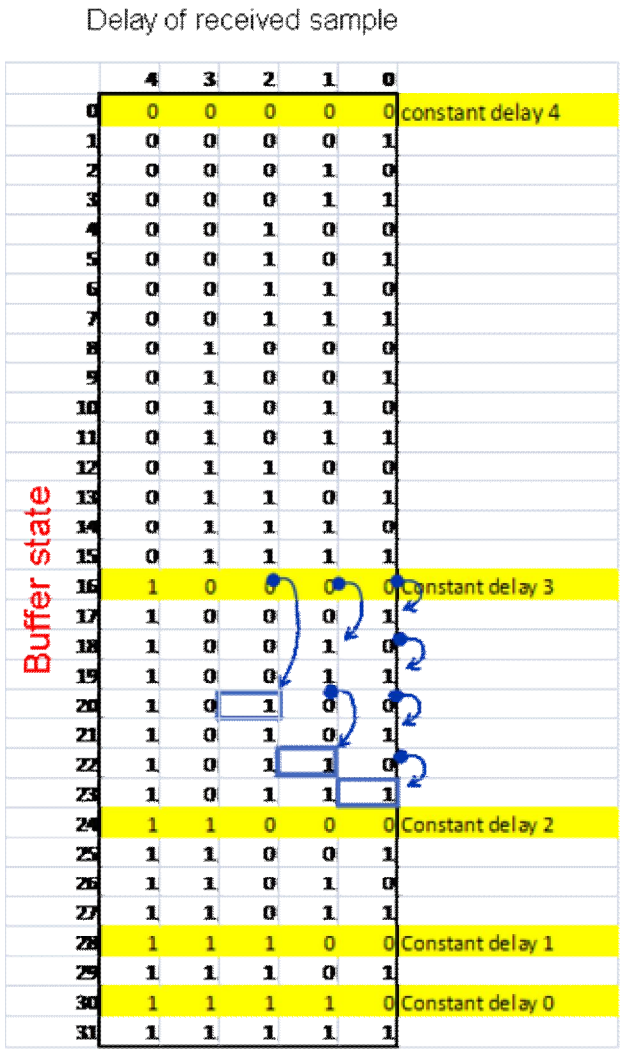
- **Efficient implementation of time-varying Kalman filter**
  - *Baramov et al, 2009*, single output systems
  - Pre-computing some quantities off-line
  - Exploiting problem structure
  - Specific representation of factorized covariance
  - About two times faster than the classical square root representation, and equally robust
  - Moderate storage requirements
- **Quasi-steady state Kalman filter**
  - *Pachner and Havlena, 2008*
- **Assumptions**
  - No data loss
  - All samples assumed to arrive with delay no greater than $d_{max.}$
- **Features**
  - The optimal filter converges
  - Asymptotic covariance depends entirely on the history of missing samples up to time $k- d_{max}$
  - Proven using Bayesian argument
  - A finite number of Kalman gains needs to be pre-computed off-line and stored

Kalman filter for delayed measurement

Honeywell

Quasi-steady state KF gains

- **Covariance matrix attains a finite number of values, if**
  - Maximum delay not exceeding d_{max}
  - Suitable initialization
- **Covariance matrices**
  - Depend on the state of the received data buffer
  - Total 2^{dmax+1} combinations (for single output)
  - Specific buffer states (persistent)
    - Sample $y(k-i)$ received for all $i \geq d$ not received for $i<d$, $1 \leq d \leq d_{max}$ e.g. [1 1 1 0 0]
    - Computed by algebraic Riccati equation
  - Other buffer states (transient)
    - Difference Riccati equation



Kalman gain for each zero entry

Quasi-steady state KF algorithm: Data step

Honeywell

- **Buffer state m**
  - Integer between 0 and 2^{dmax+1}
- **Fusing measurement of delay i**
  - **Augmented state mean**

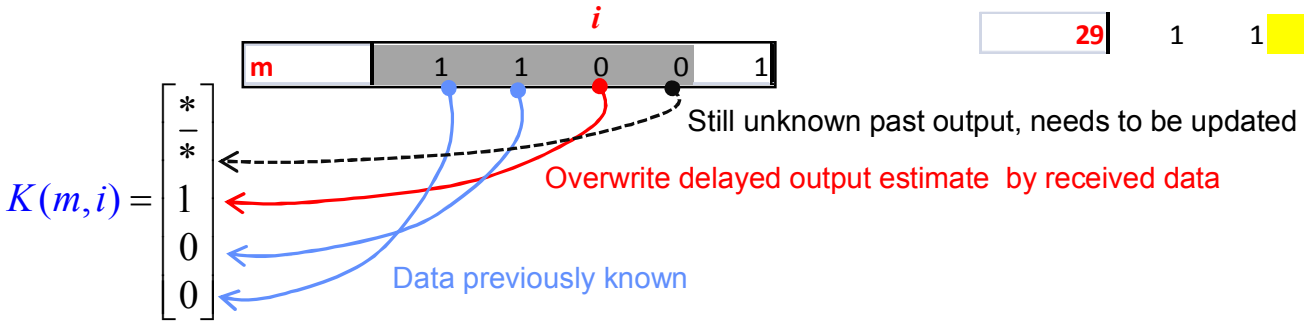
$$\hat{X}(k \mid k-1, \{y(k-j_s\}_{s=1}^{i}) = \hat{X}(k \mid k-1, \{y(k-j_s\}_{s=1}^{i-1}) + K(m,i)\left(y(k-i) - C_i X\right)$$

  - i = 0

$$K(m,0) = \frac{P_m(:,1:n_0)C_0^T}{C_0 P_m(1:n_0,1:n_0) + R_0}$$

  - i > 0

$$K(m,i) = \frac{P_m(:,n_0+i)}{P_m(n_0+i,n_0+i)}$$



  - Kalman gain $K(m,i)$ defined if the binary table entry is zero, i.e.,

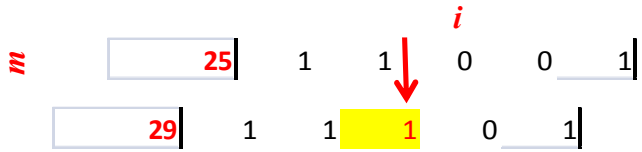
$$\mathrm{rem}\left(m, 2^{i+1}\right) < 2^i$$

  - Number of gains to store

$$2^{d_{\max}} \cdot \left(d_{\max} + 1\right)$$

  - **Buffer state update**

$$m(k \mid k-1, \{y(k-j_s\}_{s=1}^{i}) = m(k \mid k-1, \{y(k-j_s\}_{s=1}^{i-1}) + 2^i$$



Quasi-steady state KF: Time step

Honeywell

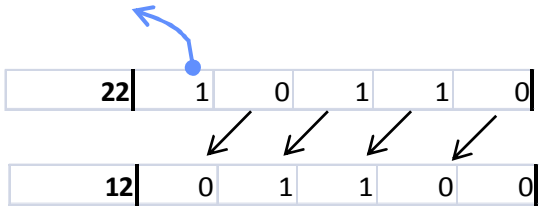
- **Time step**
  - **State mean**

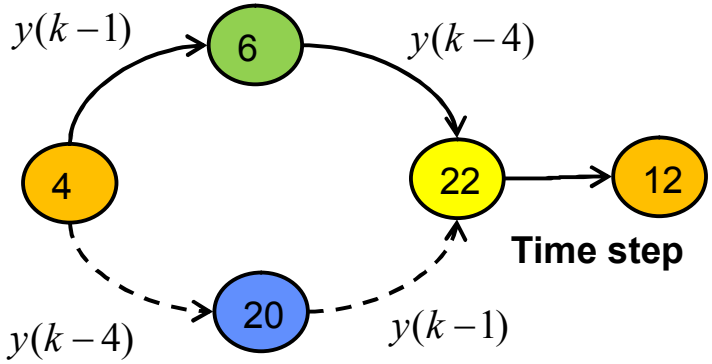
$$\hat{X}(k+1\,|\,k) = A\hat{X}\left(k\,|\,k-1, \{y(k-i)\,|\,i \in J(k)\}\right) + Bu(k)$$

  - **Buffer state**

$$m(k+1\,|\,k) = \mathrm{rem}\left(2m(k\,|\,k), 2^{d_{\max}+1}\right)$$

Must be 1 after fusing all data by assumption





Delay of received sample

Buffer state	4	3	2	1	0
0	0	0	0	0	0
1	0	0	0	0	1
2	0	0	0	1	0
3	0	0	0	1	1
4	0	0	1	0	0
5	0	0	1	0	1
6	0	0	1	1	0
7	0	0	1	1	1
8	0	1	0	0	0
9	0	1	0	0	1
10	0	1	0	1	0
11	0	1	0	1	1
12	0	1	1	0	0
13	0	1	1	0	1
14	0	1	1	1	0
15	0	1	1	1	1
16	1	0	0	0	0
17	1	0	0	0	1
18	1	0	0	1	0
19	1	0	0	1	1
20	1	0	1	0	0
21	1	0	1	0	1
22	1	0	1	1	0
23	1	0	1	1	1
24	1	1	0	0	0
25	1	1	0	0	1
26	1	1	0	1	0
27	1	1	0	1	1
28	1	1	1	0	0
29	1	1	1	0	1
30	1	1	1	1	0
31	1	1	1	1	1

Quasi-steady state Kalman filter - Conclusions

Honeywell

- **Very low on-line computational effort**
  - Almost as low as the asymptotic steady-state Kalman filter
  - Switched linear system
- **The main issue: data storage requirements**
  - Pre-computed Kalman gains need to be stored
  - The number of gains grows combinatorically with the max delay and number of outputs

$$\frac{2^{\sum_{i=1}^{m}(nd_i+1)}}{2}\left(\sum_{i=1}^{m}(nd_i+1)\right)$$

- **Suboptimal under packet loss**
  - If a sample arrives with delay greater than the length of the data buffer.
  - Occasional loss of old data does not affect the performance much, if later measurements are available
  - If no data loss occurs, the optimal filter would converge to the quasi-steady state one

Switched Suboptimal KF

Honeywell

- **Switched suboptimal KF**
  - Reducing storage requirements to an acceptable level for a loss of optimality
- **Assumptions**
  - No sample arrives later than d_{max}.
  - Number of missing samples with delay greater than d_{opt} and smaller than/equal to d_{max} is not larger than n_{miss}.
- **Basic ideas:**
  - If a sample $\mathrm{k} - \mathrm{d}_{\mathrm{opt}}$ has not arrived, it is replaced it by its estimate (i.e., no mean state correction)
  - When the particular sample arrives later, it is fused correctly
  - After all data missing earlier are received and fused, the optimality of the quasi-steady state filter is restored.
  - The gains are the same as in the optimal case, but only a limited number of those need to be stored.
  - If the assumed number of missing data is low, the reduction is significant.
- **References**
  - *Pachner at al, Prague 2010.*
  - Similar result in the time-varying Kalman filter setting and for $n_{miss} = 1$ was obtained in
  - *Larsen et al, 1998.*

Switched suboptimal KF continued

Honeywell

- **Fusing data**
  1. data with time-stamps greater than/equal to $k - d_{opt}$ as in the case of optimal switched KF
  2. Data with time stamps between $k - d_{max}$ and $k - d_{opt}$
     - fused only if they are on the missing data list.
     - After fusing, remove them from this list
  3. If sample with time stamp $k - d_{opt}$ has not arrived:
     - Include it into the missing data list
     - If length of the list exceeds the bound, remove the oldest member.

- **Suboptimality of the algorithm**
  - Missing data between $k - d_{max}$ and $k - d_{opt}$ are treated as received in the state covariance (or 'buffer state') update.
  - State mean is unaffected by missing data in this interval (or missing measurements are replaced by the estimates')
  - If the missing data list becomes empty, and no item has been removed without actually receiving the sample, ***optimality is re-gained***.

Switched-suboptimal KF continued

Honeywell

- **Implemented in the quasi-steady-state framework**
- **Reduction of buffer states**
- **Feasible buffer states**
  - Having at most n_{mis} missing data with time stamps between $k - d_{max}$ and $k - d_{opt}$
  - Eliminated states with most zero entries in the binary form: significant reduction of storage.
- **Example:**
  - $d_{max} = 4,\ d_{opt} = 1,\ n_{mis} = 1$
    - 28 gains to store; compared to 80 of the optimal case (see gray+white fields, zero entries).
  - $d_{max} = 4,\ d_{opt} = 0,\ n_{mis} = 1$ (see white fields, zero entries).
    - Remains 13 gains to store

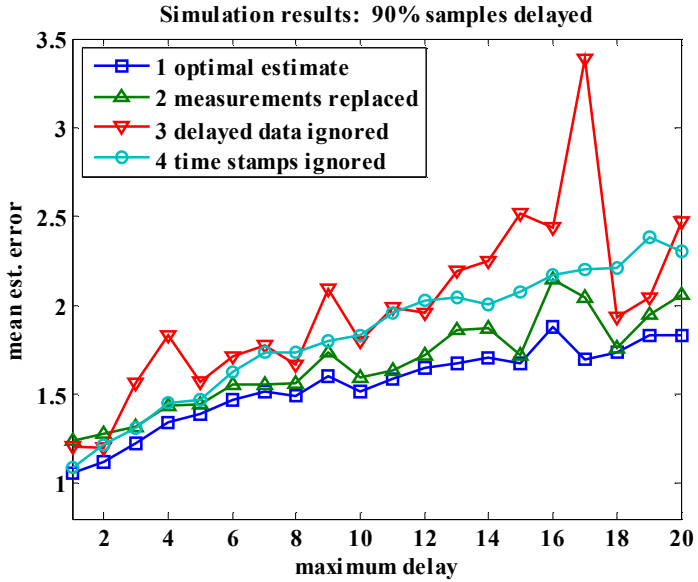
Delay of received sample

Buffer state	4	3	2	1	0
0	0	0	0	0	0
1	0	0	0	0	1
2	0	0	0	1	0
3	0	0	0	1	1
4	0	0	1	0	0
5	0	0	1	0	1
6	0	0	1	1	0
7	0	0	1	1	1
8	0	1	0	0	0
9	0	1	0	0	1
10	0	1	0	1	0
11	0	1	0	1	1
12	0	1	1	0	0
13	0	1	1	0	1
14	0	1	1	1	0
15	0	1	1	1	1
16	1	0	0	0	0
17	1	0	0	0	1
18	1	0	0	1	0
19	1	0	0	1	1
20	1	0	1	0	0
21	1	0	1	0	1
22	1	0	1	1	0
23	1	0	1	1	1
24	1	1	0	0	0
25	1	1	0	0	1
26	1	1	0	1	0
27	1	1	0	1	1
28	1	1	1	0	0
29	1	1	1	0	1
30	1	1	1	1	0
31	1	1	1	1	1

Switched-suboptimal KF - conclusion

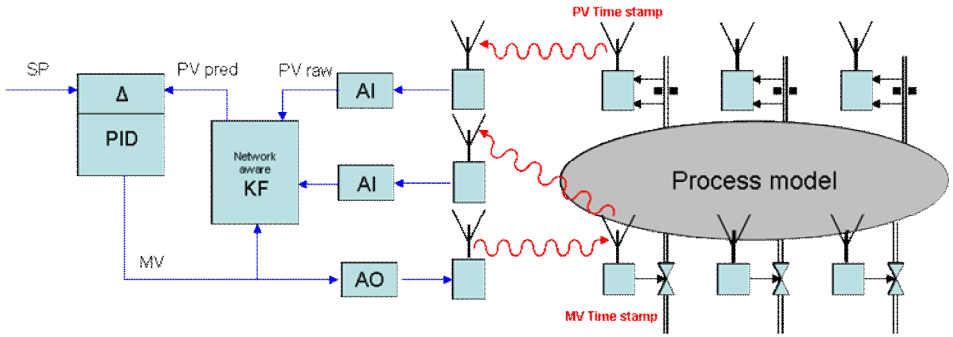
Honeywell

- **Example –random walk**



  - *Pachner et al, 2010*

- **Computationally cheap**
  - **Comparable to steady-state KF**
  - **Additional commplexity**
    - **Set of Kalman Gains**
    - **Switching logic**
    - **Bookkeeping of missing samples**
- **Suitable for the use in PID control with WSN in the loop**
- **Simulink models**
  - included in WIDE toolbox (http://ist-wide.dii.unisi.it/)



Delayed data case: References

Honeywell

- Nilsson, J. Bernardsson, B. and Wittenmark, B.: Stochastic Analysis and Control of Real-time Systems with Random Time Delays. *Automatica,* Vol34, pp57-64, 1998.
- Schenato, L.: Optimal Sensor Fusion for distributed sensors subject to random delays. *46th IEEE Conf Decision and Contr, New orleans,LA, USA, 2007*
- Schenato, L.: Optimal estimation in networked control systems subject to random delays and packet drop. *IEEE TAC,* Vol 53*, 2008, pp 1311*
- Pachner, D., Havlena, V.: An Approach to Out-Of-Sequence Measurements in Feedback Control Systems. Proceedings of 6*th IEEE International Workshop on Factory Communication Systems*, Dresden, Germany, 2008.
- Baramov, L., Pachner, D. and Havlena, V.: Kalman Filter for Systems with Communication Delay Proceedings of *1st IFAC Workshop. on Estimation and Control of Networked Systems* NecSys2009, Venice, Italy, September 2009.
- Pachner, D. Baramov, L. and Havlena, V.: Suboptimal State Estimation under Communication Delay. *Proceedings of 9th IFAC Workshop of Time Delay Systems*, Prague, Czech Republic, 2010.

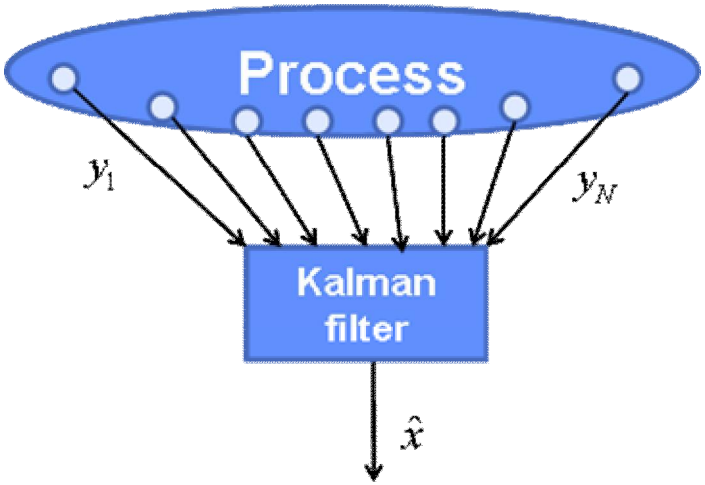
Distributed Estimation 3

CONSENSUS-BASED DISTRIBUTED KALMAN FILTER

Problem formulation

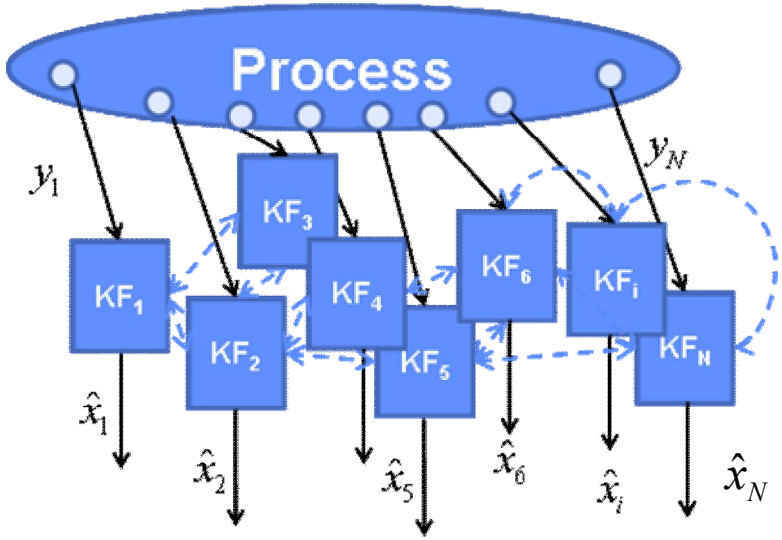
Honeywell

Centralized KF



- **Use all available process data**
  - Wireless or wire-line
- **Compute optimal state estimate**
  - **High-dimensional process data → high computational burden**

Distributed KF



- **Process observed by many local filters (agents)**
  - Estimating the full global state or its part
- **Inter-filter communication**
  - **needed to approach the globally optimal estimate**

Information filter

Honeywell

- **Centralized implementation**
  - **Transformed variables**
    - Information filter state vector

$$z(k\,|\,\cdot) = P(k\,|\,\cdot)^{-1}\hat{x}(k\,|\,\cdot)$$

    - Information matrix

$$Z(k\,|\,\cdot) = P(k\,|\,\cdot)^{-1}$$

    - Observation variables (independent)

$$\left.\begin{aligned} i_j(k) &= C_j^T R_j^{-1}(k) y_j(k) \\ S_j(k) &= C_j^T R_j^{-1}(k) C_j \end{aligned}\right\} j = 1,\ldots,N$$

  - **Data update step (filtering)**

$$Z(k\,|\,k) = Z(k\,|\,k-1) + \sum_{j=1}^{N} S_j(k)$$

$$z(k\,|\,k) = z(k\,|\,k-1) + \sum_{j=1}^{N} i_j(k)$$

  - **Time update step (prediction)**

$$P(k+1\,|\,k) = AZ(k\,|\,k)^{-1}A^T + Q$$

$$\hat{x}(k+1\,|\,k) = AZ(k\,|\,k)^{-1}z(k\,|\,k) + Bu(k)$$

- **Comparison to standard KF**
  - No inversion of the overall output covariance matrix
  - Need for state covariance inversion
  - Advantageous when state dim << N (redundant measurements)
  - Filtering step *additive* in transformed variables → suitable for distributed computing

Information filter

Honeywell

- **Decentralized implementation**
  - *Olfati-Saber 2005*
  - Assume network-wide consensus on the average of observation vectors

$$\bar{i}(k) = \frac{1}{N}\sum_{j=1}^{N} i_j(k)$$

$$\bar{S}(k) = \frac{1}{N}\sum_{j=1}^{N} S_j(k)$$

  - Local initial covariance shared

$$P_i(0|-1) = NP(0|-1)$$

- **Local filter update**

$$\begin{aligned} Z_j(k|k) &= P_j(k|k-1)^{-1} + \bar{S}(k) \\ z_j(k|k) &= P_j(k|k-1)^{-1} x_j(k|k-1) + \bar{i}(k) \\ P_j(k|k) &= AZ_j(k|k)^{-1} A^T + N \cdot Q(k) \\ x_j(k+1|k) &= AZ_j(k|k)^{-1} z_j(k|k) + Bu(k) \end{aligned}$$

  - Solution equivalent to the central KF

$$\begin{aligned} \hat{x}_j(k|k-1) &= \hat{x}(k|k-1) \\ \hat{P}_j(k|k-1) &= N \cdot \hat{P}(k|k-1) \\ j &= 1,\ldots,N \end{aligned}$$

- **How to obtain consensus on average observation values?**
  - Iterative algorithm within a KF step
  - Neighbor-to-neighbor data exchange
  - Convergence under certain connectivity assumption

Iterative consensus algorithm

Honeywell

- **Communication graph**
  - Vertices ~ local KFs
  - Edges ~ communication links
- **Assumption**
  - Graph is connected

    (there exists a path between any two vertices)
- **Iterative algorithm**
  - N_j: Set of vertices connected to vertex j by an edge
  - d_j: number of elements in N_j
  - Node update: weighted average of neighbors (and itself)

$$i_j(k,0) = i_j(k)$$

$$i_j(k,s+1) = i_j(k,s) + \frac{1}{N}\sum_{p\in N_j}\left(i_p(k,s) - i_j(k,s)\right)$$

- **Algorithm analysis**
  - Generalized setting

$$I_k(s) = \begin{bmatrix} i_1^T(k,s) \\ \vdots \\ i_N^T(k,s) \end{bmatrix}$$

  - Consensus algorithm from the global viewpoint

$$I_k(s+1) = W \cdot I_k(s)$$

  - Weight matrix W
    - w_{ij} is nonzero if there is an edge between vertices i and j, $i \neq j$
  - Alternative weighting in consensus algorithms are available

    (Xiao et al, 2005, Carli et al, 2005)

Iterative consensus algorithm

Honeywell

- Requirement for consensus

$$\lim_{s\to\infty} W^s = \frac{1}{N}\begin{bmatrix}1\\ \vdots \\ 1\end{bmatrix}\begin{bmatrix}1 & \cdots & 1\end{bmatrix}$$

- Equivalent conditions, *Xiao et al, 2005*

$$W\begin{bmatrix}1\\ \vdots \\ 1\end{bmatrix} = W^T\begin{bmatrix}1\\ \vdots \\ 1\end{bmatrix} = \begin{bmatrix}1\\ \vdots \\ 1\end{bmatrix},$$

$$r = \lambda_{\max}\left(W - \frac{1}{N}\begin{bmatrix}1\\ \vdots \\ 1\end{bmatrix}\begin{bmatrix}1 & \cdots & 1\end{bmatrix}\right) < 1$$

- W is 'doubly stochastic';
- r is the second largest eigen value of W and the convergence rate of the algorithm

- **Time-varying weights – changing the network topology**
  - *Xiao et al, 2005*
  - finite number of graphs/finite number of topologies
  - weaker connectivity conditions: graph does not need to be connected every time, but vertices active infinite number of times make an connected graph.
- **Inherent robustness of distributed estimation**
  - System survives a loss of communication link as long as the connectivity condition is satisfied.

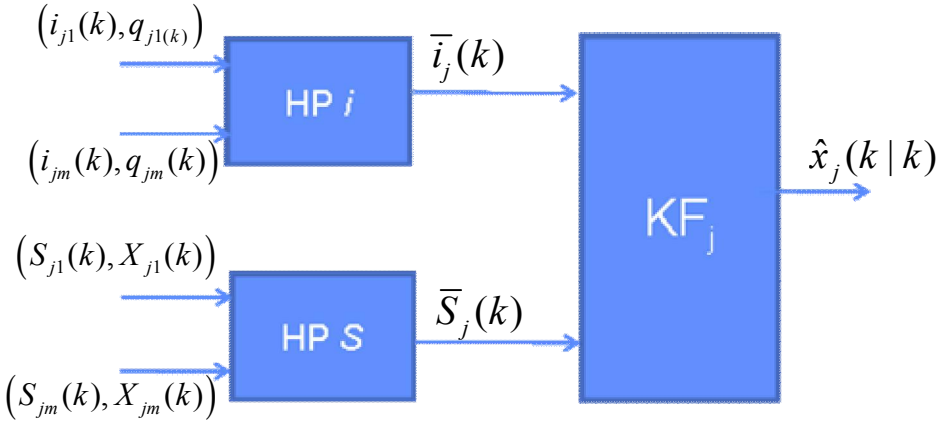
Noniterative Consensus Filter

Honeywell

- Local Kalman filters use local consensus filters to average data
  - *Spanos et al, (2005), Olfati-Saber (2008)*
  - Other options
- Consensus filters operate with the same frequency as Kalman filters
  - Consensus iterations no longer used
  - More extensive information exchange
    - Observation $i_j(k)$, $S_j(k)$
    - Consensus filter state $q_j(k)$, $Q_j(k)$
  - Filter convergence

$$\bar{i}_j \to \bar{i}, \quad \bar{S}_j \to \bar{S}$$

    - controlled by parameter β



$$q_j(k+1) = q_j(k) + \beta \sum_{j_i \in N_j} \left[\left(q_{j_i}(k) - q_j(k) \right) + \left(i_{j_i}(k) - i_j(k) \right) \right]$$
$$\bar{i}_j(k) = q_j(k+1) + i_j(k)$$

$$Q_j(k+1) = Q_j(k) + \beta \sum_{j_i \in N_j} \left[\left(Q_{j_i}(k) - Q_j(k) \right) + \left(S_{j_i}(k) - S_j(k) \right) \right]$$
$$\bar{S}_j(k) = Q_j(k+1) + S_j(k)$$

Other Noniterative Consensus Filters

Honeywell

- **Consensus on estimates instead of sensor data**
  - *Olfati-Saber (2007)*
  - Data aggregation (over neighbors)

$$i_j(k) = \sum_{j_i \in N_j} i_{j_i}(k), \quad S_j(k) = \sum_{j_i \in N_j} S_{j_i}(k),$$

  - Filtering

$$\begin{aligned} Z_j(k \mid k) &= P_j(k \mid k-1)^{-1} + S_j(k) \\ \varphi_j(k) &= \hat{x}_j(k \mid k-1) + Z_j(k \mid k)^{-1} \times \\ &\quad \left(i_j(k) - S_j(k)\hat{x}_j(k \mid k-1) \right) \end{aligned}$$

  - Consensus step

$$\hat{x}_j(k \mid k) = \varphi_j(k) + \alpha \sum_{i \in N_j} \left(\varphi_i(k) - \varphi_j(k) \right)$$

  - Ad hoc algorithm (no proof)

    … but better performance than the previous type of consensus filter

- **Further modifications**
  - Consensus on estimates algorithm
  - Integrated filtering and consensus steps
  - Rigorous derivation for continuous-time systems and Lyapunov argument

$$\begin{aligned} \hat{x}_j(k \mid k) &= \hat{x}_j(k \mid k-1) + \\ &\quad Z_j(k \mid k)^{-1} \left(i_j(k) - S_j(k)\hat{x}_j(k \mid k-1) \right) \\ &\quad + \alpha Z_j(k \mid k)^{-1} \sum_{i \in N_j} \left(\hat{x}_i(k \mid k-1) - \hat{x}_j(k \mid k-1) \right) \end{aligned}$$

Diffusion algorithm

Honeywell

- *Cattivell and Sayed, 2010*

- Similar to distributed KF with consensus on estimates
  - Different weights
  - Rigorous performance analysis and convergence proof
  - extension to distributed smoothers
- Three steps per sampling interval
  1. Kalman data update step

     (fusing measurement from all neighbors - either standard implementation or information filter)
  2. Diffusion step

     (filtered state as a convex combination of filtered states of all neighboring filters)
  3. Kalman time update step

- Algorithm
  - Diffusion weighting matrix W
    - tunable

$$w_{ij} \begin{cases} > 0 \text{ for } j \in N_i \cup i \\ 0 \quad \text{otherwise} \end{cases} \qquad \sum_j w_{ij} = 1$$

  - Data update step

$$P_i(k \mid k) = \left(P_i(k \mid k-1)^{-1} + \sum_{j \in N_i} S_j(k) \right)^{-1}$$

$$\psi_i(k \mid k) = x_i(k \mid k-1) + P(k \mid k) \sum_{j \in N_i} \left(z_j(k) - S_j(k) x(k \mid k-1) \right)$$

  - Diffusion step

$$\hat{x}_i(k \mid k) = \sum_{j \in N_i} w_{ij} \psi_j(k \mid k)$$

  - Time update step
    - … standard

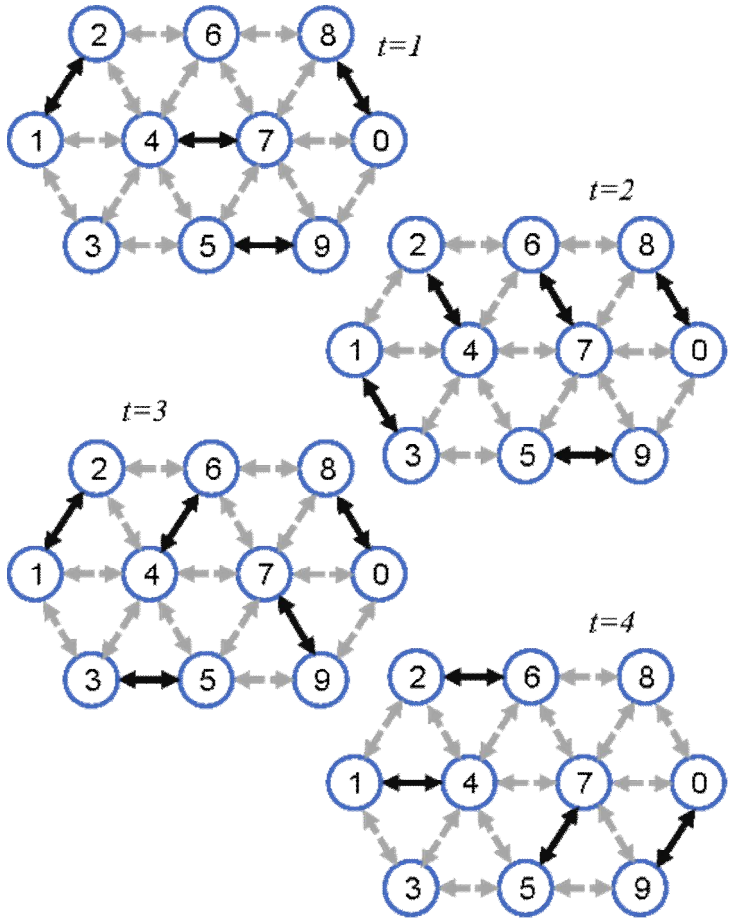
Gossip algorithm for distributed KF

Honeywell

- **Limited information exchange**
  - *Kar and Moura (2011)*
  - Communication with at most one other agent per KF period
  - Kalman filter swaps its estimate with one randomly chosen neighbor
- **Week detectability concept**
  - The system may not be detectable by individual agents
  - Week detectability = there exists a path of nodes $n_1, \ldots n_l$ so that the following matrix is invertible

$$\sum_{i=1}^{l} \left(A^{i-1} \right)^T C_{n_i}^T C_{n_i} A^{i-1}$$

- **Main results**
  - Error process is stochastically bounded
  - Estimation error of randomly selected agent converges in distribution to a unique invariant measure



Distributed KF for large-scale systems

Honeywell

Previous methods

- **All agents estimate full state**
- **Infeasible for large-scale systems**
  - Data storage
  - Communication
  - Computation
    - Inverse of a large information matrix
- **Applications seldom need the whole global state estimate stored at each agent**

Solution for large scale systems

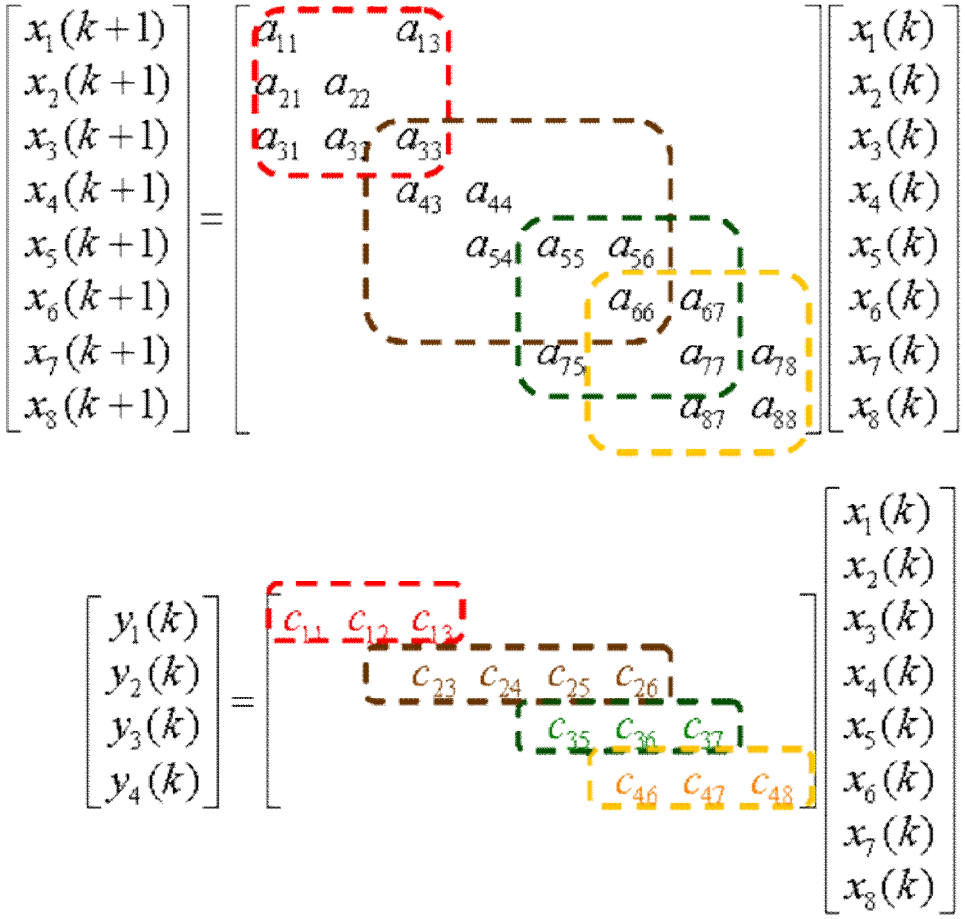
- **Local filters/agents estimate sub-spaces of the global state-space**
  - Local subspaces overlap
  - Overlapping subspaces of information vectors
  - Consensus on overlapped subspaces
- **Distributed information matrix inverse**

Distributed KF for large-scale systems

Honeywell

- *Khan and Moura, 2008*

- **System assumptions**
  - Large-scale
  - Sparse
  - Localized (minimum band-width matrix)
  - Arising in spatial discretization of distributed parameter systems, image processing,…



Distributed KF for large scale systems

Honeywell

- **Information matrix approximated by an L-banded matrix**
  - Small width may cause filter instability
  - Formula for choosing L guaranteeing stability not available
- **The inverse of an L-banded matrix is full**
- **Distributed iterative algorithm for inverting L-banded matrix**
- **Distributed algorithm for inverting Z**
  - Computes only diagonal blocks of the covariance matrix that are relevant for the prediction step
  - Limited communication with neighbors
  - Iterative scheme, modified Jacobi over relaxation method
- **Distributed algorithm for the inversion of covariance matrix**
  - Uses only block-diagonal part
  - Works on assumption that the target inverse matrix is L-banded

$$P = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} & P_{15} & P_{16} & P_{17} & P_{18} \\ * & P_{22} & P_{23} & P_{24} & P_{25} & P_{26} & P_{27} & P_{28} \\ * & * & P_{33} & P_{34} & P_{35} & P_{36} & P_{37} & P_{38} \\ * & * & * & P_{44} & P_{45} & P_{46} & P_{47} & P_{48} \\ * & * & * & * & P_{55} & P_{56} & P_{57} & P_{58} \\ * & * & * & * & * & P_{66} & P_{67} & P_{68} \\ * & * & * & * & * & * & P_{77} & P_{78} \\ * & * & * & * & * & * & * & P_{88} \end{bmatrix} = \begin{bmatrix} Z_{11} & Z_{12} & & & & & & \\ Z_{12} & Z_{22} & Z_{23} & & & & & \\ & Z_{23} & Z_{33} & Z_{34} & & & & \\ & & Z_{34} & Z_{44} & Z_{45} & & & \\ & & & Z_{45} & Z_{55} & Z_{56} & & \\ & & & & Z_{56} & Z_{66} & Z_{67} & \\ & & & & & Z_{67} & Z_{77} & Z_{78} \\ & & & & & & Z_{78} & Z_{88} \end{bmatrix}^{-1} = Z^{-1}$$

Distributed filters for large scale systems

Honeywell

- **Decomposition**
  - One measurement per subsystem
  - Independent noise for different sensor
  - States of other subsystems handled as external input
  - No direct feed-through from interconnection inputs on local measurement

$$\begin{bmatrix} x_5^{(3)}(k+1) \\ x_6^{(3)}(k+1) \\ x_7^{(3)}(k+1) \end{bmatrix} = \begin{bmatrix} a_{55} & a_{56} & . \\ . & a_{66} & a_{67} \\ a_{75} & . & a_{77} \end{bmatrix} \begin{bmatrix} x_5^{(3)}(k) \\ x_6^{(3)}(k) \\ x_7^{(3)}(k) \end{bmatrix} + \begin{bmatrix} a_{54} \\ . \\ . \end{bmatrix} x_4^{(2)}(k) + \begin{bmatrix} . \\ . \\ a_{78} \end{bmatrix} x_8^{(4)}(k)$$

- **Local variables**
  - Covariance

$$P^{(3)} = \begin{bmatrix} P_{55}^{(3)} & P_{56}^{(3)} & P_{57}^{(3)} \\ * & P_{66}^{(3)} & P_{67}^{(3)} \\ * & * & P_{77}^{(3)} \end{bmatrix}$$

  - Information matrix

$$Z^{(3)} = \begin{bmatrix} Z_{55}^{(3)} & Z_{56}^{(3)} & \\ * & Z_{66}^{(3)} & Z_{67}^{(3)} \\ & * & Z_{77}^{(3)} \end{bmatrix} \neq P^{(3)-1}$$

Distributed filters for large scale systems

Honeywell

- **Local variables**
  - Local observation vector

$$i^{(3)}(k) = \begin{bmatrix} i_5^{(3)}(k) \\ i_6^{(3)}(k) \\ i_7^{(3)}(k) \end{bmatrix},$$

$$i_i^{(j)}(k) = c_{ji} r_j^{-1} y_j(k)$$

- **Observation fusion**

$$\bar{i}^{(3)}(k) = \begin{bmatrix} i_5^{(3)}(k) + i_5^{(2)}(k) \\ i_6^{(2)}(k) + i_6^{(3)}(k) + i_6^{(8)}(k) \\ i_7^{(3)}(k) + i_7^{(8)}(k) \end{bmatrix},$$

$$\bar{i}_i(k) = \sum_{j \in N_i} c_{ji} r_j^{-1} y_j(k), \quad N_i = \{ j : c_{ji} \neq 0 \}$$

$$z^{(j)}(k \mid k) = z^{(j)}(k \mid k-1) + \bar{i}^{(j)}(k)$$

  - Consensus established immediately: observation vector for a particular state is the same for all subsystems
  - All agents that share a state talk one to another – usually not a problem when global model is localized
  - Number of communication links can be reduced ; iterative consensus algorithm need to be applied and the graph-connectivity assumption must hold for each state
  - Similar fusion is done for information matrix.

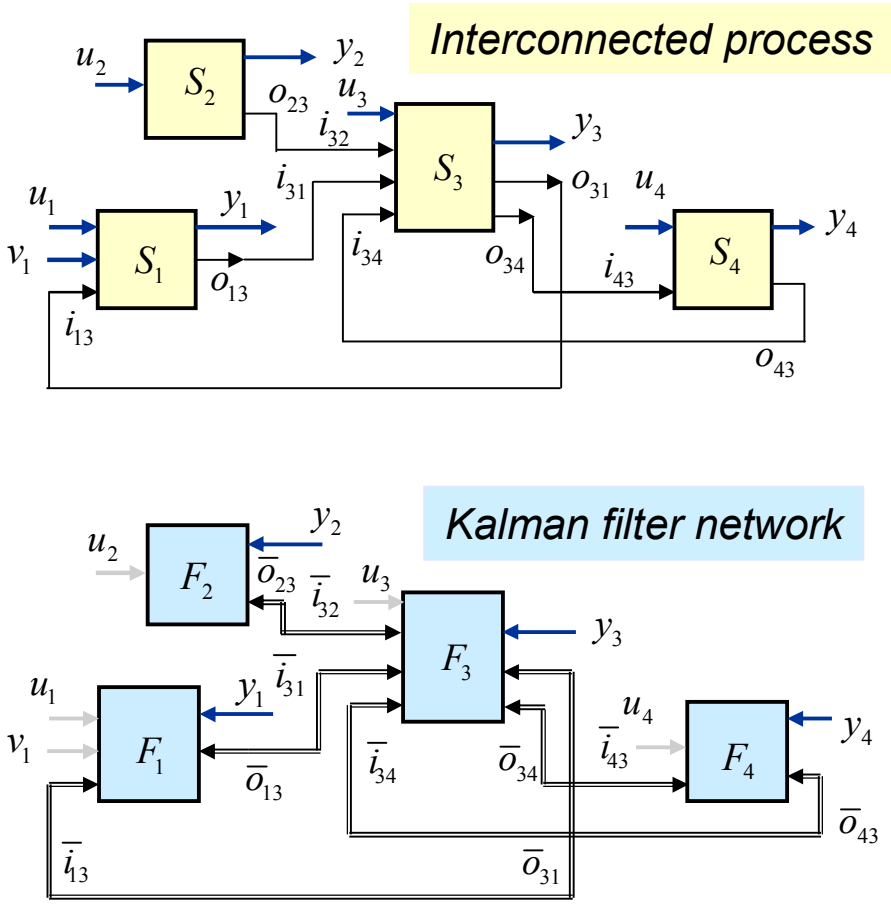
- **Prediction**
  - Takes into account covariance in coupling states obtained from neighboring subsystems

Distributed KF for interconnected systems

Honeywell

- *Baramov, Pachner, Havlena (2011)*

- **Kalman filter network**
  - Topology corresponds to the process interconnection map
  - Local filters have an access only to local measurements.
- **Local filters estimate local state & interconnection inputs**
  - Coupling inputs → the only overlaps in estimation spaces
- **Structural limitation**
  - No feed-through from interconnection inputs to coupling outputs
  - Fairly general, but mostly beneficial for sparse systems

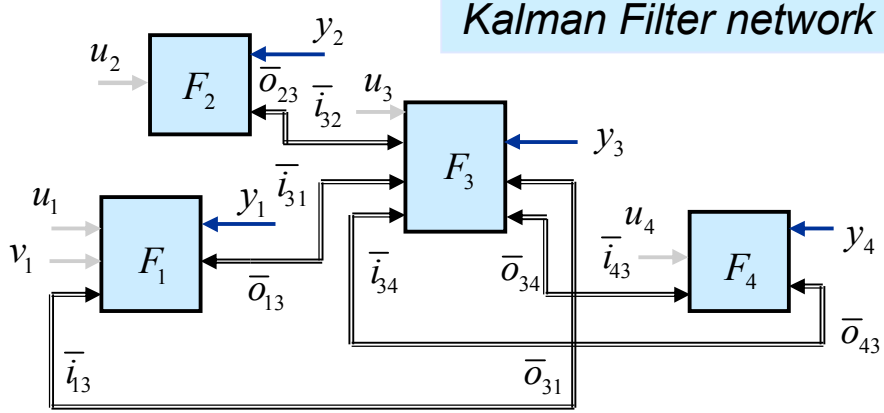


Distributed KF for interconnected systems

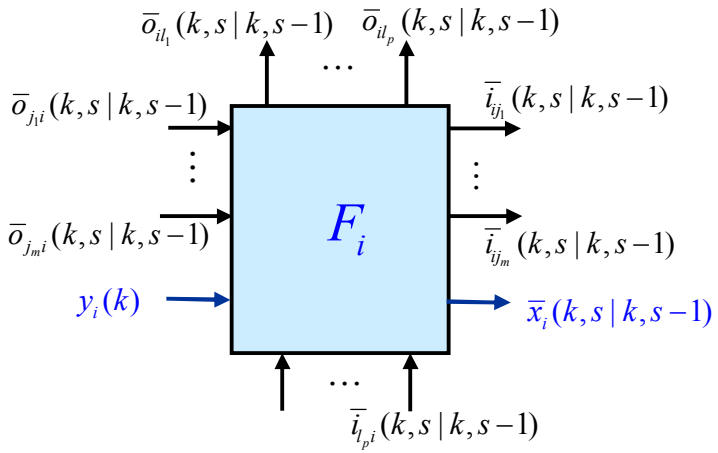
Honeywell

- **Local filters fuse coupling variables as measurement data**
  - The process measurements of these variables are not available – they are replaced by estimates from the neighboring units
- **Updating local state → updated estimate of coupling variables**
  - Kalman filtering step proceeds iteratively, when information among neighbors is exchanged
  - Argument s is introduced as the iteration count (or a second discrete time variable)
  - Iterations proceeds until a network-wide equilibrium is reached

Kalman Filter network



Local Kalman Filter



Distributed KF for interconnected systems

Honeywell

- **Interconnection input**
  - treated as a stochastic variable, mean & covariance supplied from the neighboring KF
  - This variable is correlated with the local state

$$i_{ij}(k) = \bar{o}_{ji}(k \mid k-1) + \vartheta_{ij}(k)$$

- **Local state**
  - Augmented by components related to the inputs

$$X_i = \begin{bmatrix} x_i \\ \vartheta_{ij_1} \\ \vdots \\ \vartheta_{ij_n} \end{bmatrix} \left[x_i^T \mid \vartheta_{ij_1}^T \cdots \vartheta_{ij_n}^T \right]$$

- **Local observations**
  - Locally accessed process measurements
  - Locally estimated interconnection variables

$$Y_i(k,s) = \begin{bmatrix} y_i(k) \\ \hline \bar{o}_{l_1 i}(k,s \mid k,s-1) \\ \vdots \\ \bar{o}_{l_k i}(k,s \mid k,s-1) \\ \hline \bar{\vartheta}_{ij_1}(k,s \mid k,s-1) + \bar{o}_{j_1 i}(k \mid k-1) \\ \vdots \\ \bar{\vartheta}_{ij_m}(k,s \mid k,s-1) + \bar{o}_{j_m i}(k \mid k-1) \end{bmatrix}.$$

Distributed KF for interconnected systems

Honeywell

- **Covariance matrices**
  - State-to-state

$$M_{ij}(k,s \mid k,s-1) = \mathrm{E}\left(\tilde{X}_i(k,s \mid k,s-1)\tilde{Y}_j(k,s \mid k,s-1)^T \mid k,s-1\right)$$

  - State-to-output

$$H_{ij}(k,s \mid k,s-1) = \mathrm{E}\left(\tilde{X}_i(k,s \mid k,s-1)\tilde{Y}_j(k,s \mid k,s-1)^T \mid k,s-1\right)$$

  - Output-to-output

$$W_{ij}(k,s \mid k,s-1) = \mathrm{E}\left(\tilde{Y}_i(k,s \mid k,s-1)\tilde{Y}_j(k,s \mid k,s-1)^T \mid k,s-1\right)$$

- **Initialization for s=0**
  - Straightforward from predictions $X_i(k|k\text{-}1)$ and $M_{ij}(k|k\text{-}1)$.

- **Iterative data step for s=1,2,…**
  - Kalman gain: solution to (possibly singular) equation

$$K_i(k,s)W_{ii}(k,s \mid k,s-1) = H_{ii}(k,s \mid k,s-1).$$

  - State mean update

$$\bar{X}_i(k,s+1 \mid k,s) = \bar{X}_i(k,s \mid k,s-1) + K_i(k,s)\tilde{Y}_i(k,s \mid k,s-1).$$

  - Covariance updates: Block-diagonal term

$$M_{ii}(k,s+1 \mid k,s) = M_{ii}(k,s \mid k,s-1) - K_i(k,s)W_{ii}(k,s \mid k,s-1)K_i(k,s)^T.$$

Distributed KF for interconnected systems

Honeywell

- **Iterative data step**
  - Covariance updates:

    Block-off-diagonal term

$$M_{ij}(k,s+1\,|\,k,s) = M_{ij}(k,s\,|\,k,s-1) - K_i(k,s)H_{ji}(k,s\,|\,k,s-1)^T - H_{ij}(k,s\,|\,k,s-1)K_j(k,s)^T + K_i(k,s)W_{ij}(k,s\,|\,k,s-1)K_j(k,s\,|\,k,s-1)^T.$$

  - These blocks are needed for predicting covariances of coupling variables
  - Needs Kalman gains and covariance matrices H_{ij} and W_{ij}: Obtained from j[th] KF through neighboring subsystem via multi-hop communication.
  - Updating state-to-output and output-to-output covariance matrices
    - These matrices are not readily derived from state covariances, for s>0
    - These matrices are kept as algorithm states and updated in parallel to the state covariance
  - Data update proceeds until network-wide equilibrium is detected
  - Convergence of this process can be proven
- **Prediction step**
  - State mean and covariance – obtained in a straightforward way, using a single-hop communication with the neighbors
  - Needs to know models of other subsystems

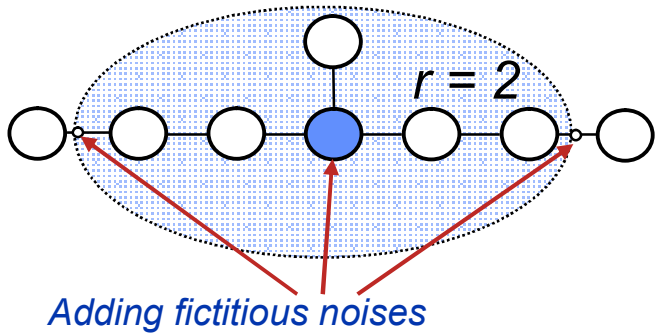
Distributed KF for interconnected systems

Honeywell

- **Algorithm summary**
  - Local agents store X_i, M_{i*}, H_{i*} and W_{i*}
  - Update these variables in the data step, an iterative, and convergent process using multi-hop communication to obtain data related to other subsystems
  - Prediction sep – one-time process using single-hop communication
  - Under some conditions, equivalent to central KF, otherwise globally suboptimal
  - Needs models of other subsystems, extensive communication and bookkeeping

Reduced communication load

- **Inter-filter communication**
  - Prescribed network radius (variables in a large network distance made independent
    - Fictitious measurement /process noise de-correlate some variables
    - Results in global covariance matrix with some zero off-diagonal blocks and increased diagonal ones
  - Replacing process information beyond the network radius by local uncertainties
    - Bayesian arguments



Distributed Kalman filter: References

Honeywell

- Xiao, L., Boyd, S. and Lall, S.: A scheme for robust distributed sensor fusion based on average consensus. *4th International Symposium on Information Processing in Sensor Networks*, 2005
- Carli, R. Chiuso, A. Schenato, L. and Zampieri, S.: Distributed Kalman filtering using consensus strategies*, IEEE CDC*, New Orleans, 2007
- Olfati-Saber, R.: Distributed Kalman Filtering for Sensor networks, *Joint ECC and IEEE CDC*, Seville, 2005
- Olfati-Saber, R.: Distributed Kalman Filter with Embedded Consensus Filters, *IEEE CDC*, New Orleans, 2007
- Khan, U.A,. Moura, J.M.F: Distributing the Kalman Filter for Large-Scale Systems, *IEEE Trans on Signal Processing*, Vol 56, 2008, 4919-4935.
- Kar, S.,. Moura, J.M.F: Gossip and Distributed Kalman Filtering: Weak Consensus Under Weak Detectability , *IEEE Trans on Signal Processing*, Vol 59, 2011, 1766-1784.
- Cattivelli, F.S. and Sayed, A.H.: Diffusion Strategies for Distributed Kalman Filtering and Smoothing. IEEE Trans on Aut Contr, Vol 55, 2010, pp2069-2084
- Baramov, L., Pachner, D., and Havlena, V.: Approximate Kalman filter for interconnected systems. *Paper under preparation,* 2011