SCAAT : Incremental Tracking with Incomplete Information

by

Gregory Welch and Gary Bishop

Presented by

Manmohan Krishna

CSE 252C, UCSD, Fall 2003

Images/Videos – Courtesy Prof. Greg Welch, Tracking Group at UNC Chapel Hill

The Central Idea

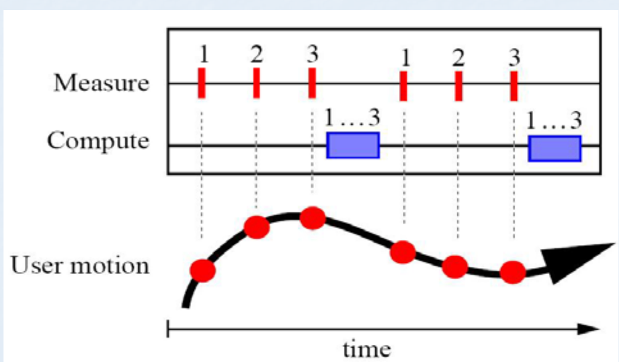
- Even incomplete observations provide *some* information
- Globally observable process can be estimated by sequentially incorporating locally unobservable measurements
- SCAAT : Single-Constraint-At-A-Time approach in an EKF framework.

Why SCAAT ?

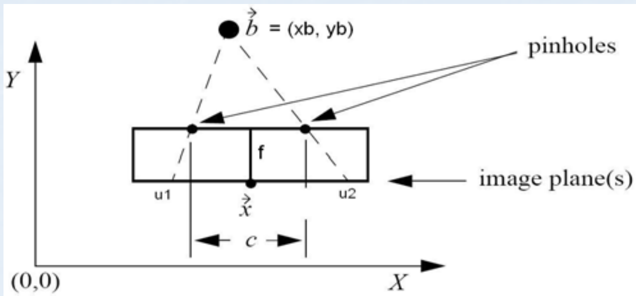
- Improved Accuracy
  - Simultaneity Assumption avoided
  - Concurrent device autocalibration
- Improved estimation rates and latencies

Simultaneity Assumption

- Measurements collected sequentially ….
- …. but treated as simultaneous for computation.



Simultaneity (Contd ….)



$$u1 = (\frac{f}{2})(\frac{2x - 2x_b - c}{y_b - y - f})$$

$$u2 = (\frac{f}{2})(\frac{2x - 2x_b + c}{y_b - y - f})$$

$$u2' = (\frac{f}{2})(\frac{2(x + \dot{x}\tau_m) - 2x_b + c}{y_b - (y + \dot{y}\tau_m) - f})$$

Simultaneity (some typical numbers)

$$f = 0.035\,m\ ,\ c = 0.2\,m\ ,\ \vec{b} = [1,2]\,m$$

$$\dot{x} = \dot{y} = 0.5\,m/s\quad ,\quad \tau_m = 10\,ms$$

Let, at time of first observation

$$\vec{x}(t) = [1,1]\ m$$

Then, at time of second observation

$$\vec{x}(t + \tau_m) = [1.005, 1.005]\ m$$

Consequently,

$$u_1 = -3.627e{-3}\ m\ ,\ u_2 = 3.828e{-3}\ m$$

So, our position estimate would be

$$\vec{x}(t + \tau_m) = [1.003, 1.026]\ m$$

Estimation Rates and Latencies

- *Nyquist Criterion*

  Sampling Frequency > (2 X Target Motion BW)

- *Virtual Environment Systems* – Visual feedback latency

SCAAT Helps ….

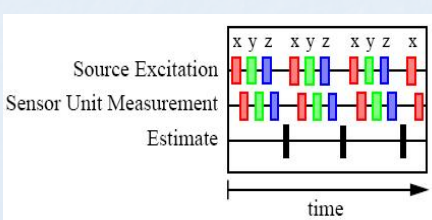
$\tau_m \rightarrow$ *Measurement time per observation*

$N \rightarrow$ *Number of Observations*

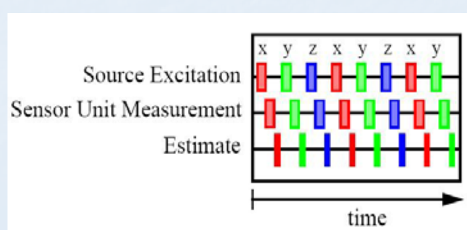
$\tau_c(N) \rightarrow$ *Computation Time*

$$\text{Estimation Rate}, \ \rho_e = \frac{1}{N\tau_m + \tau_c(N)}$$

Sequential Tracker

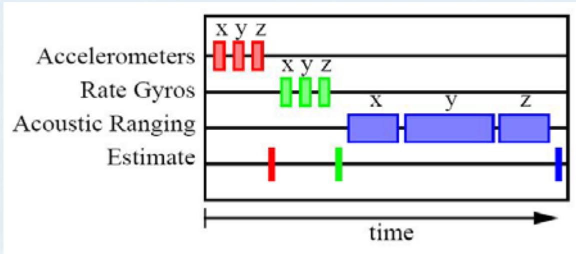


SCAAT tracker

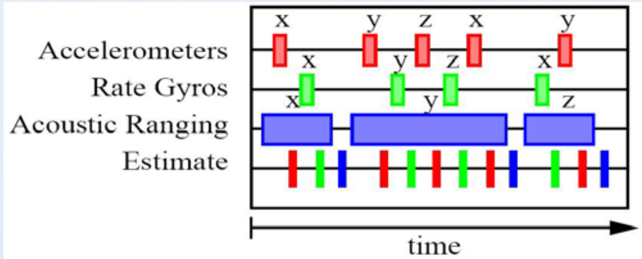


Hybrid Systems and Multi-Sensor Data Fusion

Sequential → Systems

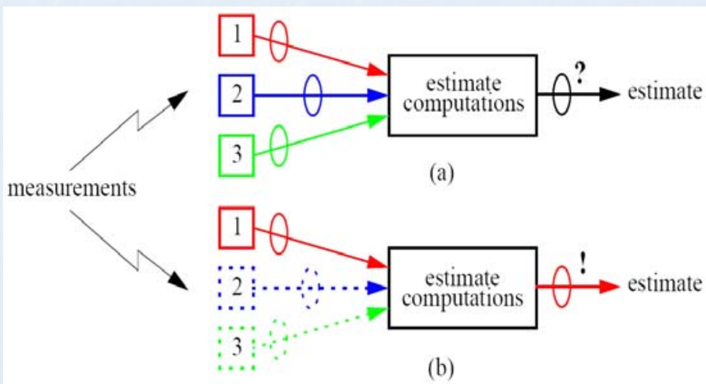


SCAAT → System



AutoCalibration

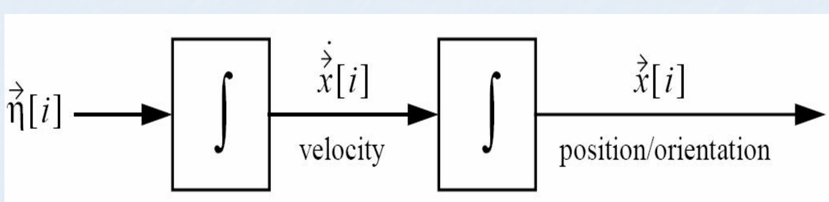
- New tracking estimate for each measurement
- Individual device imperfections segregated



SCAAT DEFINITIONS

Process Model

- Position-Velocity (PV) Model
- Accelerations $\vec{\eta}[i]$ modeled as zero-mean white noise sources,



Model State and Transitions

- Inter-sample time → $\delta(t)$
- Inter-state transition

$$\vec{x}(t) = A(\delta t)\,\vec{x}(t-\delta t) + \vec{\omega}(t)$$

- Complete target state
  - n-element internal state vector (n = 12)

$$\vec{x}(t) = [x \;\; y \;\; z \;\; \dot{x} \;\; \dot{y} \;\; \dot{z} \;\; \vec{\Delta}\phi \;\; \vec{\Delta}\theta \;\; \vec{\Delta}\psi \;\; \dot{\phi} \;\; \dot{\theta} \;\; \dot{\psi}\,]^T$$

  - 4-element external quaternion

$$\vec{\alpha} = (\alpha_w,\, (\alpha_x, \alpha_y, \alpha_z))$$

State Transition Matrix $A(\delta t)$

- Implements

$$x(t) = x(t-\delta t) + \dot{x}(t-\delta t)\,\delta t$$

$$\dot{x}(t) = \dot{x}(t-\delta t)$$

$$A(\delta t) = \begin{bmatrix} 1 & 0 & 0 & \delta t & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & \delta t & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \delta t & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \delta t & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \delta t & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \delta t \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Process Noise

- n dimensional process noise vector → $\vec{\omega}(t)$
- Normal distribution, zero-mean, white sequence
- Process noise covariance matrix

$$E\{\vec{\omega}(t)\ \vec{\omega}^T(t+\varepsilon)\} = Q, \quad \varepsilon = 0$$
$$= 0, \quad \varepsilon \neq 0$$

Measurement Model

- Predict ideal, noise-free response of each sensor-source pair
- Distinguishes SCAAT Kalman filter from well-constrained Kalman filter

SCAAT Measurements

- Pure Sense → Single scalar measurements
- Practically → One multi-dimensional constraint per estimate

For sensor σ, m_σ–dimensional constraint.

Heuristic for Choosing SCAAT Measurements

- During each KF measurement update, observe single source-sensor pair
- Extract one geometric constraint per observation

Predicting Measurement from State

- For some sensor type σ,

$\hat{z}_\sigma(t)$: Measurement Estimate Vector

$\vec{h}_\sigma()$: Measurement Function

$$\hat{z}_\sigma(t) = \vec{h}_\sigma(\vec{x}(t), \vec{\alpha}(t), \vec{b}(t), \vec{c}(t)) + \vec{\upsilon}_\sigma(t)$$

$\vec{b}(t)$: Source parameter vector (beacon)

$\vec{c}(t)$: Sensor parameter vector (camera)

Measurement Noise

- $\vec{\upsilon}_\sigma(t)$: m_σ – dimensional measurement noise vector
- $R_\sigma(t)$: $m_\sigma \times m_\sigma$ measurement noise covariance matrix

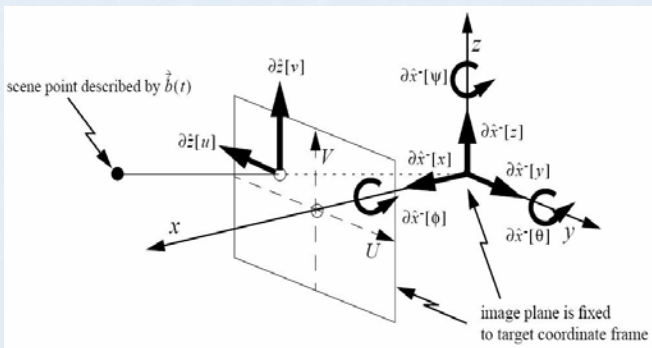
$$E\{\vec{v}_\sigma(t)\ \vec{v}_\sigma^{\,T}(t+\varepsilon)\} = R_\sigma(t), \quad \varepsilon = 0$$

$$= 0, \quad \varepsilon \neq 0$$

Measurement Jacobian

- Signifies sensitivity of measurement

$$H_\sigma(\vec{x}(t), \vec{\alpha}(t), \vec{b}(t), \vec{c}(t))\,[i, j] \;=\; \frac{\partial}{\partial \vec{x}[j]}\, \vec{h}_\sigma(\vec{x}(t), \vec{\alpha}(t), \vec{b}(t), \vec{c}(t))[i]$$



State Error Covariance

- Reflects filter's estimate of its own uncertainty

- Error in filter's state estimate,

$$\hat{\varepsilon}(t) = \vec{x}(t) - \hat{x}(t)$$

- State Error Covariance,

$$P(t) = E\{\hat{\varepsilon}(t)\hat{\varepsilon}^T(t)\}$$

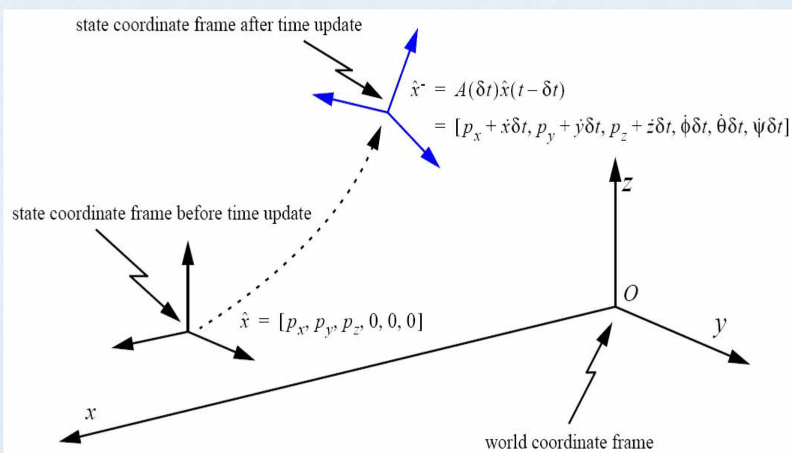
SCAAT ALGORITHM

Time Update

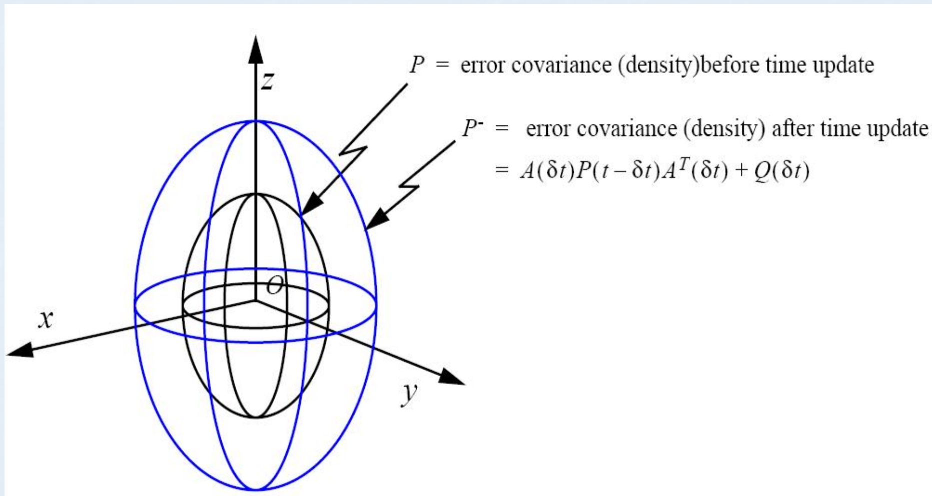
- *a priori* state and error covariance

$$\hat{x}^- = A(\delta t)\,\hat{x}(t-\delta t)$$

$$P^- = A(\delta t)\,P(t-\delta t)A^T(\delta t) + Q(\delta t)$$



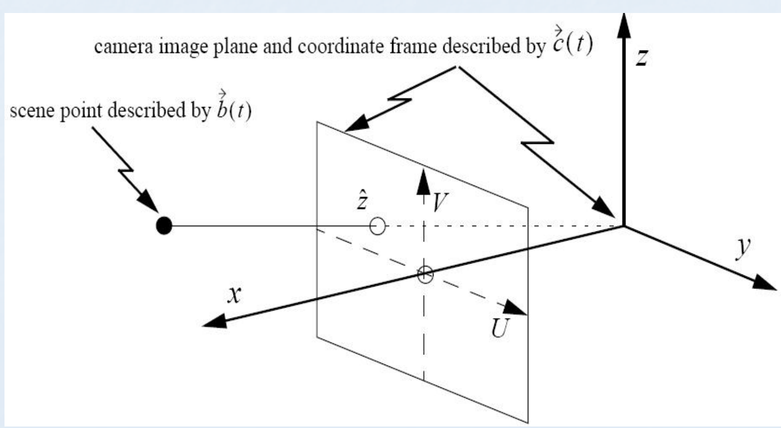
Time Update (Contd….)



Measurement Prediction

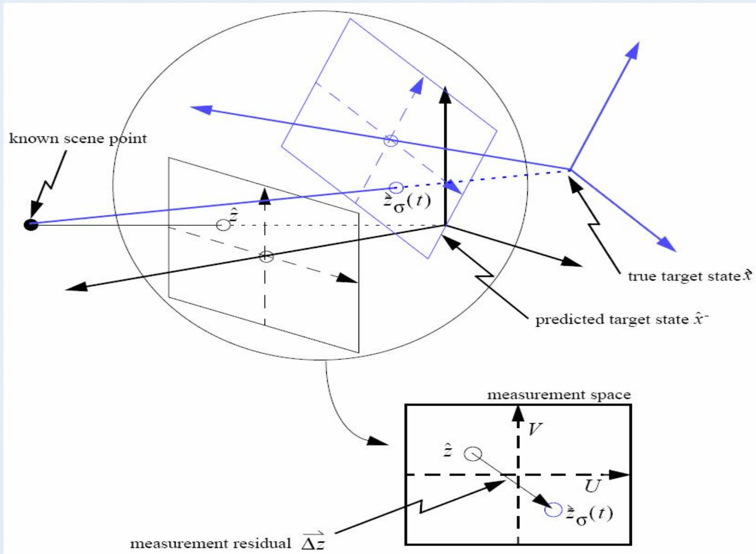
$$\hat{z} = \vec{h}_\sigma(\hat{x}^-(t),\, \hat{\alpha}^-(t),\, \vec{b}(t),\, \vec{c}(t))$$

$$H = H_\sigma(\hat{x}^-(t),\, \hat{\alpha}^-(t),\, \vec{b}(t),\, \vec{c}(t))$$



Measurement Residual

$$\Delta\vec{z} = \vec{z}_{\sigma}(t) - \hat{z}$$



Kalman Gain

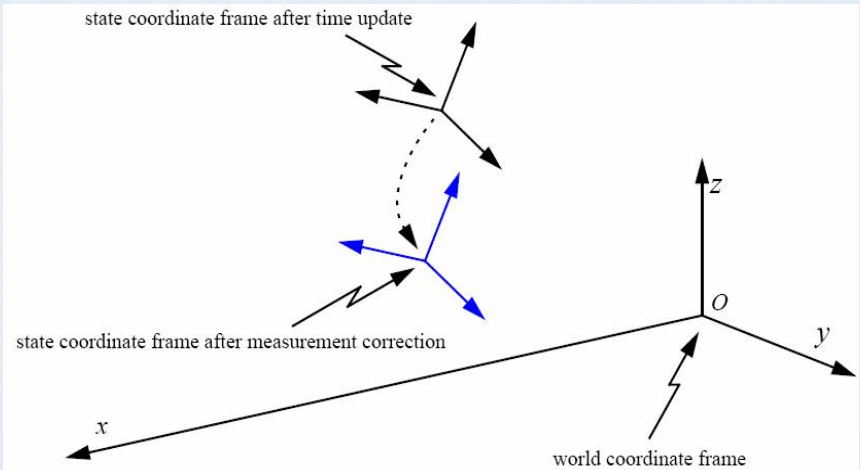
- Weights the residual in correction cycle

$$K = P^{-}H^{T}(HP^{-}H^{T} + R_{\sigma}(t))^{-1}$$

Update the State

- Compute *a posteriori* state estimate

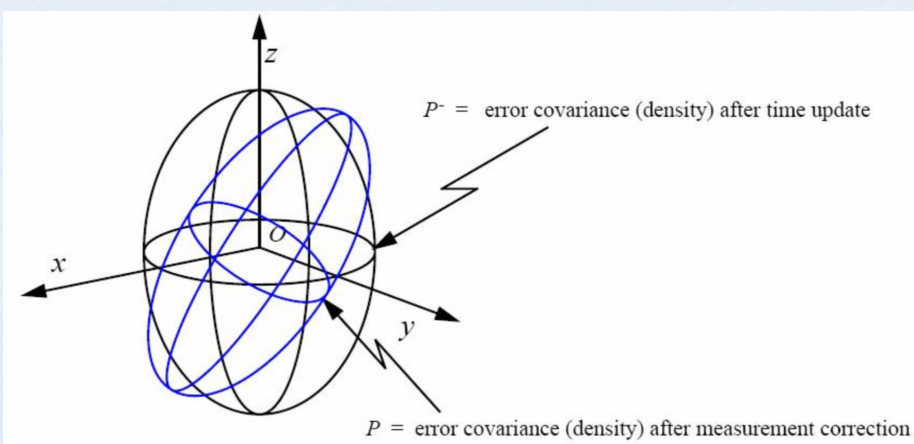
$$\vec{x}(t) = \hat{x}^- + K\,\Delta\vec{z}$$



Update Error Covariance

- Compute *a posteriori* error covariance.

$$P(t) = (I - KH)P^-$$



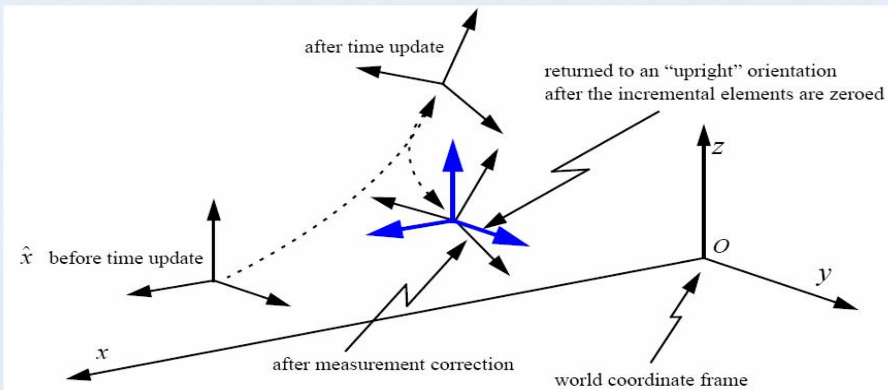
Update External Quaternion

$$\Delta\hat{\alpha} = quaternion(\hat{x}[\Delta\phi], \hat{x}[\Delta\theta], \hat{x}[\Delta\psi])$$

$$\hat{\alpha} = \hat{\alpha} \otimes \Delta\hat{\alpha}$$

Zero the incremental orientation state elements.

$$\hat{x}[\Delta\phi] = \hat{x}[\Delta\theta] = \hat{x}[\Delta\psi] = 0$$



Algorithm Summary

- Given : Initial state estimate, $\hat{x}(0)$
  External Orientation estimate, $\hat{\alpha}(0)$
  Error Covariance estimate, $P(0)$

- For each measurement $z_\sigma(t)$ from sensor σ (and corresponding source) at time t :

  1. Compute elapsed time δt
  2. Predict state and error covariance

$$\hat{x}^- = A(\delta t)\,\hat{x}(t-\delta t)$$

$$P^- = A(\delta t)\,P(t-\delta t)A^T(\delta t) + Q(\delta t)$$

Algorithm Summary (Contd ….)

3. Predict measurement and compute Jacobian

$$\hat{z} = \vec{h}_\sigma(\hat{x}^-(t), \hat{\alpha}^-(t), \vec{b}(t), \vec{c}(t))$$

$$H = H_\sigma(\hat{x}^-(t), \hat{\alpha}^-(t), \vec{b}(t), \vec{c}(t))$$

4.Compute Kalman gain

$$K = P^- H^T (HP^- H^T + R_\sigma(t))^{-1}$$

5. Compute measurement residual

$$\Delta\vec{z} = \vec{z}_\sigma(t) - \hat{z}$$

Algorithm Summary (Contd ….)

6. Correct the prediction

$$\vec{x}(t) = \hat{x}^- + K\,\Delta\vec{z}$$

$$P(t) = (I - KH)P^-$$

7. Update external orientation quaternion

$$\Delta\hat{\alpha} = quaternion(\hat{x}[\phi], \hat{x}[\theta], \hat{x}[\psi])$$

$$\hat{\alpha} = \hat{\alpha} \otimes \Delta\hat{\alpha}$$

8. Zero incremental orientation state elements

$$\hat{x}[\Delta\phi] = \hat{x}[\Delta\theta] = \hat{x}[\Delta\psi]$$

SCAAT AutoCalibration

- In effect, a distinct *device filter* for each source or sensor to be calibrated.
- Calibrate parameters in $\vec{b}(t)$ and $\vec{c}(t)$
- Notation : $\hat{x}$ → Augmented matrix/vector

Device Filters

- Device parameters : $\vec{x}_\pi[i]$, $i = 0,...., n_\pi - 1$

1.(a) Allocate n_π dimensional state vector $\hat{x}_\pi$ for the device.
(b) Initialize with *a priori* estimates (from design).

2.(a) Allocate $n_\pi \times n_\pi$ noise covariance matrix $Q_\pi(\delta t)$
(b) Initialize with expected parameter variances.

3.(a) Allocate $n_\pi \times n_\pi$ error covariance matrix P_π
(b) Initialize with level of confidence in 1. above

Revised Tracking Algorithm

- Form augmented state vector

$$\hat{x}(t-\delta t) = [\hat{x}^T(t-\delta t)\ \ \hat{x}_b^T(t-\delta t)\ \ \hat{x}_c^T(t-\delta t)]$$

- Augmented error covariance matrix

$$\hat{P}(t-\delta t) = \begin{bmatrix} P(t-\delta t) & 0 & 0 \\ 0 & P_b(t-\delta t) & 0 \\ 0 & 0 & P_c(t-\delta t) \end{bmatrix}$$

- State Transition Matrix

$$\hat{A}(\delta t) = \begin{bmatrix} A(\delta t) & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix}$$

Process Noise Matrix

$$\hat{Q}(\delta t) = \begin{bmatrix} Q(\delta t) & 0 & 0 \\ 0 & Q_b(\delta t) & 0 \\ 0 & 0 & Q_c(\delta t) \end{bmatrix}$$

AutoCalibration (Contd ….)

- Follow the original algorithm.
- Extract and save device filter portions.

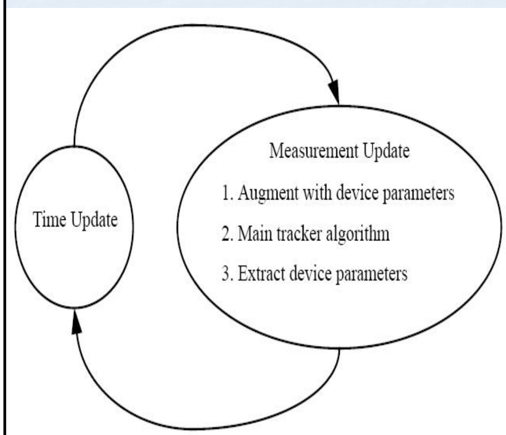
$$\hat{x}_b(t) = \hat{x}(t)[i....j]$$
$$P_b(t) = \hat{P}(t)[i....j, i....j]$$
$$\hat{x}_c(t) = \hat{x}(t)[k....l]$$
$$P_c(t) = \hat{P}(t)[k....l, k....l]$$

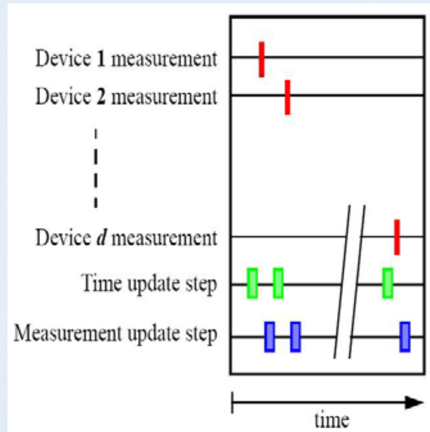
where

$$i = n+1$$
$$j = n+n_b$$
$$k = k+n_b+1$$
$$l = n+n_b+n_c$$

Tracking + Autocalibration

Autocalibration Timing Diagram





Code Optimization Strategies

- Small angles

  $$\cos(\Delta\theta) \approx 1\,,\quad \sin(\Delta\theta) \approx \Delta\theta$$

- Symmetry of $Q(\delta t),\ R_\sigma,\ P$
- Sparse Matrices – $A(\delta t)$ and Jacobian H
- Matrix inversion – reduced size of measurement vector

Filter Stability

- If there be real numbers $\alpha_1, \beta_1 > 0$ and $\alpha_2, \beta_2 < \infty$ such that for all $k \geq N$, for some $N \geq n/m$

$$\alpha_1 I \leq \sum_{i=k-N}^{k-1} A(t_k - t_{i+1}) Q(t_k - t_{i+1}) A^T(t_k - t_{i+1}) \leq \alpha_2 I$$

$$\beta_1 I \leq \sum_{i=k-N}^{k} \Gamma(i,k) \leq \beta_2 I$$

where

$$\Gamma(i,k) = A^T(t_i - t_k) H^T(t_i) R_\sigma^{-1}(t_i) H(t_i) A(t_i - t_k)$$

$$H(t_i) = H_\sigma(\vec{x}(t_i), \vec{b}(t_i), \vec{c}(t_i))$$

then the global system given by our dynamic and measurement model is uniformly asymptotically stable.

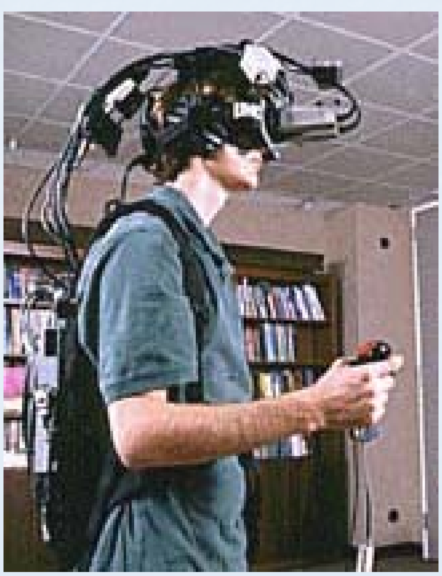
Source and Sensor Ordering Schemes

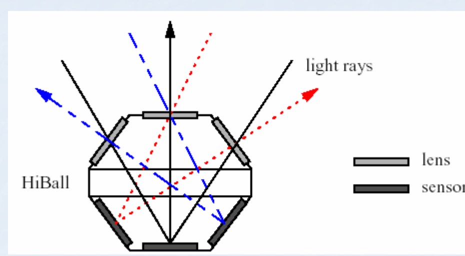
- Use a measurement scheduling algorithm
  - Better resource utilization
  - Monitor and control uncertainty in state vector
- Round-robin implementation

EXPERIMENTS AND SIMULATIONS

Hi-Ball Tracker

- Inside-out tracking





Initialization – Tracker

- 15-element state vector $\hat{x}(t) = [\,\hat{x}(t) \;\; \hat{x}_b(t)\,]$
- State Transition Matrix
  - Main tracker filter - $A(\delta t)$
  - Beacon filter - 3 X 3 Identity matrix
- Noise covariances determined off-line.
- Beacon filter state → initialized to (erroneous) position estimates.
- Beacon Error covariance matrix → initialized to

$$P_b(0)[i,j] = \begin{cases} (0.001)^2 & if\ i = j \\ 0 & otherwise \end{cases}$$

Measurement Noise

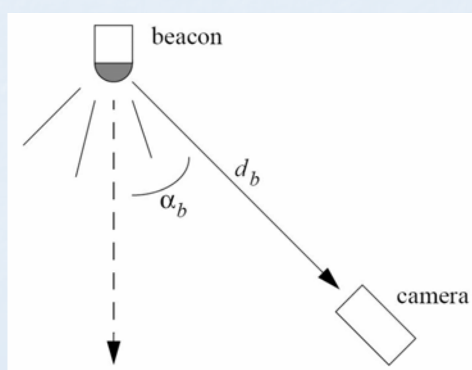
- $R_\sigma(t)$ → Uncertainty in actual camera measurement

$$R_\sigma(t)[i,j] = \begin{cases} \lambda_c & if\ i = j \\ 0 & otherwise \end{cases}$$

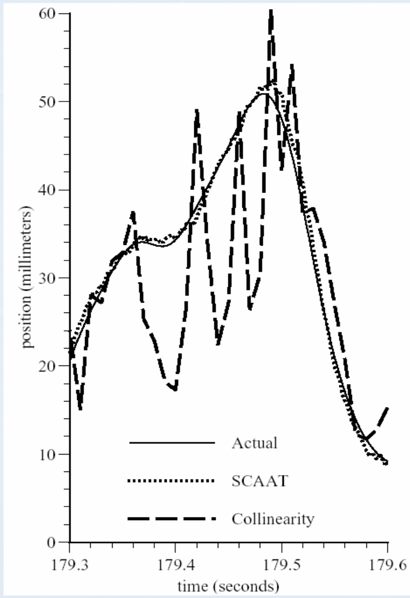
- Distance and angle-dependent variance λ_c

$$\sqrt{\lambda_c} = \frac{\sqrt{\lambda_0}\, d_b^2}{a\alpha_b^3 + b\alpha_b^2 + c\alpha_b + 1}$$

- Use previous position estimate to compute d_b and α_b

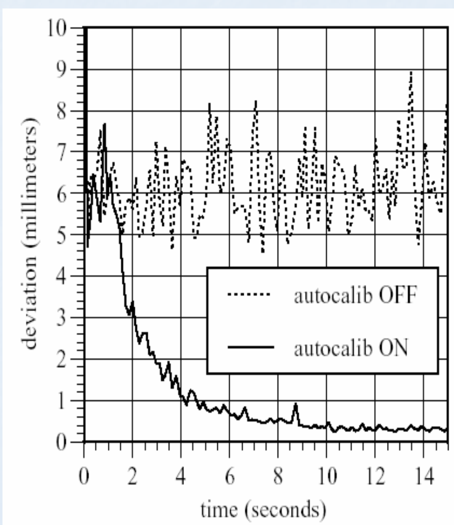


Simulation - Accuracy



- Collinearity – typically uses N = 10 observations per estimate.
- SCAAT
  - Higher update rate
  - Kalman Filtering
  - Autocalibration

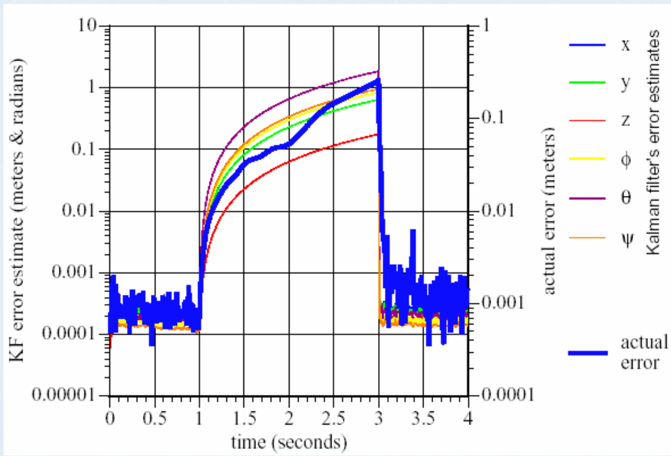
Autocalibration



- Stationary HiBall on stable platform.
- Estimate position with overhead beacons
- Deviation of estimates with time progressively decreases once autocalibration turned on.

Blocked Cameras

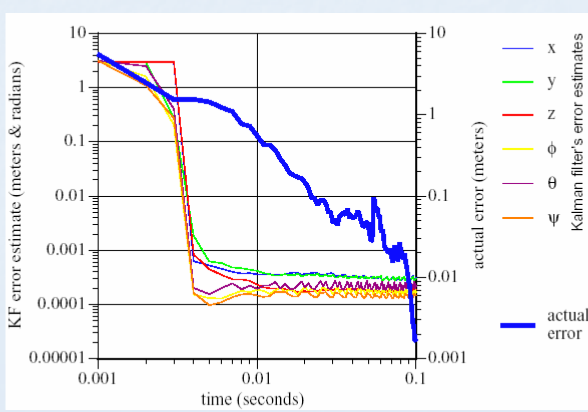
- Occlusion
- Camera not facing ceiling



The Weak Links ….

- *Cold Start*

Initialization to completely erroneous state vector sometimes leads to divergence.



Thank You !!