Tracking and recognition of airborne targets via commercial television and FM radio signals

Aaron D. Lanterman
(lanterma@ifp.wustl.edu)

Coordinated Science Laboratory
Univ. of Illinois
1308 W. Main
Urbana, IL 61801

This work was supported by DARPA Contract F49620-98-1-0498, administered by AFOSR.
http://www.ifp.uiuc.edu/~lanterma/darpa

March 31, 1999
Introduction

• Multistatic system using commercial transmissions
  — System remains covert
  — No cost of building transmitters
  — Coverage of low altitude targets

• Television signals
  — Low practical bandwidth
  — On all the time
  — Good doppler resolution
  — Poor range resolution
  — Need high SNR receivers

• FM radio
  — Better range resolution, but still not very good

• Bayesian approach (Srivastava, Grenander, Miller)
  — Prior $p(\mathbf{x})$ based on Newtonian dynamics
  — Data likelihood $p(\mathbf{y}|\mathbf{x})$
  — Inference via posterior $p(\mathbf{x}|\mathbf{y})$

• Denote target class
  $$\alpha \in \mathcal{A} = \{\text{F-15, F-18, Mig-29, Mirage, 747, Cessna, \ldots}\}$$
Track Parameterization

- Path in *inertial reference frame*
  \[ \mathbf{p}(t) = [p_1(t) \ p_2(t) \ p_3(t)]^T \in \mathbb{R}(3) \]

- Orientation relative to inertial reference frame
  \[ \mathbf{O}(t) \in SO(3) \]
  — Avoids sin’s and cos’s associated with Euler angles

- Angular velocity
  \[ \mathbf{q}(t) = [q_1(t) \ q_2(t) \ q_3(t)]^T \]

- Translational velocity in *body reference frame*
  \[ \mathbf{v}(t) = [v_1(t) \ v_2(t) \ v_3(t)]^T \]

- Positions, orientations, velocities related by
  \[
  \begin{align*}
  \dot{\mathbf{p}}(t) &= \mathbf{O}(t)\mathbf{v}(t) \\
  \dot{\mathbf{O}}(t) &= \mathbf{O}(t)\mathbf{Q}(t)
  \end{align*}
  \]

  where

  \[
  \mathbf{Q}(t) = \begin{bmatrix}
  0 & -q_3(t) & q_2(t) \\
  q_3(t) & 0 & -q_1(t) \\
  -q_2(t) & q_1(t) & 0
  \end{bmatrix}
  \]
Target Dynamics

- Velocities governed by SDEs
  \[ \dot{v}(t) + Q(t)v(t) = f(t) \]
  \[ I_m\dot{q}(t) + Q(t)I_mq(t) = \tau(t) \]

  — Translational force \( f(t) \)
  — Rotational torque \( \tau(t) \)
  — Rotational inertials \( I_m = \text{diag}(I_1, I_2, I_3) \)

- Discretization yields
  \[ \frac{v[k+1] - v[k]}{\delta t} + Q[k]v[k] = f[k] \]
  \[ I_m\frac{q[k+1] - q[k]}{\delta t} + Q[k]I_mq[k] = \tau[k] \]

  \[ p[k+1] = p[k] + \delta_t O[k]v[k] \]
  \[ O[k+1] = O[k] + \delta_t O[k]Q[k] \]  \( (1) \)

- To avoid accumulation of errors, use
  \[ O[k+1] = O[k] \exp(\delta_t Q[k]) \]
  instead of (1)
**Sample Flight Path**

- Flown in Silicon Graphics flight simulator
- Path in X-Y plane:

![X-Y path graph](image)

- Altitude (Z-coordinate):

![Altitude graph](image)
Extracted Velocities

- Body-frame velocities $v$

- Angular velocities $q$

March 31, 1999
Extracted Forces

- Translational force $\mathbf{f}_1$

- Torque $\mathbf{\tau}_1$
Extracted Forces (zooming in)

- Translational forces $\mathbf{f}$

- Torques $\mathbf{\tau}$
Fitting a Gaussian Model

- Estimated translational force covariance

\[ \Sigma_f = \begin{bmatrix} 79917 & 36421 & -18819 \\ 36421 & 96630 & -17301 \\ -18819 & -17301 & 51533 \end{bmatrix} \times 10^2 \]

\[ \Sigma_f = \begin{bmatrix} 88535 & 16860 & 8201 \\ 16860 & 39152 & 462 \\ 8201 & 462 & 6479 \end{bmatrix} \times 10 \]

using smaller chunk

- Estimated torque covariance

\[ \Sigma_T = \begin{bmatrix} 1576 & -521 & 9524 \\ -521 & 1416 & -9385 \\ 9524 & -9385 & 140464 \end{bmatrix} \times 10^2 \]

\[ \Sigma_T = \begin{bmatrix} 7585 & 2485 & -1723 \\ 2485 & 1141 & -553 \\ -1723 & -553 & 6182 \end{bmatrix} \times 10 \]

using smaller chunk

March 31, 1999
Fitting an $\alpha$-Stable Model

- $\alpha$-stable densities not generally available in closed form
- Linear combinations of $\alpha$-stable r.v.’s are $\alpha$-stable
- Characteristic exponent $\alpha \in (0, 2]$
  - Lower $\alpha$’s yield heavier tails
  - Gaussian: $\alpha = 2$
  - Cauchy: $\alpha = 1$
- Dispersion $\gamma > 0$
  - Analogous to variance
- $\alpha$ estimated by the asymptotic extreme value method of Tsihrintzis and Chysostomos (IEEE Sig. Proc., June 1996)
- $\gamma$ estimated via fractional lower order moments

<table>
<thead>
<tr>
<th></th>
<th>$\alpha$</th>
<th>$\gamma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_1$</td>
<td>1.0501</td>
<td>27.3</td>
</tr>
<tr>
<td>$f_2$</td>
<td>0.7197</td>
<td>6.7</td>
</tr>
<tr>
<td>$f_3$</td>
<td>1.0482</td>
<td>21.1</td>
</tr>
<tr>
<td>$\tau_1$</td>
<td>1.1996</td>
<td>172.7</td>
</tr>
<tr>
<td>$\tau_2$</td>
<td>1.0469</td>
<td>42.4</td>
</tr>
<tr>
<td>$\tau_3$</td>
<td>1.0008</td>
<td>207.6</td>
</tr>
</tbody>
</table>
Television Signals

• Three groups, 6 MHz spacing in each group:
  — Lower VHF, channels 2-6, 55.25-79.25 MHz
  — Upper VHF, channels 7-13, 175.25-221.25 MHz
  — UHF, channels 14-83, 471.25-885.25 MHz

• Around 85% of the power is in the carrier

• Model reflected signals as doppler shifted sinusoids

• Assume one receiving station
  — $N$-element array of isotropic sensors

• $S$ = number of transmitters

• $M$ = number of targets
Received Data

- After quadrature demodulation:

\[
\mathbf{r}(t) = \sum_{m=1}^{M} \mathbf{D}[\mathbf{p}^{(m)}(t)]b(\mathbf{p}^{(m)}(t), \mathbf{O}^{(m)}(t), \alpha^{(m)}) \\
\times \exp\{j\omega_D[\mathbf{p}(t)^{(m)}, \dot{\mathbf{p}}^{(m)}(t)]t\} + \mathbf{w}(t)
\]

- \(b\) complex reflectance
- \(\omega_D\) doppler shift
- \(\mathbf{w}\) noise
- \(\omega_D \ll \omega_C\), so use narrowband approximation

- Direction vector \(\mathbf{D}\) given by

\[
\mathbf{D}(\mathbf{p}, \omega_D) \approx \begin{bmatrix}
\exp(j\frac{\omega_C}{c} \mathbf{p}^T \mathbf{a}_1) \\
\vdots \\
\exp(j\frac{\omega_C}{c} \mathbf{p}^T \mathbf{a}_N)
\end{bmatrix}
\]

- \(\mathbf{a}_1, \ldots, \mathbf{a}_N\) = positions of sensors
Target Reflectance

• Each target type, frequency, incident angle, and observation angle gives different reflectance

• Prediction via XPATCH
  — Shooting and bouncing array approximation
  — Not accurate at low frequencies

• Prediction via FISC
  — Full method-of-moments solution
  — Uses “fast multilevel multipole algorithm”
  — Accurate at low frequencies
  — Memory requirements impractical at high frequencies

• FISC and XPATCH output $\xi$ normalized so that $|\xi|^2$ is the radar cross section. Adjust magnitude based on path loss according to the radar equation

\[ b = \text{phase}(\xi)\sqrt{P_R} = \xi \frac{\lambda \sqrt{P_T G_T G_R}}{(4\pi)^{3/2} R_T R_R} \]

— $G_T$ and $G_R$ antenna gains
— $R_T$ and $R_R$ ranges to transmitter and receiver
— $P_T$ transmitted power

• Large databases of $\xi$’s needed
Doppler Shift

- Doppler velocity $v_D = \frac{d}{dt}(R_T + R_R) = \frac{d}{dt}(\|p - s\| + \|p - r\|)$

  \[
  = \frac{d}{dt} \left( \sqrt{[p_x(t) - s_x]^2 + [p_y(t) - s_y]^2 + [p_z(t) - s_z]^2} \right) \\
  + \sqrt{[(p_x(t) - r_x)^2 + (p_y(t) - r_y)^2 + (p_z(t) - r_z)^2]} \right)
  \]

  \[
  = \left( \frac{[p_x(t)-s_x]\dot{p}_x(t)+[p_y(t)-s_y]\dot{p}_y(t)+[p_z(t)-s_z]\dot{p}_z(t)}{\sqrt{[p_x(t)-s_x]^2+[p_y(t)-s_y]^2+[p_z(t)-s_z]^2}} \right) \\
  + \left( \frac{[p_x(t)-r_x]\dot{p}_x(t)+[p_y(t)-r_y]\dot{p}_y(t)+[p_z(t)-r_z]\dot{p}_z(t)}{\sqrt{[p_x(t)-r_x]^2+[p_y(t)-r_y]^2+[p_z(t)-r_z]^2}} \right)
  \]

(2)

In practice, we compute $v_D$ using the last expression via a discrete approximation to $\dot{p}$

- Associated doppler shift is

  \[
  f_D = -\frac{v_D}{\lambda} \text{ or } \omega_D = -\frac{2\pi v_D}{\lambda}
  \]
Inference via the Posterior

- Recursive estimation
  - Extended Kalman filtering
  - Explicit nonlinear filtering (Daum)

- Jump-diffusion processes (Grenander & Miller)
  - Stochastic gradient ascent for continuous parameters
  - Jumps for discrete parameters
  - Gibbs and Metropolis style jumps

- Representation via a population of samples
  - “Conditional Density Propagation” (Blake & Isard)
  - “Bayesian bootstrap filter” (Gordon)
  - Inspired by factor sampling
  - Employ population of $N$ samples
  - Propagate samples forward under dynamical model
  - Resample population based on likelihood
Recursive Estimation

- \( \mathbf{x}[k] = \) parameters, \( \mathbf{y}[k] = \) data
- \( \mathbf{y}[k]'s \) indep. conditioned on \( \mathbf{y}[k]'s:\n
\[
p(\mathbf{y}[k], \ldots, \mathbf{y}[0]|\mathbf{x}[k], \ldots, \mathbf{x}[0]) = \prod_{i=0}^{k} p(\mathbf{y}[i]|\mathbf{x}[i]).
\]

- Recursive update:

\[
p(\mathbf{x}[k + 1]|\mathbf{y}[k + 1], \ldots, \mathbf{y}[0])
= \frac{1}{Z} p(\mathbf{y}[k + 1]|\mathbf{x}[k + 1])p(\mathbf{x}[k + 1]|\mathbf{y}[k], \ldots, \mathbf{y}[0])
= \frac{1}{Z} p(\mathbf{y}[k + 1]|\mathbf{x}[k + 1])
\times \int p(\mathbf{x}[k + 1]|\mathbf{x}[k])p(\mathbf{x}[k]|\mathbf{y}[k], \ldots, \mathbf{y}[0]) d\mathbf{x}[k]
\]

- In traditional Kalman filtering:
  — Dynamic and observation eqns. are linear
  — System inputs and noise are Gaussian
  — Densities are Gaussian
  — Just need means and covariances

- Extended Kalman filter for nonlinear systems
  — Linearize around current estimate

- Traditional nonlinear filtering
  — Needs explicit form for the density
  — Extremely complicated

March 31, 1999
The CONDENSATION Algorithm

• Independently developed by two groups
  — Blake & Isard: “Conditional Density Propagation”
  — Gordon: “Bayesian bootstrap filter”

• Inspired by factored sampling for static scenes
  — Want to sample from $p(x|y) \propto p(y|x)p(x)$
  — Suppose $p(x)$ easy to sample
  — Suppose $p(y|x)$ hard to sample but easy to evaluate
  — Draw $N$ samples $x^{(1)}, \ldots, x^{(N)}$ from $p(x)$
  — Choose from among them with probability

$$Pr\{\text{Choose } x^{(n)}\} = \frac{p(y|x^{(n)})}{\sum_{m=1}^{N} p(y|x^{(m)})}$$
CONDENSATION Con’t

- Use a pool of $N$ samples $\mathbf{x}^{(1)}[k], \ldots, \mathbf{x}^{(N)}[k]$

- Attach a probability $\pi^{(n)}[k]$ to each sample

- At each step $k$:
  1. For $n = 1 \ldots N$,
     (a) $\mathbf{x}^{\text{chosen}} = \mathbf{x}^{(n)}[k]$ with prob. $\pi^{(n)}[k]$
     (b) Draw $\mathbf{x}^{(n)}[k + 1]$ from $p(\mathbf{x}[k + 1] | \mathbf{x}^{\text{chosen}})$.
     (c) Assign $\tilde{\pi}^{(n)} = p(\mathbf{y}[k + 1] | \mathbf{x}^{(n)}[k + 1])$
  2. Let
     $$\pi^{(n)}[k + 1] = \frac{\tilde{\pi}^{(n)}}{\sum_{m=1}^{N} \tilde{\pi}^{(m)}}$$

- Compute cond. mean estimates of functions of $\mathbf{x}[k]$ by
  $$E[f(\mathbf{x}[k]) | \mathbf{y}[k], \ldots, \mathbf{y}[0]] \approx \sum_{n=1}^{N} \pi^{(n)}[k] f(\mathbf{x}^{(n)}[k])$$

March 31, 1999