Hard Combinatorial Problems in Multiple Target Tracking

and

Mitigating Computational Complexity using Methods from Analytic Combinatorics

by

Dr. Roy L. Streit
1818 Library St, Suite 600
Reston, VA 20190
r.streit@ieee.org and streit@metsci.com

Abstract

Analytic combinatorics (AC) is a classical mathematical field that in the last few years has found new applications in information fusion and multiple target tracking. Many problems in these applied fields are severely handicapped by the fact that exact solution algorithms are NP-hard. What AC contributes to these fields is two-fold. Firstly, AC formulates problems in terms of a generating function or generating functional. This formulation is exact and complete, taking the form of a single concise equation. Exact filters are derived from these functions by differentiation. Because the derivatives are exact, they too suffer from being NP-hard to compute. This is where the second contribution of AC comes into play. The derivative is rewritten as an integral, and the integral is approximated by the saddle point method. The use of saddle point methods is a classic device in AC, and it is powerful too since it leads to results not easily obtained in other ways. It is not limited to AC. It is considered an established method in the physics community, but it is virtually unknown in information fusion and target tracking. One goal of these two papers is to motivate interest in AC by showing how the methods of AC are applied in these fields. Another goal is to show that the saddle point method is a broadly applicable technique that provides principled approximations for many problems – regardless of whether or not exact solutions are NP-hard. The examples given in these papers are all drawn from papers published in the open literature.

Hard Combinatorial Problems in Multiple Target Tracking

I

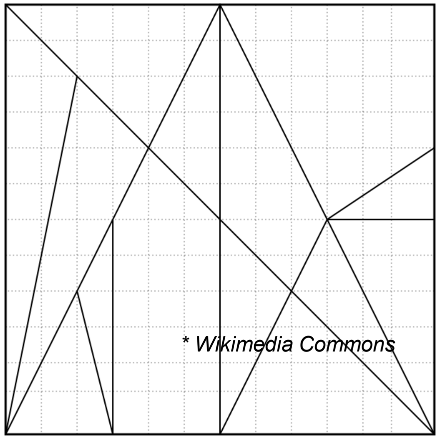
Roy Streit
Metron, Inc.
1818 Library Street
Reston, VA 20190
+1 (703) 787-8700
streit@metsci.com or r.streit@ieee.org

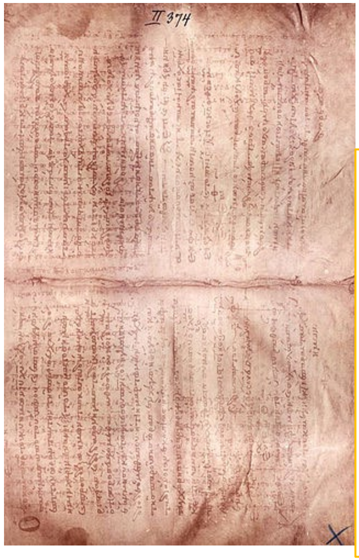
Artificial Intelligence for Military Multiple Sensor Fusion Engines

NATO Research Lecture Series SET-290, 2022
Rome 26-27 Sep ; Wachtberg 29-30 Sept; Budapest 03-04 Oct

Combinatorial Problems are Ancient

- Stomachion
    14 piece dissection
    puzzle attributed to
    Archimedes (c. 287 BC – 212 BC)
    (before NATO expansion)
- Archimedes palimpsest
    - Unique, 10th century Byzantine parchment copy
    - *In 13th century it was scraped, washed, folded in half, turned 90 deg., and overwritten with liturgical text*
- How many ways can the 14 pieces be arranged into a square?
- Enumeration is the natural method
- Is there an analytical method?
- What is "analytic combinatorics"?


* Wikimedia Commons


The Walters Museum, Baltimore, MD USA

Combinatorial Problems are Modern

- Prime Number Theorem. Asymptotic result about the "distribution" of prime numbers

$$\#\{\text{Prime numbers } \leq x\} = \frac{x}{\ln x} \quad \text{as } x \to \infty$$

- Conjectured by Gauss (1791, *age 14*) and later by Legendre (1797) and others
- Proved by complex analytic methods in 1896
  - Independently by Hadamard and de la Vallee-Poussin
- Thus began the field of "analytic number theory"
- "Elementary" arithmetic proof was discovered in 1949, independently by Selberg and Erdos

Analytic Combinatorics (AC)

- Combinatorial questions ask questions about a sequence

  $a_0, a_1, a_2, a_3, \ldots$

  where a_n is the number of "configurations" or "things" of interest
- In tracking, configurations are often "hypotheses" about the data
- Combinatorial enumeration
  - How does a_n grow as $n \to \infty$?
  - Bayesian estimation in bivariate problems
- Combinatorial optimization
  - What is the best, or optimal, configuration?
- Analytic Combinatorics approach
  - Map all hypotheses into a generating function (GF)
  - "A GF is a clothesline on which we hang up a sequence of numbers for display"
    (*generatingfunctionology*, H. Wilf, CRC Press, 2005)

Why Complex Analysis

- GFs are functions of a complex variable, z
- GFs are analytic functions – classic 19th mathematics
- Why do they help solve discrete problems?
- "Why" is for philosophers. Nonetheless … possible answers
  - Discrete problems often have hidden regularities
    - Examples in game theory, graphs, and specialized math problems ("Inevitable Randomness in Discrete Maths" by J. Beck, AMS Pubs, 2009)
  - Regularities can often show up in the "dual" (frequency) domain, z
    - GF(z^{-1}) is the *z*-transform of the sequence a_n
  - Concentration of measure
    - Much more broadly applicable than central limit theorem
  - The success of random projections for NN design
    - "Random kitchen sink" (Rahimi and Recht, 2008)
- "How" GFs work is a question for everyone

Probability Generating Function (PGF)

- Definition: $\mathbf{G}(z) = \sum_{n=0}^{\infty} a_n \, z^n$
  - z is the “indeterminate variable” For us, it is a complex variable.
  - It is a FORMAL power series – it does not need to converge
- We call it a PGF when $a_n = \Pr\{A = n\} \geq 0$ and $\mathbf{G}(1) = 1$
  - PGFs are analytic inside the disc $|z| < 1$
  - Moment generating function: $M(t) = \mathbf{G}(e^t)$
  - Characteristic function: $\Phi(\omega) = \mathbf{G}(e^{\omega\sqrt{-1}})$
- $\mathbf{G}(z)$ “encodes” probabilities – it characterizes the distribution
- To decode, differentiate $\mathbf{G}(z)$

$$\Pr\{A = n\} = \frac{1}{n!}\mathbf{G}^{(n)}(0) \equiv \frac{1}{n!}\frac{d^n}{dz^n}\bigg|_{z=0} \mathbf{G}(z)$$

- You can also extract summary statistics from the derivatives
  - Mean number: $\mathrm{E}[A] = \mathbf{G}'(1) = \frac{d}{dz}\bigg|_{z=1} \mathbf{G}(z)$

Care and Feeding of GFs

- GFs are rarely found by summing the series
- They are more often derived from first principles
  - Known recursion (AR model) – solve for the GF (e.g., Fibonacci)
  - Decomposition into smaller parts
    - Independence: GF of a sum → product of GFs
    - E.g. How many "heads" with *n* coin flips
      - Identical coins: $\mathrm{G(z)} = (q + p\,z)^n$
      - Non-identical: $\mathrm{G(z)} = (q_1 + p_1 z)\ldots(q_n + p_n z)$
    - More fun with GFs: Random numbers of randomly types of coins
      - Suppose that we flip *n* coins a random number, *N*, of times
      - Then the GF of the total number of heads is $$H_N\big(\mathrm{G(z)}\big) = H_N(\,(q_1 + p_1 z)\ldots(q_n + p_n z)\,)$$
      - "Great fleas have little fleas upon their backs …"
- GFs are also powerful models of *if-then-else* structures
- Sometimes GFs are easy to find, sometimes not

Example: Making Change

- How many ways to make €100.00 in coins?
- Expand the question to any euro value and find the GF:

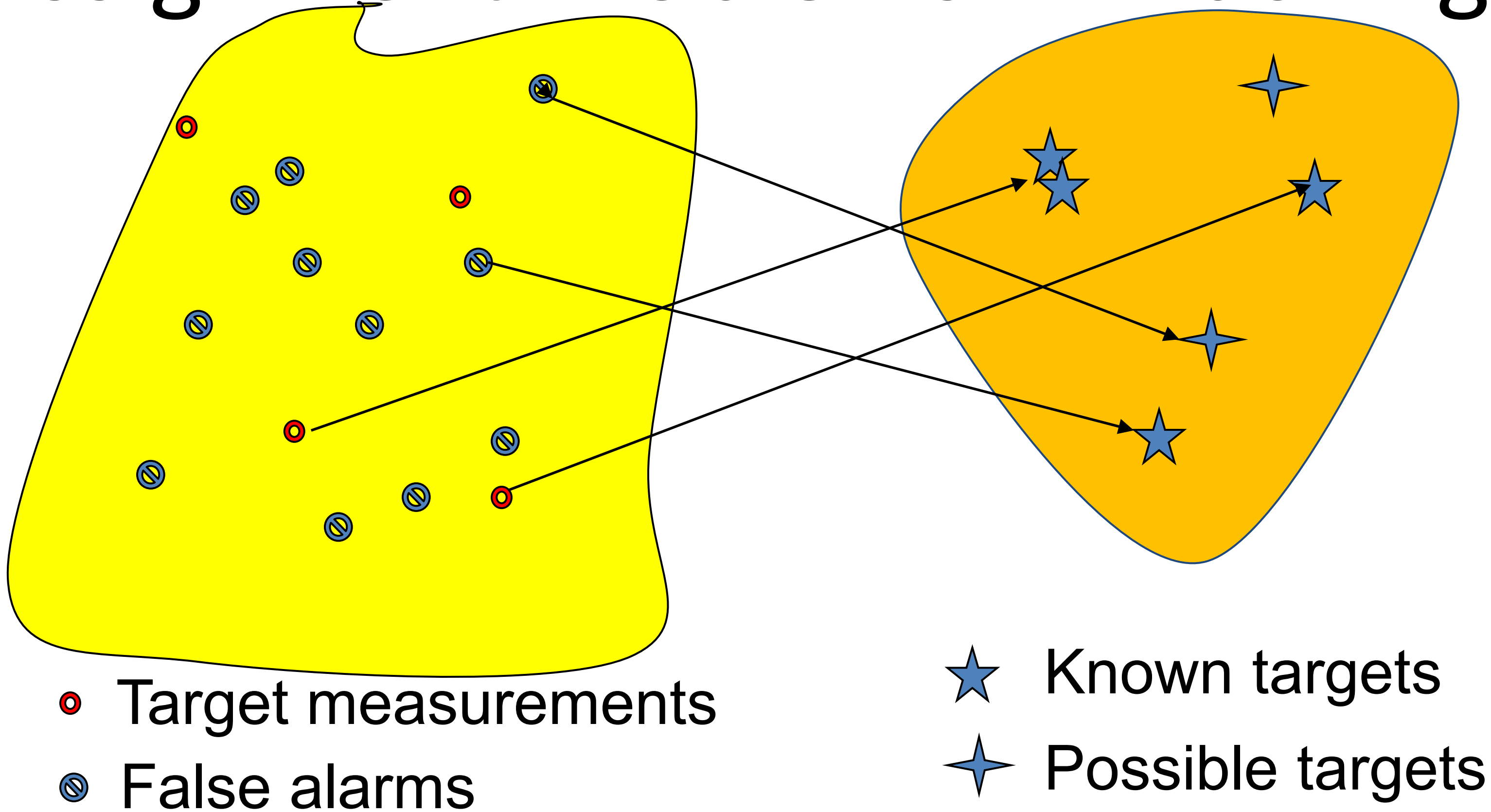
$$G(z)=\frac{1}{(1-z)(1-z^2)(1-z^5)(1-z^{10})(1-z^{20})(1-z^{50})(1-z^{100})(1-z^{200})}$$

- Exact answer is the coefficient of z^{10000}
  - To find the answer, just differentiate G(z) 10,000 times ☺

    = 1,133,873,304,647,601 ≈ 1.133 x 10^15
- Saddle point approximation (can be computed by hand)

    = 1.145 x 10^15 → *less than 1% error*
  - Homework: Find a way to enumerate the solutions
- This *almost-a-toy* problem is due to Polya (c. 1910?)
  - Partition function (Hardy and Ramanujan, 1918)
- *It's an unusually long road from here to tracking*
  - Measurement assignments → GFs → Derivatives → Cauchy Integral → Saddle point approximation → Particle filter weights

Assignment Problems in Tracking



One Hypothesis: An assignment of measurements to targets and false alarms
Likelihood function is sum over the set of all feasible hypotheses
Hugely impractical sum in many applications
Note: Two unresolved targets are depicted. Adds complexity.

Combinatorial Methods in Tracking

- Many information fusion problems are inherently combinatorial
  - Tracking – one or more targets in clutter, multisensor, and batch
  - High level fusion too: Integer Linear Programming (ILP)
- Traditional approach in tracking is to enumerate all "feasible" combinations
  - Natural and intuitive
  - Exploits domain knowledge to find and prune feasible combinations
  - Hard to use for large scale problems
- Classical alternative to enumeration
  - Use generating functions Ψ and complex analysis (LaPlace c. 1810)
- **Analytic Combinatorics**: it is equivalent to enumeration
  - Discovery Step
    - Encode the problem into a generating functional Ψ
  - Analysis Step
    - Decode Ψ by differentiation

Discovery Step

- *Key Idea*
  - Derive Ψ from first principles, exploit the problem structure
  - GFs become functions of functions (a.k.a. functionals)
  - Fear not, functionals convert to ordinary GFs given measurements
  - GFs in tracking are multivariate
    - Usually bivariate for one sensor: targets and measurements
    - Bayes Theorem has GF form := a ratio of derivatives of the GF
- Ψ incorporates _all_ the problem assumptions
  - Independent targets/objects
  - Conditional independence of measurements, sensors, targets
- Tracking filters are derived by decoding Ψ
  - No loss of information in encoding or decoding
- The factorized form of Ψ is basis for taxonomy of tracking filters

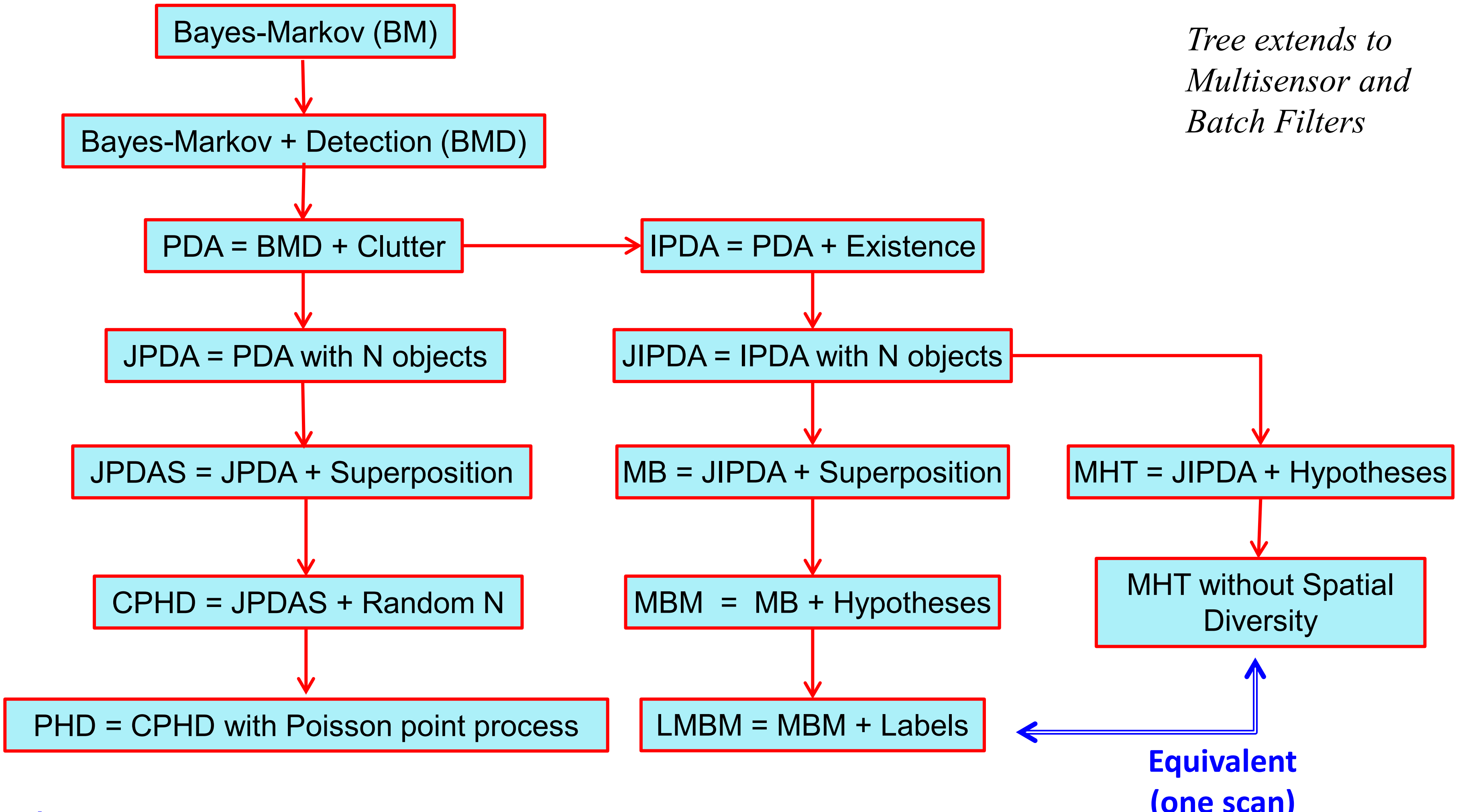
Analysis Step

- Many things can be done with Ψ *without* explicit enumeration
  - Event probabilities are the derivatives of Ψ at the origin
  - Generating function for Bayes Theorem

$$\Psi_{\text{Bayes}}(\text{targets}\,|\,\text{measurements}) = \text{ratio of derivatives of } \Psi(\text{targets, measurements})$$

  - Derivatives of Ψ_{Bayes} are the target conditional probabilities
- **Exact derivatives of Ψ are as hard to evaluate as computing exact event probabilities by enumeration!**
- *Analytic Combinatorics*
  - Recast derivatives of Ψ as integrals
    - Cauchy Residue Theorem (1825)
  - Derive approximations to the derivatives
    - Saddle point method (also called method of stationary phase)
    - Established technique in math and physics

AC Taxonomy of Tracking Filters*



*Analytic Combinatorics for Multiple Object Tracking, by R. Streit, R. Angle, and M. Efe, Springer, 2021

Justification of the "Main Line"

Classic Bayes-Markov: Exactly one target and one measurements

$$\downarrow \quad \Psi^{BM}(h,g) = \int_S \int_Y h(x)\, g(y) \underbrace{\mu(x) p(y \mid x)}_{Bayes\,Theorem} dy\, dx \; = \int_S \left\{ h(x)\, \mu(x) \left[\int_Y g(y) p(y \mid x)\, dy \right] \right\} dx$$

PDA filter: Bayes-Markov + missed target detections and false alarms

$$\downarrow \quad \Psi^{PDA}(h,g) = \Psi^{FA}(g) \int_S \left\{ h(x)\, \mu(x) \left[1 - Pd(x) + Pd(x) \int_Y g(y) p(y \mid x)\, dy \right] \right\} dx$$

JPDA filter: N targets, independent, with missed target detections and false alarms

$$\downarrow \quad \Psi^{JPDA}(h_1, \ldots., h_N, g) = \Psi^{FA}(g) \prod_{n=1}^{N} \Psi^{PDA}(h_n, g)$$

Each target has its own space: h_n

Impractical : NP - hard

JPDAS intensity filter:

$$\downarrow \quad \Psi^{JPDAS}(h,g) = \Psi^{FA}(g) \prod_{n=1}^{N} \Psi^{PDA}(h,g)$$

JPDA with **Superposition** $\Rightarrow h_1 = \ldots = h_N = h$

Fast: $O(N \times M)$

CPHD intensity filter:

$$\downarrow \quad \Psi^{CPHD}(h,g) = \Psi^{FA}(g)\, G_N\left(\Psi^{PDA}(h,g) \right)$$

JPDAS with random N with GF $G_N(z)$

Fast

PHD intensity filter:

$$\Psi^{PHD}(h,g) = \Psi^{FA}(g)\, G_{Poisson}\left(\Psi^{PDA}(h,g) \right)$$

CPHD with Poisson distribution on N

Fast

End of the Beginning – Complexity Worsens

- Multisensor

Single target MS/PDA is NP-hard

$$\Psi^{\mathrm{MS/PDA}}(h, \underbrace{g_1, \ldots, g_L}) = \Psi^{\mathrm{FA}}(g) \int_S \left\{ h(x)\, \mu(x) \prod_{\ell=1}^{L} \left[1 - Pd_\ell(x) + Pd_\ell(x) \int_{Y_\ell} g_\ell(y)\, p_\ell(y \mid x)\, dy \right] \right\} dx$$

$\Rightarrow$ MS/PHD intensity is NP-hard

$$\Psi^{\mathrm{MS/PHD}}(h, g) = \Psi^{\mathrm{FA}}(g)\; G_{Poisson}\left(\Psi^{\mathrm{MS/PDA}}(h, g_1, \ldots, g_L) \right)$$

- Batch of length K
  - GF is a K-deep nest of the form Ψ = G(G(…(G(h,g))…))
  - With Poisson distributions, GF is exponential tower of height K
- "Extended" targets are not point scatterers
  - Unknown number of targets, each with an unknown # scatterers
  - GF is an exponential tower, under Poisson assumptions
- Unresolved targets
  - Evaluating the "natural" GF is itself NP-hard

NP-hard* Likelihood Functions

- Likelihood functions are often combinatorial
  - E.g., they are sums over feasible assignments
  - Too many assignments → NP-hard
- At scale, NP-hard likelihoods are roadblocks
  - Approximations are inevitable
- SMC particle filters compute likelihoods
- Particle intensity filters compute intensities for each particle
- Either way, exact calculation is doomed for NP-hard problems
- Use saddle point approximation
  - Stationary phase, “stationary action”
  - It can be remarkably fast and accurate
  - It is 100% free of all enumerations

* *Loosely speaking, a problem is NP-hard if it is impractical to solve using any known algorithm*.

Analytic Combinatorics Method

- Find the probability generating functional, denoted Ψ
- Given the data $\{y_1,\ldots,y_M\}$
- Reduce PGFL to a multivariate function, Ψ(β)

→ Substitute a train of Dirac deltas: $\sum_{m=1}^{M} \beta_m \delta_{y_m}$

→ Deltas replace integrals with samples of the integrand

- SMC particle filter weights are derivatives of Ψ(β) at 0

$$w(x) = Weight\ of\ a\ particle\ at\ x$$

$$\propto \frac{d^M}{d\beta_1 \cdots d\beta_M}\bigg|_{\beta=0} \Psi(\beta_1,\cdots,\beta_M)$$

- Particle weights for intensity filters are ratios of derivatives
- No information loss in this formulation
- Terms in derivative map 1-to-1 to feasible combinations
- BUT → we never take these derivatives

Saddle Point Approximation

- Write derivatives as Cauchy contour integrals

$$w(x) \propto \frac{1}{(2\pi i)^M} \oint_{C(r_1)} \cdots \oint_{C(r_M)} \frac{\Psi(\beta)}{\beta_1 \cdots \beta_M} \frac{d\beta_1 \cdots d\beta_M}{\beta_1 \cdots \beta_M}$$

- Integrand is *guaranteed* to have a unique saddle point
- This is a point $\widehat{\beta}_{1:M} > 0$ such that the Taylor series

$$\frac{\Psi(\beta)}{\beta_1 \cdots \beta_M} \equiv e^{\phi(\beta)} = \exp\left(\phi(\hat{\beta}_{1:M}) + \boxed{\begin{matrix} LINEAR\ TERM\ IS \\ ZERO\ AT\ \hat{\beta}_{1:M} \end{matrix}} \right.$$

This property defines the saddle point

$$\left. + \frac{1}{2}\left[\beta_{1:M} - \hat{\beta}_{1:M}\right]^T \begin{bmatrix} Hessian\ Matrix \\ of\ \phi\ at\ \hat{\beta} \end{bmatrix} \left[\beta_{1:M} - \hat{\beta}_{1:M}\right] + \text{H.O.T.} \right)$$

- Almost Gaussian → if contours pass through saddle pt.
- Everything is exact and error-free up to this point

One measurement

r = 0.13

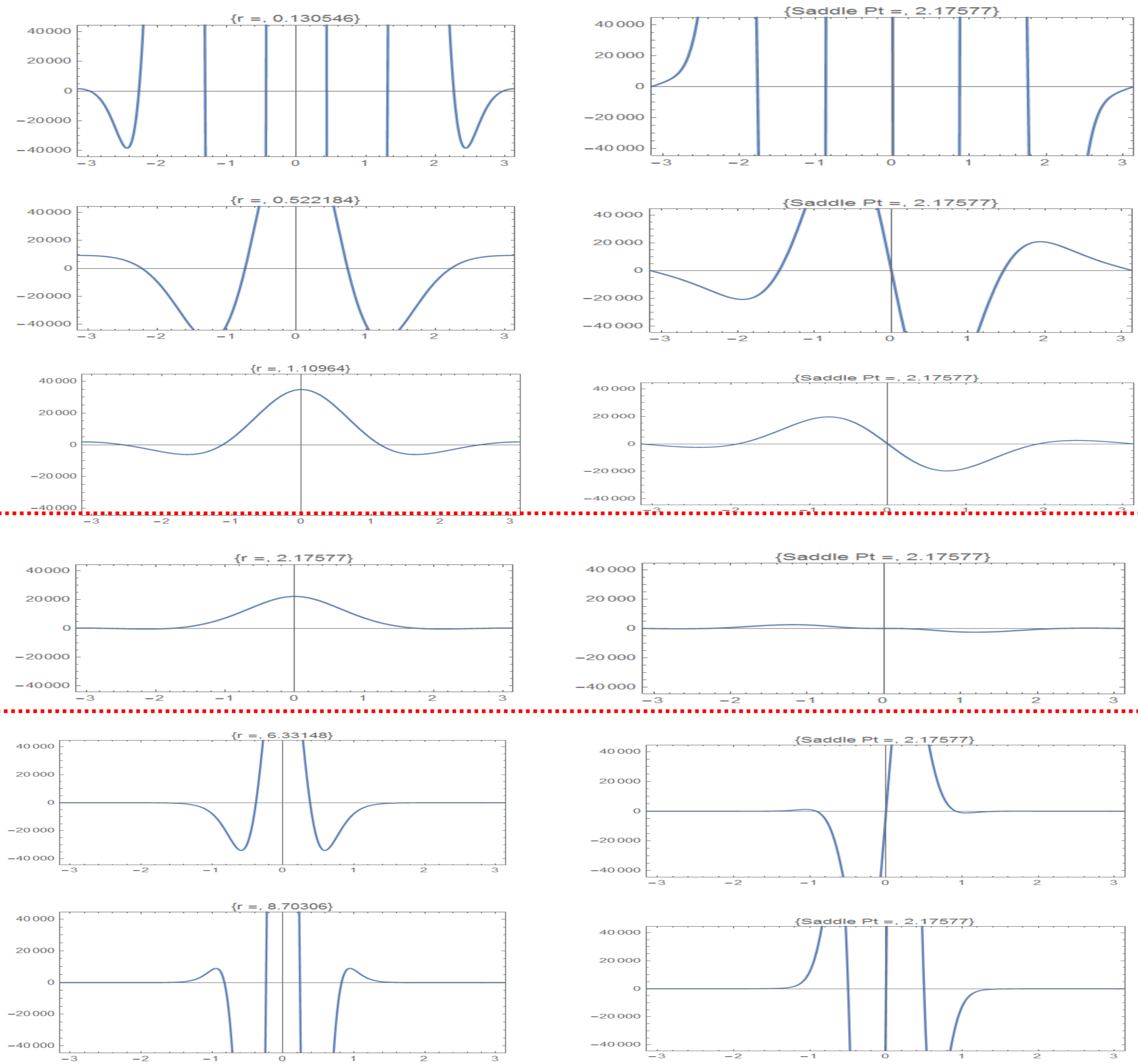
r = 0.52

r = 1.1

r = 2.17

r = 6.3

r = 8.7



Real parts have same integral

Imaginary parts integrate to 0

Why Find the Saddle Point?

Gaussian Shape only at Saddle Pt

Approximation and Bound

- The saddle point *approximation*

$$w(x) = p\left(y_{1:M} \,\middle|\, \begin{matrix} particle \\ at\ \ x \end{matrix} \right) \approx \frac{\Psi(\hat{\beta})}{(2\pi)^{M/2}\sqrt{\det\left(Diag(\hat{\beta})^T H_\phi(\hat{\beta})\, Diag(\hat{\beta}) \right)}}$$

  - The approximation is asymptotically accurate when there is a large parameter and certain “admissibility” conditions hold.
  - No obvious large parameter here

- The saddle point *bound*

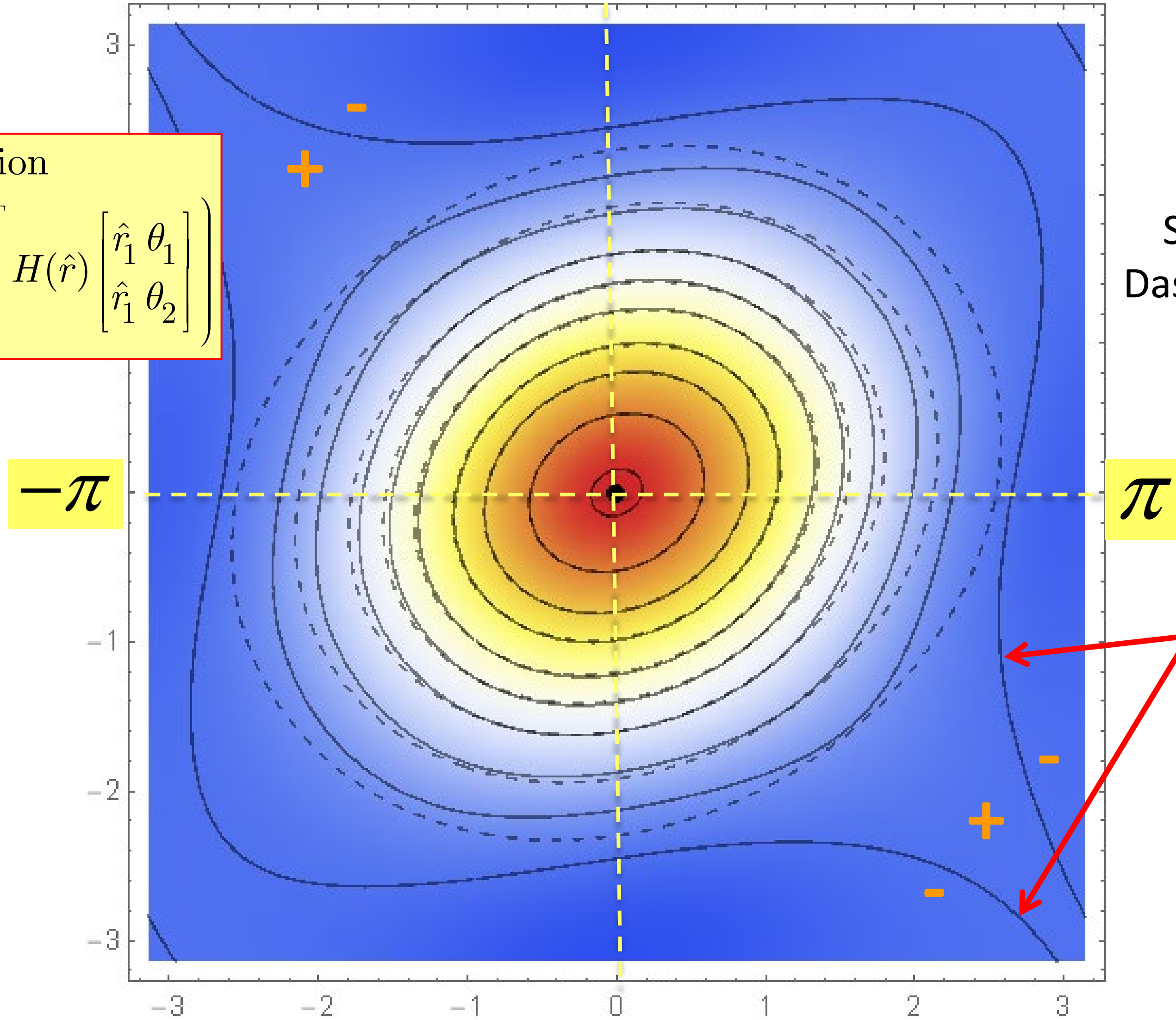
$$w(x) = p\left(y_{1:M} \,\middle|\, \begin{matrix} particle \\ at\ \ x \end{matrix} \right) \leq \frac{\Psi(\hat{\beta})}{(2\pi)^{M/2}}$$

  - The bound always holds, no large parameter, no need to satisfy admissibility conditions.

JPDA Example – 2 Measurements, 4 Targets

Approximation

$$\propto \exp\left(-\frac{1}{2}\begin{bmatrix}\hat{r}_1\,\theta_1\\ \hat{r}_1\,\theta_2\end{bmatrix}^T H(\hat{r})\begin{bmatrix}\hat{r}_1\,\theta_1\\ \hat{r}_1\,\theta_2\end{bmatrix}\right)$$



Contours:

Solid – *Exact*

Dashed – *Approx*

Zero Contour Lines

GFs as "Action"

- Physics defines the "Action" $S = \int_{t_1}^{t_2} \mathcal{L}(q(t), \dot{q}(t), t)\, dt$
  - Use calculus of variations to minimize $\mathcal{L}$ = Lagrangian
    - Euler-Lagrange equations
  - "The Statistical Physics of Data Assimilation and Machine Learning," Henry Abarbanel, Cambridge Univ Press, 2022
- Analytic combinatorics
  - Paths are the circles of radii β
  - Define the action as

$$S(\beta) = \log \frac{\Psi(\beta)}{\beta_1 \cdots \beta_M} = \log \Psi(\beta) - \log(\beta_1 \cdots \beta_M) \equiv T - V$$

  - Minimize the action to find the best paths (circles)

$$\hat{\beta} = \underset{\beta}{\arg\min} \left\{ \log \Psi(\beta) - \log(\beta_1 \cdots \beta_M) \right\}$$

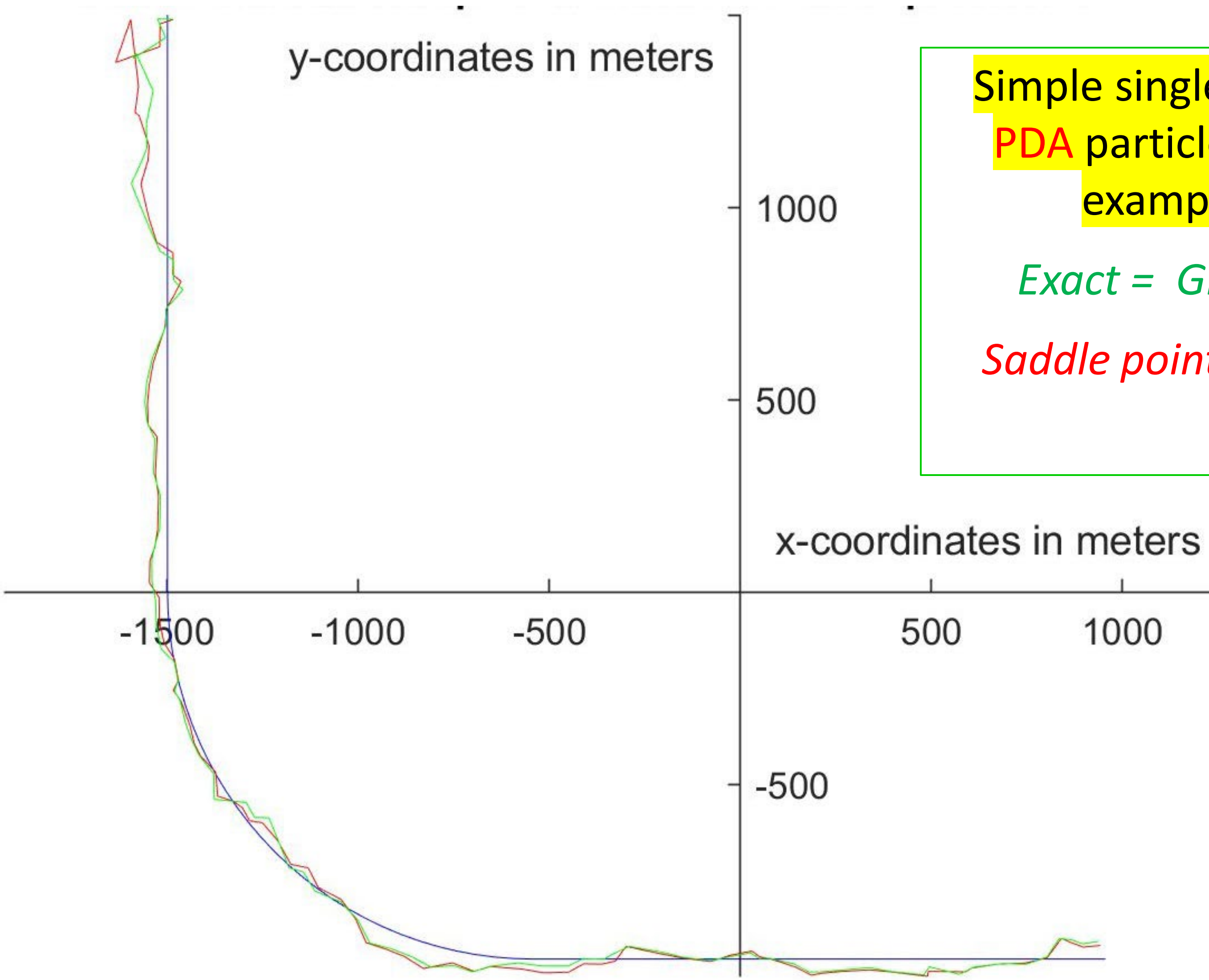
- The combinatorial action is at or near the saddle point

Fixed Point Iteration for the Saddle Point*

- Necessary conditions for the saddle point
  - Set the gradient of $log\Psi(\beta) - \sum_m \log \beta_m = 0$
  - This sets up a natural fixed-point style iteration

- Example: JPDA
  - Computing the fixed point for JPDA is **fast**. The fixed-point iteration converges monotonically.
  - Approximating particle weights using the saddle point *bound* is essentially **linear** complexity
  - The complexity of the saddle point *approximation* for a particle weight is governed by det(Hessian). It is shown in the paper (using Weinstein–Aronszajn identity) that this complexity is **either $\boldsymbol{O(N^3)}$ or $\boldsymbol{O(M^3)}$**, whichever is smaller.

* ”On particle filters with high complexity combinatorial likelihood functions,” S. Ferguson, J. Silver, R. Streit, ISIF Fusion Conference, Linkoping, Sweden, July 2022



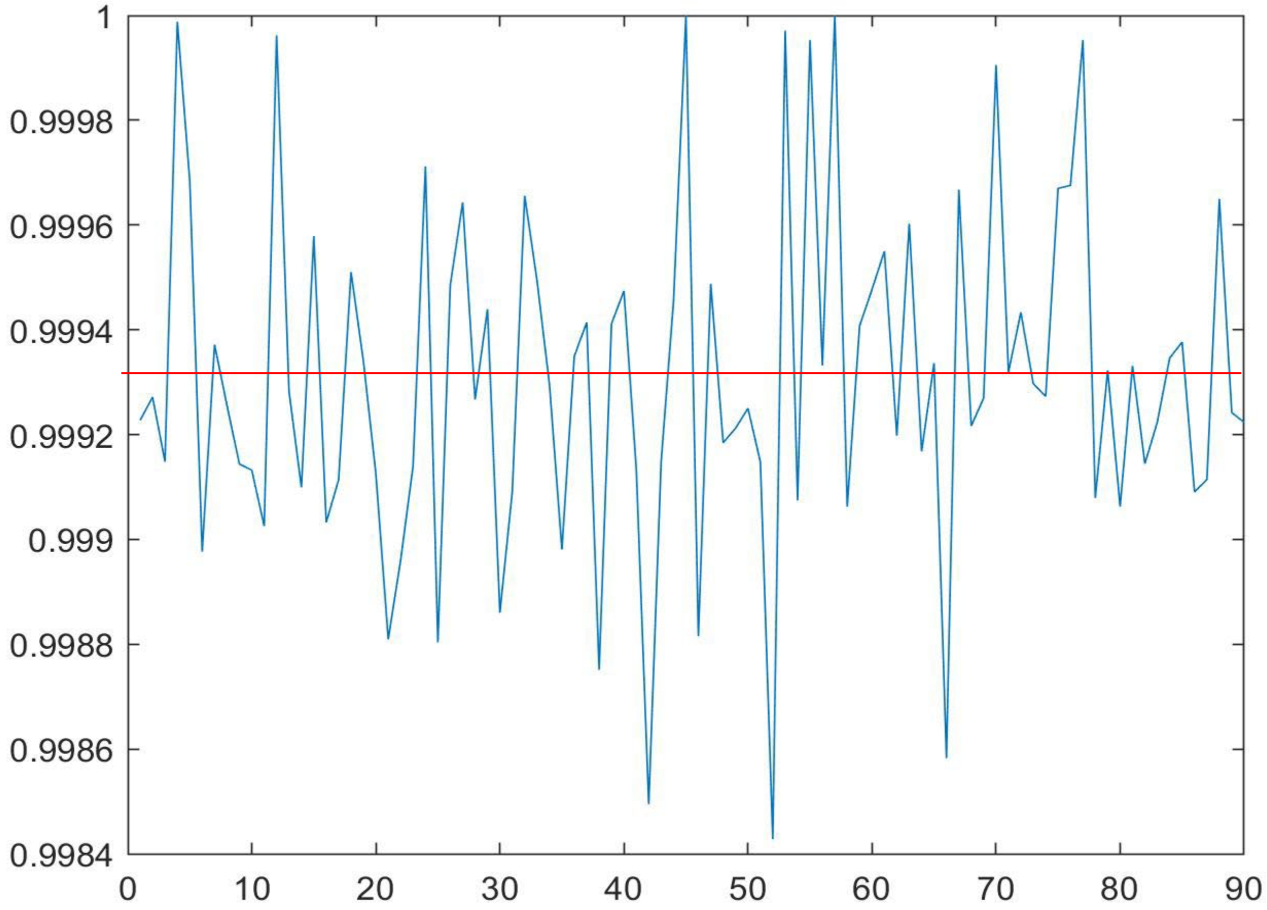
Simple single target PDA particle filter example

Exact = GREEN

Saddle point = RED

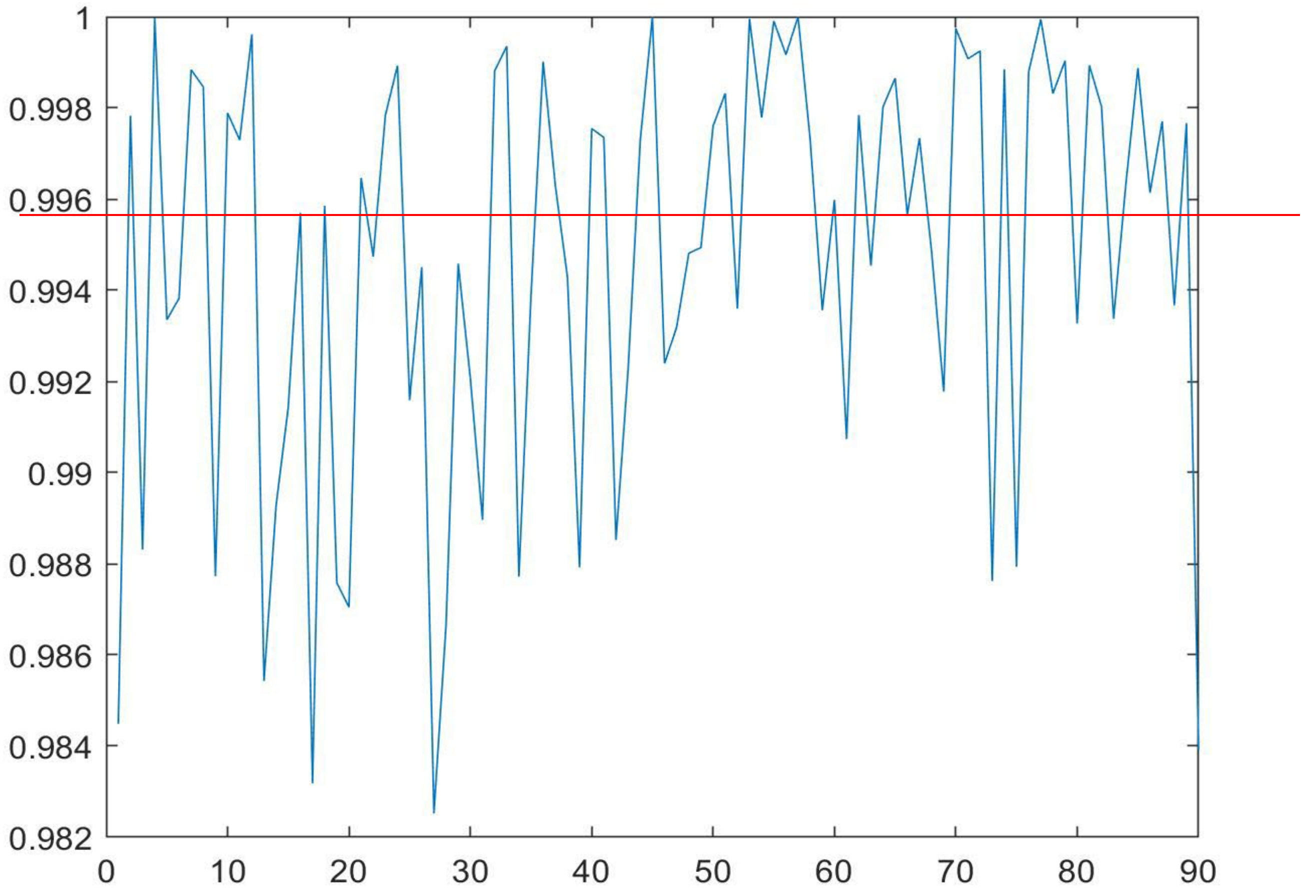
Particle Weight Correlation

Exact to saddle point



Particle Weight Correlation

Exact to saddle bound



Concluding Remarks

- Saddle point method avoids all enumeration
- Applicable to likelihood functions with a known probability generating function
- The long road from AC to tracking applications
  - Measurement assignments ➔ GFs ➔ Derivatives ➔ Cauchy Integral ➔ Saddle point approximation ➔ Particle filter weights
- First example is NP-hard filter: JPDA
  - Fast fixed-point calculation for each particle
  - Hessian-free saddle **bound** approximation
    --- *essentially linear*
  - Saddle **point** approximation
  - High correlation of exact and approximate particle weights
- How to evaluate approximation when exact is NP-hard?
- More examples in the next lecture

Backup

Machine-Precision Numerical First Derivative for Free

It looks like magic and feels like magic,

but it is for real

The Theory is Simple

- Given a function $f(x)$ that is analytic in neighborhood of $x_0 \in \mathbb{R}$
- Taylor series expansion, with sufficiently small $\varepsilon \in \mathbb{R}$

$$f(x_0 + i\varepsilon) = f(x_0) + f'(x_0)(i\varepsilon) + \tfrac{1}{2!} f''(x_0)(i\varepsilon)^2 + \tfrac{1}{3!} f'''(x_0)(i\varepsilon)^3 + H.O.T.$$

- Imaginary part is

$$\operatorname{Im} f(x_0 + i\varepsilon) = f'(x_0)\varepsilon - \tfrac{1}{3!} f'''(x_0)\varepsilon^3 + H.O.T.$$

- Divide by ε and rearrange terms:

$$f'(x_0) = \frac{1}{\varepsilon} \operatorname{Im} f(x_0 + i\varepsilon) + O\left(\varepsilon^2\right)$$

2^{nd} order accuracy

- To find the derivative of f at x_0 on the real line:
  - Perturb x_0 in the complex direction
  - Evaluate the function in **complex arithmetic**
- *Accurate to machine precision* !
- Central finite difference is also 2^{nd} order accurate
  - In practice, choosing ε is tricky subtraction errors for small ε

Not All Second Order Methods Are Created Equal

Table C.1 Complex step and central-difference estimates of f' at $x_0 = -0.6784$; the most accurate significant digit (rounded) is printed in bold font.

ϵ	Complex Step	Central-Difference
1e-04	-0.002284**9**33782882690	-0.002284**9**56126530346
1e-05	-0.0022849**4**4840968425	-0.0022849**5**077590805
1e-06	-0.00228494**5**1549592	-0.0022849**4**233821307
1e-07	-0.002284944952**6**53912	-0.002284**9**54447873133
1e-08	-0.0022849449526**6**4632	-0.002284**9**27802520542
1e-09	-0.0022849449526**6**5128	-0.00228**4**394895468722
1e-10	-0.0022849449526**6**4094	-0.0022**8**2618538629320
1e-11	-0.0022849449526**6**4482	-0.002**3**09263891220326
1e-12	-0.0022849449526**6**4966	-0.001332267629550188
1e-13	-0.0022849449526**6**5370	+0.008881784197001250
1e-14	-0.0022849449526**6**5749	-0.088817841970012500
1e-15	-0.0022849449526**6**5591	-0.888178419700125300
1e-16	-0.0022849449526**6**5591	0.
Exact	-0.002284944952665635	-0.002284944952665635

$$f(x) = \frac{\exp(\cos(x) - 2\sin(x))}{1 + x^2}$$

Central difference: $f'(x_0) \approx \dfrac{f(x_0 + h) - f(x_0 - h)}{2h}$

Applications in Tracking

- Find the GFL and secular function of the problem
- Adopt a particle filter model
- Need the weights of every particle
- Weights are ratios of derivatives of the secular function
- Use the complex step method
- JPDAS example used the method to evaluate weights of the particles in the intensity function
  - Used the method exactly as presented here
- Cross-derivatives require a *multi-complex* step method:
  - Unnecessary to take symbolic derivatives at all (!)
  - Computational complexity – becomes the question
  - How hard is it to evaluate the secular function?

Mitigating Computational Complexity using Methods from Analytic Combinatorics

II

Roy Streit
Metron, Inc.
1818 Library Street
Reston, VA 20190
+1 (703) 787-8700
streit@metsci.com or r.streit@ieee.org

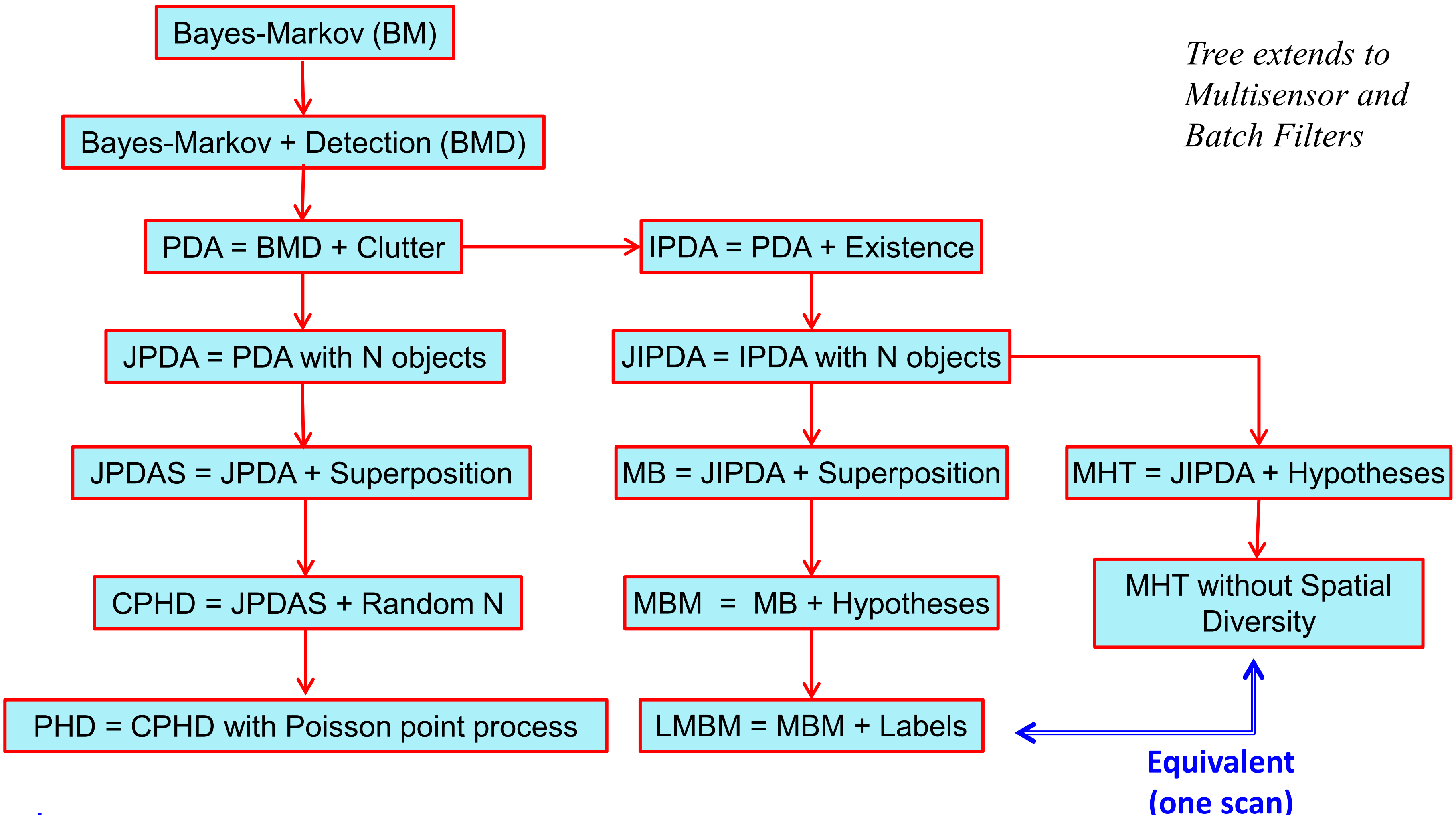
Artificial Intelligence for Military Multiple Sensor Fusion Engines

NATO Research Lecture Series SET-290, 2022
Rome 26-27 Sep ; Wachtberg 29-30 Sept; Budapest 03-04 Oct

NP-hard Likelihood Functions

- Likelihood functions are often combinatorial
  - E.g., they are sums over feasible assignments
  - Too many assignments → NP-hard
- NP-hard likelihoods are roadblocks at scale
  - Approximations are inevitable
- Mitigate complexity by using saddle point approximation
  - Stationary phase, “stationary action”
  - It can be exceedingly fast and accurate
  - It is 100% free of all enumerations
- Useful in high level information fusion

AC Taxonomy of Tracking Filters*



* *Analytic Combinatorics for Multiple Object Tracking*, by R. Streit, R. Angle, and M. Efe, Springer, 2021

Examples

- JPDA with Superposition
  - Fast, not NP-hard
- Unresolved targets
  - Crossing and parallel tracks
- Multiple target tracking (JiFi)
  - Single sensor
  - Multiple sensor, bearings-only example

JPDAS and PHD Intensity

Target state superposition

Secular Function for JPDAS

- JPDA with N non-identical targets and false alarms

$$\Psi_{\text{JPDA}}(h_1, \ldots, h_n, g) = \left(\prod_{n=1}^{N} \Psi_{\text{BMD}}(h_n, g)\right) \Psi_{\text{FA}}(g)$$

- JPDAS is JPDA with N identical superposed targets

$$\Psi_{\text{JPDAS}}(h, g) = \Psi_{\text{JPDA}}(h, \ldots, h, g) = \left(\Psi_{\text{BMD}}(h, g)\right)^N \Psi_{\text{FA}}(g)$$

$\longrightarrow$ Superposition dramatically changes the PGFL

- $\longrightarrow$ Dirac delta trains for one superposed target and M measurements

$$h(x) = 1 + \alpha_1 \, \delta(x - x_1) \quad \text{and} \quad g(y) = \sum \beta_m \, \delta(y - y_m)$$

$\longrightarrow$ Secular function of GFL of exact Bayes posterior with M measurements

$$\Psi_{\text{JPDAS}}(\alpha, \beta) = \Big(Linear(\alpha) + Bilinear(\alpha, \beta)\Big)^N \times e^{Linear(\beta)}$$

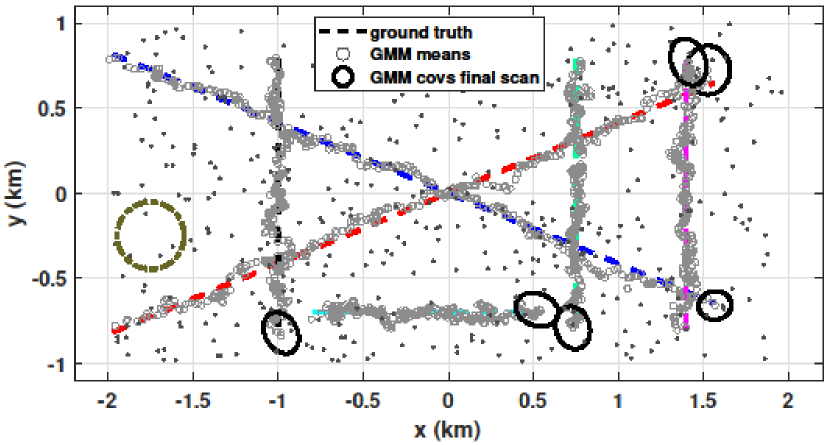
- Mixed derivative w.r.t. β gives sum of Elementary Symmetric Polynomials
  - Complexity is $O\big(MN\big)$
- First derivative of the result w.r.t. α_1 is proportional to the intensity at x_1

$\longrightarrow$ Complex step method is fast and very accurate (often attributed to Cleve Moler)

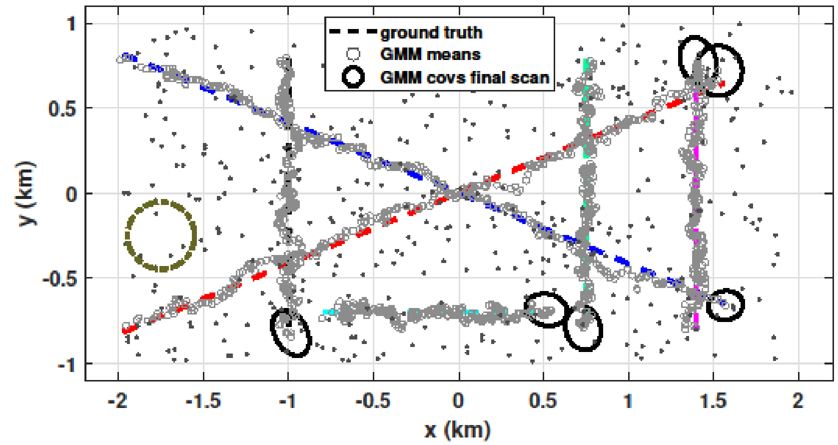
Example

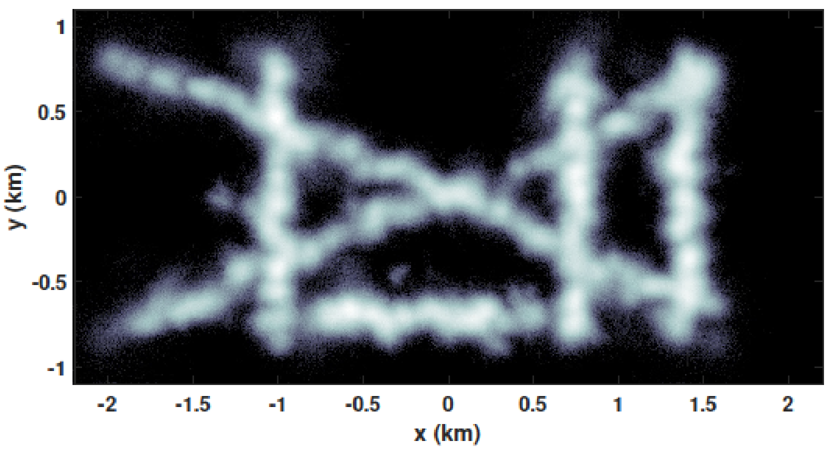
- Six nearly constant velocity targets in the plane
- Pd = 0.9 at all scans at 1 second intervals
- Linear-Gaussian 4D (pos-vel) models for convenience
- Position measurements only, equal variance in x and y
- Poisson clutter with mean of 75. Translates into 0.66 clutter points per 3 sigma_measurement radius circle (on average)
- No gating
- Particle filter implementations with 100,000 particles at each scan
- Filters implemented -- Same data used for both
  - Standard PHD
  - JPDAS
- States estimated using GMM Matlab function (R2017b)
  - No effort was made to extract target tracks
- Heat map of particles accumulated over all 240 scans

JPDAS

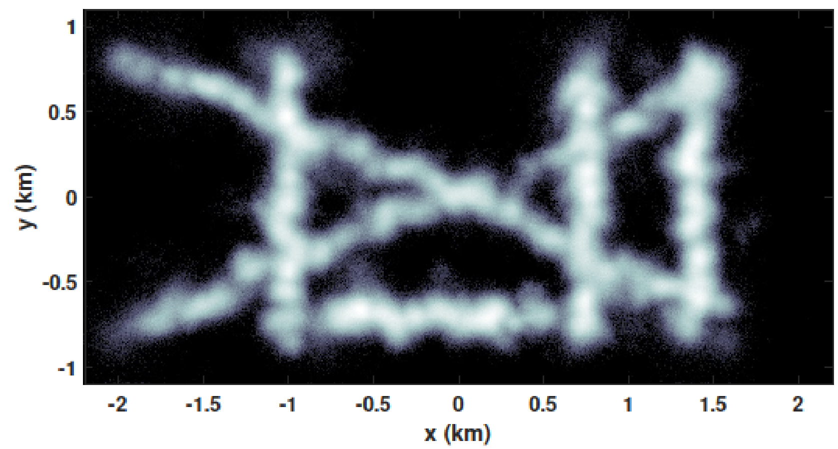


PHD





Heat maps



Superposition is stronger influence on intensity than the PMF of object number
Unexpected result since JPDAS is a special case of CPHD

Unresolved Targets/ Merged Measurements

Ignoring the mismatch between the tracker the nature of the point measurements has consequences

GF for Two Unresolved Targets

- If resolvable, each target generates at most one measurement
- If unresolvable, together they generate at most one merged measurement
- Changes the measurement generating function but nothing else!
- Given $r(x_1, x_2) = \Pr\{\text{targets at } x_1 \text{ and } x_2 \text{ are resolved}\}$
- By total probability theorem, the measurement GF is the probabilistic mixture

$$\Psi(g|x_1,x_2) = \underbrace{r(x_1,x_2)\,\Psi^{\text{Res}}(g|x_1,x_2)}_{\text{Objects Resolved}} + \underbrace{(1-r(x_1,x_2))\,\Psi^{\text{UnRes}}(g|x_1,x_2)}_{\text{Objects } NOT \text{ Resolved}}$$

where

$$\Psi^{\text{Res}}(g|x_1,x_2) = \left(1 - Pd^1(x_1) + Pd^1(x_1)\int_Y g(y)p^1(y|x_1)dy\right) \times \begin{pmatrix}\text{same form}\\ \text{for target \#2}\end{pmatrix}$$

$$\Psi^{\text{UnRes}}(g|x_1,x_2) = 1 - Pd^{\text{UnRes}}(x_1,x_2) + Pd^{\text{UnRes}}(x_1,x_2)\int_Y g(y)p^{\text{UnRes}}(y|x_1,x_2)\,dy$$

$$Pd^{\text{UnRes}}(x_1,x_2) = \Pr\{\text{unresolved targets at } x_1 \text{ and } x_2 \text{ generate one merged measurement}\}$$

May incorporate models of target strength (radar cross-section)

Examples

- JPDA with two targets
- Target and measurement models much like in previous example
  - Nearly constant speed
  - Number of clutter points per 3 sigma_meas circles is 0.24
- One target is 10 dB "stronger" than the "weaker" one

$$r(x_1, x_2) = 1 - \exp\left(-\tfrac{1}{2}\left\|H(x_1 - x_2)\right\|\right) = \Pr\left\{\text{targets at } x_1 \text{ and } x_2 \text{ are resolved}\right\}$$

$$p^{\text{UnRes}}(y \mid x_1, x_2) = N\left(y \,\middle|\, H(\underbrace{w x_1 + (1-w) x_2}),\ R^{\text{UnRes}}\right)$$

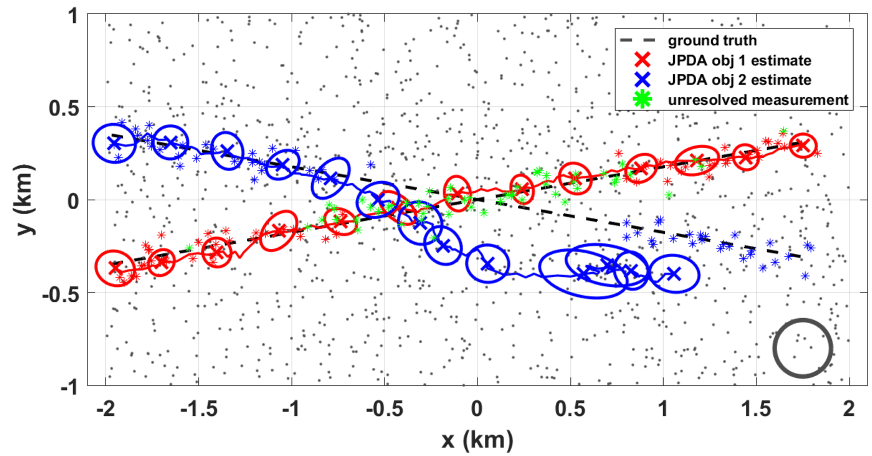
Strong | Weak

$$10 \log\left(\frac{w}{1-w}\right) = 10\,dB$$

- Two examples
  - Crossing targets
  - Parallel targets

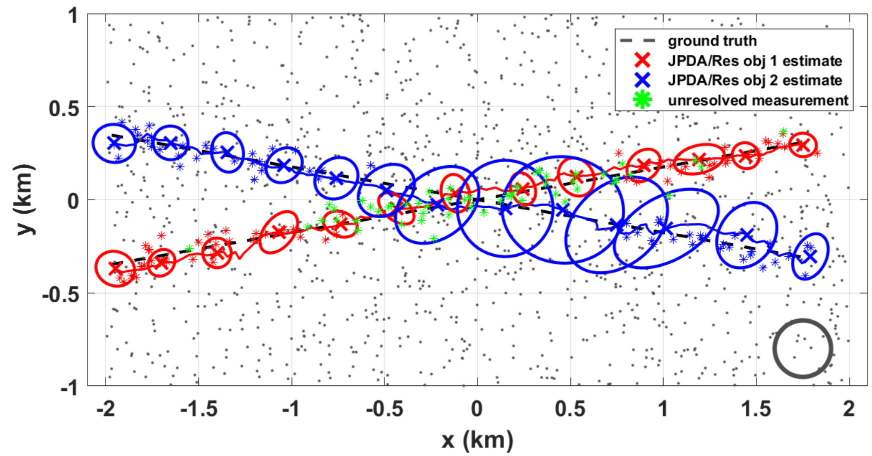
Strong and Weak Crossing Targets

JPDA filter
<u>without</u>
unresolved
measurement model



Loses weak target track almost immediately. Anticipated behavior.

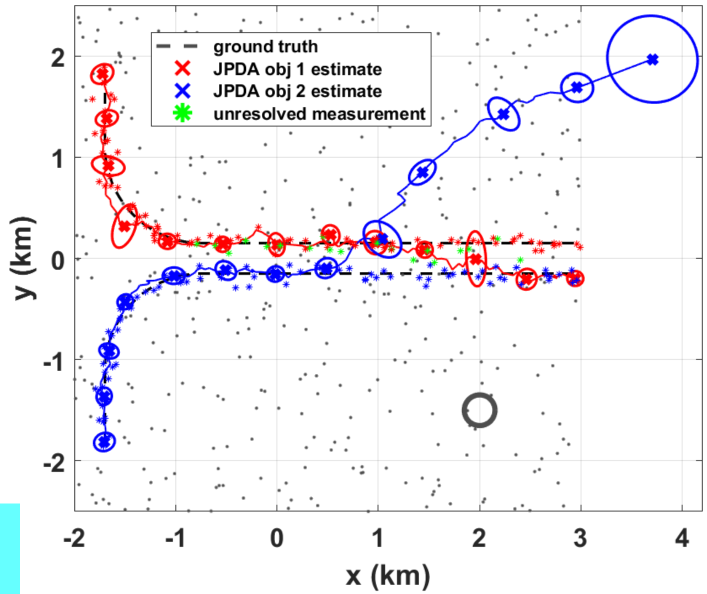
JPDA filter
<u>with</u>
unresolved
measurement model



Maintains both tracks. Inflates variance on weak target near the cross, but weak target is not “seduced” by the strong one.

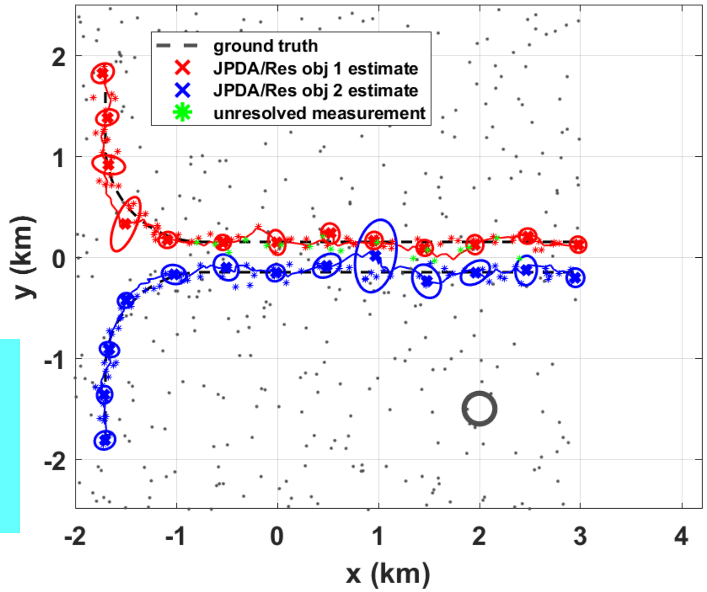
Strong and Weak Parallel Targets

JPDA filter
without
unresolved
measurement model



Weak target track, once seduced by the strong target, is lost. Anticipated behavior.

JPDA filter
with
unresolved
measurement model



Maintains both tracks. Inflates variance on weak target, but weak target is not "seduced" by the strong one.

General Problem of Unresolved Targets

- Suppose there are N targets
- Any combination of them could be unresolved
  - Each needs a Pr{unresolved} function
  - Each needs a GFL
- The overall GFL is a sum of 2^N GFLs
  - This GFL itself is NP-hard
  - The two-target approach is intractable in general
- Alternative
  - Modify Drummond's measurement peak picking rule (~1965)
    - "at most one measurement per target"
  - Allow each sensor report to have "multiplicity"
  - Changes the combinatorial problem
  - The resulting GFL is tractable

JPDA-PHD/Intensity Filter

JiFi (Joint iFilter)

255 / 221 / 0

"JPDA Intensity Filter for Tracking Multiple Extended Objects in Clutter," R. Streit, ISIF FUSION Conference, Heidelberg, Germany, July 2016

PHD Intensity Filter

- PHD = Intensity = $f_{k-1|k-1}(x) = E\left[N(x) \,\middle|\, \text{all measurements up to and including time } t_{k-1}\right]$

  $\cong$ Poisson approximation used to close the Bayesian recursion at each step

- Predicted target intensity at x

  $$f_{k|k-1}(x) = E[\text{New targets}] + E[\text{Old targets (thinned by death and moved to new states)}]$$

- Information update at x

$$f_{k|k}(x) = \underbrace{\left(1-P_D(x)\right) f_{k|k-1}(x)}_{\text{Intensity of Undetected targets}} + \sum_{j=1}^{m} \frac{\text{Expected number of targets @ } x \text{ with measurement } z_j}{\text{Expected number of all targets \textbf{\textit{and}} FA with measurement } z_j}$$

$$E\left[\mathrm{N}\{\text{targets @ } x \text{ gave rise to the } j\text{-th measurement, } z_j\}\right]$$

(Mahler, 2003)

$$= \frac{p_k(z_j \mid x)\, P_k^D(x)\, f_{k|k-1}(x)}{\underbrace{\lambda_k^{Prior}(z_j)}_{\text{FA Intensity}} + \underbrace{\int_S p_k(z_j \mid s)\, P_k^D(s)\, f_{k|k-1}(s)\, ds}_{\text{measurement Intensity from all targets}}}$$

Interpretation in Stone, Streit, Corwin, Bell, Bayesian Multiple Target Tracking, 2014, p. 179

JPDA Intensity Filter* (JiFi)

- JPDA: n is number of target Groups $\Rightarrow$ no birth or death of Groups
- Given intensity functions *for every target Group*

$$f^{\mathbf{i}}_{k-1|k-1}(x) \cong E\left[N(x) \text{ for target group } \mathbf{i} \mid \text{all measurement up to and including time } t_{k-1} \right]$$

- Predicted intensity *for every target group*

$$f^{\mathbf{i}}_{k|k-1}(x) = E[\text{New targets in group } \mathbf{i}] + E[\text{Old targets in group } \mathbf{i} \text{ (thinned by death and moved to new states)}]$$

- Bayesian information update *for every target group*

$$f^{\mathbf{i}}_{k|k}(x) = \underbrace{\left(1 - P^{\mathbf{i}}_{D}(x)\right) f^{\mathbf{i}}_{k|k-1}(x)}_{\text{Intensity of undetected targets in group } \mathbf{i}} + \sum_{j=1}^{m} \frac{\text{Expected number of targets @ } x \text{ in group } \mathbf{i} \text{ with measurements } z_j}{\text{Expected number of } all \text{ targets in } all \text{ groups and FA with measurement } z_j}$$

$$\Pr\{\text{a target in group } \mathbf{i} \text{ at } x \text{ gave rise to the } j\text{-th measurement, } z_j\}$$

$$= \frac{p^{\mathbf{i}}_k(z_j \mid x)\, P^{D\,\mathbf{i}}_k(x)\, f^{\mathbf{i}}_{k|k-1}(x)}{\underbrace{\lambda^{Prior}_k(z_j)}_{\text{FA Intensity}} + \sum_{\text{all groups}} \underbrace{\int_{S_{\mathbf{i}}} p^{\mathbf{i}}_k(z_j \mid s)\, P^{D\,\mathbf{i}}_k(s)\, f^{\mathbf{i}}_{k|k-1}(s)\, ds}_{\text{measurement Intensity from targets in group } \mathbf{i}}}$$

*R. Streit, "JPDA intensity filter for tracking multiple extended objects in clutter," ISIF Fusion Conf., Heidelberg, July 2018

Single Sensor JiFi

- *N* = specified number of objects — JPDA
- Heterogeneous objects
  - Each has its own state space
  - Each has an unknown number of highlights
  - Highlights are volatile – use a PHD intensity filter
- One sensor
  - Measurements are of object highlights or clutter points
  - Assignment problem — highlight-to-object
- Analytic Combinatorics (AC) uses generating functions (GFs)
  - GF for JiFi

$$\Psi_{\mathrm{JiFi}}(h_{1:N}, g) = \Psi_{Clutter}(g) \prod_{n=1}^{N} \Psi_{\mathrm{PHD}\ Object-Highight}(h_n, g)$$

Different object state spaces

Same highlight space

Multiple Sensor JiFi*

- *L* = number of heterogeneous sensors
  - Different measurement spaces
  - Different (independent) clutter processes
- Spatial diversity is important assumption
  - Different sensors see different object highlights
  - Sensor processes are statistically independent
- GF for Multiple Sensor JiFi

$$\Psi_{\mathrm{MS/JiFi}}(h_{1:N}, g_{1:L}) = \prod_{\ell=1}^{L} \Psi_{\mathrm{JiFi}}(h_{1:N}, g_\ell)$$

$$= \prod_{\ell=1}^{L} \left\{ \Psi_{Clutter}(g_\ell) \prod_{n=1}^{N} \Psi_{\mathrm{PHD}\ Object-Highight}(h_n, g_\ell) \right\}$$

Different object state spaces

Different highlight spaces

*Angle and Streit, “Multisensor JiFi tracking of extended objects,” ISIF Fusion Conf, July 2019

Deriving the Multiple Sensor JiFi

- Different measurements from each sensor

$$\left\{ y_{\left(\ell,\, j_\ell\right)} :\ j_\ell = 1, \ldots, m_\ell \right\} \qquad \ell = 1, \ldots, L$$

- Substitute
  - For each sensor: weighted train of Dirac deltas at the measurements

$$g_\ell(y) = \sum_{j_\ell=1}^{m_\ell} \beta_{\left(\ell, j_\ell\right)}\ \delta_{\mathrm{Dirac}}\left(y - y_{\left(\ell, j_\ell\right)} \right)$$

  - For each object:

$$h_n(x) = 1 + \alpha_n \delta_{\mathrm{Dirac}}\left(x - x_n \right)$$

- Result is the “secular function”
  - Ordinary multivariate analytic function of the weights α and β
  - Bayesian posterior intensity function
  - → Logarithmic derivative of the secular function
  - “Multisensor JiFi Tracking of Extended Objects,” Angle and Streit, 2019 ISIF Fusion Conference, Ottawa, 2019

Multisensor JiFi Recursion

- Intensity functions at the previous scan

  for each object: $f_n(x_n) \quad n = 1, \ldots, N$
- Predict object intensity functions at current scan

  for each object: $f_n^+(x_n)$
- Update intensity function for every object-sensor **pair**

  $f_{n\ell}(x_n) =$ PHD filter with sensor ℓ measurements

  -- probability of detection Pd

  -- measurement likelihood function

  -- predicted intensity function $f_n^+(x_n)$
- Bayesian update of intensity function for each object

  $$f_n(x_n) = \sum_{\ell=1}^{L} f_{n\ell}(x_n)$$
- Closes the Bayesian recursion

Single Sensor JiFi – Two Group Example

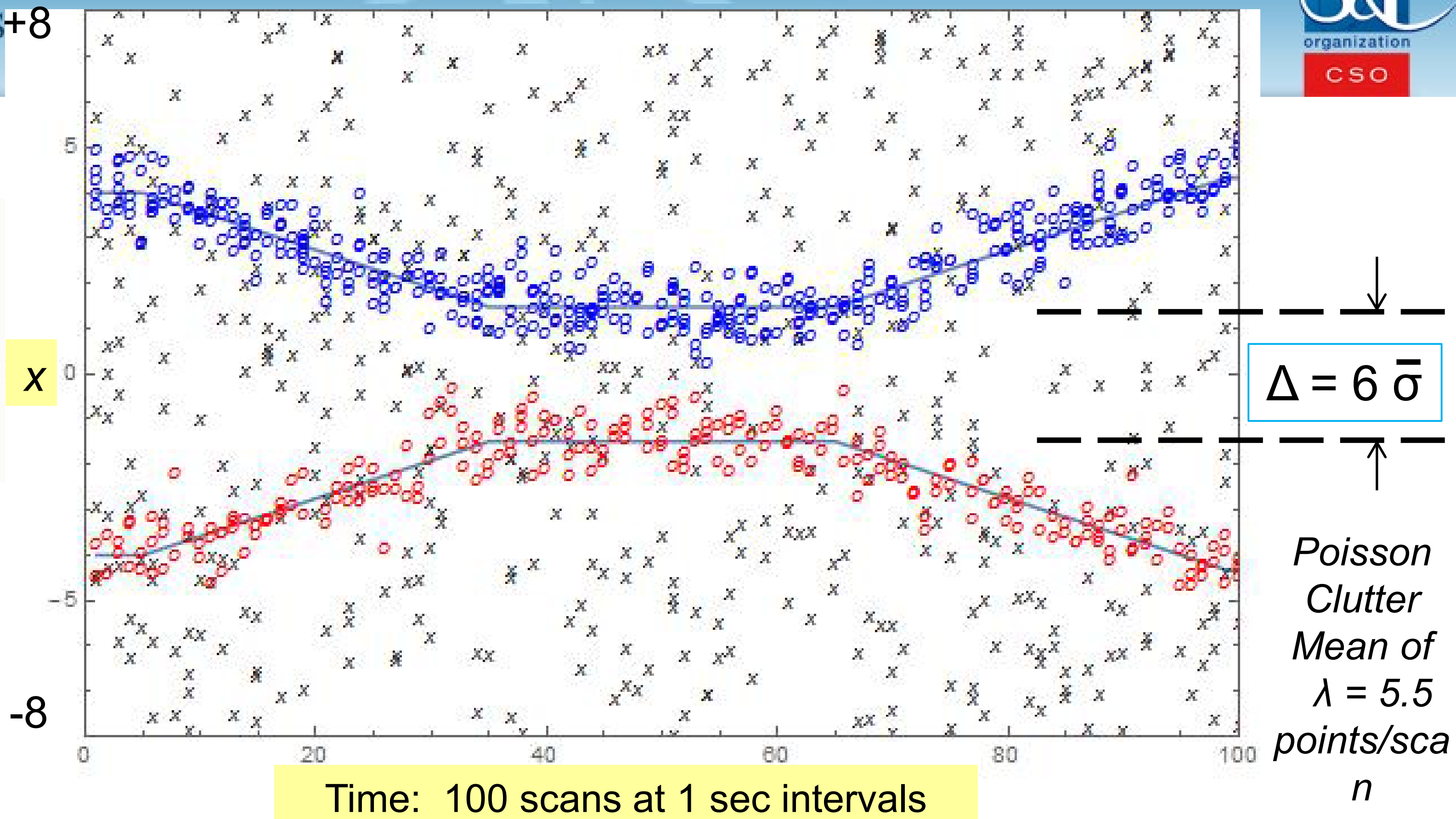
- Simulated target groups in 1-D
  - State space is position only on the interval [– 8, 8]
- Each group has a different maximum number of highlights
  - Number of detected highlights per group is binomially distributed
  - Individual highlights are Gaussian distributed about the group center
  - The groups have different spreads
- Nearly constant motion

  $p(x_k \mid x_{k\text{-}1}) = \text{Gaussian}(x_k \mid x_{k\text{-}1}, \text{process noise})$
- Measurements z are of individual target highlights
  - Measurements are i.i.d. conditioned on target position

    $p(z \mid x_k) = \text{Gaussian}(z \mid x_k, \text{measurement noise})$

- Clutter is uniform Poisson point process on [– 8, 8]
- Filters are implemented on a fine 1-D grid
  - No Monte Carlo particle approximations, Gaussian mixtures, etc

Well Separated Groups

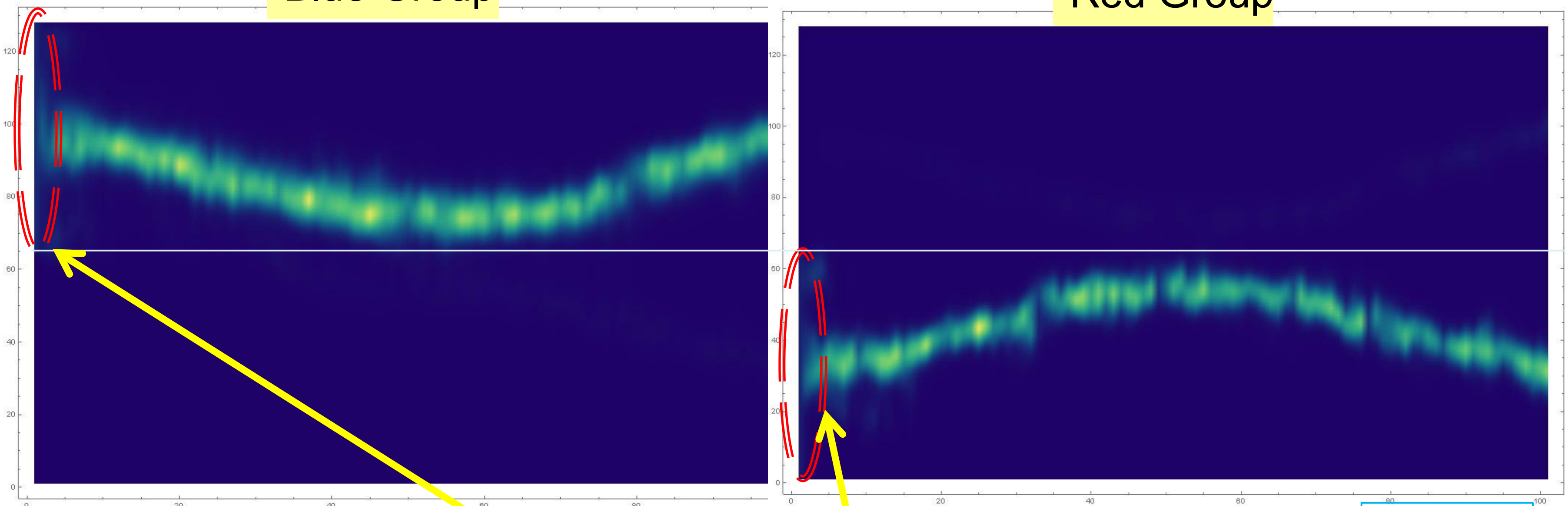


	Blue Target	Red Target
Group Extent, σ	0.5	0.5
Maximum number of targets in group	8	3
Pd of individual targets	0.5	0.8
Target process noise, σ	0.2	0.2
Measurement noise, σ	0.5	0.5

JiFi Outputs

Blue Group

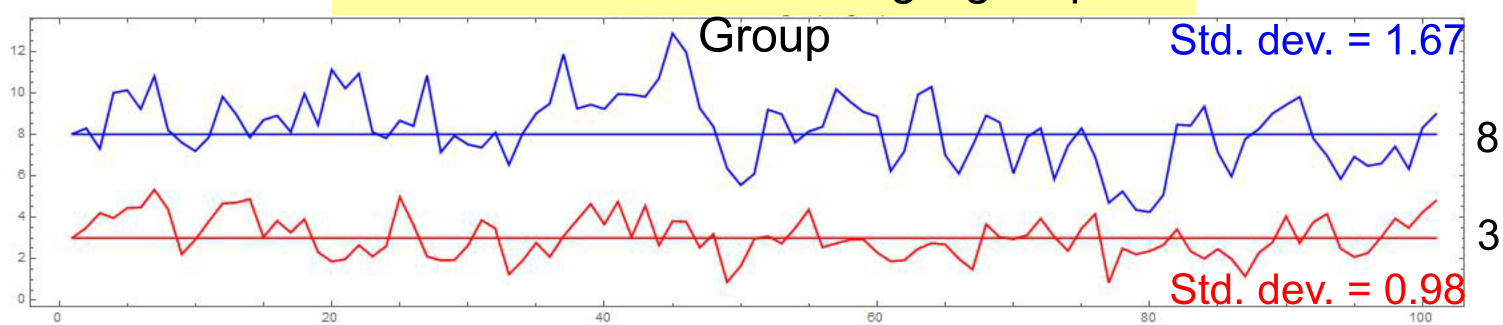
Red Group



Initializations are diffuse: Blue on [0, 4] and Red on [-4, 0]

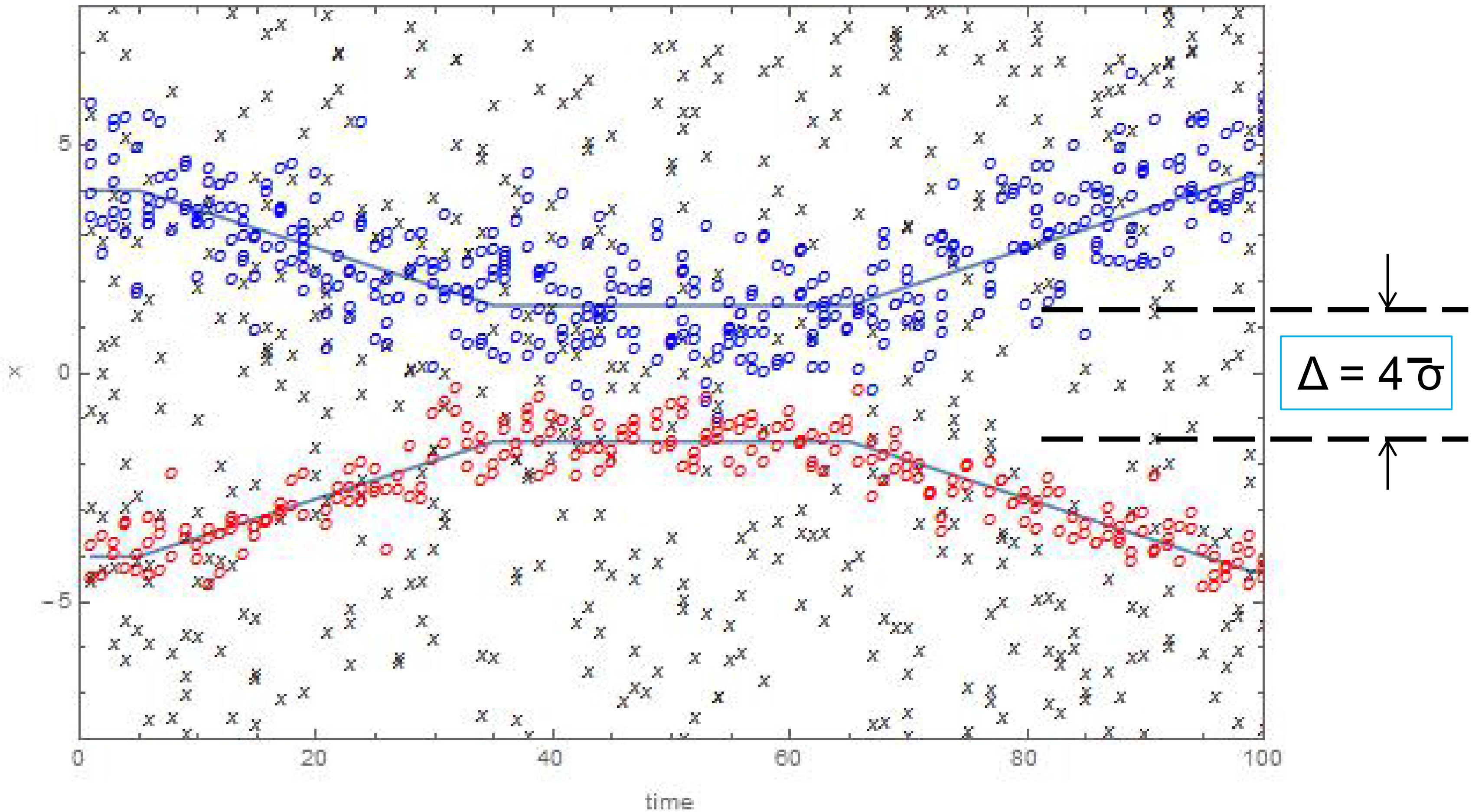
Δ = 6σ̄

Estimated number of Highlights per Group

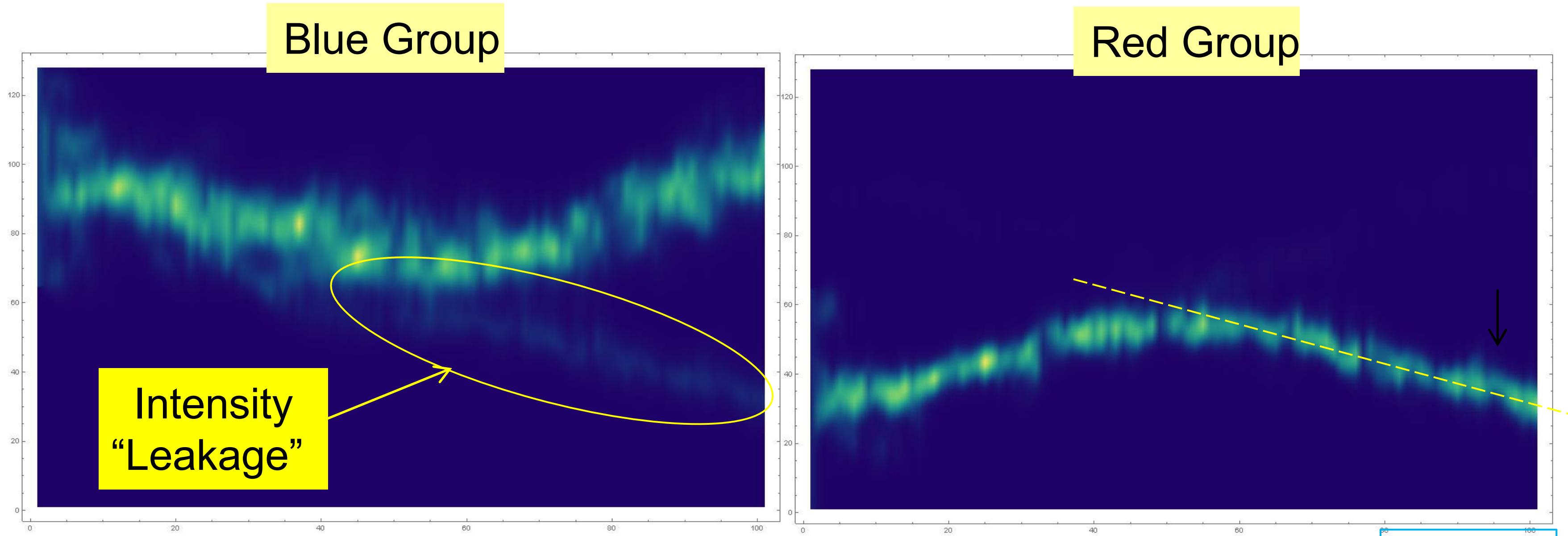


Keep everything the same but make the Blue group twice as "Wide"

Double the Spread of the Blue Group

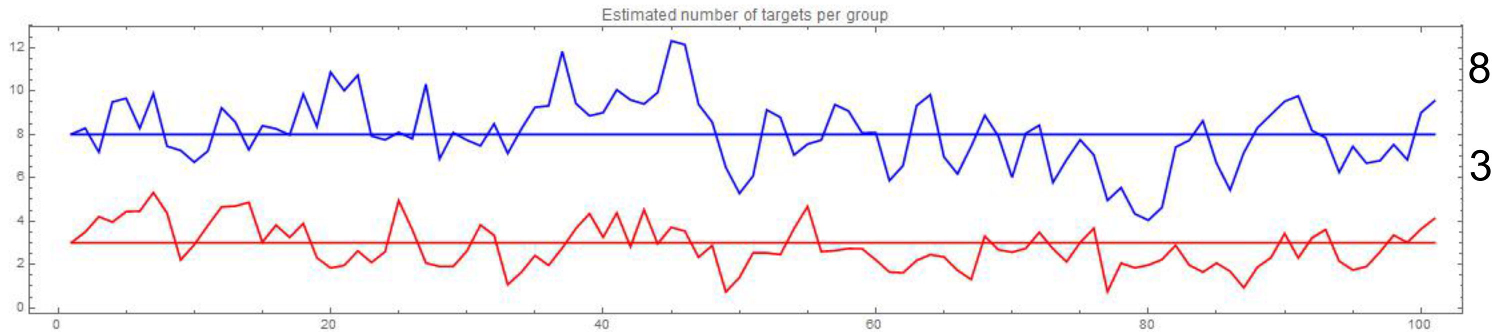


Double Blue Target Spread – Intensity Leakage

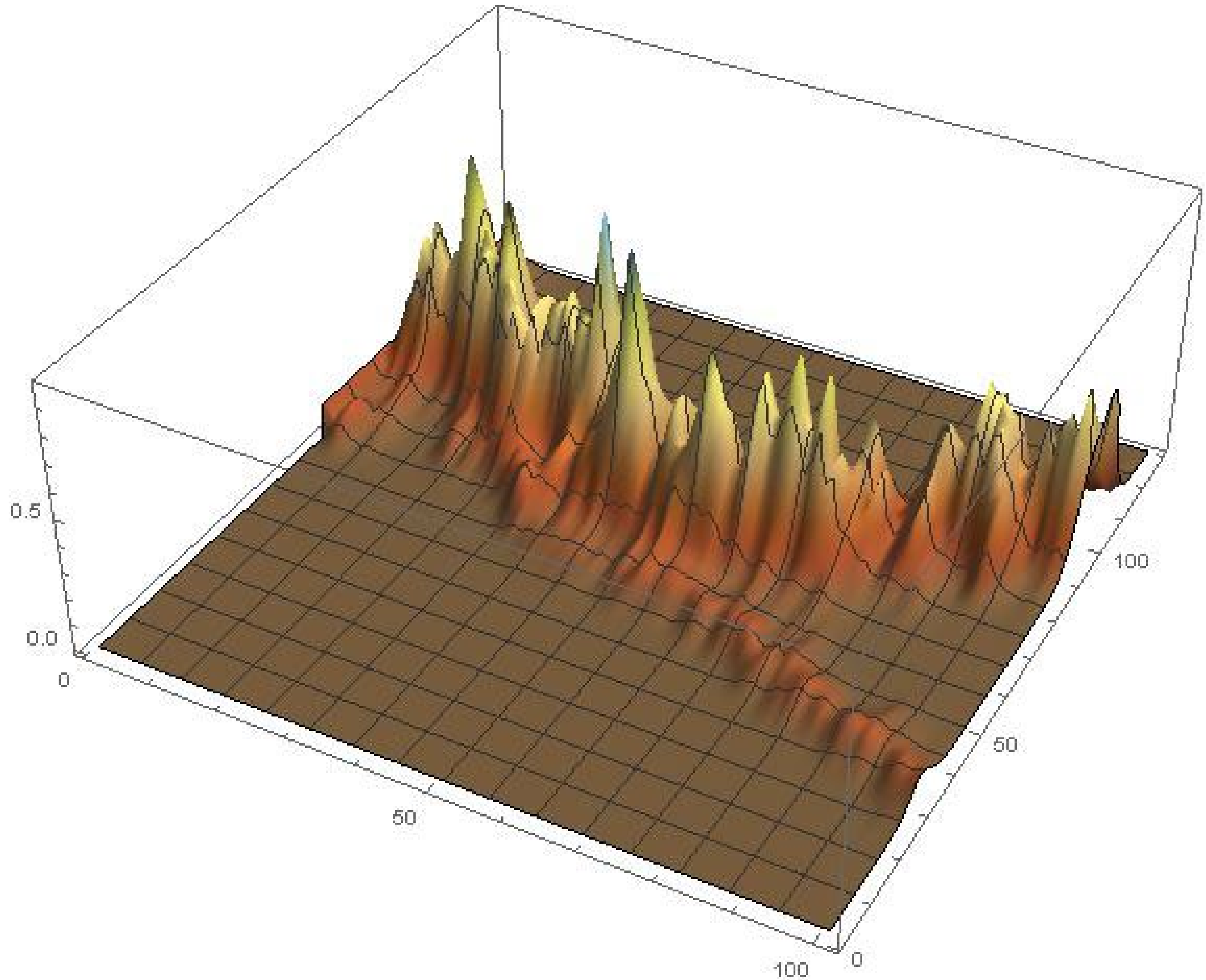


Estimated number of Highlights per Group

$\Delta = 4\bar{\sigma}$

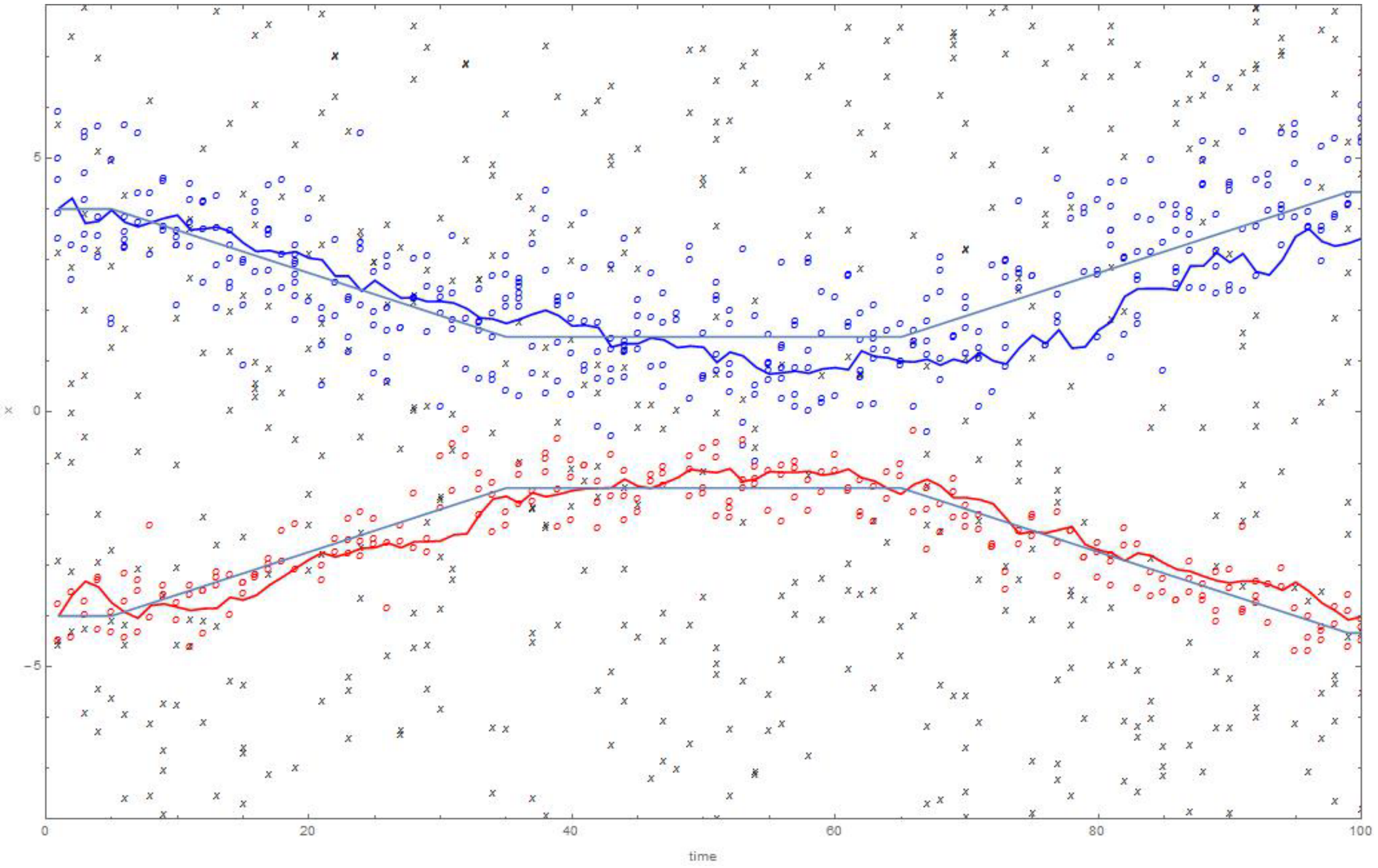


Intensity Leakage – 3-D Plot



Target-Specific Centroids

- Compute the weighted centroid of each target's intensity
- Plot centroids as function of time
- Leakage causes tracking bias



Multiple Sensor JiFi Examples

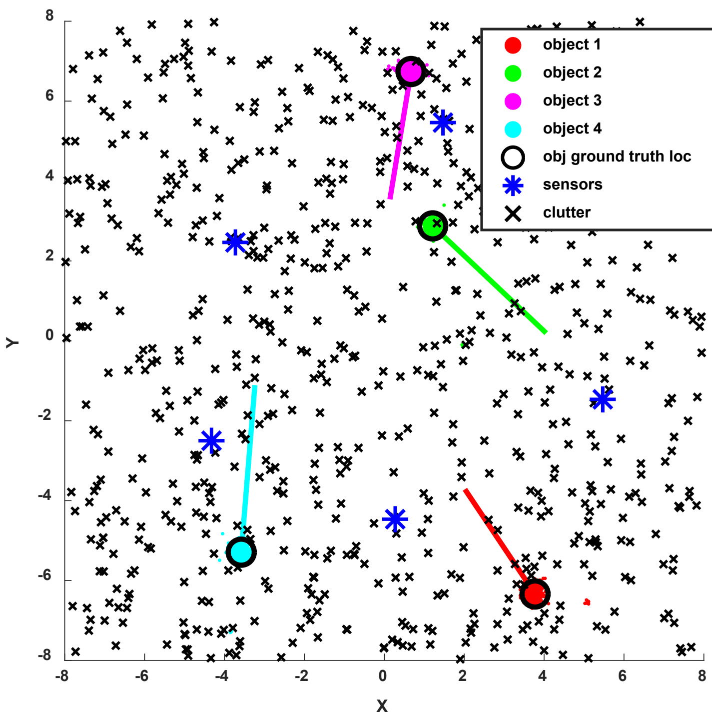
- Five sensors
  - Example 1: Position-only measurements x-y
  - Example 2: Bearings-only measurements
- Four objects – nearly constant velocity in x-y plane
- Simulated object highlights
  - Each object has 8 potentially detectable highlights per sensor, except one which has 5
    - Three objects have 5x8=40 highlights total and one has 5x5=25
  - Highlight Pd = 0.5 for each sensor
  - IID Gaussian distributed about the object's "point of interest"
  - Highlights are resampled at each scan – do not persist scan to scan

- Particle filter with 10,000 particles (display random subset of 1000)
  - Number of scans = 100
  - Initiate particles for each object
    - Uniformly distributed over "large" object-specific region
    - Little overlap in the regions
    - Box-shaped region in 1st example; Star-shaped in the 2nd (details in Fusion 2019 paper)

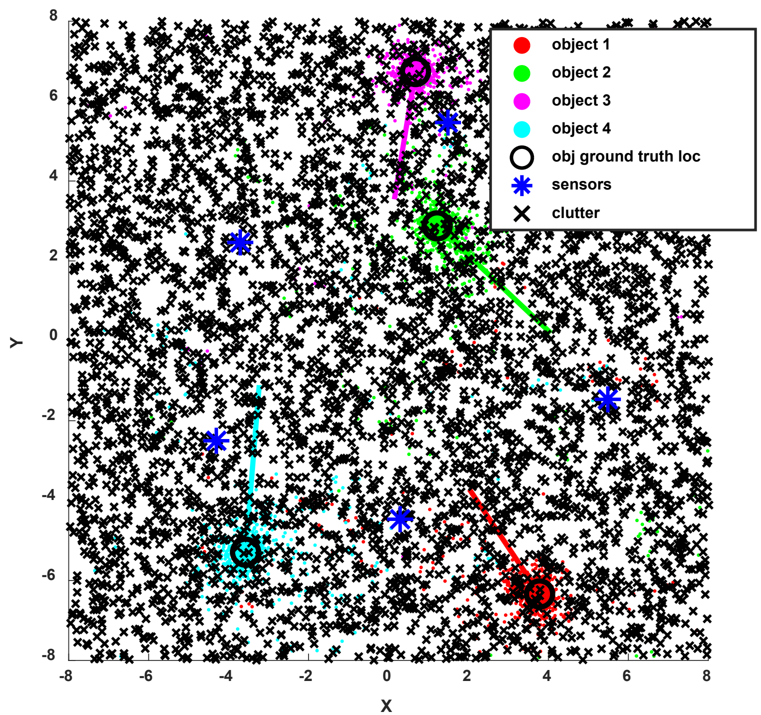
Example 1 – Position-Only

- Use different colors for each object's particles
- Region of regard is R = [-8,+8] x [-8,+8]
- FA is IID uniformly distributed over R at each scan

150 FA points per sensor per scan



1000 FA points per sensor per scan

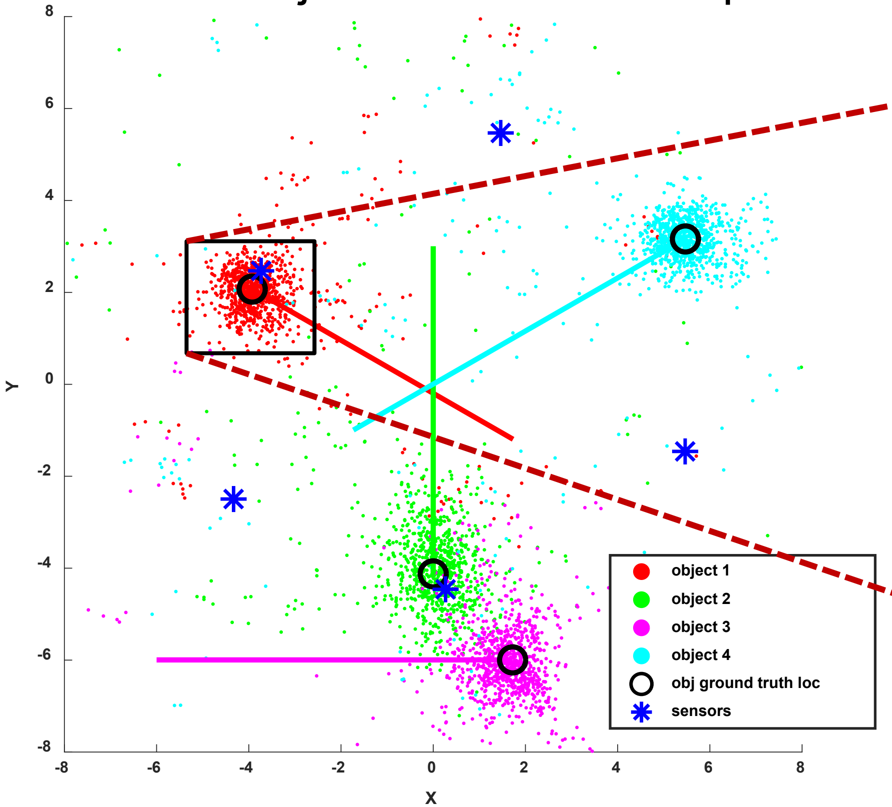


Object particles (COLOR) are less dispersed for FA = **150** compared to FA = **1000**

Example 2 – Bearings-Only

- Bearing clutter is NOT depicted

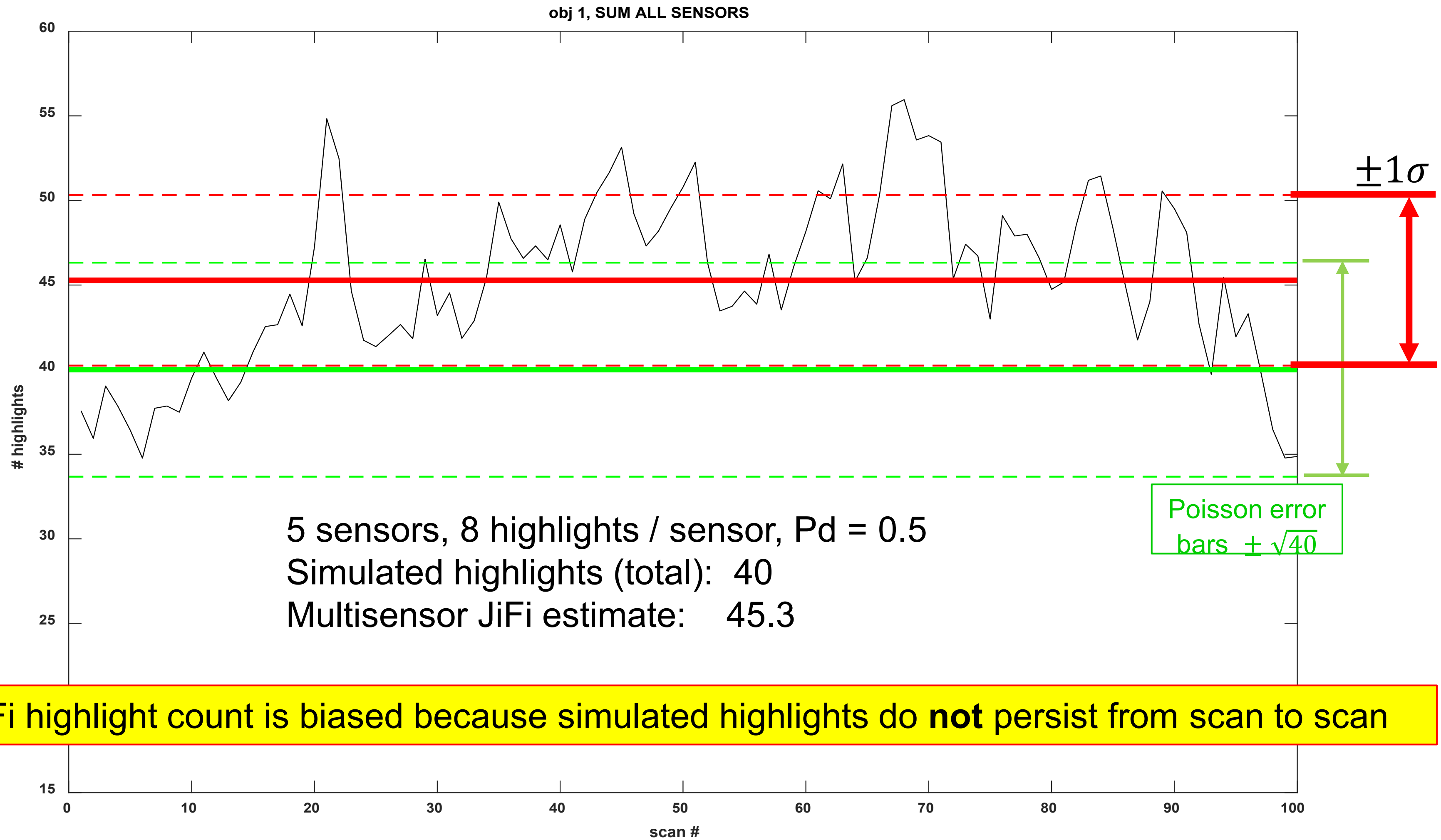
100 FA points per sensor per scan
Different trajectories from Example 1



Leakage – object particles become entangled due to earlier close encounters

Leakage: Look close to see mix
5 green, 6 purple, 8 blue

Estimated Total Highlight Count for Object #1: Example 2



JiFi highlight count is biased because simulated highlights do **not** persist from scan to scan

Multiple Sensor JiFi

- Multisensor JiFi is low computation complexity
- Makes few assumptions about object structure
- Works with sensors with low observability
  - Bearings-only sensors
- Spatial diversity assumption
  - Important – if the goal is to estimate highlights
  - Less important – if the goal is to track the objects
- Spatial leakage
  - Occurs when object tracks are too close for too long
  - Analogous to leakage in time-frequency analysis

Applications of Saddle Point

- Intensity is analogous to power spectrum
  - Whitening the intensity function for known targets
- Notched filters and the pair correlation function

  In the PHD intensity filter, conditioned on a target at x_1,

$$PHD(x \mid x_1) = \underbrace{\left[1 - \frac{C(x_1, x_2)}{PHD(x_1) PHD(x_2)} \right]}_{\textit{Pair correlation function}} PHD(x)$$

  - Pair correlation too difficult to compute for other filters, but can be approximated by saddle point methods
- Batch processing over K scans of data
  - Track before detect strategy
  - GFL is a K-deep "exponential tower"
    - not amenable to symbolic methods
  - Can be approximated by saddle point methods

Concluding Remarks

- Analytic Combinatorics (AC) and saddle points
  - A story about generating functions
  - Solving NP-hard problems with easy saddle point approximations
  - Exact solutions when sensor and other models are imperfect
- Benefits of AC to tracking
  - Unified methodology organized by AC
    - Classical Bayes-Markov, PDA, JPDA, CPHD, PHD
    - JIPDA, MB, MBM, LMBM, and MHT
  - New hybrid filters -- JiFi and SuperJiFi
  - New ways to formulate known problems (e.g., unresolved targets)
  - New and classical approximations
- AC is a Bridge to High Level Information Fusion
  - Integer linear programming
  - Natural language processing
  - Approximate common subgraph