BRIAN BECKMAN. 6 NOV 2017

KALMAN FOLDING

GOALS

- Show Kalman Filtering to SDEs
- Show Functional Programming to Scientists

- Kalman Folding is a novel synthesis of the two

KALMAN FILTERING

- TRACKING: estimating dynamic states from observations
- SYSTEM IDENTIFICATION: estimating constants from observations
- SENSOR FUSION: combine multiple sources of observation

- ∃ many variations: linear vs. non-linear, Gaussian vs. non-Gaussian
- It's regression, a fundamental instance of machine learning

FUNCTIONAL PROGRAMMING

- fold Separates Data Processing from Data Delivery
- Makes programming more like mathematics

AXIOMS FOR THIS TALK

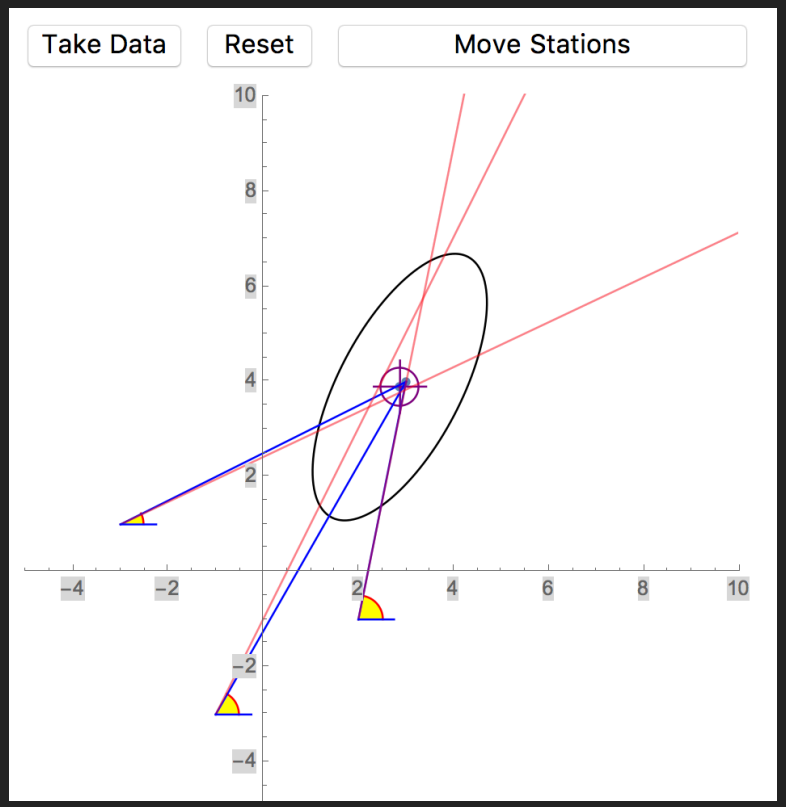
- Assumed without debate:
  - Functional Programming is good
  - Mathematica is good
  - Python is better
  - Kalman filtering is good

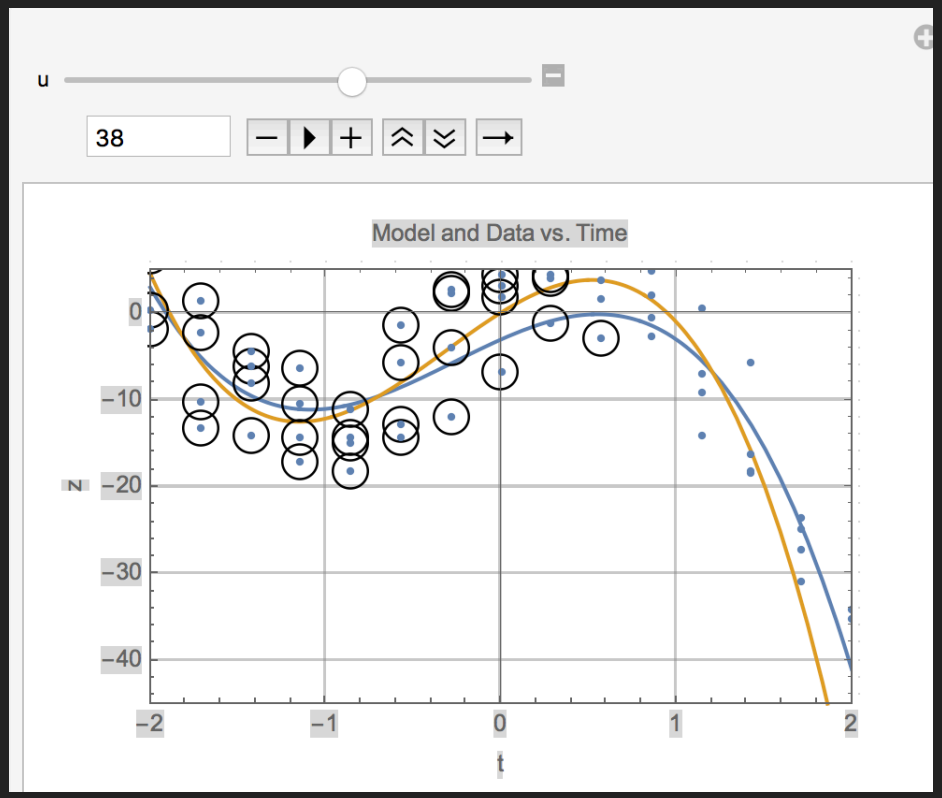
LINEAGE, ORIGINS, PROVENANCE

	Batch	iteractive, reactive	functions
iterator		foreach subscribe	z = observation
L_0	count = N	count	$\{z0, z\} \mapsto z0 + z^0$
L_1	mean = $\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$	sum	$\{z1, z\} \mapsto z1 + z^1$
		mean	$\{(z0, z1), z\} \mapsto \frac{z1+z^1}{z0+z^0}$

L_2	ssr = $\sum (x_i - \bar{x})^2$	sum sqrs	$\{z2, z\} \mapsto z2 + z^2$
		naive ssr	$\{(z0, z1, z2), z\} \mapsto \left(z2 + z^2\right) - \frac{\left(z1+z^1\right)^2}{z0+z^0}$
		Welford's	$\{(z0, z1, ssr), z\} \mapsto$ 0 if (z0 == 0) else $ssr + \left(z - \frac{z1}{z0}\right)\left(z - \frac{z1+z^1}{z0+z^0}\right)$
fit	least-squares regression	Kalman Potter SRIF EKF UKF ΣPF ...	$\{(x, P), (z, A)\} \mapsto \left(K.(z - A.x), P - K.D.K^\intercal\right)$ (new x, new P), where $K = P.A^\intercal.D^{-1}$ $D = \left(I + A.P.A^\intercal\right)$

APPLICATIONS: TRACKING (STATE) SYSTEM IDENTIFICATION (PARAMETERS)





MOTIVATING EXAMPLE

- States and Parameters of a spinning dashpot

TAKEAWAYS

- Lagrangian = Kinetic Energy – Potential Energy
- State-Space Form = 1st –order representation
- Stiff System: time step is too large
- Fold updates quantities one obsn at a time
- Foldable = fn (state, obsn) ~~> state
- Fold abstracts data processing (the foldable) from data delivery (the sequence of obsns)
- Kalman (states ↦ obsns) = obsns ↦ states
- Propagator = Euler integration for linearized systems
- Obsn Model: predicts obsns from states
- Data Fusion combines multiple obsn models in one filter
- Kalman = incremental least-squares regression
- Set low a-priori covar for unobservable states and parameters
- Set integration rate higher than filter rate