Parameterizing a Type 1 Diabetes ODE Model

Christina Catlett, Daniel Shenker, Rachel Wander, Maya Watanabe

outline

- what is the t1d model?
  - Why is this model useful
    - developing potential treatments - Rachel :)
  - what is diabetes onset - Daniel :)
  - what is the structure of the model? - Maya :)
  - what is the overall goal? - Christina :)
- what does it mean to parameterize
  - what are the parameters vs. variables - TEAM 2
  - slide about differences/similarities
    - why would we use these two methods
    - combine two methods - TEAM 1
  - 2 ways - example based explanation (less math)
    - Kalman filters - TEAM 2
    - MCMC - TEAM 1
- (Lotka Volterra?)

Type 1 diabetes

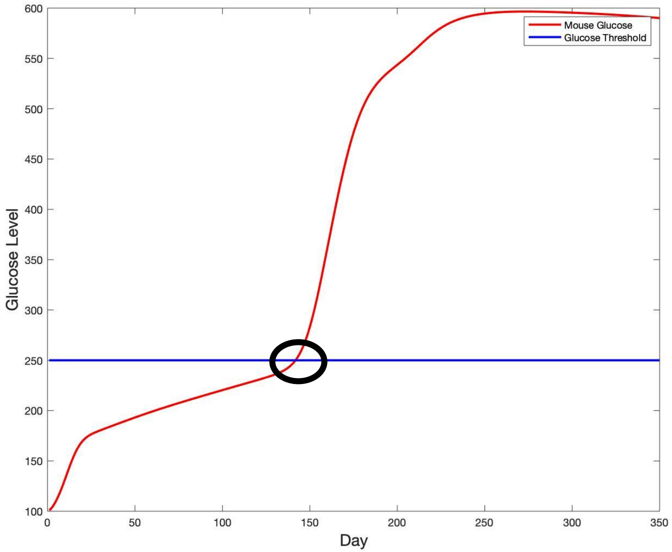
- Autoimmune disease characterized by inability to regulate blood glucose
- Can be managed, but no cure
- *Apoptotic wave* during early development cause immune response, ability to respond determines diabetes onset

Why model T1D?

- Modeling T1D helps give us a better understanding of the progression of the disease, which can be useful in determining possible treatments
- Using the model, we can simulate possible treatments and see determine whether or not we expect them to be successful
- This is especially helpful for treatments that are time sensitive

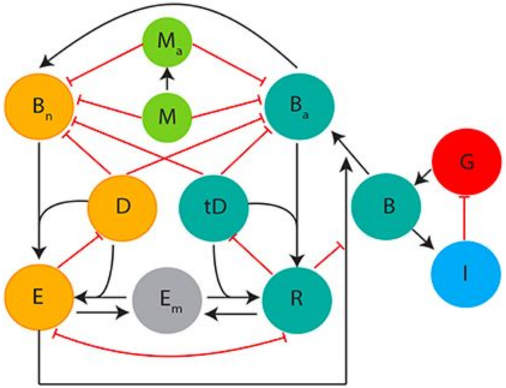
Diabetes Onset

- Onset determined by Glucose levels
  - In mice, threshold ~ 250 mg/dl
- Various theories about what causes onset
  - Model uses apoptotic wave



T1D Model

- 12 equation nonlinear ODE system
  - models immune cell interactions & relationships
- Time-dependent
- Bistable system:
  - healthy vs. disease state
  - transition determined by immune cell interactions
- Single compartment model - pancreas
- Non-obese diabetic (NOD) mice
- Glucose data - easy to collect



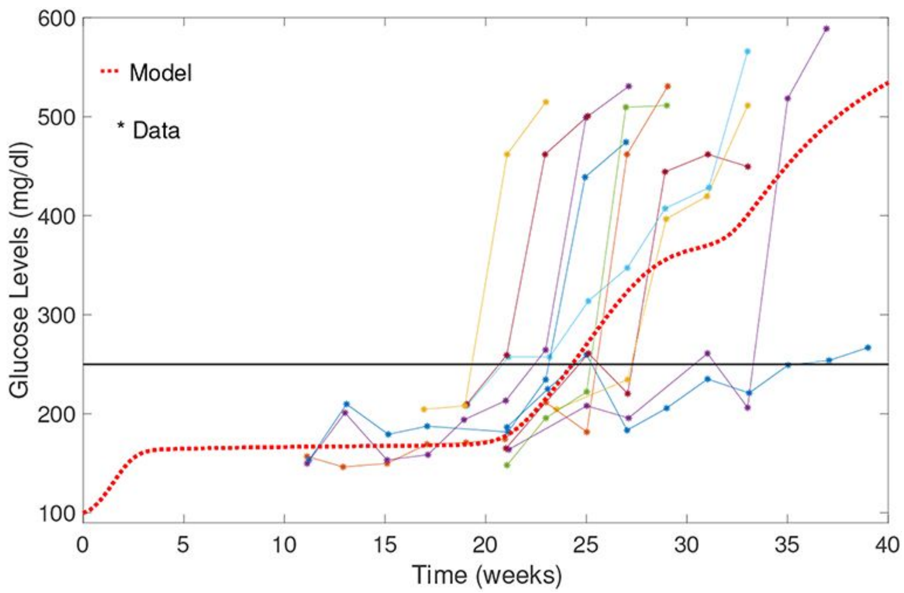
Variable	Description
B	Healthy β-cell population
Ba	Apoptotic β-cell population
Bn	Necrotic β-cell population
D	Immunogenic DC population
tD	Tolerogenic DC population
E	Effector T cell population
Em	Memory T cell population
R	Regulatory T cell population
Ma	Active macrophage population
M	Macrophage populations
G	Glucose levels
I	Insulin levels

Shtylla et al. 2019. A Mathematical Model for DC Vaccine Treatment of Type 1 Diabetes

Goal

- In paper:
  - Most parameters estimated from existing literature
- Accuracy is important
  - Parameters sit ‘on the edge’
  - Very sensitive
- Explain varying mouse behavior
  - Progressive vs. acute

Task: Use MCMC and Unscented Kalman Filters (UKF) to estimate parameters for system based on observed glucose in mice



What Does it Mean to Parametrize

- Differential equations are made up of variables and parameters
  - Variables are states within the system that change
  - Parameters are constants within the system

Example: In the differential equation

$$dx/dt = ax^2 + bx$$

x is a variable and a and b are parameters.

- Often, when you are trying to create a model for a biological phenomena, the parameters are not known
- Parameterizing is the process of trying to find the parameters that best fit a model
- Often, this process involves comparing model predictions of a phenomena to actual observations

Why Use Two Methods?

- MCMC
  - Bayesian approach, all data known at start
  - Parametrization on population level
  - Parameter distributions
- Kalman Filters
  - Adaptive, learns data point by data point
  - Parametrization on individual mouse level
  - Single parameter set
- *Potential to compare, combine methods*

Kalman Filter General Idea

- Discrete time, state space model
- Two sets of data: **latent states (x)** and **observables (y)**
  a. What I want to know versus what I do know
- Two step algorithm
  1. Propagate forward in time using knowledge of system
  2. Use observable to update estimate and system parameters
     a. "How well did we do? How can we improve for next time?"
- Tunable model set up
  - Noise, covariances

Modeling the System

Transition Matrix

Latent States

Process Noise

$$x_{k+1} = F_{k+1,k}x_k + w_k$$

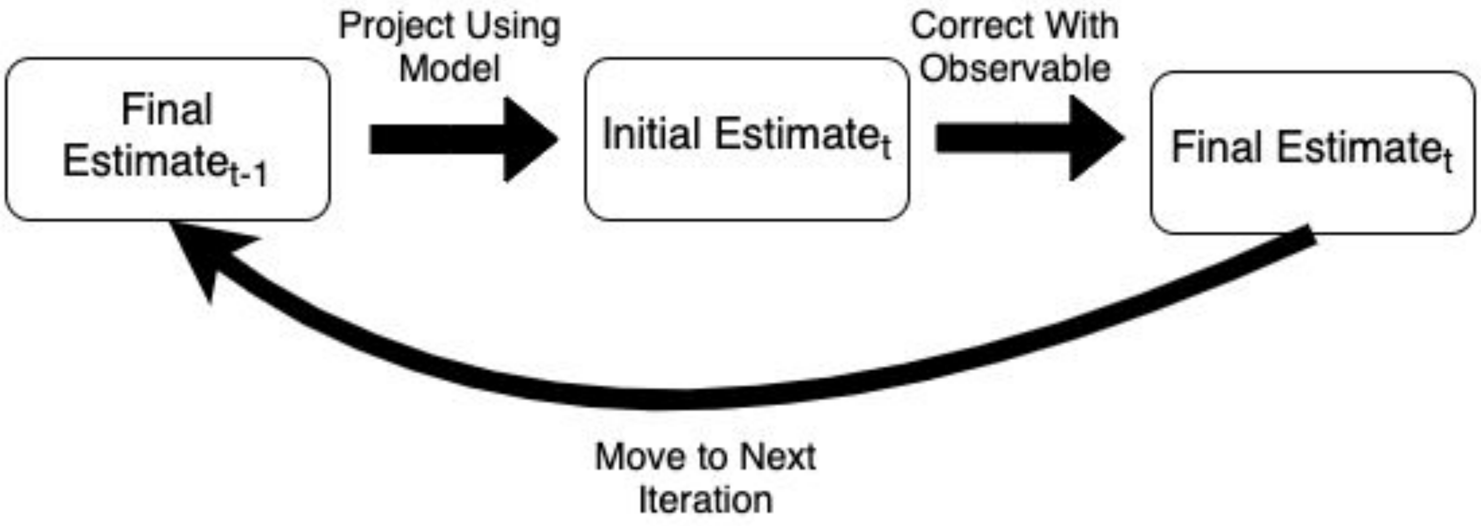
Observables

$$y_k = H_k x_k + v_k$$

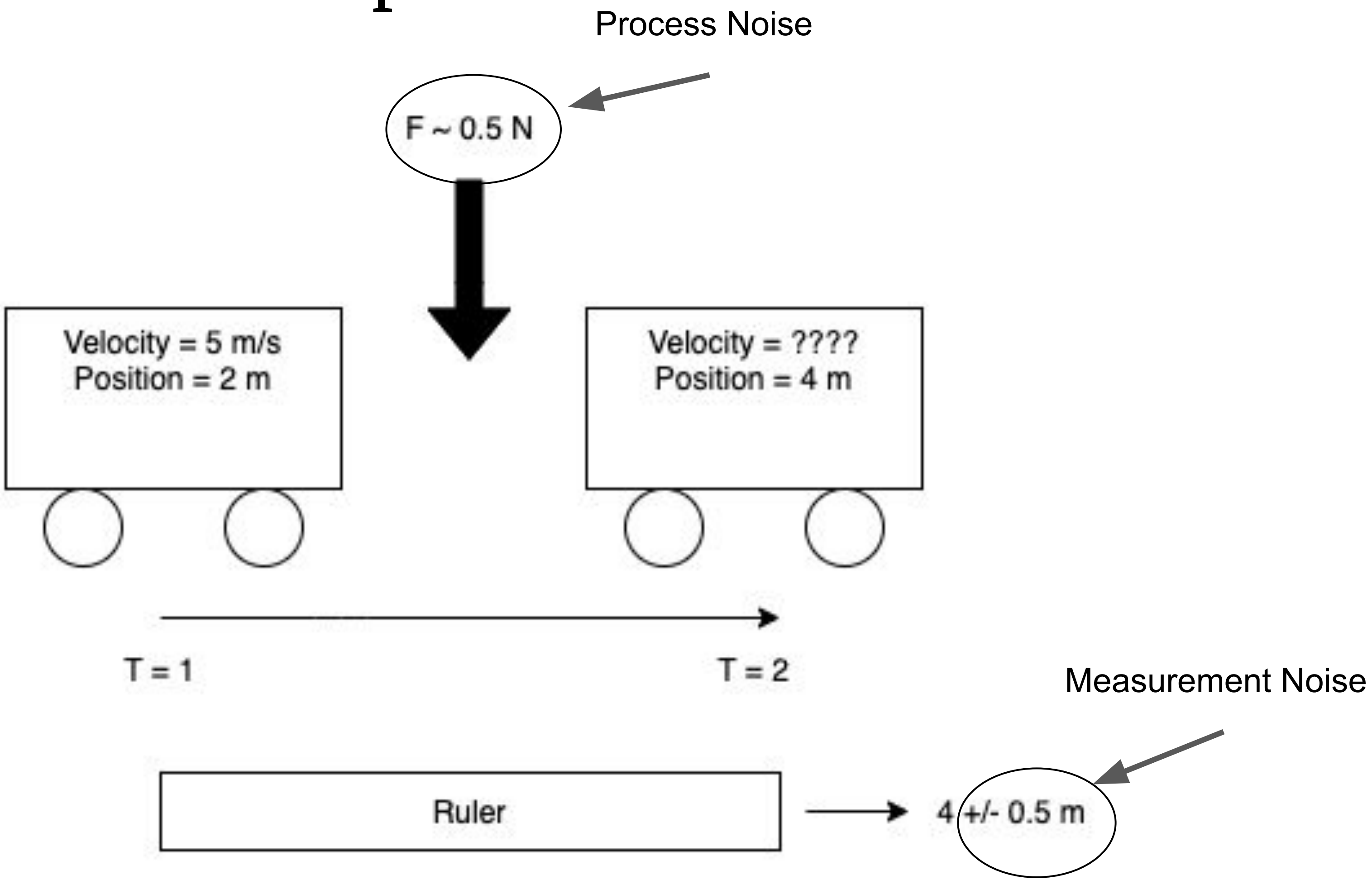
Measurement Noise

Measurement Matrix

Big Picture Diagram



Kalman Filter Example



Markov Chain Monte Carlo (MCMC)

- Parameter estimation based on maximum likelihood
  - Bayesian inference
- General idea
  - Propose parameters sets (*Markovian*)
  - Accept ‘good’ proposals
  - Reject ‘bad’ ones
  - Build a distribution of accepted ‘good’ parameters (*Monte Carlo*)

Bayes' Theorem

$$P(\theta|D) = P(\theta)P(D|\theta)$$

$P(\theta)$ - prior
$P(D|\theta)$ - likelihood
$P(\theta|D)$ - posterior

D - data

θ - parameter set

In terms of parameterization...

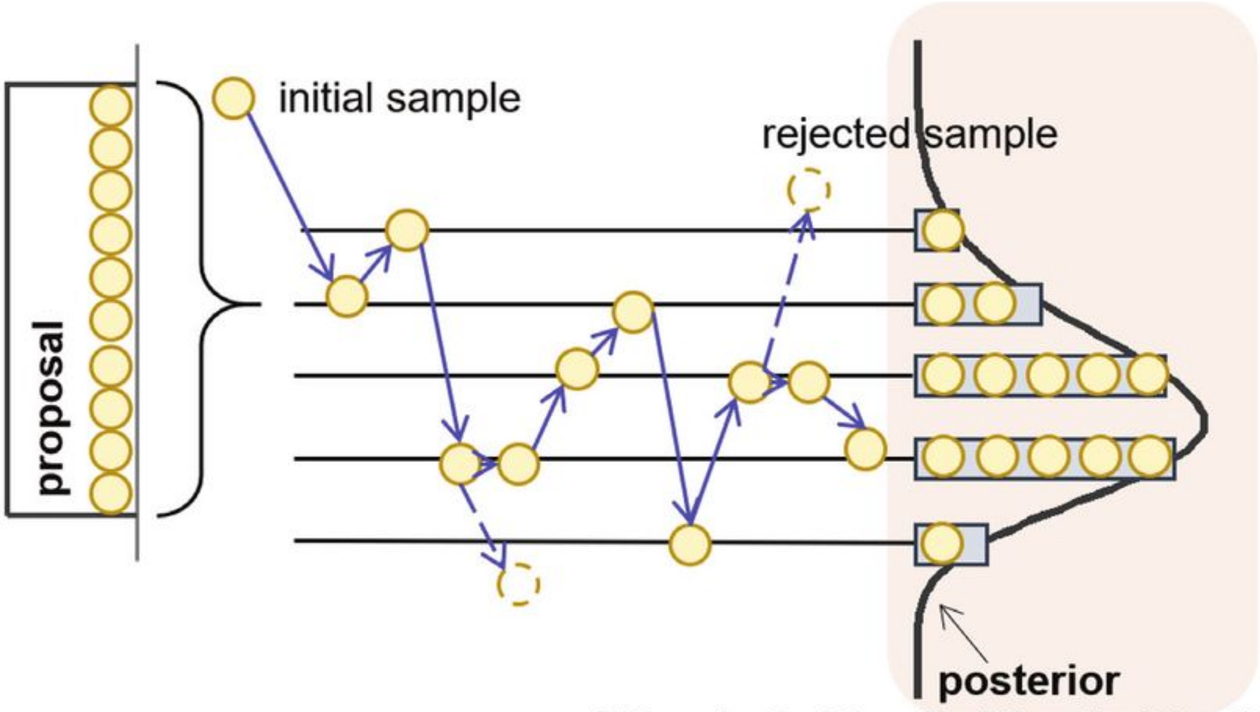
- *posterior* = estimated/sampled parameters
- *likelihood* = relationship between data points (noise)
  - tells us *how* to sample
- *prior* = knowledge about seeing parameters *before* seeing data
  - uninformative (uniform)
  - informative
- *proposal* = sampling distribution of parameter values

Metropolis-Hastings

1. Choose θ^0 (initial guess for parameter values)
    - calculate posterior
2. For each iteration of the Markov Chain:
    - Sample candidate θ^* from proposal distribution
        - calculate posterior
    - Accept or reject the proposed θ^* based on the value of a **ratio**
        - $\varrho = \dfrac{\textit{posterior of } \theta^*}{\textit{posterior of current } \theta}$

          if $\varrho \geq 1$: accept the transition
          if $\varrho < 1$: only accept with a probability of ϱ
3. Stop sampling after sufficient # of samples (10,000 typical)
4. Compute pdf of parameters using MCMC parameter samples

Big Picture Diagram

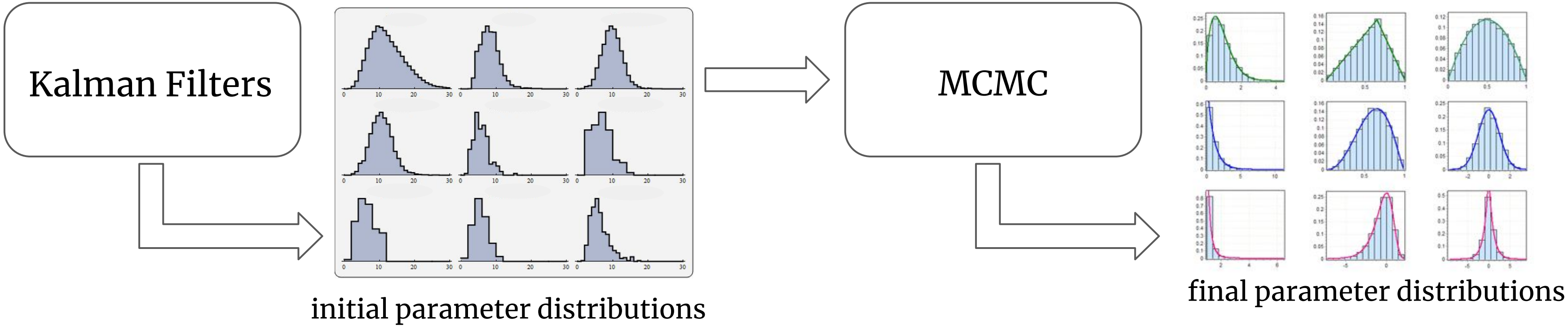


Dong et al. 2019. Prognostics 102: Efficient Bayesian-Based Prognostics Algorithm in MATLAB.

Combining the Methods

Goal: Use post-Kalman Filter output as pre-MCMC input

1. Use Kalman Filters to parameterize model
   a. Fit distributions to output values
2. Use distributions as informative priors for MCMC
3. Parameterize with MCMC
4. Fit model



Our progress

- Wrapping up
  - Early start (May)
  - Both teams successful in parameterizing T1D model
- Organization & Writing
  - Produce tutorial for parameterization using Predator-Prey (simpler model) as example
  - Clean and organize code for future use