Week 5 Summary

Subteam 2

June 2020

1 Introduction

This week we made our transition to applying the Unscented Kalman Filter (UKF) techniques to real-world mice data. The dataset we were using is from Li et al and provides us with Glucose readings taken at various time points, which are given in weeks. Our goal is to apply both the Joint and Dual UKF's to each mouse individually in order to create parameter estimates. Then, we would like to approximate distributions for the parameters using these values. We believe that understanding the distribution of the chosen parameters can assist the MCMC techniques being explored by subteam 1.

2 Parameters

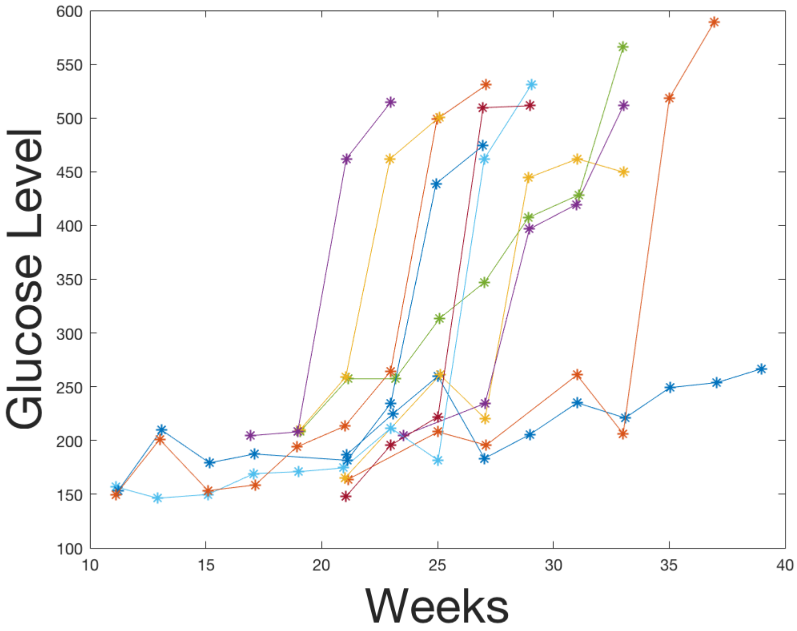
The T1D model consists of 42 parameters. Thus, it is unreasonable to approximate all of them. In order to create a more reasonable subset of parameters to estimate, we use the sensitivity analysis previously completed by An. This analysis isolates 7 parameters that are most important for the model and are thus the ones we have chosen to focus on. The parameters are:

- D_{ss} - dendritic cells in the pancreas
- Q_{panc} - pancreas volume
- S_I - rate glucose taken up in proportion to insulin
- G_I - insulin saturation point
- α_B - beta cell production rate
- μ_r - effector and regulatory cell interaction
- μ_e - effector and regulatory cell interaction

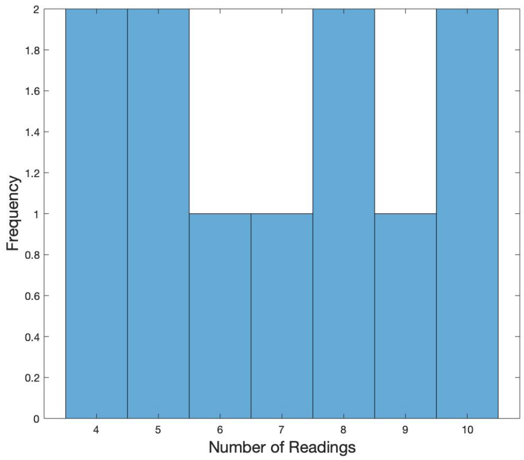
The other parameters were held constant at their baseline values, allowing us to focus exclusively on these 7.

3 Data

The dataset we worked with this week is from Li et al. It consists of 11 NOD mice, each with a set of glucose readings taken at times stated in weeks. The full dataset can be visualized by:



We can see that most of the mice follow a similar trajectory: they are relatively flat for the first couple readings, followed by a large spike to signify that the T1D has onset and remain at this point for the remainder of the readings.



Two important statistics to consider are the number of readings per mice as well as the time between readings. The average number of readings per mice is **6.91** and the average time between readings is **2.15** weeks. Furthermore, as evidenced by the histogram below, the number of readings is not necessarily centered around 6.91, but rather follows a relatively uniform distribution.

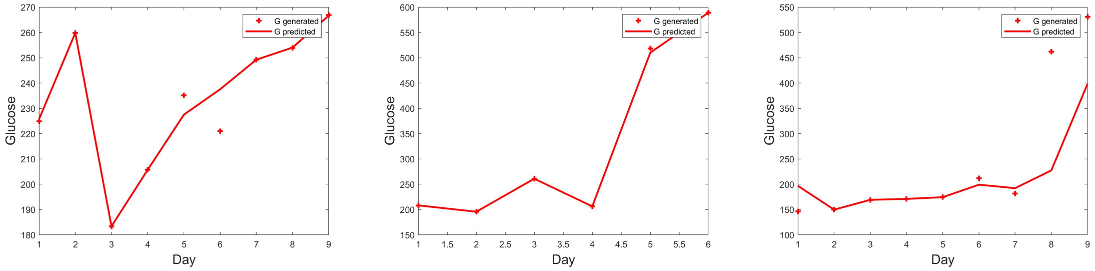
There are a couple things to keep in mind moving forward in terms of the quality of the dataset we have been given. First, the average number of readings is relatively low. Since the Kalman Filter's performance improves as more and more datapoints are used, this means that we are limited in the number of times our parameters are able to move. Second, we are working with a limited sample of 11 mice. This means that, particularly when looking at distributions of parameters, we must keep in mind that we have a limited number of runs available to us. Lastly, the average time span of 2.15 weeks between readings poses some challenges. The ODE's we are working with are based on changes per day, meaning that between readings we are jumping over ≈ 14 time steps. Since a lot of changes in state can occur in this length of time, this is important to consider moving forward.

4 Individual Mouse

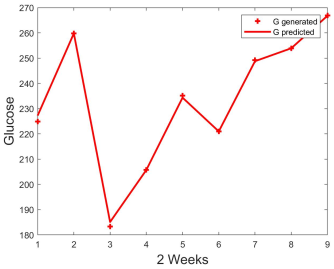
4.1 Glucose Estimation

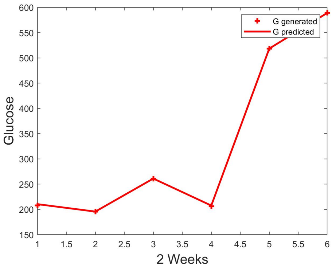
One of the drawbacks of this model is that since we have only one observable, glucose, it can be tricky to get a clear look into what the model is doing. However, looking at the results of the state estimation of the glucose compared to the actual data points can give us a look as to how the overall filter is performing relative to the data.

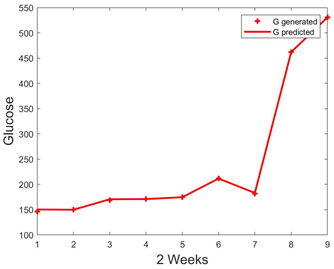
Below, we have some of the state estimation for glucose for the joint filter for Mouse 1, Mouse 2, and Mouse 6 shown from left to right:



As we can see from these, most of the estimation is pretty good, sometimes with the exception of one or two data points. This trend is pretty consistent no matter what the initial parameters are. However, changing the cholesky factorization step (as described in a later section) cleans up some of the errors, as shown below - the same mice are shown:

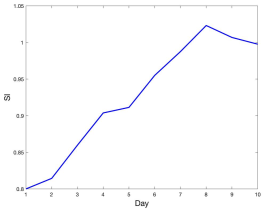


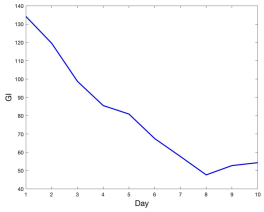




4.2 Parameter Estimation

Here we can see two examples of the movement of the parameter estimates for a single mouse:





It is clear that the parameters are moving as a result of the algorithm, however it is very much unclear if they have converged. In general, looking at a single mouse is difficult because of the low number of data points. As a result, we need mechanisms to gain more of a holistic understanding. We have explored two approaches to doing so. First, we have looked at the distributions of final parameter estimates across all the mice. Second, we have ran the algorithm multiple times for a single mouse in hopes of seeing more convergence. Both of these approaches will now be detailed.

5 Parameter Distributions

Our goal is to create distributions of the final outputted parameter values by running both our joint and dual UKF on individual mice. Our process was, for each mouse, to do the following:

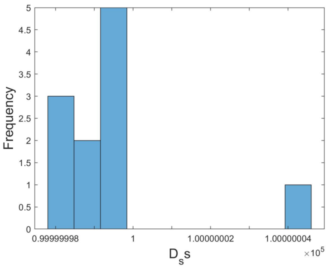
1. Identify the first available reading, and call the time at this point t
2. Simulate, using the T1D ODE's, up to point t and use the states at this point as the initial states.
3. Run the Joint and Dual UKF independently (there is **no** interaction between the 2 algorithms

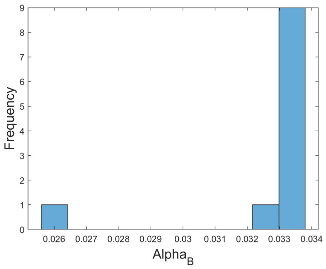
Then, we can plot histograms of the final parameter estimates in hopes of seeing a general trend.

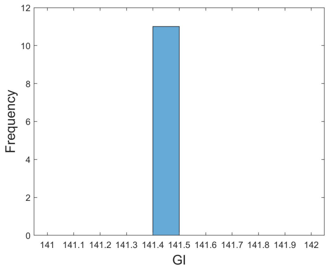
5.1 Joint Estimation

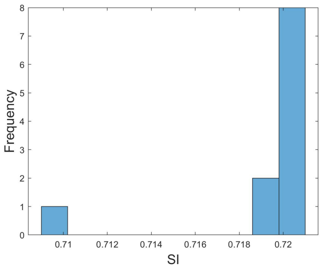
We ran the Joint Unscented Kalman filter on each mouse with a couple different sets of initial parameters and initial covariance matrices. This allowed us to see the effects of these initial conditions on the movement of the parameters. Below, we have the parameter distribution histograms with the different initial conditions. In both cases, the covariance matrix for the states is set as a diagonal matrix of ones.

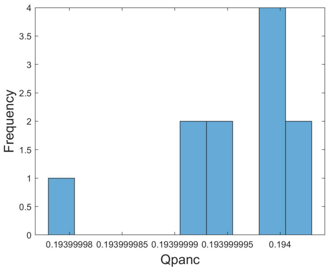
First, we have the case where the parameter covariance matrix is set to a matrix of ones and the initial conditions are set to the baseline parameter values.

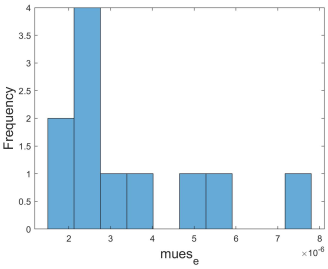


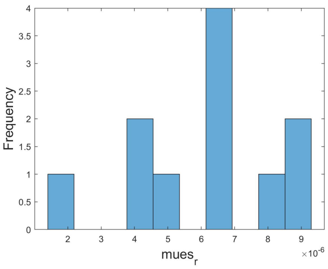










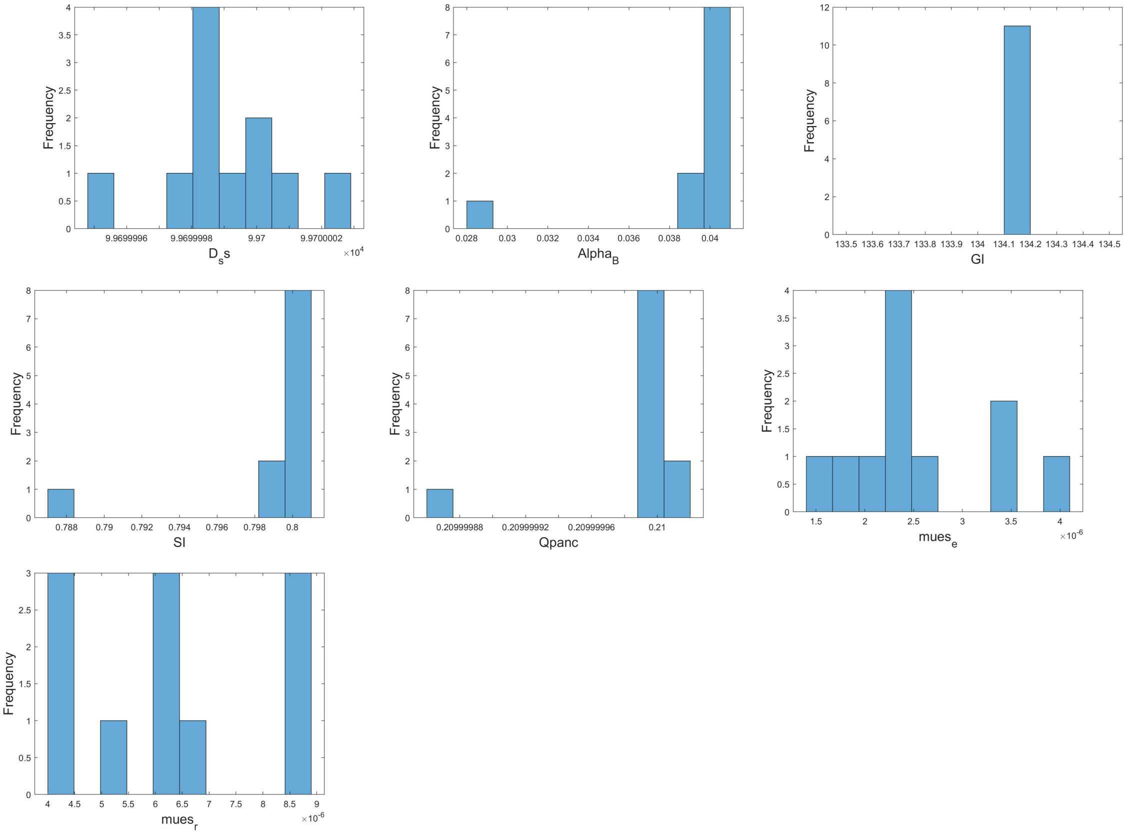


Next, we have a parameter covariance matrix that is set to

$$P_{param_0} = \begin{bmatrix} 10^5 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 10 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 10000 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 10 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 10 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 10^{-2} & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 10^{-2} \end{bmatrix}$$

and initial parameter values set to:
$D_s s$ - 99700
α_B - 0.0334
G_I - 134
S_I - 0.8
$Q_p anc$ - 0.210
μ_e - 2e-6
μ_R - 2e-6

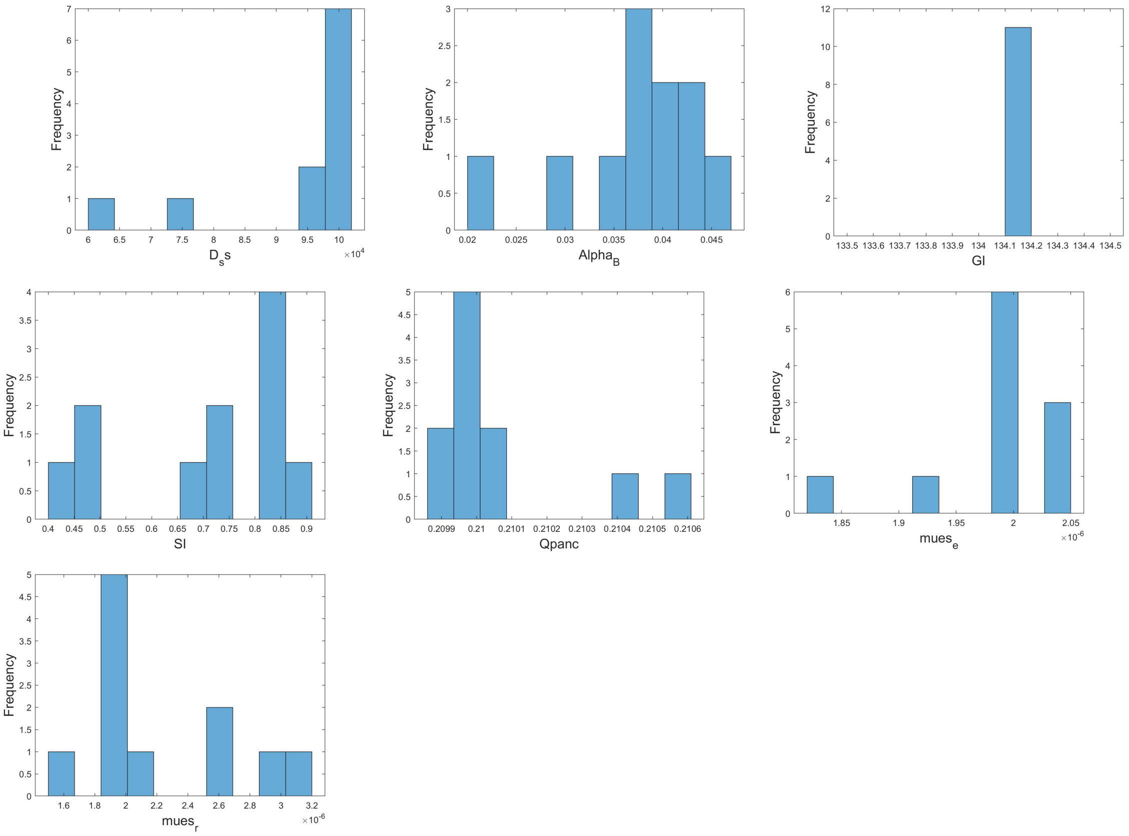
NOTE: for a more detailed description of this choice of covariance matrix please see the **Dual Estimation** section below



The following distributions were initialized to the same parameter, but had a cholesky default of

$$\begin{bmatrix}10^5 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 10 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 10000 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 10 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 10 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 10^{-2} & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 10^{-2}\end{bmatrix}$$

instead of the previous default of a diagonal matrix with the diagonal of .001.



5.2 Dual Estimation

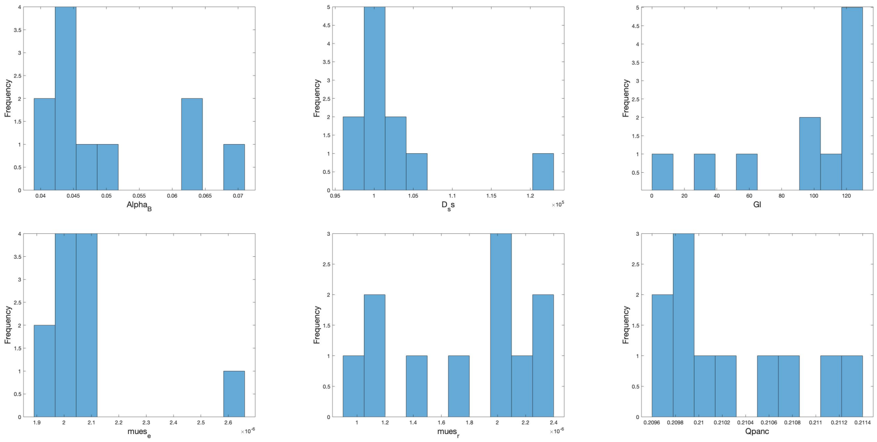
First let us discuss the set up for the Dual UKF that was used. Here, we will discuss the setup of the initial covariance matrix for parameters. The rest of the matrices, both for noise and for covariance of states, can be seen in our Github repo. The covariance matrix of parameters was set as follows:

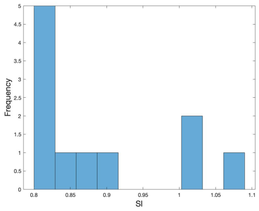
$$P_{param_0} = \begin{bmatrix} 10^5 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 10 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 10000 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 10 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 10 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 10^{-2} & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 10^{-2} \end{bmatrix}$$

The values across the diagonal represent the various of individual parameters, in the order that they are listed in the **parameters** section of this document. These values control the amount of movement an individual parameter can have and were chosen in an ad hoc fashion. However, they appear to have a lot of control over the final estimates of our parameters, so it is something we are interested in understanding more moving forward. On the off diagonal we have placed 1's in order to allow for some covariance between parameters. Once again, we are, at this moment, not entirely sure of the relationships between the parameters and is something we will spend much of our time next week on.

The initial conditions for parameters that were used is as follows:
$D_s s$ - 99700
α_B - 0.0334
G_I - 134
S_I - 0.8
$Q_p anc$ - 0.210
μ_e - 2e-6
μ_R - 2e-6

Using this set up and then running the Dual UKF on each mouse results in the following parameter distributions:





In general, we do see groupings or patterns where the parameters appear to want to go. However, we do not have full confidence in these results for two main reasons:

1. We only have 11 data points to plot for each parameter.
2. Based on our images in the **Individual Mouse** section, we cannot be confident that these are the true final parameter values that we have converged to. Since we have a limited number of readings, it is unlikely that we have been able to reach parameter convergence. Thus, looking at distributions after so few time steps is premature.

Rather than looking at distributions of parameters, we now will attempt to run the UKF multiple times on a single mouse in hopes of seeing more parameter convergence.

6 Comparison of Results

6.1 Comparison of Joint with Different Parameters

Above, we see examples of parameter histograms for the joint ukf with two different sets of initial parameters. Looking at them, it seems in most cases the initial parameters had the biggest impact. The distributions look very similar, but are usually centered around a different value, likely due to the different initial parameter. There are a couple exceptions to this. One is that the $alpha_B$ parameter is spread out over a larger range in the second set of distributions than the first. Since this parameter was initialized to the same value in both cases, this is likely due to the fact the covaraince assigned to this parameter is larger. Another exception is the μ_E and μ_R parameters, where in the second set of distributions, the range is significantly smaller. This is likely due to the fact that the covariance matrix in this case defined the variance between them to be extremely small, especially since they were initialized to the same parameter.

6.2 Comparison of Joint with different Cholesky steps

One of the steps in the ukf algorithm is the cholesky factorization of the covariance matrix. In the way that our code is set up, there in this step, the matrix defaults to a default value if it is not a positive definite matrix. This is a value that we are able to manipulate. From debugging our code, it seems that the covariance matrix must default to this value often. We ran the joint ukf with

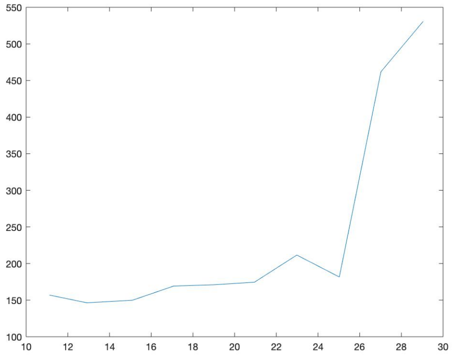
two different default matrices for the cholesky step. The results can be viewed above. Here, we will discuss those results.
We can see that with this new default matrix, the distributions stay similar but the range becomes larger except in the case of μ_E and μ_R where it becomes smaller. This makes sense because in that case, the covariance value for those variables changes the least. Also, it now looks like the distribution for those two variables is centered around the baseline value, which suggests that this version is performing better.

6.3 Comparison of Dual and Joint

Looking at the Dual and Joint with the same initial values (the third set of joint with the new cholesky factorization) it seems like the Joint is performing better. The distributions are more likely to center around the baseline value and in some cases, the ranges are smaller. One interesting discrepancy is that in the joint, the G_I parameter will not move, no matter the initial condition, while it moves a lot in the dual often away from the baseline. If we could find out why these two very different effects occur, we might discover more about the inner workings of the filter.

7 Multiple Iterations on Single Mouse - HAS ONLY BEEN DONE WITH DUAL SO FAR

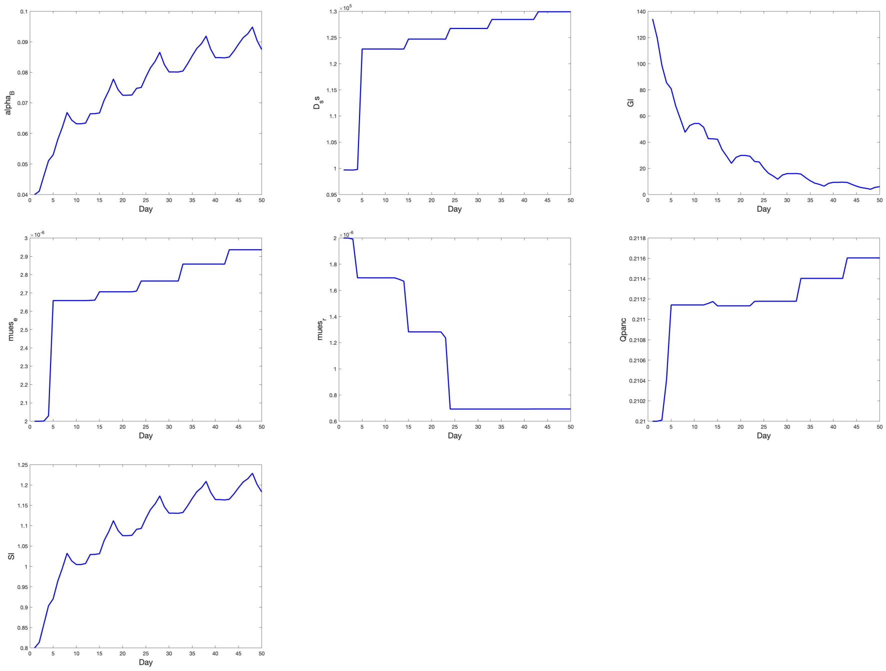
To test the effect of multiple iterations on a single mouse, we first needed to select a mouse we have deemed "average" to use for this section. We have chosen to use mouse 6. This is do to its high amount of readings (10) and that its glucose readings follow a shape similar to the other samples. The glucose for mouse 6 looks like the following.



Next, we need to select the number of run-throughs of the UKF that we will put Mouse 6 through. We have chosen to use 5 iterations because at this point the process is still computationally efficient and we can get a sense for the trajectory of the parameters. Thus, our process will be to run the UKF on the Mouse 6 data 5 times where the initial parameters for run j are set to be the final parameter values for run $j-1$.

7.1 Using the Dual UKF

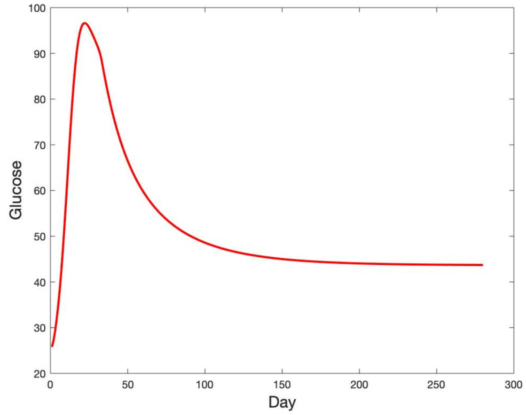
We can look at the parameter values as a function of the number of readings we have used in the following figures:



There are a few things to note here. First, there are a total of 50 data points since we have 5 runs, each with 10 readings. Next, the results do seem to suggest that the parameters are getting much closer to convergence than when we only ran the algorithm once. Although there is still movement in parameter values, the net change from start to finish for each run decreases as we run the algorithm more and more times. The clearest example of convergence is in parameter $mues_r$. Here, the parameter no longer changes whatsoever past reading 25 or so. Part of our experimentation moving forward will be in increasing the number of runs and seeing what effect this has on our parameter outputs.

7.2 Validation

At this point, we have parameter estimates for mouse 6 after running the algorithm 5 times. It is now important to understand the biological plausability of these parameters. In order to do so, we have plotted the entire system of ODE's using these final parameter values. Unfortunately, this exposed a large issue in our parameter values. The best example of this is the graph of glucose values under these parameters:



This, however, is not the behavior we would expect. Since this is an NOD mouse with an apoptotic wave, the Glucose level should spike to and remain above 250. However, we instead see a slight spike and then Glucose lowering to a steady state much lower than it should. Our goal is to now understand why this is occurring.

7.3 Possible Solutions

We are going to explore multiple avenues with respect to obtaining parameter estimates that are more biologically feasible:

1. Lowering the variances of the parameters: since the parameters appear to be deviating too far from the point which is biologically feasible, we need to control how much the parameters can move around. We can do this by playing around the variances of the parameters and, in particular, lowering them.

2. At this point we are estimating 7 parameters while keeping the rest constant. However, parameters in the model are inherently related to one another, something we are not currently accounting for. In order to deal with this, we will first run the algorithm but this time estimate **all** parameters. Additionally, we are going to take a deeper look at the sensitivity analysis that has been done on the T1D model. This way we can hopefully see which parameters are dependent on the 7 we are currently estimating and thus add those to our estimation as well.

8 Next Steps

Our first task is to better understand the range in which our parameters of interest, we may now exceed past the 7 we have identified, should move. Once we have done this, we would like to combine our methods of multiple iterations on single mice and plotting parameter distributions into a single workflow by running the UKF (both joint and dual) on each mouse x amount of times and then creating histograms of the final parameter estimates.

We then hope to apply these distributions to use as priors in the MCMC techniques.