Pomona College

Senior Thesis in Mathematics

An Exploration of Kalman Filters

Author:
Lindsey Tam

Advisor:
Dr. Blerta Shtylla

Submitted to Pomona College in Partial Fulfillment
of the Degree of Bachelor of Arts

December 10, 2019

Abstract

Kalman Filters create predictions for linear systems by recursively generating and correcting predictions according to incoming data. This process enables estimates to become progressively more accurate without relying on large amounts of initial data. Nonlinear forms of the Kalman Filter exist and include the Extended Kalman Filter, the Unscented Kalman Filter, and the Dual Unscented Kalman Filter. This paper explores the theory behind each of these filters and implements a few examples of the Unscented Kalman Filter.

Contents

Chapter 1

Introduction

A Kalman Filter, so named from its developer Rudolph Kalman, is a data prediction method that estimates the value of unknown parameters of linear dynamical systems. This is a recursive predictive-corrective process that allows Kalman Filters to continuously generate predictions about state variables without relying on large amounts of initial data. In the case of nonlinear systems, alternative forms of the Kalman Filter were developed, including the Extended Kalman Filter, the Unscented Kalman Filter, and the Dual Unscented Kalman Filter. The Extended Kalman filter linearizes the nonlinear system around the mean by taking the Jacobean of the nonlinear function. Navigation and signal processing are the main applications of the Extended Kalman Filter. Instead of linearizing the system around a single points, the Unscented Kalman Filter utilizes many points, known as Sigma points. The Sigma points then undergo a nonlinear transformation known as the Unscented Transform. The Dual Unscented Kalman Filter is similar to the Unscented Kalman Filter, but also estimates the value of some unknown parameter in the system. Applications of the Unscented and Dual Unscented Kalman Filters include modeling biological systems.

Chapter 2

Kalman Filters

The Kalman Filter (KF) generates predictions for linear dynamical systems [**?**]. Assuming the given data is noisy and Gaussian [**?**, **?**], the first predictive step assumes knowledge of initial states and the model process, which takes the form of matrix F. The next step involves calculating the covariance in order to calculate the Kalman Gain, which is a measure of how much the estimate should be changed given actual measurements of the system. The corrective step utilizes the Kalman Gain and the measured system outputs to gauge whether to depend more on the model or on the data.

This process can be done recursively, allowing the model to become progressively more accurate as more data is added. Therefore, overtime, it is expected that the model will converge with the actual system measurements.

2.1 Kalman Filter Algorithm

Variables in the Kalman Filter		
Variable	Description	Dimensions
x	State variables	$d_x \times 1$
y	Output Vector	$d_y \times 1$
u	System Inputs	$d_u \times 1$
v	Measurement Noise	$d_y \times 1$
w	Process Noise	$d_x \times 1$
F	State function	$d_x \times d_x$
H	Observation function	$d_y \times d_x$
G	Input function	$d_x \times d_u$
K	Kalman Gain	$d_x \times d_y$
Q	Process noise covariance	$d_x \times d_x$
R	Measurement noise covariance	$d_y \times d_y$
P	Covariance matrix	$d_x \times d_x$

1. Initialize the state estimate (x_0) and the initial state covariance matrix

(P_0)

$$x_0 = \mathbb{E}[x_0] = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

$$P_0 = \begin{bmatrix} var(x_1) & \dots & cov(x_1, x_n) \\ \vdots & \ddots & \vdots \\ cov(x_n, x_1) & \dots & var(x_n) \end{bmatrix}$$

Enough should be known about the modeled system to generate these values. In general, x_k denotes the model's estimate of the state variables, but when initializing these values, it is acceptable to use actual measurements of the system.

A state covariance matrix is a symmetrical positive semi-definite square matrix whose diagonals correspond to the variance of a variable at location i and elsewhere is the covariance of the pairwise elements. In practice, covariance matrices help us better understand the spread of data. For the case of the KF, calculating the state covariance is necessary for computing Kalman Gain.

2. Generate a prediction (x_{k+1})

$$x_{k+1} = Fx_{k-1} + Gu_{k-1} + w_{k-1}$$

State matrix F and Input matrix G is how the system is defined. Every state variable contained in x_k is defined by a linear differential equation. These linear differential equations can be used to generate the F and G matrices. Therefore, F should be a square matrix whose dimension is equal to the number of states variables and G can be a matrix (or in some cases a vector), depending on the dimension of input vector u.

u_k is an input vector, which is a measurable value that helps define the system, but is not contained in the state vector. Depending on how the system is defined, u_k can be a constant value or it can be a value dependent on time step k.

w_k is the process noise vector at time k. Process noise can be thought of as the model's accuracy. When process noise is 0, it implies that the model is perfectly accurate and does not have to correct for incoming system measurements. On the other hand, high process noise will essentially restate the system based on incoming measurements. w_k has the same dimensions as x_k, allowing us to identify whether or not to adjust the equations for the state variables. When all values of w_k equal 0, it implies that the linear differential equations we are using to define the state variables have no error.

3. Use the state covariance matrix ($P_{k|k-1}$) to calculate Kalman Gain (K_k)

$$P_{k|k-1} = FP_{k-1}F^T + Q_{k-1}$$

$$K_k = P_{k|k-1}H_K^T(H_kP_{k|k-1}H_K^T + R_k)^{-1}$$

   H is the observation matrix, which enables the state variables to be linearly transformed to match the outputs of the system. The dimensions of H reflects which state variables have measurable values in the system. It is not assumed that every state variable is measurable, so H allows us to compare the measurable state variables to x_k. Simple applications of H include creating matrices with 0's and 1's, with 1's denoting that a state variable is measurable and a 0's representing non-measurable states. In other cases, H is an integer used as a scaling factor.

   The Kalman Gain is the main component of the corrective aspect of the KF. The Kalman Gain is a measure of how much to change the model based on incoming data. Low values of the Kalman Gain imply the model is accurate while higher values indicate the model should adjust based on the incoming data. Calculating Kalman Gain involves using Q, which is process noise covariance of w_k, R, which is measurement noise covariance, and H, which is also used in transforming the output vector in a later step. From this equation, one can see that balancing Q and R is critical for model performance. Larger values of Q indicate higher modeling error, which leads to a higher Kalman Gain and increased model correction. On the other hand, large values of R imply high measurement error, leading to a lower Kalman Gain and less model correction.

4. Calculate the transformed output vector ($\hat{y}_k$) and correct the prediction

$$\hat{y}_k = H_kx_k + v_k$$

$$x_k = x_{k-1} + K_k(y_k - \hat{y}_k)$$

   y_k, also known as the output vector, is the actual measurements of the system and $\hat{y}_k$ is a way of transforming the prediction (using H) into a format that can be compared with y_k.The term v_k, also known as measurement noise, is also added to account for measurement error. Subtracting the actual measurements from the predicted measurements is necessary, because y_k does not always contain measurements for all variables in state vector x_K. The quantity $y_k - \hat{y}_k$ is also known as measurement residual or innovation.

5. Update the covariance matrix

$$P_k = (I - K_KH_k)P_{k|k-1}$$

   P_k will be used in the next iteration of the filter.

Chapter 3

Extended Kalman Filters

The Extended Kalman Filter (EKF) is the non-linear version of the Kalman Filter. For the most part, the EFK Algorithm is nearly identical to the KF algorithm. The critical difference is lies in defining the state function. The EKF uses the Jacobean to linearly approximate the non-linear function around the mean of the Gaussian distribution. While this is a functional method, it can become inefficient when dealing with complex, higher order systems. The Jacobian becomes harder to compute and using a singular point to make estimates of state is insufficient. Applications of the EKF include navigation.

3.1 Extended Kalman Filter Algorithm

Variables in the Extended Kalman Filter		
Variable	Description	Dimensions
x	State variables	$d_x \times 1$
y	Output Vector	$d_y \times 1$
u	System Inputs	$d_u \times 1$
v	Measurement Noise	$d_y \times 1$
w	Process Noise	$d_x \times 1$
f	None linear state function	$d_x \times d_x$
F	State Function	$d_x \times d_x$
h	Non linear observation function	$d_y \times d_x$
H	Linearized observation function	$d_y \times d_x$
G	Input Matrix	$d_x \times d_u$
K	Kalman Gain	$d_x \times d_y$
Q	Process noise covariance	$d_x \times d_x$
R	Measurement noise covariance	$d_y \times d_y$
P	Covariance matrix	$d_x \times d_x$

1. Initialize the state estimate (x_0) and the initial covariance matrix (P_0)

$$x_0 = \mathbb{E}[x_0] = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

$$P_0 = \begin{bmatrix} var(x_1) & \dots & cov(x_1, x_n) \\ \vdots & \ddots & \vdots \\ cov(x_n, x_1) & \dots & var(x_n) \end{bmatrix}$$

2. Calculate the Jacobean

$$F = \frac{\partial f}{\partial x}$$

$$H = \frac{\partial h}{\partial x}$$

Here, f is the non-linear state function and F is the linear approximation of f. On that same vein, h is the non linear observation while H is linearized observation.

3. Generate a prediction

$$x_{k+1} = f(x_{k-1}, u_{k-1})$$

4. Use the state covariance matrix ($P_{k|k-1}$) to calculate Kalman Gain (K_k)

$$P_{k|k-1} = FP_{k-1}F^T + Q_{k-1}$$

$$K_k = P_{k|k-1}H_K^T(H_kP_{k|k-1}H_K^T + R_k)^{-1}$$

Note that we are using F, not the linear approximation f.

5. Correct the prediction

$$\hat{y}_k = H(x_k + v_k)$$

$$x_k = x_{k-1} + K_k(y_k - \hat{y}_k)$$

6. Update the covariance matrix

$$P_k = (I - K_KH_k)P_{k|k-1}$$

Chapter 4

Unscented Kalman Filters

The Unscented Kalman Filter (UKF) does not require using Jacobians to linearize non-linear functions. Instead, the UKF uses the Unscented Transform (UT). This process involves using sigma points, which are represented in a Sigma Point Matrix and represent the normal distribution of the data. The number of sigma points is deterministic and depends on the dimensions of the system. In general, a UFK will have 2 $d_x + 1$ sigma points, where d_x represents the dimension of the state vector [**?**, **?**, **?**]. The covariance and weights of these sigma points are calculated. These sigma points undergo a non-linear transformation, resulting in a posterior distribution that is not normal [**?**, **?**] . We are able to approximate the normal distribution of the posterior distribution using the weights and covariance that were calculated prior to the transformation.

Variables in the Unscented Kalman Filter		
Variable	Description	Dimensions
x	Vector containing state variables	$d_x \times 1$
χ	Sigma Point Matrix	$d_x \times 2d_x + 1$
F	State Transition Matrix	$d_x \times d_x$
H	Observation Matrix	$d_y \times d_x$
G	Input Matrix	$d_x \times d_u$
u	Input Vector	$d_u \times 1$
P	Covariance matrix	$d_x \times d_x$
k	Time step	1×1
w	Process Noise Vector	$d_x \times 1$
α	Controls spread of sigma points	$0 < \alpha < 1$
β	Adjust sigma point weight	$0 \leq \beta$
κ	Sigma point weighting constant	$0, 3 - d_q, 0 \leq \kappa \leq 3$
λ	Scaling parameter for sigma point weights	scalar

1. Initialize state vector and covariance

$$\hat{x}_0 = \mathbb{E}[x_0] \tag{4.1}$$

$$P_{x_0} = \mathbb{E}[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T] \tag{4.2}$$

2. Calculate sigma points

$$\chi_{\ 0,k-1} = \hat{x}_{k-1} \tag{4.3}$$

$$\chi_{\ i,k-1} = \hat{x}_{k-1} + (\sqrt{(n_x + \lambda_x)P_{x_{k-1}}})_i, \qquad i = 1, \ldots, n_x \tag{4.4}$$

$$\chi_{\ i,k-1} = \hat{x}_{k-1} - (\sqrt{(n_x + \lambda_x)P_{x_{k-1}}})_i, \qquad i = n_x + 1, \ldots, 2n_x, \tag{4.5}$$

3. Calculate the weights for each sigma point

   Weights can have positive or negative values, but will ultimately sum to 1 [**?**]

$$\lambda_x = \alpha_x^2(n_x + \kappa_x) - n_x \tag{4.6}$$

$$W_{x,0}^{(m)} = \frac{\lambda_x}{n_x + \lambda_x} \tag{4.7}$$

$$W_{x,0}^{(c)} = \frac{\lambda_x}{n_x + \lambda_x} + (1 - \alpha_x^2 + \beta_x) \tag{4.8}$$

$$W_{x,i}^{(m)} = W_{x,i}^{(c)} = \frac{\lambda_x}{2(n_x + \lambda_x)}, i = 1, \ldots, 2n_x \tag{4.9}$$

$$W_{x,0}^{\text{aug},(m)} = \frac{\lambda_x}{2n_x + \lambda_x} \tag{4.10}$$

$$W_{x,0}^{\text{aug},(c)} = \frac{\lambda_x}{2n_x + \lambda_x} + (1 - \alpha_x^2 + \beta_x) \tag{4.11}$$

$$W_{x,i}^{\text{aug},(m)} = W_{x,i}^{\text{aug},(c)} = \frac{\lambda_x}{2(2n_x + \lambda_x)}, i = 1, \ldots, 4n_x \tag{4.12}$$

$$\tag{4.13}$$

4. Perform a non-linear transformation on the sigma points

$$\hat{w}_k = \hat{w}_{k-1} \tag{4.14}$$

$$\chi^*_{\ i,k|k-1} = f(\chi_{\ i,k-1}, \hat{w}_{k-1}) \tag{4.15}$$

5. Calculate the mean and covariance of the transformed sigma points

$$\hat{x}^*_{\ k|k-1} = \sum_{i=0}^{2n_x} W_{x,i}^{(m)} \chi^*_{\ i,k|k-1} \tag{4.16}$$

$$P_{x_{k|k-1}} = \sum_{i=0}^{2n_x} W_{x,i}^{(c)} (\chi^*_{\ i,k|k-1} - \hat{x}^*_{\ k|k-1})(\chi^*_{\ i,k|k-1} - \hat{x}^*_{\ k|k-1})^T + R_v \tag{4.17}$$

6. Re-calculate sigma points

$$\chi_{\ i,k|k-1} = {\chi^*}_{i,k|k-1}, i = 0, ..., 2n_x \tag{4.18}$$

$$\chi_{\ i,k|k-1} = {\chi^*}_{0,k|k-1} + (\sqrt{(n_x + \lambda_x) P_{x_{k|k-1}}})_i, i = 2n_x + 1, \ldots, 3n_x \tag{4.19}$$

$$\chi_{\ i,k|k-1} = {\chi^*}_{0,k|k-1} - (\sqrt{(n_x + \lambda_x) P_{x_{k|k-1}}})_i, i = 3n_x + 1, \ldots, 4n_x \tag{4.20}$$

7. Generate prediction

$$\hat{x}_{k|k-1} = \sum_{i=0}^{4n_x} W_{x,i}^{\text{aug},(m)} \chi_{\ i,k|k-1} \tag{4.21}$$

$$\mathcal{Y}_{k|k-1} = h(\chi_{k|k-1}, \hat{w_k}) \tag{4.22}$$

$$\hat{y}_{k|k-1} = \sum_{i=0}^{4n_x} W_{x,i}^{\text{aug},(m)} \mathcal{Y}_{\ i,k|k-1} \tag{4.23}$$

8. Calculate Kalman Gain

$$P_{y_k} = \sum_{i=0}^{4n_x} W_{x,i}^{\text{aug},(c)} (\mathcal{Y}_{i,k|k-1} - \hat{y}_{k|k-1})(\mathcal{Y}_{i,k|k-1} - \hat{y}_{k|k-1})^T + R_n \tag{4.24}$$

$$P_{x_k y_x} = \sum_{i=0}^{4n_x} W_{x,i}^{\text{aug},(c)} (\chi_{\ i,k|k-1} - {\hat{x}^*}_{k|k-1})(\mathcal{Y}_{i,k|k-1} - \hat{y}_{k|k-1})^T \tag{4.25}$$

$$K_k = P_{x_k y_x} P_{y_k}^{-1} \tag{4.26}$$

9. Update Estimate and state covariance

$$P_{x_k} = P_{x_{k|k-1}} - K_k P_{y_k} K_k^{-1} \tag{4.27}$$

$$\hat{x}_k = \hat{x}_{k|k-1} + K_k (y_k - \hat{y}_{k|k-1}), \tag{4.28}$$

Chapter 5

Dual Unscented Kalman Filters

Used for simultaneously estimating state and parameter in dynamical systems [?, ?, ?]. Assumes parameters are identifiable (true value can be obtained from a finite number of observations) [?]. Has 2 distinct filters: one for state and the other for parameter [?, ?]

Variables in the Dual Unscented Kalman Filter		
Variable	Description	Dimensions
x	Vector containing state variables	$d_x \times 1$
χ	Sigma Point Matrix	$d_x \times 2d_x + 1$
F	State Transition Matrix	$d_x \times d_x$
H	Observation Matrix	$d_y \times d_x$
G	Input Matrix	$d_x \times d_u$
u	Input Vector	$d_u \times 1$
P	Covariance matrix	$d_x \times d_x$
k	Time step	1×1
v	Process Noise Vector	$d_x \times 1$
w		

1. Initialize state vector, parameter vector, covariance, and parameter covariance

$$\hat{x}_0 = \mathbb{E}[x_0] \tag{5.1}$$

$$\hat{w}_0 = \mathbb{E}[w_0] \tag{5.2}$$

$$P_{x_0} = \mathbb{E}[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T] \tag{5.3}$$

$$P_{w_0} = \mathbb{E}[(w_0 - \hat{w}_0)(w_0 - \hat{w}_0)^T] \tag{5.4}$$

2. Calculate sigma points

$$\mathcal{W}_{0,k|k-1} = \hat{w}_{k|k-1} \tag{5.5}$$

$$\mathcal{W}_{i,k|k-1} = \hat{w}_{k|k-1} + (\sqrt{(n_w + \lambda_w)P_{w_{k|k-1}}})_i, i = 1, \ldots, n_w \tag{5.6}$$

$$\mathcal{W}_{i,k|k-1} = \hat{w}_{k|k-1} - (\sqrt{(n_w + \lambda_w)P_{w_{k|k-1}}})_i, i = n_w + 1, \ldots, 2n_w \tag{5.7}$$

3. Calculate the weights for each sigma point

$$\lambda_x = \alpha_x^2(n_x + \kappa_x) - n_x \tag{5.8}$$

$$W_{x,0}^{(m)} = \frac{\lambda_x}{n_x + \lambda_x} \tag{5.9}$$

$$W_{x,0}^{(c)} = \frac{\lambda_x}{n_x + \lambda_x} + (1 - \alpha_x^2 + \beta_x) \tag{5.10}$$

$$W_{x,i}^{(m)} = W_{x,i}^{(c)} = \frac{\lambda_x}{2(n_x + \lambda_x)}, i = 1, \ldots, 2n_x \tag{5.11}$$

$$W_{x,0}^{\text{aug},(m)} = \frac{\lambda_x}{2n_x + \lambda_x} \tag{5.12}$$

$$W_{x,0}^{\text{aug},(c)} = \frac{\lambda_x}{2n_x + \lambda_x} + (1 - \alpha_x^2 + \beta_x) \tag{5.13}$$

$$W_{x,i}^{\text{aug},(m)} = W_{x,i}^{\text{aug},(c)} = \frac{\lambda_x}{2(2n_x + \lambda_x)}, i = 1, \ldots, 4n_x \tag{5.14}$$

$$\lambda_w = \alpha_w^2(n_w + \kappa_w) - n_w \tag{5.15}$$

$$W_{w,0}^{(m)} = \frac{\lambda_w}{n_w + \lambda_w} \tag{5.16}$$

$$W_{w,0}^{(c)} = \frac{\lambda_w}{n_w + \lambda_w} + (1 - \alpha_w^2 + \beta_w) \tag{5.17}$$

$$W_{w,i}^{(m)} = W_{w,i}^{(c)} = \frac{\lambda_w}{2(n_w + \lambda_w)} i = 1, \ldots, 2n_w \tag{5.18}$$

4. Perform a non-linear transformation on the sigma points

$$\mathcal{Y}_{i,k|k-1} = h(f(\hat{x}_{k-1}, \mathcal{W}_{i,k|k-1}), \mathcal{W}_{i,k|k-1}) \tag{5.19}$$

$$\hat{y}_{k|k-1} = \sum_{i=0}^{2n_w} W_{w,i}^{(m)} \mathcal{Y}_{i,k|k-1} \tag{5.20}$$

5. Calculate the mean and covariance of the transformed sigma points

6. Re-calculate sigma points

7. Generate prediction

8. Calculate Kalman Gain

$$P_{y_k} = \sum_{i=0}^{2n_w} W_{w,i}^{(c)} (\mathcal{Y}_{i,k|k-1} - \hat{y}_{k|k-1})(\mathcal{Y}_{i,k|k-1} - \hat{y}_{k|k-1})^T + R_n \tag{5.21}$$

$$P_{w_k y_x} = \sum_{i=0}^{2n_w} W_{w,i}^{(c)} (\mathcal{W}_{i,k|k-1} - \hat{w}_{k|k-1})(\mathcal{Y}_{i,k|k-1} - \hat{y}_{k|k-1})^T \tag{5.22}$$

$$K_k = P_{w_k y_x} P_{y_k}^{-1} \tag{5.23}$$

9. Update parameter covariance

$$P_{w_k} = P_{w_{k|k-1}} - K_k P_{y_k} K_k^{-1} \tag{5.24}$$

10. Update parameter estimate

$$\hat{w}_k = \hat{w}_{k|k-1} + K_k (y_k - \hat{y}_{k|k-1}), \tag{5.25}$$

Chapter 6

A Metabolites Example

6.1 Foo section

A metabolite is small structure that is the byproduct of the body's metabolism. To better understand the path of metabolites, an UKF was utilized for biomedical pathway modeling. Data regarding metabolites is highly influenced by noise, which is an UKF can handle while other approaches, such as regression and annealing, fail. Since the model is complex, considering that there are many unknown parameters and many metabolites in the system, utilizing an UKF is more effective than an EKF. Initial data contained information about metabolites at various time steps.

Variables in the Unscented Kalman Filter		
Variable	Description	Dimensions
x	Vector containing state variables	$d_x \times 1$
χ	Sigma Point Matrix	$d_x \times (2d_x + 1)$
F	State Transition Matrix	$d_x \times d_x$
H	Observation Matrix	$d_y \times d_x$
G	Input Matrix	$d_x \times d_u$
u	Input Vector	$d_u \times 1$
P	Covariance matrix	$d_x \times d_x$
k	Time step	1×1
w	Parameter vector	$d_q \times 1$
v	Process Noise Vector	$d_x \times 1$
α	Controls spread of sigma points	1
β	Adjust sigma point weight	$0 \leq \beta$
κ	Sigma point weighting constant	$3 - d_q = -14$
λ	Scaling parameter for weights	1

1. Initialize state vector and covariance

$$\hat{x}_0 = \mathbb{E}[x_0] = \begin{bmatrix} 4 \\ 1 \\ 3 \\ 4 \end{bmatrix} \tag{6.1}$$

$$P_{x_0} = \mathbb{E}[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^T] = \begin{bmatrix} .1 & 0 & 0 & 0 \\ 0 & .1 & 0 & 0 \\ 0 & 0 & .1 & 0 \\ 0 & 0 & 0 & .1 \end{bmatrix} \tag{6.2}$$

2. Calculate sigma points

$$\chi_{\ 0,k-1} = \hat{x}_{k-1} \tag{6.3}$$

$$\chi_{\ i,k-1} = \hat{x}_{k-1} + (\sqrt{(n_x + \lambda_x)P_{x_{k-1}}})_i, \qquad i = 1, \ldots, n_x \tag{6.4}$$

$$\chi_{\ i,k-1} = \hat{x}_{k-1} - (\sqrt{(n_x + \lambda_x)P_{x_{k-1}}})_i, \qquad i = n_x + 1, \ldots, 2n_x, \tag{6.5}$$

3. Calculate the weights for each sigma point

   Weights can have positive or negative values, but will ultimately sum to 1 [**?**]

$$\lambda_x = \alpha_x^2(n_x + \kappa_x) - n_x \tag{6.6}$$

$$W_{x,0}^{(m)} = \frac{\lambda_x}{n_x + \lambda_x} \tag{6.7}$$

$$W_{x,0}^{(c)} = \frac{\lambda_x}{n_x + \lambda_x} + (1 - \alpha_x^2 + \beta_x) \tag{6.8}$$

$$W_{x,i}^{(m)} = W_{x,i}^{(c)} = \frac{\lambda_x}{2(n_x + \lambda_x)}, i = 1, \ldots, 2n_x \tag{6.9}$$

$$W_{x,0}^{\text{aug},(m)} = \frac{\lambda_x}{2n_x + \lambda_x} \tag{6.10}$$

$$W_{x,0}^{\text{aug},(c)} = \frac{\lambda_x}{2n_x + \lambda_x} + (1 - \alpha_x^2 + \beta_x) \tag{6.11}$$

$$W_{x,i}^{\text{aug},(m)} = W_{x,i}^{\text{aug},(c)} = \frac{\lambda_x}{2(2n_x + \lambda_x)}, i = 1, \ldots, 4n_x \tag{6.12}$$

$$\tag{6.13}$$

4. Perform a non-linear transformation on the sigma points

$$\hat{w}_k = \hat{w}_{k-1} \tag{6.14}$$

$$\chi^*_{i,k|k-1} = f(\chi_{\ i,k-1}, \hat{w}_{k-1}) \tag{6.15}$$

5. Calculate the mean and covariance of the transformed sigma points

$$\hat{x}^*{}_{k|k-1} = \sum_{i=0}^{2n_x} W_{x,i}^{(m)} \chi^*{}_{i,k|k-1} \tag{6.16}$$

$$P_{x_{k|k-1}} = \sum_{i=0}^{2n_x} W_{x,i}^{(c)} (\chi^*{}_{i,k|k-1} - \hat{x}^*{}_{k|k-1})(\chi^*{}_{i,k|k-1} - \hat{x}^*{}_{k|k-1})^T + R_v \tag{6.17}$$

6. Re-calculate sigma points

$$\chi\ {}_{i,k|k-1} = \chi^*{}_{i,k|k-1}, i = 0, ..., 2n_x \tag{6.18}$$

$$\chi\ {}_{i,k|k-1} = \chi^*{}_{0,k|k-1} + (\sqrt{(n_x + \lambda_x) P_{x_{k|k-1}}})_i, i = 2n_x + 1, \ldots, 3n_x \tag{6.19}$$

$$\chi\ {}_{i,k|k-1} = \chi^*{}_{0,k|k-1} - (\sqrt{(n_x + \lambda_x) P_{x_{k|k-1}}})_i, i = 3n_x + 1, \ldots, 4n_x \tag{6.20}$$

7. Generate prediction

$$\hat{x}_{k|k-1} = \sum_{i=0}^{4n_x} W_{x,i}^{\text{aug},(m)} \chi\ {}_{i,k|k-1} \tag{6.21}$$

$$\mathcal{Y}_{k|k-1} = h(\chi_{k|k-1}, \widehat{w_k}) \tag{6.22}$$

$$\hat{y}_{k|k-1} = \sum_{i=0}^{4n_x} W_{x,i}^{\text{aug},(m)} \mathcal{Y}\ {}_{i,k|k-1} \tag{6.23}$$

8. Calculate Kalman Gain

$$P_{y_k} = \sum_{i=0}^{4n_x} W_{x,i}^{\text{aug},(c)} (\mathcal{Y}_{i,k|k-1} - \hat{y}_{k|k-1})(\mathcal{Y}_{i,k|k-1} - \hat{y}_{k|k-1})^T + R_n \tag{6.24}$$

$$P_{x_k y_x} = \sum_{i=0}^{4n_x} W_{x,i}^{\text{aug},(c)} (\chi\ {}_{i,k|k-1} - \hat{x}^*{}_{k|k-1})(\mathcal{Y}_{i,k|k-1} - \hat{y}_{k|k-1})^T \tag{6.25}$$

$$K_k = P_{x_k y_x} P_{y_k}^{-1} \tag{6.26}$$

9. Update Estimate and state covariance

$$P_{x_k} = P_{x_{k|k-1}} - K_k P_{y_k} K_k^{-1} \tag{6.27}$$

$$\hat{x}_k = \hat{x}_{k|k-1} + K_k (y_k - \hat{y}_{k|k-1}), \tag{6.28}$$

Chapter 7

A Diabetes Example

Data assimilation (DA) uses mechanistic and empirical data to train machine learning models to make predictions. In the context of type 2 diabetes, DA was effective in providing recommendations on maintaining certain glucose thresholds, smoothing gaps in sparse finger-prick data, and estimating the state of patient phenotypes through constant estimation of physiological parameters. In this thesis, we explore how DA utilizes Bayesian Inference, a process of gauging the probability of an event, in this case glucose levels, given the known probabilities of events from the empirical data. Accuracy also depends on the estimation of relevant physiological processes; the mechanistic model uses differential equations to represent these processes by monitoring 3 variables (remote insulin, plasma insulin, and glucose) in estimating exchange rates. DA allows empirical and physiological data to work in tandem, resulting prognoses that are not only more accurate than other models, but also personalized for the patient and requiring less data. Further research in implementing DA in biomedicine has the potential to provide better patient care.