Data Fusion with Linear Kalman Filter
Dynamic Systems

Steven Dumble, PhD

1 Differential Equations

A differential equation is an equation that relates one or more functions together and their derivatives. Many of the processes that occur in the world can be expressed as differential equations, generally in representation of physical quantities, the deviates represent the rate of change of that quantity and the differential equation describes the relationship between the two. These processes occur in nature, physics, biology, economics, engineering and many other areas. Issac Newton invented calculus and used it to described classical mechanics to express the relationship between position, velocity and acceleration as a differential equation with respect to time. For example, if x is the position of a particle, then its velocity v is how quickly the position is changing with time:

$$v = \dot{x} = \frac{dx}{dt}$$

The same thing can be said about velocity and acceleration a:

$$a = \dot{v} = \frac{dv}{dt}$$

These differential equations are commonly called the equation of motion.

1.1 Differential Equation Types

Differential equations can be broken down into two types, Ordinary Differential Equations (ODE) or Partial Differential Equations (PDE).

Ordinary Differential Equations are an equation which contains only a single variable and its derivations. Most problems encountered in physics are ODE, for example the equations of motion are all a function of derivative(s) of a single variable: position.

$$v = \frac{dx}{dt}$$

$$a = \frac{d^2x}{dt^2}$$

Here the t time is the independent variable and x, v, a are the dependant variables as they depend on t. These are the types of differential equations we will focus on for data fusion and state estimation, as we usually want to track how the variable evolve over time (with time being the independent variable).

Partial Differential Equations are equations which contains multiple variables and their derivations. An example is the Laplace Equation:

$$\frac{\partial u}{\partial x^2} + \frac{\partial u}{\partial y^2} = 0$$

which is used in many areas of physics, such as heat and fluid dynamics.

1.2 Differential Equation Order

The equation order of an differential equation is determined by the highest derivative. If an equation contains only first order derivatives, then it is a first order differential equation, likewise if it contains a second order derivate, then it is a second order differential equation.

So going back to our equations of motion, the first equation for velocity is a first order differential equation, while the equation for acceleration is a second order differential equation.

$$v = \frac{dx}{dt}$$

$$a = \frac{d^2x}{dt^2}$$

It should also be noted that we can turn the second order differential equation for acceleration into a first order differential equation, if we change the independent variable to velocity instead of position:

$$a = \frac{dv}{dt}$$

You can then described the complete second order system as two first order differential equations.

1.3 Linear vs Non-Linear Differential Equations

A linear system is a system which output changes proportionally with in the input, non-linear systems do not. Linear equations conform with the additivity and homogeneity properties:

$$\text{Additivity:} \;\; f(u+v) = f(u) + f(v)$$

$$\text{Homogeneity:} \;\; f(su) = sf(u)$$

A linear ODE is when both sides of the equation are a linear combination of the dependent variable and its derivatives. They can be written in the form:

$$a_n(x)\frac{d^n y}{dx^n} + \cdots + a_1(x)\frac{dy}{dx} + a_0(x)y = b(x)$$

2 Dynamic Systems

A system is a collection of interrelated entities that can be considered as a whole. If the different properties or attributes of that system change with time, then it is considered a dynamic system. The process is how the system evolves over time.

The differential equations that make up the system are the state equations of the dynamic system. The state variables of the system are the dependant variables of the state equations.

Consider the system of time varying, first order differential equations:

$$
\begin{aligned}
\dot{x}_1 &= f_1(t, x_1, x_2, \cdots, x_n, u_1, u_2, \cdots, u_m) \\
\dot{x}_2 &= f_2(t, x_1, x_2, \cdots, x_n, u_1, u_2, \cdots, u_m) \\
\dot{x}_3 &= f_3(t, x_1, x_2, \cdots, x_n, u_1, u_2, \cdots, u_m) \\
&\vdots \\
\dot{x}_n &= f_n(t, x_1, x_2, \cdots, x_n, u_1, u_2, \cdots, u_m)
\end{aligned}
$$

There are m states, so there are m state equations, each equation is a function of time t, and the m number of states x_i and the n number of inputs u_j.
This can compactly be written in vector form as:

$$\dot{x}(t) = f(t, x(t), u(t))$$

where:

$$
\begin{aligned}
x(t) &= [x_1(t), x_2(t), \cdots, x_n(t)]^T \\
u(t) &= [u_1(t), u_2(t), \cdots, u_m(t)]^T
\end{aligned}
$$

Many of the processes that occur in the world can be expressed as linear or non-linear differential equations. We can describe these processes in state-space form which then allow us to use different mathematical tools to extract useful information and perform various analysis on the system. If the know the state of the system for the current time and all the current and future inputs to that system, then we can predict the values of the future states and outputs of that system and a lot more.

3 Continuous and Discrete Time

There are two ways to represent time in dynamic models, both of which are equivalent in terms of results, just different mechanization. Models can be represented in continuous-time or discrete-time.
Continuous time is the natural way of expressing time as a scalar real value t, models of dynamic systems calculate the instantaneous rates or derivatives $\dot{x}(t)$, so the models

are differential equations. To obtain the state of the system for some time in the future, the differential equations have to be integrated to find the state.

$$x(t) = \int_0^t f(t, x(t), u(t))\, dt$$

Discrete time systems break the continuous time t into a number of small steps of time Δt where the current time is expressed as a integer k multiplication of the timestep such that $t = k\Delta t$. So the time, state, and input are all represented at distict steps in time such that t_k, x_k and u_k are the time, state and input respectively for time step k. Discrete time dynamic systems do not calculate the derivate $\dot{x}(t)$ rather they calculate the new state for the next time step $k + 1$ in the future such that:

$$x_{k+1} = f(t_k, x_k, u_k)$$

The integration step is not needed as in continuous-time systems, however the discrete model can only be calculated for discrete chucks of time while the continuous-time system can be calculated for any time.

4 Mathematical Models

The state space representation of dynamic systems can be expressed in continuous time or discrete time. Each time representation many consist of linear or non-linear models that may or may not be time varying.

4.1 Continuous Non-Linear Model

This is the most general form of a continuos time dynamic system, any time of continuous system may be represented in this form.

$$\dot{x}(t) = f(t, x(t), u(t))$$

- Continuous Time (calculates state rates $\dot{x}$)
- Time Varying (function of t)
- General function (represented as a general $f(\cdot)$)

4.2 Continuous Linear Model

This is the version of the continuous time model that can be represented as a linear operation of the state vector and input vector.

$$\dot{x}(t) = \mathbf{A}(t)x(t) + \mathbf{B}(t)u(t)$$

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \vdots \\ \dot{x}_n(t) \end{bmatrix} = \begin{bmatrix} a_{11}(t) & a_{12}(t) & \dots & a_{1n}(t) \\ a_{21}(t) & a_{22}(t) & \dots & a_{2n}(t) \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1}(t) & a_{n2}(t) & \dots & a_{nn}(t) \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix} + \begin{bmatrix} b_{11}(t) & b_{12}(t) & \dots & b_{1r}(t) \\ b_{21}(t) & b_{22}(t) & \dots & b_{2r}(t) \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1}(t) & b_{n2}(t) & \dots & b_{nr}(t) \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_r(t) \end{bmatrix}$$

- Continuous Time (calculates state rates $\dot{x}$)
- Time Varying ($\mathbf{A}$ and $\mathbf{B}$ are a function of t)
- Linear function (represented as a linear matrix operation)

4.3 Discrete Non-Linear Model

This is the most general form of a discrete time dynamic system, any time of discrete system may be represented in this form.

$$x_{k+1} = f(t_k, x_k, u_k)$$

- Discrete Time (calculates next time step state x_{k+1})
- Time Varying (function of timestep k)
- General function (represented as a general $f(\cdot)$)

4.4 Discrete Linear Model

This is the version of the discrete time model that can be represented as a linear operation of the state vector and input vector.

$$x_{k+1} = \mathbf{F}_k x_k + \mathbf{G}_k u_k$$

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}_{k+1} = \begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1n} \\ f_{21} & f_{22} & \cdots & f_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ f_{n1} & f_{n2} & \cdots & f_{nn} \end{bmatrix}_k \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}_k + \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1r} \\ g_{21} & g_{22} & \cdots & g_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ g_{n1} & g_{n2} & \cdots & g_{nr} \end{bmatrix}_k \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_r \end{bmatrix}_k$$

- Discrete Time (calculates next time step state x_{k+1})
- Time Varying ($\mathbf{A}$ and $\mathbf{B}$ are a function of timestep k)
- Linear function (represented as a linear matrix operation)

4.5 Model Summary

A summary of how the different mathematical models of dynamics systems can be represented is shown in the following table.

	Continuous	Discrete
Time invariant		
General	$\dot{x}(t) = f\left(x(t), u(t)\right)$	$x_k = f\left(x_{k-1}, u_{k-1}\right)$
Linear	$\dot{x}(t) = \mathbf{A}x(t) + \mathbf{B}u(t)$	$x_k = \mathbf{F}x_{k-1} + \mathbf{G}u_{k-1}$
Time varying		
General	$\dot{x}(t) = f\left(t, x(t), u(t)\right)$	$x_k = f\left(k, x_{k-1}, u_{k-1}\right)$
Linear	$\dot{x}(t) = \mathbf{A}(t)x(t) + \mathbf{B}(t)u(t)$	$x_k = \mathbf{F}_{k-1}x_{k-1} + \mathbf{G}_{k-1}u_{k-1}$

5 Continuous to Discrete Conversions

In general, an time invariant, continuous-time, deterministic linear system can be represented as:

$$\dot{x}(t) = \mathbf{A}x(t) + \mathbf{B}u(t) \tag{1}$$

where x is the state vector, u is the control vector and the matrices $\mathbf{A}$ and $\mathbf{B}$ are commonly referred to as the system matrix and control matrix respectively. The continuous-time solution is given by:

$$x(t) = e^{\mathbf{A}(t-t_0)}x(t_0) + \int_{t_0}^{t} e^{\mathbf{A}(t-\tau)}\mathbf{B}u(\tau)d\tau \tag{2}$$

where t_0 is the initial time of the system, which is usually just set to be zero. The first term matrix exponential in the above equation is commonly referred to as the state-transition matrix, as it is the matrix that describes how the state changes from an initial condition to a new condition at the end of the time period *without* any control involvement.

To convert the above system into a discrete form, we want to substitute the discrete time relationship $t_k = t_{k-1} + \Delta t$ into the continuous-time solution, so $t_0 = t_{k-1}$ is the previous time step and $t = t_k$ is the current time step with $\Delta t = t_k - t_{k-1}$ so that:

$$x(t_k) = e^{\mathbf{A}(t_k-t_{k-1})}x(t_{k-1}) + \int_{t_{k-1}}^{t_k} e^{\mathbf{A}(t_k-\tau)}\ \mathbf{B}u(\tau)d\tau \tag{3}$$

If we assume that the control input $u(t)$ is held constant over the time period ΔT and let $\alpha = \tau - t_{k-1}$ then:

$$x(t_k) = e^{\mathbf{A}\Delta t}x(t_{k-1}) + \int_0^{\Delta t} e^{\mathbf{A}(\Delta t - \alpha)} d\alpha \, \mathbf{B}u(t_{k-1}) \tag{4}$$

$$= e^{\mathbf{A}\Delta t}x(t_{k-1}) + e^{\mathbf{A}\Delta t} \int_0^{\Delta t} e^{-\mathbf{A}\alpha} d\alpha \, \mathbf{B}u(t_{k-1}) \tag{5}$$

$$= \mathbf{F}x(t_{k-1}) + \mathbf{G}u(t_{k-1}) \tag{6}$$

where the discrete-time matrices are:

$$\mathbf{F} = e^{\mathbf{A}\Delta t} \tag{7}$$

$$\mathbf{G} = \mathbf{F} \int_0^{\Delta t} e^{-\mathbf{A}\alpha} d\alpha \, \mathbf{B} \tag{8}$$

$$= \mathbf{F}\left[\mathbf{I} - e^{-\mathbf{A}\Delta t}\right] \mathbf{A}^{-1}\mathbf{B} \;\text{ if } \mathbf{A}^{-1} \text{ exists} \tag{9}$$

The matrix exponential term $e^{\mathbf{A}t}$ can be computed or approximated a number of different ways. One useful definition is:

$$e^{\mathbf{A}t} = \sum_{j=0}^{\infty} \frac{(\mathbf{A}t)^j}{j!} \tag{10}$$

If the above summation is expanded then it can be written as:

$$e^{\mathbf{A}t} = (\mathbf{A}t)^0 + (\mathbf{A}t)^1 + \frac{(\mathbf{A}t)^2}{2!} + \frac{(\mathbf{A}t)^3}{3!} + \ldots \tag{11}$$

A simple approximation that may be applied is to ignore the higher order terms, such that:

$$e^{\mathbf{A}t} \approx \mathbf{I} + \mathbf{A}t \tag{12}$$

This approximation may or may not be appropriate depending on the system matrix $\mathbf{A}$, if the higher order terms are significant then a different method would be needed to be taken.

6 Numerical Simulation of Models

In order to solve differential equations with respect to time, the integral of the equations needs to be computed. This process is called integration and in simulation it is usually carried out numerically because an exact analytical solution can not usually be found.

6.1 Continuous System Simulation

To simulate continuous systems a numerical integration method needs to be used. The simplest method is called *Euler First Order Integration.* This method approximates the continuous response of the system by assuming the derivatives can be held constant over a small interval in time, such that:

$$x(t + \Delta t) = x(t) + \dot{x}(t)\Delta t$$

The state x is stepped forward in small chucks of time Δt, this process is carried out until time t reaches the end time of the simulation. The steps of the integration process can be summarized as:

1. Assume an Initial Condition $x(0)$
2. Calculate the State Rates $\dot{x}(t) = f(t, x(t), u(t))$
3. Integrate the State $x(t + \Delta t) = x(t) + \dot{x}(t)\Delta t$
4. Update the Time $t \rightarrow t + \Delta T$
5. Repeat steps 2-4 as needed

The characteristics of *Euler First Order Integration* that should be noted are:

- Simplest Integration Method
- Least Accurate
- Time step must be small to capture fast dynamics
- Can be numerically unstable (Solution accuracy get exponentially worse with time)

6.2 Discrete System Simulation

Discrete equations don't need an integration process as the discrete equations themselves already solve for the next step in time, rather these equations only need to be stepped forward in time to get the system response. The simulation of discrete models is very straight forward, no integration is needed, the recursive solution just needs to be stepped forward from the start of the simulation until the desired time.

1. Assume an Initial Condition x_0
2. Step the Solution $x_{k+1} = f(t_k, x_k, u_k)$
3. Update the Timestep $k \rightarrow k + 1$
4. Repeat steps 2-3 as needed