Introduction to Kalman Filtering and Its Applications

Prof Ka C Cheok
Electrical & Systems Engineering Dept,
School of Engineering & Computer Science,
Oakland University,
Rochester, MI 48309-4401

SUMMARY

Kalman filtering (KF) techniques have been applied in numerous scientific investigations and engineering applications. Most tracking systems, such as GPS, radar, sonar, optical, infrared sensor systems, use Kalman filters to smooth information acquired from corrupted data reported from sensors and simultaneously compensate for the motion involved during the measurements. Kalman filter is also used in many systems involving multi-sensor fusion, feedback control and prediction (forecast) schemes. These applications have led to many successful commercial and military applications.

The basic idea behind Kalman filtering is to combine a system model (the differential or difference equation that describes the dynamical motion of the system being investigated) and the measurement model (the observations made of the variables of the system) in an optimum manner, to compute the best estimate of the present state of the system. In certain cases, the KF can also be used to predict future states with reasonable accuracy.

Since 1960, a large number of articles have been published in conferences, journals and books, with regards to Kalman filtering, algorithms and its applications. The vast literature often focuses on advanced variations of the subject that can sometimes be vague to uninitiated readers. In this seminar, the underlying principles of Kalman filter will be explained in depth and made simple and clear with illustrative examples. Practical examples of KF application to state estimation, tracking, control and sensor fusion systems will be presented.

The topics to be covered include: Essential algebraic and differential math; essential statistics; system models; measurement models; observer theory; optimal estimation (derivation of KF); forms of KF (analog & discrete); KF algorithms; extended KF; practical aspects of KF; KF trackers; and applications. Application examples will involve demonstration with Matlab simulation and animation of tracking systems with GPS, radar and/or sonar, and with an actual inertial stabilization platform control system.

CONTENT

SECTION 1

SECTION 2

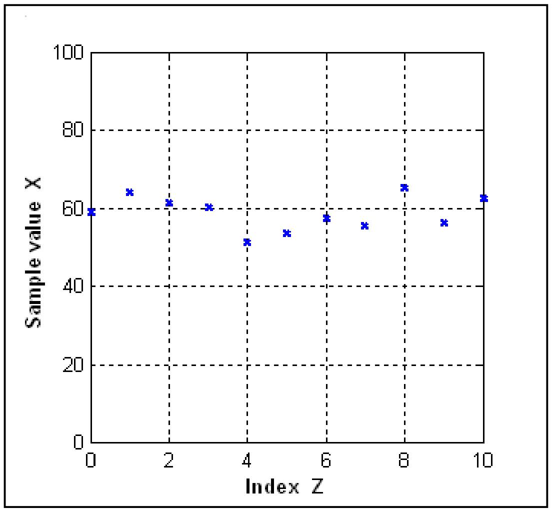
SECTION 3

SECTION 4

1. RANDOM SIGNALS & VARIABLES

1.1. Random Scalar Variable

1.1.1. Means, Covariance and Standard Deviation



$$Z = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{bmatrix}$$

$$X = \begin{bmatrix} 69.72 & 59.46 & 68.05 & 59.02 & 66.09 & 66.57 & 53.32 & 57.87 & 60.41 & 64.36 & 61.38 \end{bmatrix}$$

Sample mean (a.k.a. average or expectation):

$$\overline{X} = mean(X) = \frac{1}{11}\left(69.72 + 59.46 + \quad \ldots \quad + 64.36 + 61.38\right) = 60.08$$

Sample covariance:

$$\begin{aligned} q &= \operatorname{cov}(X) = \sigma^2 \\ &= \frac{1}{11}\left((69.72 - \overline{x})^2 + (59.46 - \overline{x})^2 + \quad \ldots \quad + (64.36 - \overline{x})^2 + (61.38 - \overline{x})^2\right) = 38.5991 \end{aligned}$$

Sample standard deviation:

$$\sigma = \left(\operatorname{cov}(X)\right)^{1/2} = q^{1/2} = 6.21$$

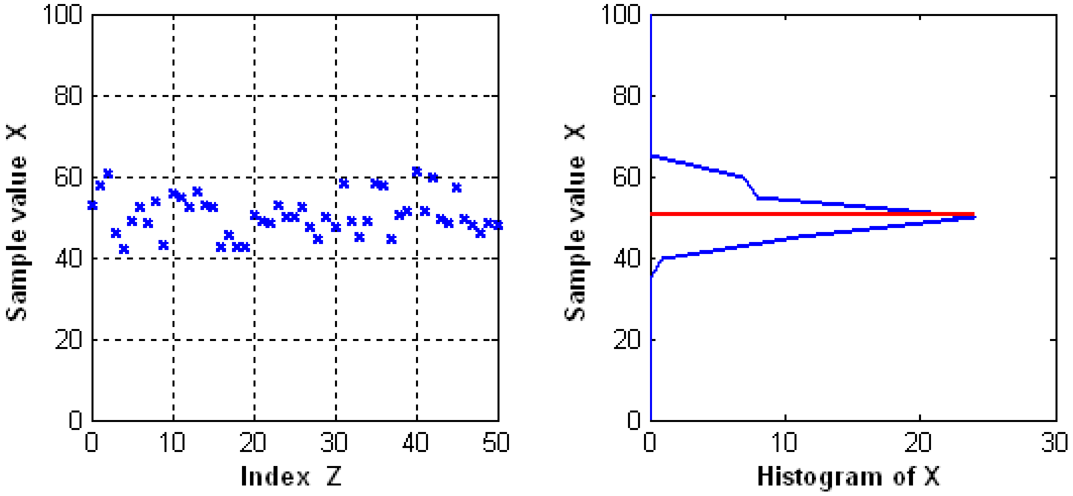
In general, a random scalar variable can be characterized by the following statistics

$$Z = \begin{bmatrix} z_1 & z_2 & \cdots & z_p \end{bmatrix} \qquad X = \begin{bmatrix} x_1 & x_2 & \cdots & x_p \end{bmatrix}$$

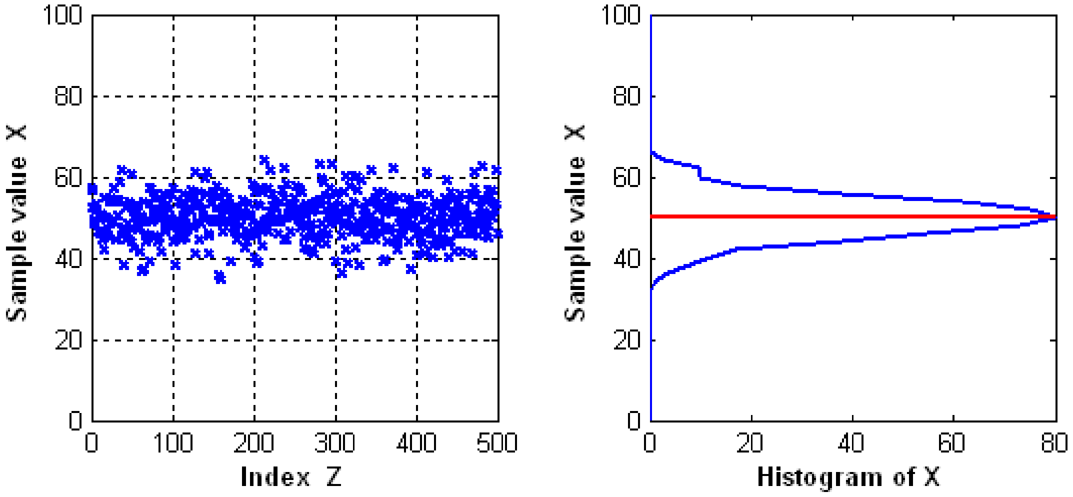
$$\overline{x} = \frac{1}{p}\sum_{i=1}^{p} x_i \qquad q = \frac{1}{p}\left(\sum_{i=1}^{p}(x_i - \overline{x})^2\right) \qquad \sigma = q^{1/2}$$

1.1.2. Histogram and Distribution Function

n = 50 points of random data and its histogram/distribution



500 points of random data and its histogram/distribution

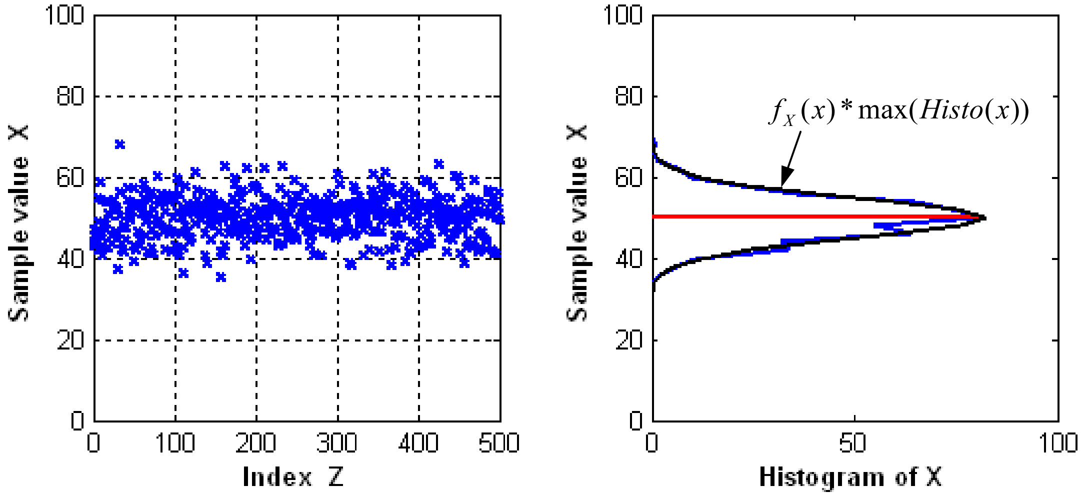


1.1.3. Central Limit Theorem:

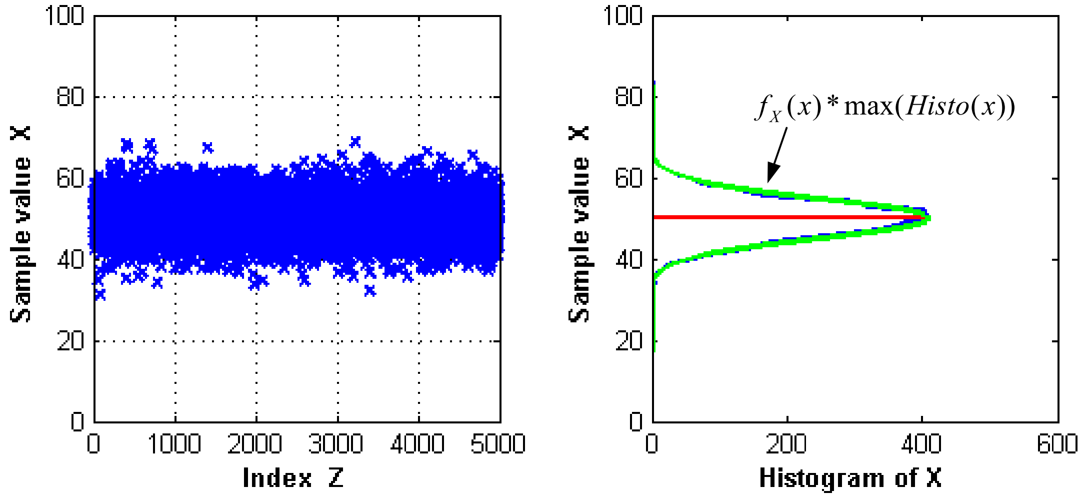
The histogram/distribution of a noisy process is approximately a **Gaussian function** given by

$$f_X(x) = e^{-(x-\bar{x})^2/(2\sigma^2)}$$

500 points of random data, histogram/distribution & Gaussian function



5000 points of random data, histogram/distribution & Gaussian function

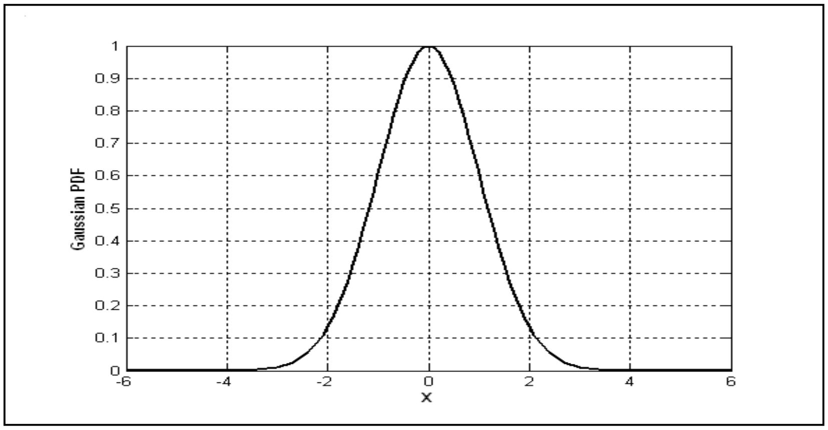


1.1.4. Probability Density Function (PDF)

The probability distribution function of noise is the normalized histogram of the data. Following are examples of various statistical pdf's

Gaussian Distribution.

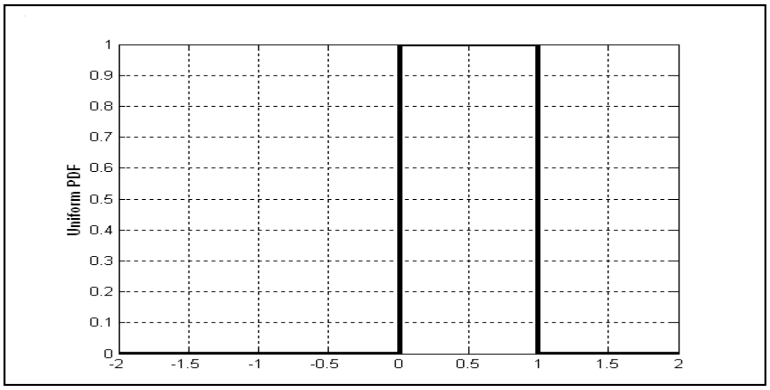
$$f_X(x) = e^{-(x-\overline{x})^2/(2\sigma^2)}$$



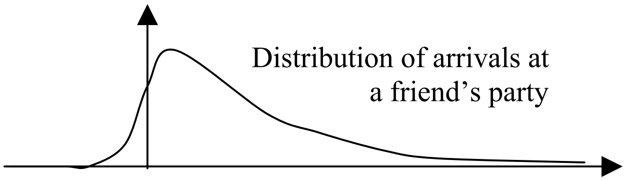
$$f_X(x) = e^{-(x-\overline{x})^2/(2\sigma^2)} \quad with\ \ \overline{x} = 0\ \ \&\ \ \sigma = 1$$

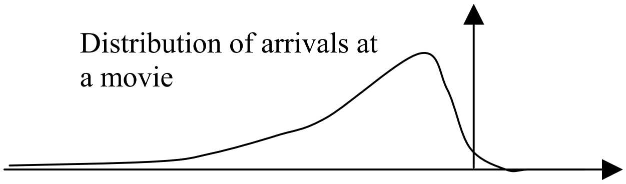
Uniform Distribution.

$$f_X(x) = \begin{cases} 1, & 0 \le x \le 1 \\ 0, & otherwise \end{cases}$$



Rayleigh Distribution, etc.

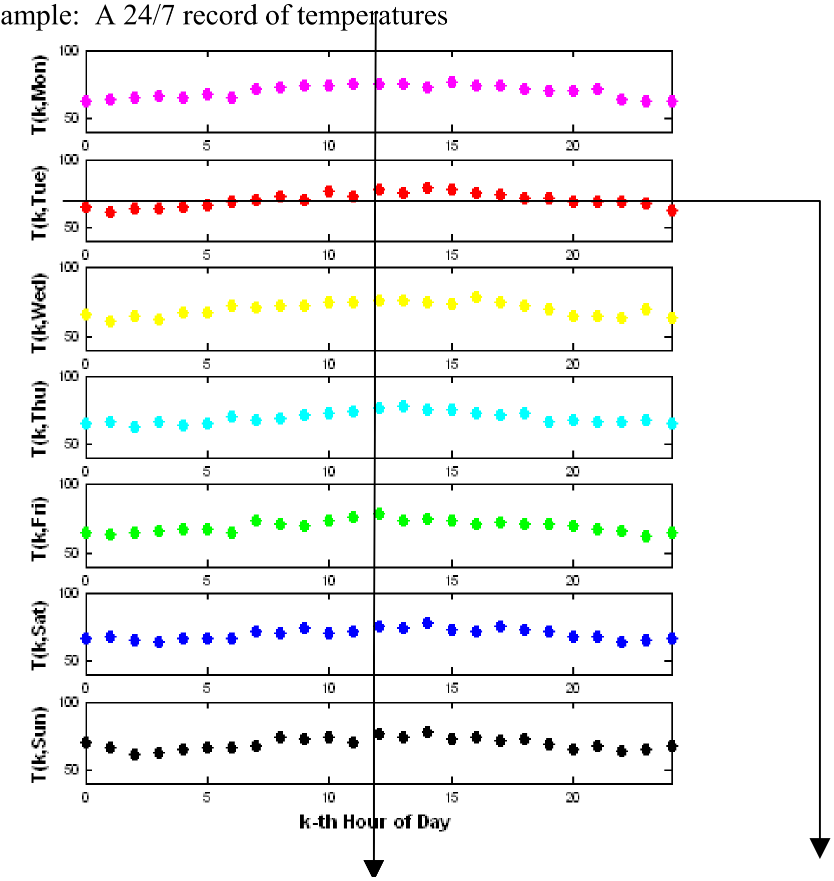


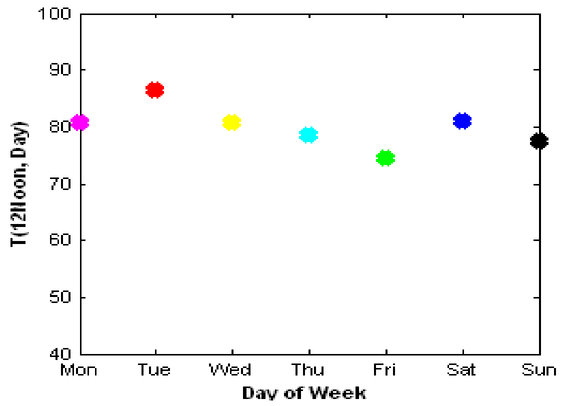


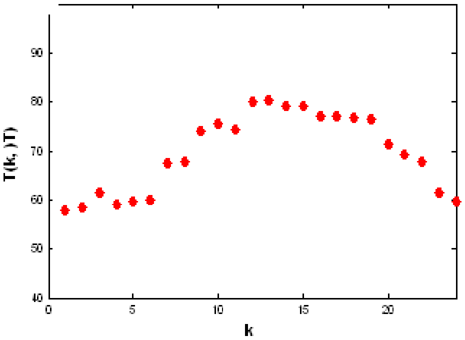
Bottom Line: **A random variable is associated with a mean, a covariace, a std dev and a pdf.**

1.2. Time Series Statistics versus Event Ensemble Statistics

An example: A 24/7 record of temperatures







Time series statistics is based on the data collected over a continual period.

E.g. The average temperature of the day is 70°F.

$$Average\left(T(Tuesday)\right) = \frac{1}{24}\left(T(0,Tue)+T(1,Tue)+\cdots+T(24,Tue)\right)$$

Event ensemble statistics is based on a collection of data for an event.

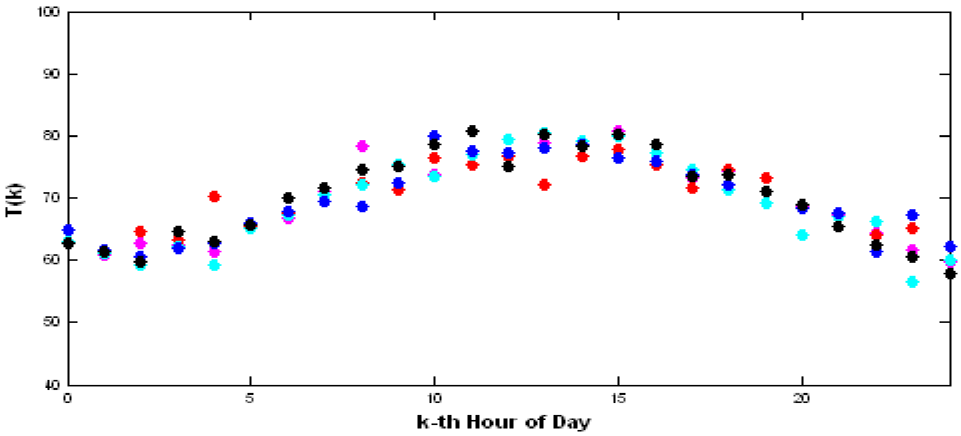
E.g. The **expected** temperature at 12 noon over the week is 82°F.

$$\overline{T}(Noon) = \mathrm{E}\left[T(Noon,Day)\right] = \text{Expectation of } T \text{ at Noon,Any Day}$$

$$= \frac{1}{7}\left(T(Noon,Mon)+T(Noon,Tue)+\cdots+T(Noon,Sun)\right)$$

1.3. Time Series of Scalar Random Variables

If we superpose the T(k, Day) where Day = Mon, Tue, …, Sun, we'd have



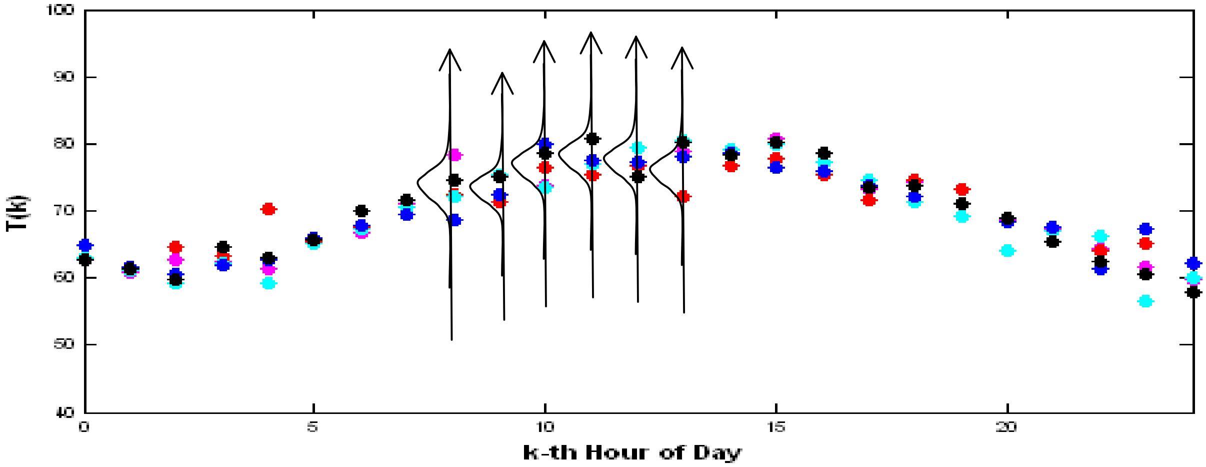
T(k) would be a scalar random variable and has a non-predictable single value at each k. We associate it with event ensemble statistics

$$\bar{T}(k) = \mathrm{E}\left[T(k)\right] = \text{expectation of } T(k) = \text{mean of } T(k)$$

$$\sigma^2_{T(k)} = \mathrm{cov}(T(k)) = \mathrm{E}\left[\left(T(k) - \bar{T}(k)\right)^2\right]$$

$$\sigma_{T(k)} = \text{std dev}(T(k)) = \sqrt{\mathrm{cov}(T(k))}$$

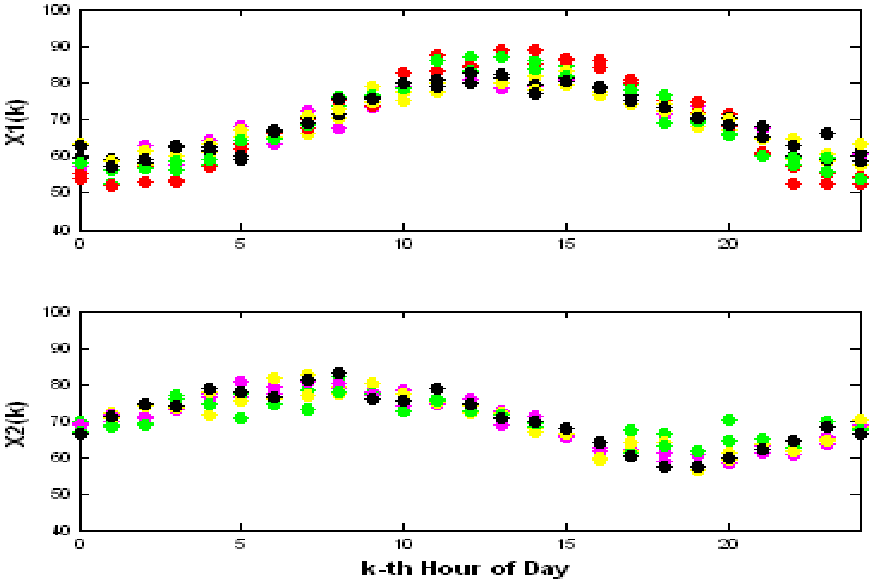
$$f_T(t) = \text{pdf of } T(k) \qquad \text{E.g., } f_T(t) = e^{-\left(T(k)-\bar{T}(k)\right)^2 / \left(2\sigma^2_{\bar{T}(k)}\right)} \text{ if } T(k) \text{ is Gaussian}$$



Bottom Line:	**A random variable, such as T(k), at each instance k is associated with a mean, a covariace, a std dev and a pdf.**

1.4. Time Series of Random Vector Variables

Example:



Let's define the **random vector variable**

$$\mathbf{X}(k) = \begin{bmatrix} X_1(k) \\ X_2(k) \end{bmatrix}.$$

The **means of X(k)** is denoted as

$$\bar{\mathbf{X}}(k) = \mathrm{E}\left[\mathbf{X}(k)\right] = \mathrm{E}\begin{bmatrix} \bar{X}_1(k) \\ \bar{X}_2(k) \end{bmatrix} = \begin{bmatrix} \mathrm{E}[\bar{X}_1(k)] \\ \mathrm{E}[\bar{X}_2(k)] \end{bmatrix}$$

The **covariance of X(k)** is denoted as

$$\begin{aligned}\mathbf{Q}(k) &= \mathrm{E}\left[\begin{bmatrix} X_1(k)-\bar{X}_1(k) \\ X_2(k)-\bar{X}_2(k) \end{bmatrix}\begin{bmatrix} X_1(k)-\bar{X}_1(k) \\ X_2(k)-\bar{X}_2(k) \end{bmatrix}'\right] = \mathrm{E}\left[\begin{bmatrix} X_1(k)-\bar{X}_1(k) \\ X_2(k)-\bar{X}_2(k) \end{bmatrix}\begin{bmatrix} X_1(k)-\bar{X}_1(k) & X_2(k)-\bar{X}_2(k) \end{bmatrix}\right] \\ &= \begin{bmatrix} \mathrm{E}\left(\left(X_1(k)-\bar{X}_1(k)\right)^2\right) & \mathrm{E}\left(\left(X_1(k)-\bar{X}_1(k)\right)\right)\left(X_2(k)-\bar{X}_2(k)\right) \\ \mathrm{E}\left(\left(X_2(k)-\bar{X}_2(k)\right)\right)\left(X_1(k)-\bar{X}_1(k)\right) & \mathrm{E}\left(\left(X_2(k)-\bar{X}_2(k)\right)^2\right) \end{bmatrix} \\ &= \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix}\end{aligned}$$

1.5. Dependency of Random Vector Variables

The noise in signals can be unrelated to or independent of each other, if they are generated from different sources. Examples of independent noise can be found in different sensors; for example, potentiometer tachogenerator, accelerometers, angular rate sensors, radar, sonar, etc.

If the two random variables in $\mathbf{X}(k)=\begin{bmatrix} X_1(k) \\ X_2(k) \end{bmatrix}$ are independent, then their covariance is a diagonal matrix; i.e.,

$$\begin{aligned}\mathbf{Q}(k) &= \begin{bmatrix} \mathrm{E}\left(\left(X_1(k)-\bar{X}_1(k)\right)^2\right) & \mathrm{E}\left(\left(X_1(k)-\bar{X}_1(k)\right)\right)\left(X_2(k)-\bar{X}_2(k)\right) \\ \mathrm{E}\left(\left(X_2(k)-\bar{X}_2(k)\right)\right)\left(X_1(k)-\bar{X}_1(k)\right) & \mathrm{E}\left(\left(X_2(k)-\bar{X}_2(k)\right)^2\right) \end{bmatrix} \\ &= \begin{bmatrix} q_{11} & 0 \\ 0 & q_{22} \end{bmatrix}\end{aligned}$$

This property extends to a n-th dimensional **X***(k)*.

1.6. Summary of Stochastic Signals

Random signals are statistical in nature and can be associated with expectation, such as mean, covariance, standard deviation, probability density distribution, etc.

What time shall we break for lunch? Around noon, give or take five minutes. … There you go… Human behavior is often stochastic in nature,

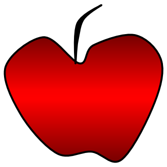
2. MODELS OF DYNAMIC SYSTEMS

A story about Newton

Differential-integral calculus is a tool developed to describe physics of dynamic systems. An apple fell on top of Isaac Newton's and he said "Man, that was painful." I touched his head and felt a bump. "I wonder how many Newton of force acted on to produce the bump on my head." So he decided to calculate the forces involved. That's how we end up with Newtonian law of physics for describing motions of dynamic systems.

Fact about an apple

A force of one Newton is what you feel when you hold up a small Mackintosh apple in your palm. Calculations: 10 Mackintosh apples weight approximately 1Kg, which is 1Kg * 9.81m/s^2 = 9.81N. Therefore, 1 apple exerts approx 1N under gravity in static condition.

Mackintosh  = 0.1 Kg ~ = 1 N

Quote of the day:

Lord Kelvin (the Kelvinator guy) once said:

> **If you can describe a technical or scientific concept with math and numbers, then you can understand the idea in a rigorous manner If not, you cannot precisely explain the notion.**

Since we owe much of founding ideas on refrigeration, air conditioning and heating to Lord Kelvin, I think we can trust him.

1st question of the day:

Can animal count? If so, up to how many?

2nd question of the day:

I think having 10 fingers is a nature's freak. What do you think?
I think we should have 8 fingers, because then the octal system is more natural.
2, 4, 8, 16, 32, 1/2 , 1/4 , 1/8, 1/16 Binary, Octal, Hex, …

Joke of the day:

There are three kinds of people in the world. Those who can count, and those who can't.

2.1. Alpha- Beta Model

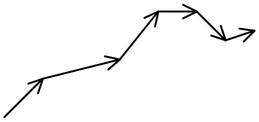
2.1.1. Brownian Motion

A 2nd-order Brownian motion is simply a motion that accelerates with sparing spurious impulses. That is

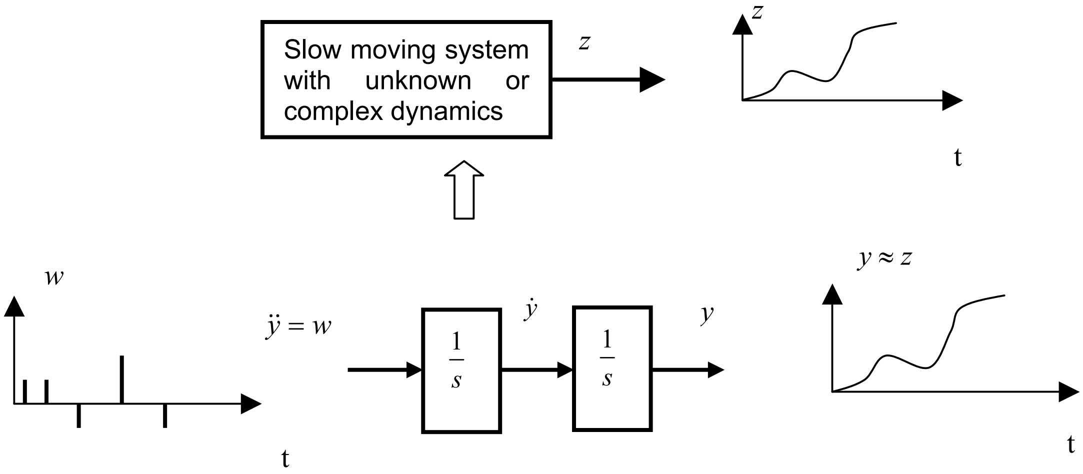
$$\ddot{y} = w, \quad w \sim (0, q) \quad .$$

w is a sparse random noise with zero mean and covariance q .

In 2-D it produces movement that may look like this



This type of characteristics is used to loosely or approximately describe motions of systems that move slowly and whose dynamics are not known or too complicated to model. For example



Slow moving system with unknown or complex dynamics

The motion can be expressed in state-space representation as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

2.1.2. Continuous-time $\alpha - \beta$ model.

In literature on estimation and tracking of targets, this model is referred to as an $\alpha-\beta$ type dynamics. To represent a more stochastic (random) motion, the model is often subjected to influence of output measurement noise and process noise as follows:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + v$$

2.1.3. Discrete-time model.

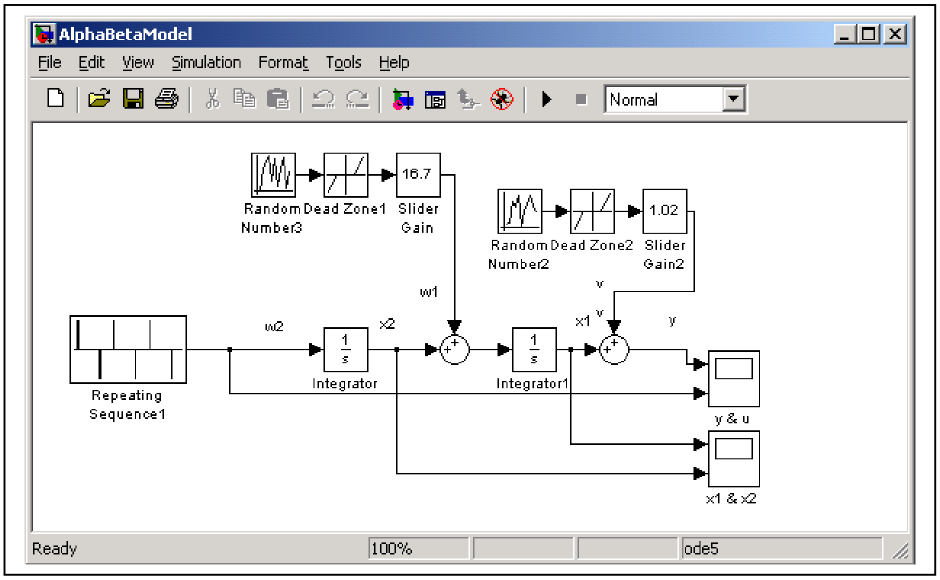
A discrete-time description of the $\alpha-\beta$ model is given by

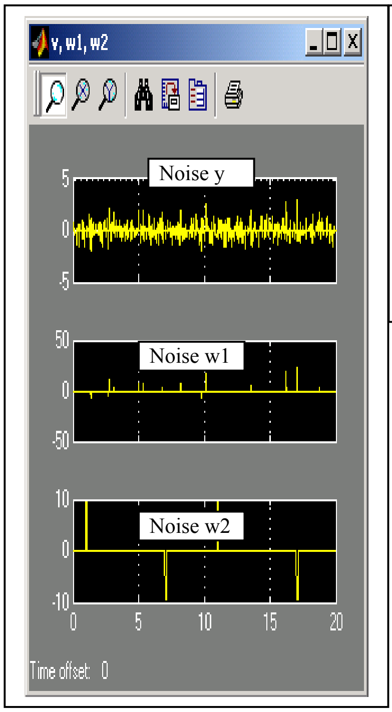
$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_k + \begin{bmatrix} T & 0 \\ 0 & T \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}_k$$

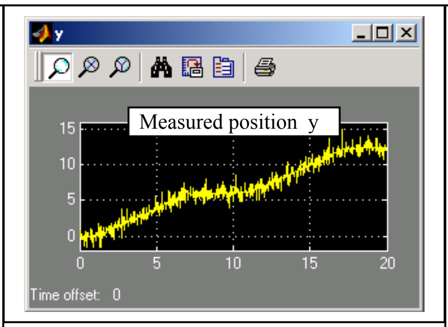
$$y_k = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_k + v_k$$

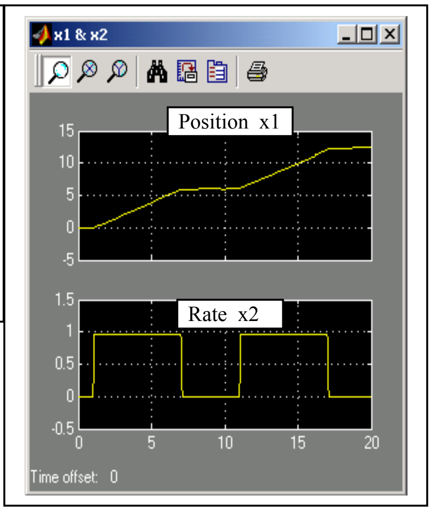
Such model has proven to be very useful for approximating behavior of slow moving system with unknown or complex dynamics. Examples are found in the solutions to many radar applications, prediction and tracking problems.

2.1.4. Simulink demonstration of a $\alpha-\beta$ dynamic model

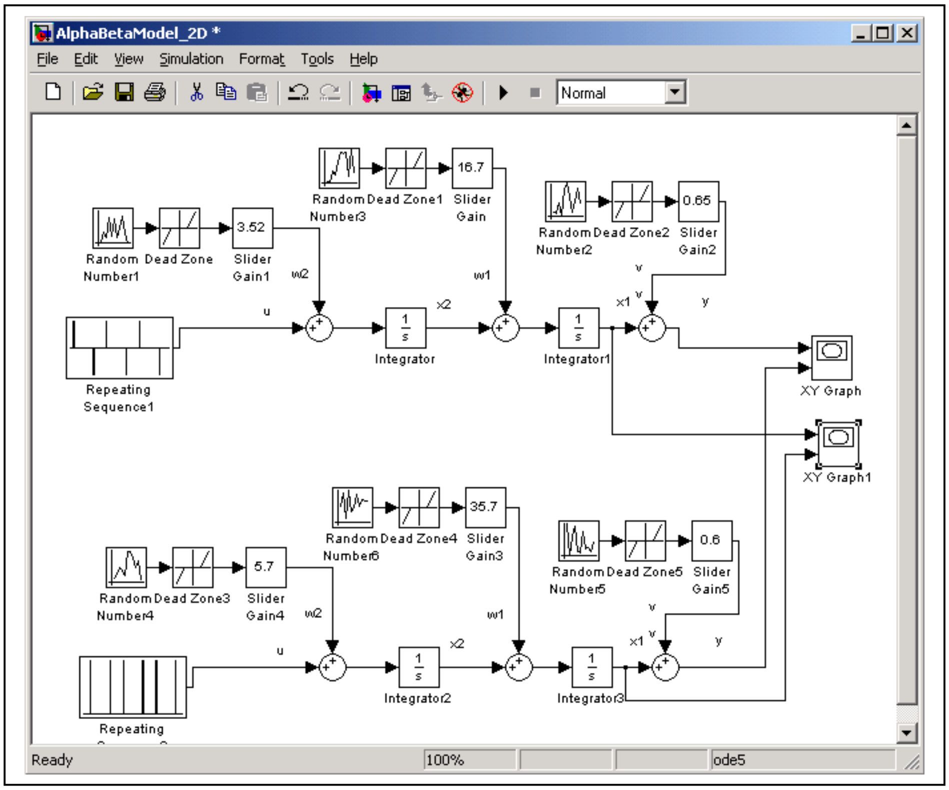


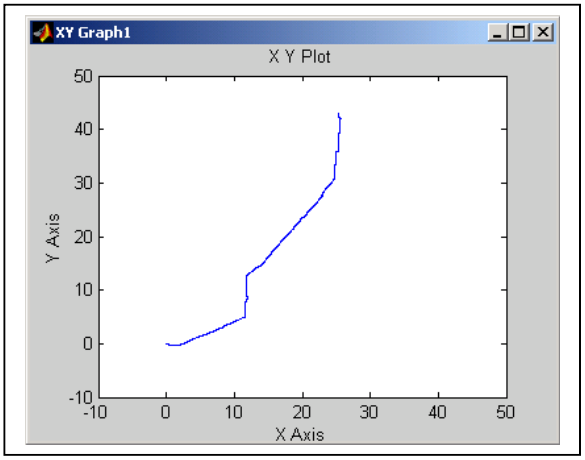


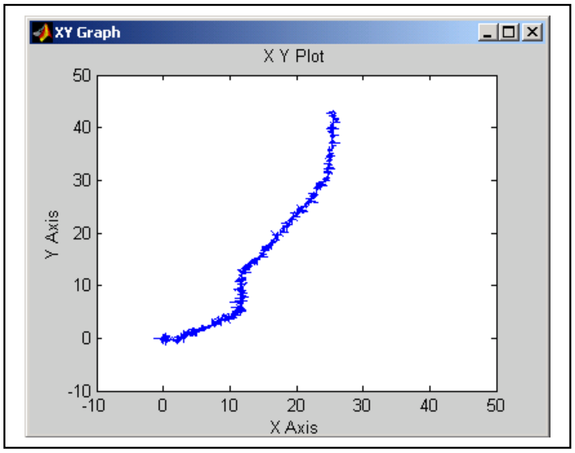




Simulink demonstration of a 2-D Brownian motion with two independent α–β model.







2.2. Physics-Based Model

Physics-based models are equations derived from applying **fundamental principles of science** or **physical laws** to describe the systems at hand. An example below illustrates this.

2.2.1. Math Model for Permanent Magnet DC Motor

2.2.2. Physical components

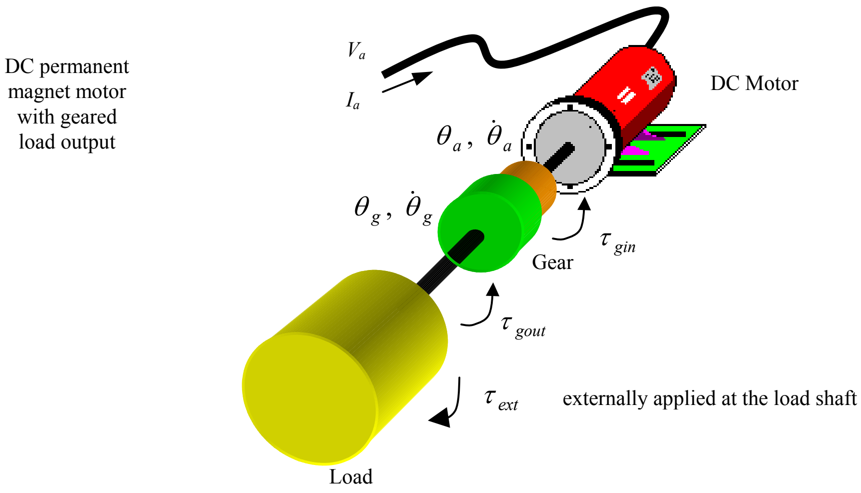

DC permanent magnet motor with geared load output
V_a
I_a
DC Motor
θ_a, $\dot{\theta}_a$
θ_g, $\dot{\theta}_g$
τ_{gin}
Gear
τ_{gout}
τ_{ext}
externally applied at the load shaft
Load

Figure 1. Physical setup of a permanent magnet dc motor driven by a voltage input

2.2.3. Variables & Parameters

Variables

v_a	armature voltage [V]
i_a	armature current [A]
τ_a	torque produced by armature [N. m]
τ_{gin}	torque applied at gear input [N. m]
τ_{gout}	torque produced at gear output [N. m]
τ_{ext}	external torque acting at gear shaft [N. m]
$\dot{\theta}_a$	speed of armature shaft [rad / s]
$\dot{\theta}_g$	speed of gear shaf [rad./s]

Parameters

K_b	back emf coefficient [V. s / rad]
K_t	torque coefficient [N. m / A]
R_a	resistance of armature circuit [Ω]
L_a	inductance of armature circuit [H]
N_g	gear ratio [rad / rad]
J_a	moment of inertia of armature [kg m^2]
J_g	moment of inertia of gear / load [kg m^2]
D_a	coefficient of viscous friction at armature [N m s / rad]
D_g	coefficient of viscous friction at gear [N m s / rad]

2.2.4. Fundamental principles or physical laws

Armature circuit: (Kirchoff voltage law)

$$L_a \frac{di_a}{dt} = -R_a i_a - v_b + v_a$$

Back emf (Electromechanical property)

$$v_b = K_b \dot{\theta}_a$$

Armature torque: (Electromechanical property)

$$\tau_a = K_t i_a$$

Armature motion: (Newton's law)

$$J_a \ddot{\theta}_a = -D_a \dot{\theta}_a + \tau_a - \tau_{sticka} - \tau_{gin}$$

Gear ratio

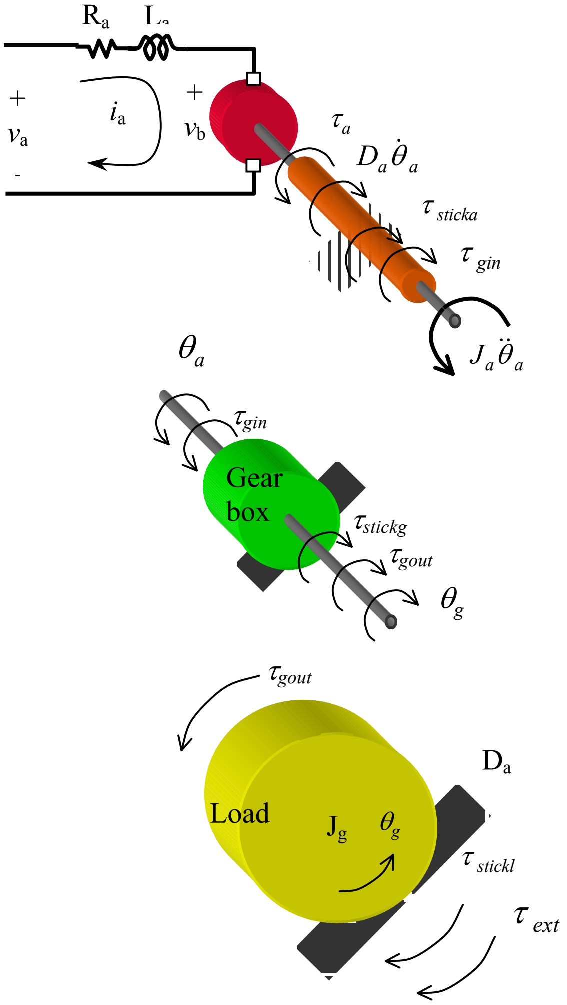
$$\dot{\theta}_a = N_g \dot{\theta}_g$$

Gear torque: Conservation of power (work done)

$$\tau_{gin} \dot{\theta}_a = \tau_{gout} \dot{\theta}_g + \tau_{stickg} \dot{\theta}_g$$

Gear shaft motion

$$J_g \ddot{\theta}_g = -D_g \dot{\theta}_g + \tau_{gout} - \tau_{stickl} - \tau_{ext}$$



Equivalent lumped equation of motion

$$\left(J_g + J_a N_g^2\right)\ddot{\theta}_g = -\left(D_g + D_a N_g^2\right)\dot{\theta}_g + N_g K_t i_a - \tau_{stickl} - N_g \tau_{sticka} - \tau_{stickg} - \tau_{ext}$$

2.2.5. Lumped Parameter Dynamics State Equations

We can show that the above physical relationships can be lumped (combined) into an equivalent dynamics state equations given by:

$$L_a \frac{di_a}{dt} = -R_a i_a - K_{beq}\dot{\theta}_g + v_a$$

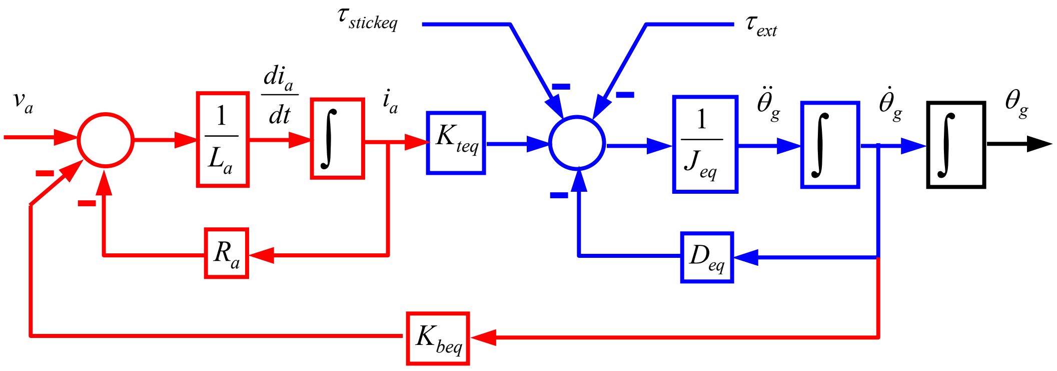
$$J_{eq}\ddot{\theta}_g = -D_{eq}\dot{\theta}_g + K_{teq} i_a - \tau_{stickeq} - \tau_{ext}$$

where

$$J_{eq} = J_g + J_a N_g^2 \qquad D_{eq} = D_g + D_a N_g^2$$

$$K_{beq} = K_b N_g \qquad K_{teq} = N_{eq} K_t \qquad \tau_{stickeq} = N_g \tau_{sticka} + \tau_{stickg} + \tau_{stickl}$$

2.2.6. Block diagram representation of the dynamics.



Equivalent dynamic state diagram

2.2.7. State space equation

A state space model can be expressed as

$$\begin{bmatrix} \ddot{\theta}_g \\ \frac{di_a}{dt} \end{bmatrix} \begin{bmatrix} \frac{-D_{eq}}{J_{eq}} & \frac{K_{teq}}{J_{eq}} \\ \frac{{}_{beq}}{L_a} & \frac{{}_{a}}{L_a} \end{bmatrix} \begin{bmatrix} \dot{\theta}_g \\ i_a \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{L_a} \end{bmatrix} v_a + \begin{bmatrix} -1 \\ 0 \end{bmatrix} \left(\tau_{stickeq} + \tau_{ext}\right)$$

3. DISCRETE-TIME KALMAN FILTERS FOR LINEAR SYSTEMS

3.1. 1st Order Systems

3.1.1. Problem Statement:

Consider a dynamic system whose behavior can be approximately modeled by a stochastic 1st order difference equation

Dynamics equation $x_{k+1} = ax_k + bu_k + gw_k$

Measurement equation $z_k = cx_k + v_k$

where

x_k = scalar state of the system

u_k = scalar control input to the system

w_k = scalar random noise input

z_k = scalar measured output of the system

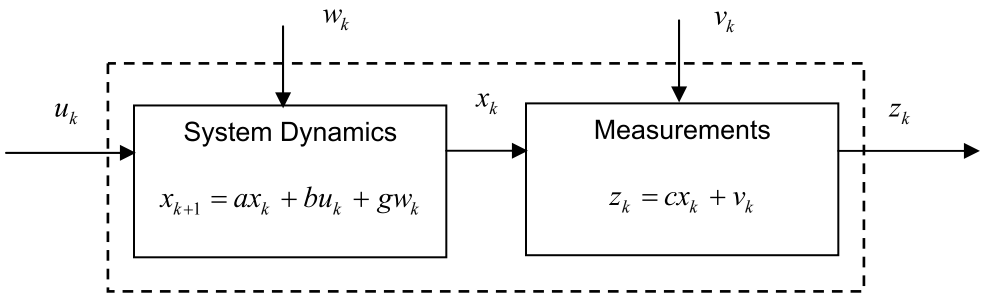
v_k = scalar random noise in the measurement

a = characteristic parameter of the system

b = input gain parameter

g = noise gain parameter

c = output gain parameter



We assume that we know the statistics for the noise w_k & v_k, and initial condition x_0 , which are as follows:

Mean of w_k:	$\bar{w}_k = E(w_k) = 0$	Cov of w_k:	$q = \text{cov}(w_k) = E\left((w_k - \bar{w}_k)^2\right) \geq 0$
Mean of v_k:	$\bar{v}_k = E(v_k) = 0$	Cov of v_k:	$r = \text{cov}(v_k) = E\left((v_k - \bar{v}_k)^2\right) > 0$
Mean of x_0:	$\bar{x}_0 = E(x_0)$	Cov of x_0:	$p_0 = \text{cov}(x_0) = E\left((x_0 - \bar{x}_0)^2\right) > 0$

The random variables w_k , v_k, & x_0 are assumed to be independent.

The system is assumed to be observable and controllable; i.e., $b \neq 0$ & c $\neq 0$.

We would like to derive and formulate a discrete-time Kalman filter to estimate the state x_k from knowing from the noisy measurement y_k and the control input u(k).

3.1.2. Derivation of DTKF:

Let

$$x_{k|k} \triangleq \text{a current estimate of } x_k \text{ based on info available up till time k}$$
$$x_{k+1|k} \triangleq \text{a predicted estimate of } x_{k+1} \text{ based on info available up till time k}$$
$$x_{k|k-1} \triangleq \text{a current estimate of } x_k \text{ based on info available up till time k-1}$$

Using the above measurement equation and dynamic equation, the formulation of a DTKF is as follows:

$$\begin{aligned} &\text{Measurement update:} \qquad && x_{k|k} = x_{k|k-1} + K_k\left(z_k - cx_{k|k-1}\right) \\ &\text{Dynamics update:} && x_{k+1|k} = ax_{k|k} + bu_k \end{aligned}$$

where K_k is yet to be determined. To find K_k, we consider the errors and their covariance between the estimates ($x_{k|k}$, $x_{k+1|k}$, $x_{k|k-1}$) and the true state (x_k)

$$\begin{aligned} \tilde{x}_{k|k} &\triangleq x_k - x_{k|k} \qquad & p_{k|k} &\triangleq \mathrm{E}\left[\tilde{x}_{k|k}^2\right] \\ \tilde{x}_{k+1|k} &\triangleq x_{k+1} - x_{k+1|k} & p_{k+1|k} &\triangleq \mathrm{E}\left[\tilde{x}_{k+1|k}^2\right] \\ \tilde{x}_{k|k-1} &\triangleq x_k - x_{k|k-1} & p_{k|k-1} &\triangleq \mathrm{E}\left[\tilde{x}_{k|k-1}^2\right] \end{aligned}$$

We would like to find a K_k such that these errors will approach zero. We observe that the predicted error behave according to

$$\begin{aligned} \tilde{x}_{k+1|k} &\triangleq x_{k+1} - x_{k+1|k} \\ &= \left(ax_k + bu_k + gw_k\right) - \left(ax_{k|k} + bu_k\right) \\ &= \left(ax_k + bu_k + gw_k\right) - \left(a\left(x_{k|k-1} + K_k\left(z_k - cx_{k|k-1}\right)\right) + bu_k\right) \\ &= \left(ax_k + bu_k + gw_k\right) - \left(a\left(x_{k|k-1} + K_k\left(\left(cx_k + v_k\right) - cx_{k|k-1}\right)\right) + bu_k\right) \\ &= a\left(x_k - x_{k|k-1}\right) + gw_k - aK_kc\left(x_k - x_{k|k-1}\right) - aK_kv_k \\ &= a\left(1 - K_kc\right)\tilde{x}_{k|k-1} + gw_k - aK_kv_k \end{aligned}$$

If we square the predicted error, we'd get

$$\begin{aligned} \left(\tilde{x}_{k+1|k}\right)^2 = {} & \left(a\left(1-K_kc\right)\tilde{x}_{k|k-1}\right)^2 + a\left(1-K_kc\right)\tilde{x}_{k|k-1}gw_k - a\left(1-K_kc\right)\tilde{x}_{k|k-1}aK_kv_k \\ & + gw_ka\left(1-K_kc\right)\tilde{x}_{k|k-1} + \left(gw_k\right)^2 - gw_kaK_kv_k \\ & - aK_kv_ka\left(1-K_kc\right)\tilde{x}_{k|k-1} - aK_kv_kgw_k + \left(aK_kv_k\right)^2 \end{aligned}$$

Note that an expectation operation on this equation will yield covariance for the prediction estimation error. Also if the variables are independent, then the expectation operation yields a zero cross covariance. The result would be

$$\begin{aligned}\mathrm{E}\left(\left(\tilde{x}_{k+1|k}\right)^2\right) &= \mathrm{E}\left(\left(a\left(1-K_k c\right)\tilde{x}_{k|k-1}\right)^2\right) + \mathrm{E}\left(a\left(1-K_k c\right)\tilde{x}_{k|k-1} g w_k\right) - \mathrm{E}\left(a\left(1-K_k c\right)\tilde{x}_{k|k-1} a K_k v_k\right)\\ &+\mathrm{E}\left(g w_k a\left(1-K_k c\right)\tilde{x}_{k|k-1}\right) + \mathrm{E}\left(\left(g w_k\right)^2\right) - \mathrm{E}\left(g w_k a K_k v_k\right)\\ &-\mathrm{E}\left(a K_k v_k a\left(1-K_k c\right)\tilde{x}_{k|k-1}\right) - \mathrm{E}\left(a K_k v_k g w_k\right) + \mathrm{E}\left(\left(a K_k v_k\right)^2\right)\end{aligned}$$

Removing cross-covariance of independent variables yields

$$\mathrm{E}\left(\left(\tilde{x}_{k+1|k}\right)^2\right) = \left(a\left(1-K_k c\right)\right)^2 \mathrm{E}\left(\left(\tilde{x}_{k|k-1}\right)^2\right) + g^2 \mathrm{E}\left(\left(w_k\right)^2\right) + \left(a K_k\right)^2 \mathrm{E}\left(\left(v_k\right)^2\right)$$

In terms of defined covariance, this remaining terms leads to

$$p_{k+1|k} = \left(a\left(1-K_k c\right)\right)^2 p_{k|k-1} + g^2 q + \left(a K_k\right)^2 r$$

We see that there is an opportunity to choose K_k such that the error covariance $p_{k+1|k}$ is minimized. The calculus calls for taking the derivative of $p_{k+1|k}$ with respect to K_k , and setting the derivative to zero.

$$\begin{aligned}\frac{dp_{k+1|k}}{dK_k} &= 2\left(a\left(1-K_k c\right)\right)\left(-ac\right)p_{k|k-1} + 2\left(a K_k\right)ar\\ &= 2aa\left(-cp_{k|k-1} + \left(ccp_{k|k-1}+r\right)K_k\right)\\ &= 0\end{aligned}$$

This yields the so-called Kalman gain

$$K_k = \frac{cp_{k|k-1}}{c^2 p_{k|k-1} + r}$$

It minimizes the covariance of the prediction estimate error $p_{k+1|k} = \mathrm{E}\left[\tilde{x}^2_{k+1|k}\right] = \mathrm{E}\left[\left(x_{k+1} - x_{k+1|k}\right)^2\right]$.

Note that the quadratic (having square terms) nature of the problem ensures that the result is a minimum. This can be verified by checking that the second derivative of $p_{k+1|k}$ with respect to K_k is positive. Indeed,

$$\frac{d^2 p_{k+1|k}}{dK_k^{\ 2}} = 2aa\left(ccp_{k|k-1} + r\right) > 0$$

We have just derived a Kalman filter formula for the 1st order discrete-time system. (Hurray!)

3.1.3. Time-Varying Kalman Filter for a 1st Order System

An algorithm for programming the KF, based on the above derivation, is as follows:

1st order Kalman Filter Algorithm with Time Varying Kalman Gain

Step 1. Set the initial conditions: $x_{0|-1} = \overline{x}_0$ & $p_{0|-1} = p_0$.

Compute the initial Kalman gain $K_0 = \dfrac{cp_{0|-1}}{c^2 p_{0|-1} + r}$

Set the index $k = 0$

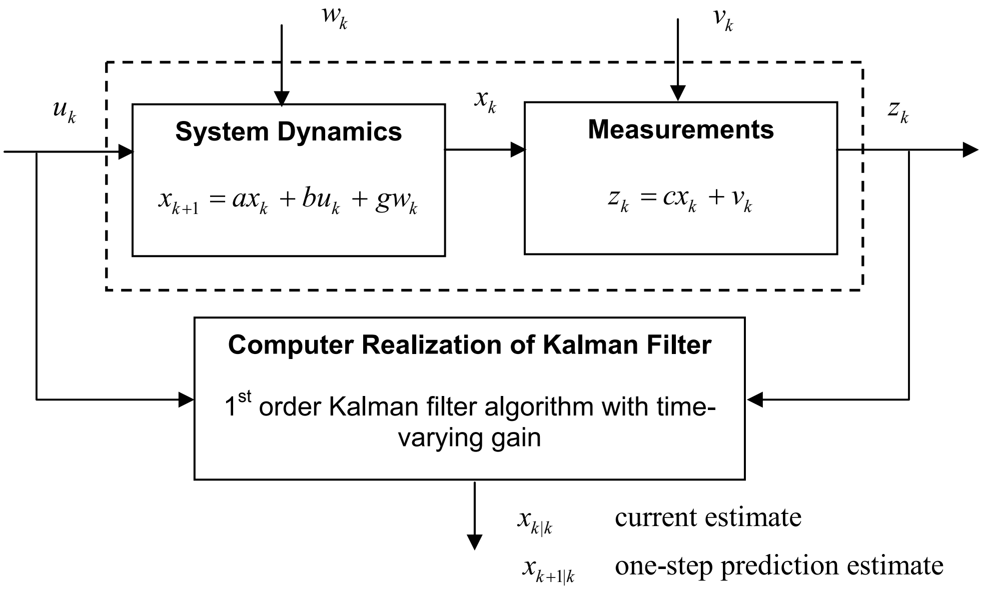
Step 2. Measure z_k & compute the current estimate (measurement update) $x_{k|k} = x_{k|k-1} + K_k\left(z_k - cx_{k|k-1}\right)$

Step 3. Sample u_k and compute the prediction estimate (dynamics update) $x_{k+1|k} = ax_{k|k} + bu_k$

Step 4. Compute the prediction estimate covariance $p_{k+1|k} = \left(a\left(1 - K_k c\right)\right)^2 p_{k|k-1} + g^2 q + \left(aK_k\right)^2 r$

Step 5. Compute the next Kalman gain $K_{k+1} = \dfrac{cp_{k+1|k}}{c^2 p_{k+1|k} + r}$

Step 6. Set k to k +1, and repeat from Step 2.


w_k
v_k
u_k
System Dynamics
$x_{k+1} = ax_k + bu_k + gw_k$
x_k
Measurements
$z_k = cx_k + v_k$
z_k
Computer Realization of Kalman Filter
1st order Kalman filter algorithm with time-varying gain
$x_{k|k}$ current estimate
$x_{k+1|k}$ one-step prediction estimate

3.1.4. Constant Gain Kalman Filter for a 1st Order System

In the case where the system parameters and statistics of random variables have constant parameters, the Kalman filter gain is also a constant parameter. That is, K_k becomes K which is solved from the steady state discrete-time Riccati equation. There is no longer a need to compute the covariance $p_{k+1|k}$. This leads to a very simple KF algorithm

1st order Kalman Filter Algorithm with Constant Kalman Gain

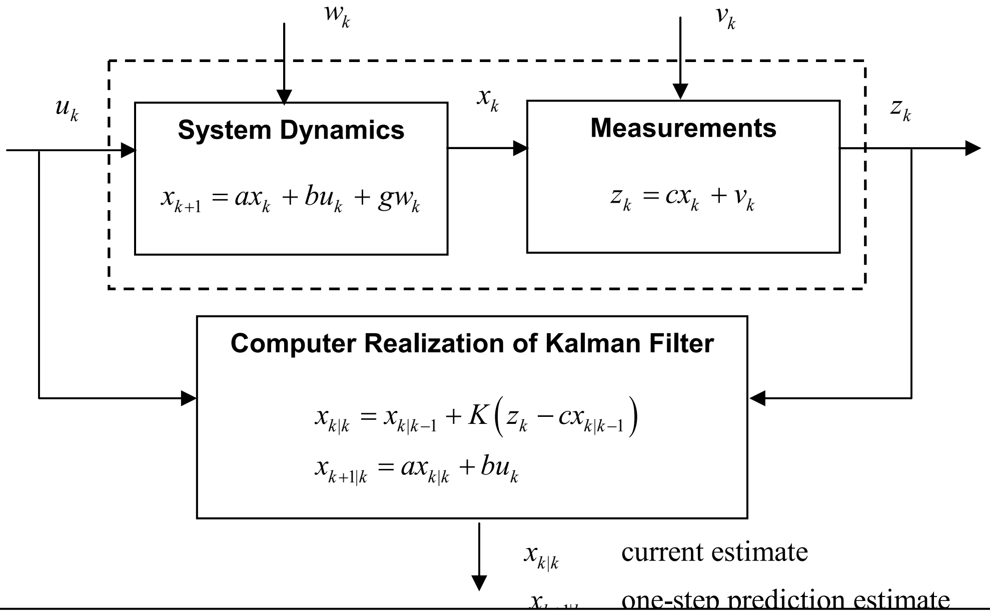
Step 1. Calculate the constant Kalman gain K by solving for p & K from the nonlinear equations

$$p = \left(a\left(1-Kc\right)\right)^2 p + g^2 q + \left(aK\right)^2 r$$
$$K = \frac{cp}{c^2 p + r}$$

Step 2. Measure z_k & compute the current estimate (measurement update) $x_{k|k} = x_{k|k-1} + K\left(z_k - cx_{k|k-1}\right)$

Step 3. Sample u_k and compute the prediction estimate (dynamics update) $x_{k+1|k} = ax_{k|k} + bu_k$

Step 4. Set k to k +1, and repeat from Step 2.


w_k
v_k
u_k
System Dynamics
$x_{k+1} = ax_k + bu_k + gw_k$
x_k
Measurements
$z_k = cx_k + v_k$
z_k
Computer Realization of Kalman Filter
$x_{k|k} = x_{k|k-1} + K\left(z_k - cx_{k|k-1}\right)$
$x_{k+1|k} = ax_{k|k} + bu_k$
$x_{k|k}$ current estimate
$x_{k+1|k}$ one-step prediction estimate

Bottom Line: **To apply a KF, formulate the systems model with random variables in the configuration shown, and specify the parameters *a, b, c, q & r.* The constant KF algorithm is very straightforward. The time-varying KF is slightly more sophisticated as it requires additional computation of the error covariance.**

3.2. 2nd Order Systems

Before we consider the n-th order case, we present the case of a 2nd order system to illustrate a matrix & vector calculus for the DTKF.

3.2.1. Problem Statement:

Consider a 1-input 1-output 2nd discrete-time order system:

Dynamics equation
$$\begin{bmatrix} x_{1,k+1} \\ x_{2,k+1} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_{1,k} \\ x_{2,k} \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} u_k + \begin{bmatrix} w_{1,k} \\ w_{2,k} \end{bmatrix}$$

Measurement equation
$$z_k = \begin{bmatrix} c_1 & c_2 \end{bmatrix} \begin{bmatrix} x_{1,k} \\ x_{2,k} \end{bmatrix} + v_k$$

where

$$\begin{bmatrix} w_{1,k} \\ w_{2,k} \end{bmatrix} \sim \left(\bar{w}_k, \mathbf{Q}\right) \qquad \bar{w}_k = \mathrm{E}\begin{bmatrix} w_{1,k} \\ w_{2,k} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad \mathbf{Q} = \mathrm{cov}\left(\begin{bmatrix} w_{1,k} \\ w_{2,k} \end{bmatrix}\right) = \begin{bmatrix} q_{11} & 0 \\ 0 & q_{22} \end{bmatrix} \geq 0$$

$$v_k \sim \left(\bar{v}_k,\ r\right) \qquad \bar{v}_k = 0 \qquad r > 0$$

$$\begin{bmatrix} x_{1,0} \\ x_{2,0} \end{bmatrix} \sim \left(\bar{x}_0,\ \mathbf{P}_0\right) \qquad \bar{x}_0 = E\left(\begin{bmatrix} x_{1,0} \\ x_{2,0} \end{bmatrix}\right) \qquad \mathbf{P}_0 = \mathrm{cov}\left(\begin{bmatrix} x_{1,0} \\ x_{2,0} \end{bmatrix}\right) = \begin{bmatrix} p_{11} & 0 \\ 0 & p_{22} \end{bmatrix}_0 > 0$$

The random variables are assumed to be independent of each other.

The system is assumed to be observable and controllable.

Formulate a discrete-time Kalman filter to estimate the state x_k from knowing from the noisy measurement z$_k$ and the control input u(k).

3.2.2. Derivation of DTKF:

Let

$$\begin{bmatrix} x_{1,k|k} \\ x_{2,k|k} \end{bmatrix} \text{ a current estimate of } \begin{bmatrix} x_{1,k} \\ x_{2,k} \end{bmatrix} \text{ based on info available up till time k}$$

$$\begin{bmatrix} x_{1,k+1|k} \\ x_{2,k+1|k} \end{bmatrix} \text{ a predicted estimate of } \begin{bmatrix} x_{1,k} \\ x_{2,k} \end{bmatrix} \text{ based on info available up till time k}$$

$$\begin{bmatrix} x_{1,k|k-1} \\ x_{2,k|k-1} \end{bmatrix} \text{ a current estimate of } \begin{bmatrix} x_{1,k} \\ x_{2,k} \end{bmatrix} \text{ based on info available up till time k-1}$$

Using the above measurement equation and dynamic equation, the formulation of a DTKF is as follows:

Measurement update: $$\begin{bmatrix} x_{1,k|k} \\ x_{2,k|k} \end{bmatrix} = \begin{bmatrix} x_{1,k|k-1} \\ x_{2,k|k-1} \end{bmatrix} + \begin{bmatrix} K_{1,k} \\ K_{2,k} \end{bmatrix} \left(z_k - \begin{bmatrix} c_1 & c_2 \end{bmatrix} \begin{bmatrix} x_{1,k|k-1} \\ x_{2,k|k-1} \end{bmatrix} \right)$$

Dynamics update: $$\begin{bmatrix} x_{1,k+1|k} \\ x_{2,k+1|k} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{bmatrix} \begin{bmatrix} x_{1,k|k} \\ x_{2,k|k} \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} u_k$$

where $\begin{bmatrix} K_{1,k} \\ K_{2,k} \end{bmatrix}$ is yet to be determined. Define the errors and their covariance between the estimates and the true state as

$$\begin{bmatrix} \tilde{x}_{1,k|k} \\ \tilde{x}_{2,k|k} \end{bmatrix} \triangleq \begin{bmatrix} x_{1,k} \\ x_{2,k} \end{bmatrix} - \begin{bmatrix} x_{1,k|k} \\ x_{2,k|k} \end{bmatrix} \qquad \begin{bmatrix} p_{11,k|k} & p_{12,k|k} \\ p_{21,k|k} & p_{22,k|k} \end{bmatrix} \triangleq \mathrm{E}\left[\begin{bmatrix} \tilde{x}_{1,k|k} \\ \tilde{x}_{2,k|k} \end{bmatrix} \begin{bmatrix} \tilde{x}_{1,k|k} \\ \tilde{x}_{2,k|k} \end{bmatrix}' \right]$$

$$\begin{bmatrix} \tilde{x}_{1,k+1|k} \\ \tilde{x}_{2,k+1|k} \end{bmatrix} \triangleq \begin{bmatrix} x_{1,k+1} \\ x_{2,k+1} \end{bmatrix} - \begin{bmatrix} x_{1,k+1|k} \\ x_{2,k+1|k} \end{bmatrix} \qquad \begin{bmatrix} p_{11,k+1|k} & p_{12,k+1|k} \\ p_{21,k+1|k} & p_{22,k+1|k} \end{bmatrix} \triangleq \mathrm{E}\left[\begin{bmatrix} \tilde{x}_{1,k+1|k} \\ \tilde{x}_{2,k+1|k} \end{bmatrix} \begin{bmatrix} \tilde{x}_{1,k+1|k} \\ \tilde{x}_{2,k+1|k} \end{bmatrix}' \right]$$

$$\begin{bmatrix} \tilde{x}_{1,k|k-1} \\ \tilde{x}_{2,k|k-1} \end{bmatrix} \triangleq \begin{bmatrix} x_{1,k} \\ x_{2,k} \end{bmatrix} - \begin{bmatrix} x_{1,k|k-1} \\ x_{2,k|k-1} \end{bmatrix} \qquad \begin{bmatrix} p_{11,k|k-1} & p_{12,k|k-1} \\ p_{21,k|k-1} & p_{22,k|k-1} \end{bmatrix} \triangleq \mathrm{E}\left[\begin{bmatrix} \tilde{x}_{1,k|k-1} \\ \tilde{x}_{2,k|k-1} \end{bmatrix} \begin{bmatrix} \tilde{x}_{1,k|k-1} \\ \tilde{x}_{2,k|k-1} \end{bmatrix}' \right]$$

We would like to find a K_k such that these errors will approach zero. It can be shown that the predicted error behave according to

$$\begin{aligned}
\begin{bmatrix} \tilde{x}_{1,k+1|k} \\ \tilde{x}_{2,k+1|k} \end{bmatrix} &\triangleq \begin{bmatrix} x_{1,k+1} \\ x_{2,k+1} \end{bmatrix} - \begin{bmatrix} x_{1,k+1|k} \\ x_{2,k+1|k} \end{bmatrix} \\
&= \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_{1,k} \\ x_{2,k} \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} u_k + \begin{bmatrix} w_{1,k} \\ w_{2,k} \end{bmatrix} - \begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{bmatrix} \begin{bmatrix} x_{1,k|k} \\ x_{2,k|k} \end{bmatrix} - \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} u_k \\
&= \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_{1,k} \\ x_{2,k} \end{bmatrix} + \begin{bmatrix} w_{1,k} \\ w_{2,k} \end{bmatrix} - \begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{bmatrix} \left[\begin{bmatrix} x_{1,k|k-1} \\ x_{2,k|k-1} \end{bmatrix} + \begin{bmatrix} K_{1,k} \\ K_{1,k} \end{bmatrix} \left(z_k - \begin{bmatrix} c_1 & c_2 \end{bmatrix} \begin{bmatrix} x_{1,k|k-1} \\ x_{2,k|k-1} \end{bmatrix} \right) \right] \\
&= \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \left[\begin{bmatrix} x_{1,k} \\ x_{2,k} \end{bmatrix} - \begin{bmatrix} x_{1,k|k-1} \\ x_{2,k|k-1} \end{bmatrix} \right] + \begin{bmatrix} w_{1,k} \\ w_{2,k} \end{bmatrix} - \begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{bmatrix} \left[\begin{bmatrix} K_{1,k} \\ K_{1,k} \end{bmatrix} \left(\left(\begin{bmatrix} c_1 & c_2 \end{bmatrix} \begin{bmatrix} x_{1,k} \\ x_{2,k} \end{bmatrix} + v_k \right) - \begin{bmatrix} c_1 & c_2 \end{bmatrix} \begin{bmatrix} x_{1,k|k-1} \\ x_{2,k|k-1} \end{bmatrix} \right) \right] \\
&= \left[\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} - \begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{bmatrix} \begin{bmatrix} K_{1,k} \\ K_{1,k} \end{bmatrix} \begin{bmatrix} c_1 & c_2 \end{bmatrix} \right] \left[\begin{bmatrix} x_{1,k} \\ x_{2,k} \end{bmatrix} - \begin{bmatrix} x_{1,k|k-1} \\ x_{2,k|k-1} \end{bmatrix} \right] + \begin{bmatrix} w_{1,k} \\ w_{2,k} \end{bmatrix} - \begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{bmatrix} \begin{bmatrix} K_{1,k} \\ K_{1,k} \end{bmatrix} v_k \\
&= \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \left[\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} K_{1,k} \\ K_{1,k} \end{bmatrix} \begin{bmatrix} c_1 & c_2 \end{bmatrix} \right] \left[\begin{bmatrix} x_{1,k} \\ x_{2,k} \end{bmatrix} - \begin{bmatrix} x_{1,k|k-1} \\ x_{2,k|k-1} \end{bmatrix} \right] + \begin{bmatrix} w_{1,k} \\ w_{2,k} \end{bmatrix} - \begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{bmatrix} \begin{bmatrix} K_{1,k} \\ K_{1,k} \end{bmatrix} v_k
\end{aligned}$$

$$\begin{bmatrix}\tilde{x}_{1,k+1|k}\\ \tilde{x}_{2,k+1|k}\end{bmatrix} \square \begin{bmatrix}x_{1,k+1}\\ x_{2,k+1}\end{bmatrix} - \begin{bmatrix}x_{1,k+1|k}\\ x_{2,k+1|k}\end{bmatrix}$$

$$= \begin{bmatrix}a_{11} & a_{12}\\ a_{21} & a_{22}\end{bmatrix}\left[\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} - \begin{bmatrix}K_{1,k}\\ K_{1,k}\end{bmatrix}\begin{bmatrix}c_1 & c_2\end{bmatrix}\right]\begin{bmatrix}\tilde{x}_{1,k|k-1}\\ \tilde{x}_{2,k|k-1}\end{bmatrix} + \begin{bmatrix}w_{1,k}\\ w_{2,k}\end{bmatrix} - \begin{bmatrix}a_{11} & a_{21}\\ a_{12} & a_{22}\end{bmatrix}\begin{bmatrix}K_{1,k}\\ K_{1,k}\end{bmatrix}v_k$$

Eliminating the cross covariance of independent variables, it can be shown that the covariance of $\begin{bmatrix}\tilde{x}_{1,k+1|k}\\ \tilde{x}_{2,k+1|k}\end{bmatrix}$ is given by

$$\mathrm{E}\left(\begin{bmatrix}\tilde{x}_{1,k+1|k}\\ \tilde{x}_{2,k+1|k}\end{bmatrix}\begin{bmatrix}\tilde{x}_{1,k+1|k}\\ \tilde{x}_{2,k+1|k}\end{bmatrix}'\right)$$

$$\square\ \mathrm{E}\left(\begin{bmatrix}a_{11} & a_{12}\\ a_{21} & a_{22}\end{bmatrix}\left[\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} - \begin{bmatrix}K_{1,k}\\ K_{1,k}\end{bmatrix}\begin{bmatrix}c_1 & c_2\end{bmatrix}\right]\begin{bmatrix}x_{1,k|k-1}\\ x_{2,k|k-1}\end{bmatrix}\begin{bmatrix}x_{1,k|k-1}\\ x_{2,k|k-1}\end{bmatrix}'\begin{bmatrix}a_{11} & a_{12}\\ a_{21} & a_{22}\end{bmatrix}\left[\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} - \begin{bmatrix}K_{1,k}\\ K_{1,k}\end{bmatrix}\begin{bmatrix}c_1 & c_2\end{bmatrix}\right]'\right)$$

$$+\mathrm{E}\left(\begin{bmatrix}w_{1,k}\\ w_{2,k}\end{bmatrix}\begin{bmatrix}w_{1,k}\\ w_{2,k}\end{bmatrix}'\right) + \mathrm{E}\left(\left[\begin{bmatrix}a_{11} & a_{21}\\ a_{12} & a_{22}\end{bmatrix}\begin{bmatrix}K_{1,k}\\ K_{1,k}\end{bmatrix}v_k\right]\left[\begin{bmatrix}a_{11} & a_{21}\\ a_{12} & a_{22}\end{bmatrix}\begin{bmatrix}K_{1,k}\\ K_{1,k}\end{bmatrix}v_k\right]'\right)$$

That is the **prediction estimation error covariance** is given by

$$\begin{bmatrix}p_{11,k+1|k} & p_{12,k+1|k}\\ p_{21,k+1|k} & p_{22,k+1|k}\end{bmatrix}$$

$$= \begin{bmatrix}a_{11} & a_{12}\\ a_{21} & a_{22}\end{bmatrix}\left[\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} - \begin{bmatrix}K_{1,k}\\ K_{1,k}\end{bmatrix}\begin{bmatrix}c_1 & c_2\end{bmatrix}\right]\begin{bmatrix}p_{11,k|k-1} & p_{12,k|k-1}\\ p_{21,k|k-1} & p_{22,k|k-1}\end{bmatrix}\left(\begin{bmatrix}a_{11} & a_{12}\\ a_{21} & a_{22}\end{bmatrix}\left[\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} - \begin{bmatrix}K_{1,k}\\ K_{1,k}\end{bmatrix}\begin{bmatrix}c_1 & c_2\end{bmatrix}\right]\right)'$$

$$+\begin{bmatrix}Q_{11}\\ Q_{21}\end{bmatrix}\begin{bmatrix}Q_{12}\\ Q_{22}\end{bmatrix} + \begin{bmatrix}a_{11} & a_{21}\\ a_{12} & a_{22}\end{bmatrix}\begin{bmatrix}K_{1,k}\\ K_{1,k}\end{bmatrix}\begin{bmatrix}r\end{bmatrix}\left(\begin{bmatrix}a_{11} & a_{21}\\ a_{12} & a_{22}\end{bmatrix}\begin{bmatrix}K_{1,k}\\ K_{1,k}\end{bmatrix}\right)'$$

Again we see that there is an opportunity to choose $\begin{bmatrix}K_{1,k}\\ K_{1,k}\end{bmatrix}$, such that some measure of the error covariance $\begin{bmatrix}p_{11,k+1|k} & p_{12,k+1|k}\\ p_{21,k+1|k} & p_{22,k+1|k}\end{bmatrix}$ is minimized. We can use the trace of $\begin{bmatrix}p_{11,k+1|k} & p_{12,k+1|k}\\ p_{21,k+1|k} & p_{22,k+1|k}\end{bmatrix}$ as a measure. The calculus for taking the derivative of $trace\begin{bmatrix}p_{11,k+1|k} & p_{12,k+1|k}\\ p_{21,k+1|k} & p_{22,k+1|k}\end{bmatrix}$ with respect to a vector $\begin{bmatrix}K_{1,k}\\ K_{1,k}\end{bmatrix}$ requires background in gradient matrices and vectors. It can be shown that

$$\frac{d\left(trace\begin{bmatrix}p_{11,k+1|k} & p_{12,k+1|k}\\ p_{21,k+1|k} & p_{22,k+1|k}\end{bmatrix}\right)}{d\begin{bmatrix}K_{1,k}\\ K_{1,k}\end{bmatrix}}$$

$$=\begin{bmatrix}a_{11} & a_{12}\\ a_{21} & a_{22}\end{bmatrix}\left(-2\begin{bmatrix}c_1 & c_2\end{bmatrix}\begin{bmatrix}p_{11,k|k-1} & p_{12,k|k-1}\\ p_{21,k|k-1} & p_{22,k|k-1}\end{bmatrix}\left(\left[\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}-\begin{bmatrix}K_{1,k}\\ K_{1,k}\end{bmatrix}\begin{bmatrix}c_1 & c_2\end{bmatrix}\right]\right)^{'}+2\begin{bmatrix}r\end{bmatrix}\left(\begin{bmatrix}K_{1,k}\\ K_{1,k}\end{bmatrix}\right)^{'}\right)\begin{bmatrix}a_{11} & a_{21}\\ a_{12} & a_{22}\end{bmatrix}^{'}$$

$$=2\begin{bmatrix}a_{11} & a_{12}\\ a_{21} & a_{22}\end{bmatrix}\left(\left(\begin{bmatrix}c_1 & c_2\end{bmatrix}\begin{bmatrix}p_{11,k|k-1} & p_{12,k|k-1}\\ p_{21,k|k-1} & p_{22,k|k-1}\end{bmatrix}\begin{bmatrix}c_1\\ c_2\end{bmatrix}+\begin{bmatrix}r\end{bmatrix}\right)\begin{bmatrix}K_{1,k}\\ K_{1,k}\end{bmatrix}^{'}-\begin{bmatrix}c_1 & c_2\end{bmatrix}\begin{bmatrix}p_{11,k|k-1} & p_{12,k|k-1}\\ p_{21,k|k-1} & p_{22,k|k-1}\end{bmatrix}\right)\begin{bmatrix}a_{11} & a_{21}\\ a_{12} & a_{22}\end{bmatrix}^{'}$$

To find an extremum, we set the derivative to a null vector of appropriate dimension:

$$\left(\begin{bmatrix}c_1 & c_2\end{bmatrix}\begin{bmatrix}p_{11,k|k-1} & p_{12,k|k-1}\\ p_{21,k|k-1} & p_{22,k|k-1}\end{bmatrix}\begin{bmatrix}c_1\\ c_2\end{bmatrix}+\begin{bmatrix}r\end{bmatrix}\right)\begin{bmatrix}K_{1,k}\\ K_{1,k}\end{bmatrix}^{'}-\begin{bmatrix}c_1 & c_2\end{bmatrix}\begin{bmatrix}p_{11,k|k-1} & p_{12,k|k-1}\\ p_{21,k|k-1} & p_{22,k|k-1}\end{bmatrix}=\begin{bmatrix}0 & 0\end{bmatrix}$$

or

$$\begin{bmatrix}K_{1,k}\\ K_{1,k}\end{bmatrix}^{'}=\left(\begin{bmatrix}c_1 & c_2\end{bmatrix}\begin{bmatrix}p_{11,k|k-1} & p_{12,k|k-1}\\ p_{21,k|k-1} & p_{22,k|k-1}\end{bmatrix}\begin{bmatrix}c_1\\ c_2\end{bmatrix}+\begin{bmatrix}r\end{bmatrix}\right)^{-1}\begin{bmatrix}c_1 & c_2\end{bmatrix}\begin{bmatrix}p_{11,k|k-1} & p_{12,k|k-1}\\ p_{21,k|k-1} & p_{22,k|k-1}\end{bmatrix}$$

We again have arrived at the Kalman gain, this time for a second order system:

$$\begin{bmatrix}K_{1,k}\\ K_{1,k}\end{bmatrix}=\begin{bmatrix}p_{11,k|k-1} & p_{12,k|k-1}\\ p_{21,k|k-1} & p_{22,k|k-1}\end{bmatrix}\begin{bmatrix}c_1\\ c_2\end{bmatrix}\left(\begin{bmatrix}c_1 & c_2\end{bmatrix}\begin{bmatrix}p_{11,k|k-1} & p_{12,k|k-1}\\ p_{21,k|k-1} & p_{22,k|k-1}\end{bmatrix}\begin{bmatrix}c_1\\ c_2\end{bmatrix}+\begin{bmatrix}r\end{bmatrix}\right)^{-1}$$

3.2.3. Time-Varying Kalman filter for a 2nd order system

2nd order Kalman Filter Algorithm with Time Varying Kalman Gain

Step 1. Set the initial conditions: $\begin{bmatrix} x_{1,0|-1} \\ x_{2,0|-1} \end{bmatrix} = \begin{bmatrix} \bar{x}_{1,0} \\ \bar{x}_{2,0} \end{bmatrix}$ & $\begin{bmatrix} p_{11,0|-1} & p_{12,0|-1} \\ p_{21,0|-1} & p_{22,0|-1} \end{bmatrix} = \begin{bmatrix} p_{11,0} & p_{12,0} \\ p_{21,0} & p_{22,0} \end{bmatrix}$.

Compute the initial Kalman gain

$$\begin{bmatrix} K_{1,0} \\ K_{2,0} \end{bmatrix} = \frac{\begin{bmatrix} c_1 & c_2 \end{bmatrix}\begin{bmatrix} p_{11,0|-1} & p_{12,0|-1} \\ p_{21,0|-1} & p_{22,0|-1} \end{bmatrix}}{\begin{bmatrix} c_1 & c_2 \end{bmatrix}\begin{bmatrix} p_{11,0|-1} & p_{12,0|-1} \\ p_{21,0|-1} & p_{22,0|-1} \end{bmatrix}\begin{bmatrix} c_1 \\ c_2 \end{bmatrix} + r}$$

Set the index $k = 0$

Step 2. Measure z_k & compute the current estimate (measurement update)

$$\begin{bmatrix} x_{1,k|k} \\ x_{2,k|k} \end{bmatrix} = \begin{bmatrix} x_{1,k|k-1} \\ x_{2,k|k-1} \end{bmatrix} + \begin{bmatrix} K_{1,k} \\ K_{2,k} \end{bmatrix}\left(z_k - \begin{bmatrix} c_1 & c_2 \end{bmatrix}\begin{bmatrix} x_{1,k|k-1} \\ x_{2,k|k-1} \end{bmatrix}\right)$$

Step 3. Sample u_k and compute the prediction estimate (dynamics update)

$$\begin{bmatrix} x_{1,k+1|k} \\ x_{2,k+1|k} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{bmatrix}\begin{bmatrix} x_{1,k|k} \\ x_{2,k|k} \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} u_k$$

Step 4. Compute the prediction estimate covariance

$$\begin{bmatrix} p_{11,k+1|k} & p_{12,k+1|k} \\ p_{21,k+1|k} & p_{22,k+1|k} \end{bmatrix}$$

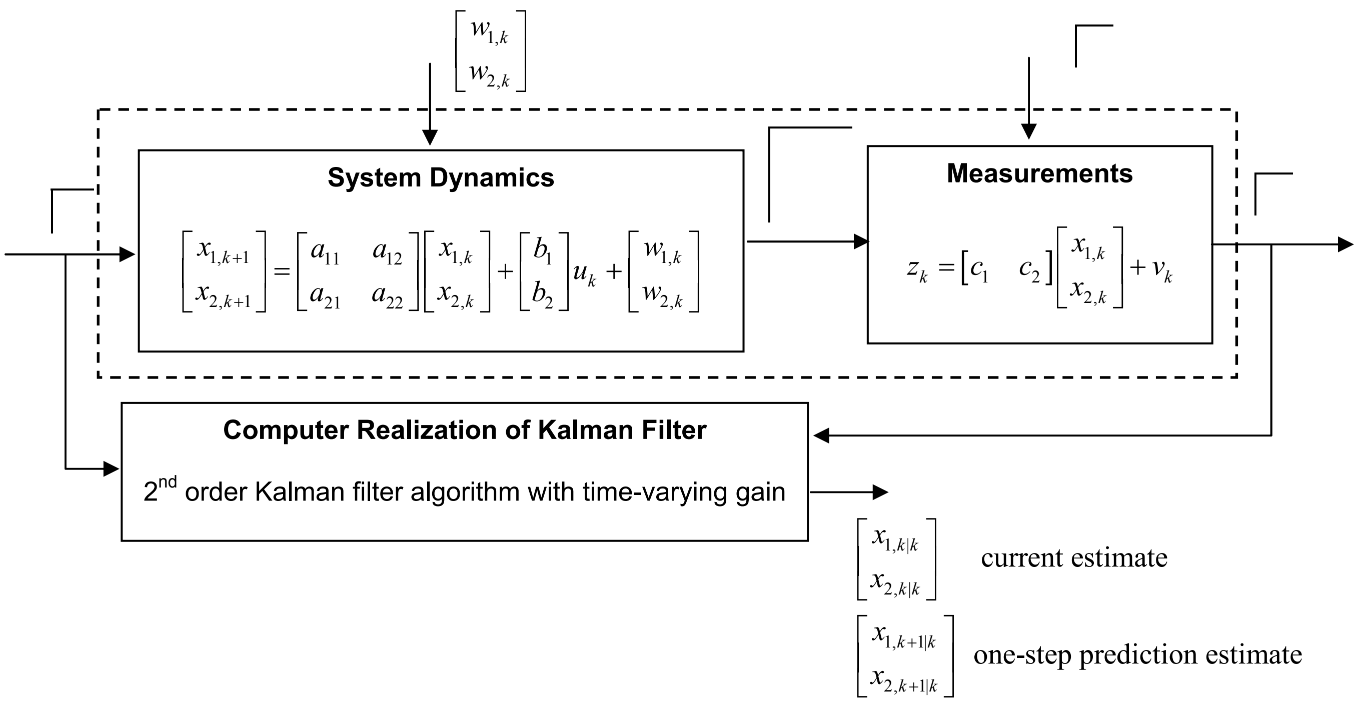
$$= \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}\left[\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} K_{1,k} \\ K_{1,k} \end{bmatrix}\begin{bmatrix} c_1 & c_2 \end{bmatrix}\right]\begin{bmatrix} p_{11,k|k-1} & p_{12,k|k-1} \\ p_{21,k|k-1} & p_{22,k|k-1} \end{bmatrix}\left(\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}\left[\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} K_{1,k} \\ K_{1,k} \end{bmatrix}\begin{bmatrix} c_1 & c_2 \end{bmatrix}\right]\right)'$$

$$+ \begin{bmatrix} Q_{11} \\ Q_{21} \end{bmatrix}\begin{bmatrix} Q_{12} \\ Q_{22} \end{bmatrix} + \begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{bmatrix}\begin{bmatrix} K_{1,k} \\ K_{1,k} \end{bmatrix}\begin{bmatrix} r \end{bmatrix}\left(\begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{bmatrix}\begin{bmatrix} K_{1,k} \\ K_{1,k} \end{bmatrix}\right)'$$

Step 5. Compute the next Kalman gain CHANGE k to k+1

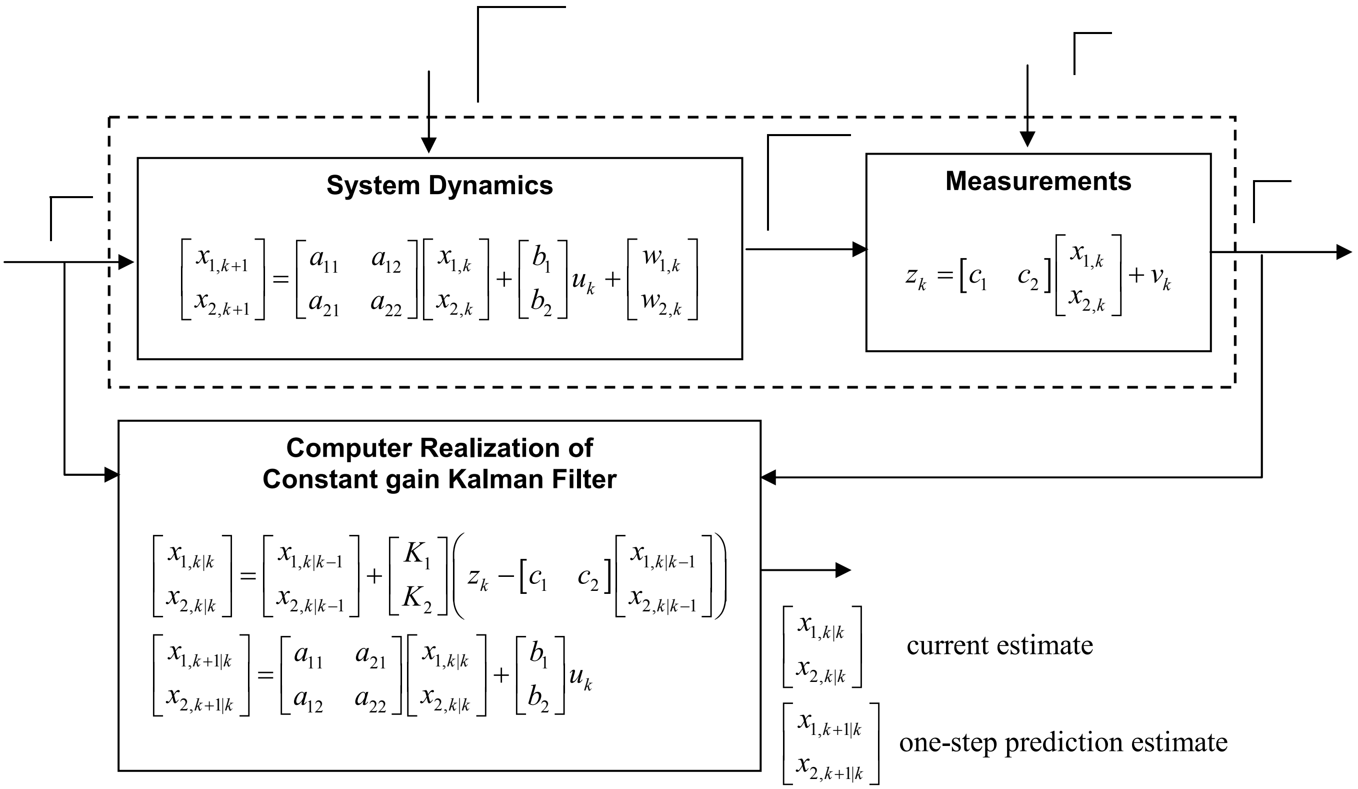
$$\begin{bmatrix} K_{1,k} \\ K_{1,k} \end{bmatrix} = \begin{bmatrix} p_{11,k|k-1} & p_{12,k|k-1} \\ p_{21,k|k-1} & p_{22,k|k-1} \end{bmatrix}\begin{bmatrix} c_1 \\ c_2 \end{bmatrix}\left(\begin{bmatrix} c_1 & c_2 \end{bmatrix}\begin{bmatrix} p_{11,k|k-1} & p_{12,k|k-1} \\ p_{21,k|k-1} & p_{22,k|k-1} \end{bmatrix}\begin{bmatrix} c_1 \\ c_2 \end{bmatrix} + \begin{bmatrix} r \end{bmatrix}\right)^{-1}$$

Step 6. Set k to k +1, and repeat from Step 2.


$\begin{bmatrix} w_{1,k} \\ w_{2,k} \end{bmatrix}$
System Dynamics
$\begin{bmatrix} x_{1,k+1} \\ x_{2,k+1} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_{1,k} \\ x_{2,k} \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} u_k + \begin{bmatrix} w_{1,k} \\ w_{2,k} \end{bmatrix}$
Measurements
$z_k = \begin{bmatrix} c_1 & c_2 \end{bmatrix} \begin{bmatrix} x_{1,k} \\ x_{2,k} \end{bmatrix} + v_k$
Computer Realization of Kalman Filter
2nd order Kalman filter algorithm with time-varying gain
$\begin{bmatrix} x_{1,k|k} \\ x_{2,k|k} \end{bmatrix}$ current estimate
$\begin{bmatrix} x_{1,k+1|k} \\ x_{2,k+1|k} \end{bmatrix}$ one-step prediction estimate

3.2.4. Constant Gain Kalman Filter for a 2nd O

In the case of time-invariant systems, the 2nd order constant gain Kalman filter simplifies to


System Dynamics
$\begin{bmatrix} x_{1,k+1} \\ x_{2,k+1} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x_{1,k} \\ x_{2,k} \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} u_k + \begin{bmatrix} w_{1,k} \\ w_{2,k} \end{bmatrix}$
Measurements
$z_k = \begin{bmatrix} c_1 & c_2 \end{bmatrix} \begin{bmatrix} x_{1,k} \\ x_{2,k} \end{bmatrix} + v_k$
Computer Realization of
Constant gain Kalman Filter
$\begin{bmatrix} x_{1,k|k} \\ x_{2,k|k} \end{bmatrix}$ current estimate
$\begin{bmatrix} x_{1,k+1|k} \\ x_{2,k+1|k} \end{bmatrix}$ one-step prediction estimate

$$\begin{bmatrix} x_{1,k|k} \\ x_{2,k|k} \end{bmatrix} = \begin{bmatrix} x_{1,k|k-1} \\ x_{2,k|k-1} \end{bmatrix} + \begin{bmatrix} K_1 \\ K_2 \end{bmatrix} \left(z_k - \begin{bmatrix} c_1 & c_2 \end{bmatrix} \begin{bmatrix} x_{1,k|k-1} \\ x_{2,k|k-1} \end{bmatrix} \right)$$

$$\begin{bmatrix} x_{1,k+1|k} \\ x_{2,k+1|k} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{bmatrix} \begin{bmatrix} x_{1,k|k} \\ x_{2,k|k} \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} u_k$$

3.3. nth Order Systems

3.3.1. Problem Statement:

A system that can be modeled as a linear n-th order discrete-time system with r-inputs and m-outputs can be expressed similarly in the preceding sections as

$$\text{Dynamics equation} \qquad \mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_\mathbf{k} + \mathbf{w}_\mathbf{k}$$

$$\text{Measurement equation} \qquad \mathbf{z}_k = \mathbf{C}\mathbf{x}_k + \mathbf{v}_k$$

where

$$\begin{array}{lll} \mathbf{w}_k \sim \left(\overline{\mathbf{w}}_k, \mathbf{Q}\right) & \overline{\mathbf{w}}_k = \mathbf{0} & \mathbf{Q} \geq 0 \\ \mathbf{v}_k \sim \left(\overline{\mathbf{v}}_k, \mathbf{R}\right) & \overline{\mathbf{v}}_k = 0 & \mathbf{R} > 0 \\ \mathbf{x}_0 \sim \left(\overline{\mathbf{x}}_0, \mathbf{P}_0\right) & \overline{\mathbf{x}}_0 = E\left(\mathbf{x}_0\right) & \mathbf{P}_0 > 0 \end{array}$$

The dimensions of the variables and parameters are:

$$\begin{array}{llll} \mathbf{x}_k \in \mathbb{R}^{n\times 1} & & \mathbf{A} \in \mathbb{R}^{n\times n} & \mathbf{Q} \in \mathbb{R}^{n\times n} \\ \mathbf{u}_\mathbf{k} \in \mathbb{R}^{r\times 1} & \mathbf{z}_k \in \mathbb{R}^{m\times 1} & \mathbf{B} \in \mathbb{R}^{n\times r} & \mathbf{R} \in \mathbb{R}^{m\times m} \\ \mathbf{w}_\mathbf{k} \in \mathbb{R}^{n\times 1} & \mathbf{v}_k \in \mathbb{R}^{m\times 1} & \mathbf{C} \in \mathbb{R}^{m\times n} & \mathbf{P}_0 \in \mathbb{R}^{n\times n} \end{array}$$

The random variables are assumed to be independent of each other.

The system is assumed to be observable and controllable.

Formulate a discrete-time Kalman filter to estimate the state $\mathbf{x}_k$ from knowing from the noisy measurement $\mathbf{z}_k$ and the control input $\mathbf{u}_\mathbf{k}$.

3.3.2. Formulation of DTKF:

Let

$$\mathbf{x}_{k|k} \triangleq \text{a current estimate of } \mathbf{x}_k \text{ based on info available up till time k}$$

$$\mathbf{x}_{k+1|k} \triangleq \text{a predicted estimate of } \mathbf{x}_k \text{ based on info available up till time k}$$

$$\mathbf{x}_{k|k-1} \triangleq \text{a current estimate of } \mathbf{x}_k \text{ based on info available up till time k-1}$$

Using the above measurement equation and dynamic equation, the formulation of a DTKF is as follows:

$$\text{Measurement update:} \qquad \mathbf{x}_{k|k} = \mathbf{x}_{k|k-1} + \mathbf{K}_k\left(\mathbf{z}_k - \mathbf{C}\mathbf{x}_{k|k-1}\right)$$

$$\text{Dynamics update:} \qquad \mathbf{x}_{k+1|k} = \mathbf{A}\mathbf{x}_{k|k} + \mathbf{B}u_k$$

where $\mathbf{K}_k$ is yet to be determined. Define the errors and their covariance between the estimates and the true state as

$$\begin{aligned} \tilde{\mathbf{x}}_{k|k} &= \mathbf{x}_k - \mathbf{x}_{k|k} \\ \tilde{\mathbf{x}}_{k+1|k} &= \mathbf{x}_{k+1} - \mathbf{x}_{k+1|k} \\ \tilde{\mathbf{x}}_{k|k-1} &= \mathbf{x}_k - \mathbf{x}_{k|k-1} \end{aligned} \qquad \begin{aligned} \mathbf{P}_{k|k} &\square \mathrm{E}\left[\left(\tilde{\mathbf{x}}_{k|k}\right)\left(\tilde{\mathbf{x}}_{k|k}\right)'\right] \\ \mathbf{P}_{k+1|k} &\square \mathrm{E}\left[\left(\tilde{\mathbf{x}}_{k+1|k}\right)\left(\tilde{\mathbf{x}}_{k+1|k}\right)'\right] \\ \mathbf{P}_{k|k-1} &\square \mathrm{E}\left[\left(\tilde{\mathbf{x}}_{k|k-1}\right)\left(\tilde{\mathbf{x}}_{k|k-1}\right)'\right] \end{aligned}$$

We would like to find a $\mathbf{K}_k$ such that these errors will approach zero. It can be shown that the predicted error behave according to

$$\tilde{\mathbf{x}}_{k+1|k} = \mathbf{A}\left[\mathbf{I} - \mathbf{KC}\right]\tilde{\mathbf{x}}_{k|k-1} + \mathbf{w}_k - \mathbf{A}\mathbf{K}_k\mathbf{v}_k$$

Eliminating the cross covariance of independent variables, it can be shown that the **prediction estimation error covariance** is given by

$$\mathbf{P}_{k+1|k} = \mathbf{A}[\mathbf{I} - \mathbf{K}_k\mathbf{C}]\mathbf{P}_{k|k-1}[\mathbf{I} - \mathbf{K}_k\mathbf{C}]'\mathbf{A}' + \mathbf{Q} - \mathbf{A}\mathbf{K}_k\mathbf{R}\mathbf{K}_k'\mathbf{A}'$$

We can choose $\mathbf{K}_k$, such that the trace of the error covariance $\mathbf{P}_k$ is minimized. It can be shown that

$$\frac{d\left(trace\left[\mathbf{P}_{k+1|k}\right]\right)}{d\left[\mathbf{K}_k\right]} = \mathbf{A}\left[-2\mathbf{C}\mathbf{P}_{k|k-1}\left[\mathbf{I} - \mathbf{K}_k\mathbf{C}\right]' + 2\mathbf{R}\mathbf{K}_k'\right]\mathbf{A}' = 2\mathbf{A}\left[[\mathbf{C}\mathbf{P}_{k|k-1}\mathbf{C}' + \mathbf{R}]\mathbf{K}_k' - \mathbf{C}\mathbf{P}_{k|k-1}\right]\mathbf{A}'$$

To find an extremum, we set the derivation to a null vector of appropriate dimension, which then yields the Kalman gain

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{C}'\left[\mathbf{C}\mathbf{P}_{k|k-1}\mathbf{C}' + \mathbf{R}\right]^{-1}$$

3.3.3. Time-Varying Kalman filter for a n^th order system

***n^th* order Kalman Filter Algorithm with Time Varying Kalman Gain**

Step 1. Set the initial conditions: $\mathbf{x}_{0|-1} = \overline{\mathbf{x}}_0 \quad \& \quad \mathbf{P}_{0|-1} = \mathbf{P}_0 .$

Compute the initial Kalman gain $\mathbf{K}_0 = \mathbf{P}_{0|-1}\mathbf{C}'\left[\mathbf{C}\mathbf{P}_{0|-1}\mathbf{C}' + \mathbf{R}\right]^{-1}$

Set the index $k = 0$

Step 2. Measure z_k & compute the current estimate (measurement update)

$$\mathbf{x}_{k|k} = \mathbf{x}_{k|k-1} + \mathbf{K}_k\left(\mathbf{z}_k - \mathbf{C}\mathbf{x}_{k|k-1}\right)$$

Step 3. Sample u_k and compute the prediction estimate (dynamics update)

$$\mathbf{x}_{k+1|k} = \mathbf{A}\mathbf{x}_{k|k} + \mathbf{B}u_k$$

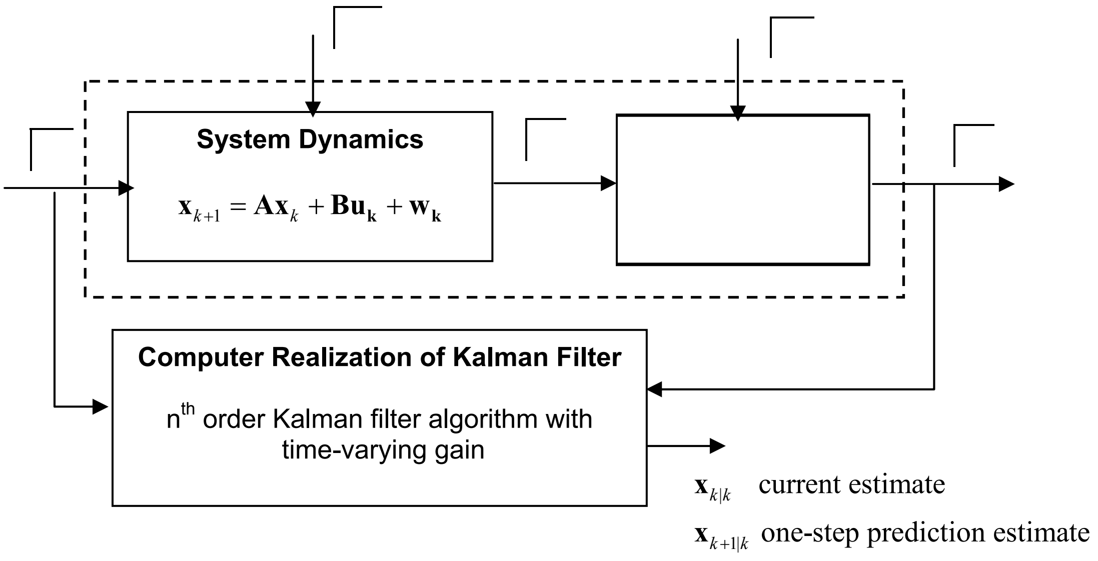
Step 4. Compute the prediction estimate covariance

$$\mathbf{P}_{k+1|k} = \mathbf{A}[\mathbf{I} - \mathbf{K}_k\mathbf{C}]\mathbf{P}_{k|k-1}[\mathbf{I} - \mathbf{K}_k\mathbf{C}]'\mathbf{A}' + \mathbf{Q} - \mathbf{A}\mathbf{K}_k\mathbf{R}\mathbf{K}_k'\mathbf{A}'$$

Step 5. Compute the next Kalman gain

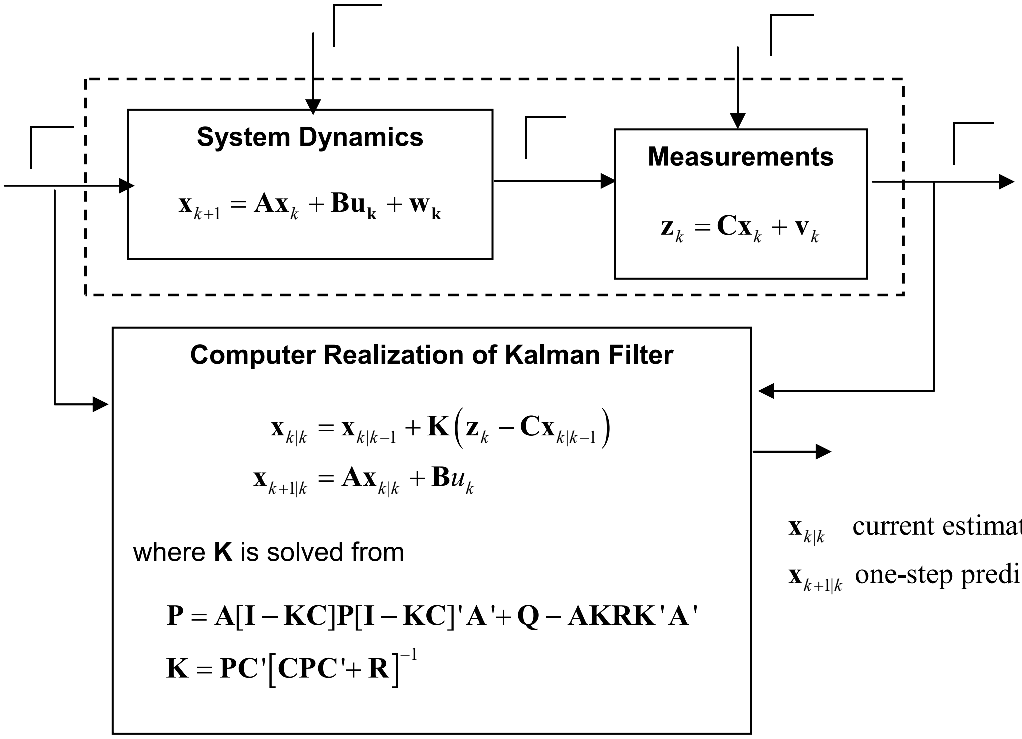
$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{C}'\left[\mathbf{C}\mathbf{P}_{k|k-1}\mathbf{C}' + \mathbf{R}\right]^{-1}$$

Step 6. Set k to k +1, and repeat from Step 2.


System Dynamics
$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_\mathbf{k} + \mathbf{w}_\mathbf{k}$
Computer Realization of Kalman Filter
n^th order Kalman filter algorithm with time-varying gain
$\mathbf{x}_{k|k}$ current estimate
$\mathbf{x}_{k+1|k}$ one-step prediction estimate

3.3.4. Constant Gain Kalman filter for a nth order system

In the case of time-invariant systems, the constant gain Kalman filter simplifies to



$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_\mathbf{k} + \mathbf{w}_\mathbf{k}$$

$$\mathbf{z}_k = \mathbf{C}\mathbf{x}_k + \mathbf{v}_k$$

$$\mathbf{x}_{k|k} = \mathbf{x}_{k|k-1} + \mathbf{K}\left(\mathbf{z}_k - \mathbf{C}\mathbf{x}_{k|k-1}\right)$$
$$\mathbf{x}_{k+1|k} = \mathbf{A}\mathbf{x}_{k|k} + \mathbf{B}u_k$$

where **K** is solved from

$$\mathbf{P} = \mathbf{A}[\mathbf{I} - \mathbf{K}\mathbf{C}]\mathbf{P}[\mathbf{I} - \mathbf{K}\mathbf{C}]'\mathbf{A}' + \mathbf{Q} - \mathbf{A}\mathbf{K}\mathbf{R}\mathbf{K}'\mathbf{A}'$$
$$\mathbf{K} = \mathbf{P}\mathbf{C}'\left[\mathbf{C}\mathbf{P}\mathbf{C}' + \mathbf{R}\right]^{-1}$$

$\mathbf{x}_{k|k}$ current estimate

$\mathbf{x}_{k+1|k}$ one-step prediction estimate

Summary of Discrete-Time Kalman Filter

System Model:

Dynamics equation $\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_\mathbf{k} + \mathbf{w}_\mathbf{k}$

Measurement equation $\mathbf{z}_k = \mathbf{C}\mathbf{x}_k + \mathbf{v}_k$

Kalman Filter Equations:

Measurement update: $\mathbf{x}_{k|k} = \mathbf{x}_{k|k-1} + \mathbf{K}_k\left(\mathbf{z}_k - \mathbf{C}\mathbf{x}_{k|k-1}\right)$

Dynamics update: $\mathbf{x}_{k+1|k} = \mathbf{A}\mathbf{x}_{k|k} + \mathbf{B}u_k$

Kalman gain: $\mathbf{K}_k = \mathbf{P}_k\mathbf{C}'\left[\mathbf{C}\mathbf{P}_k\mathbf{C}' + \mathbf{R}\right]^{-1}$

Prediction error covariance: $\mathbf{P}_{k+1|k} = \mathbf{A}[\mathbf{I} - \mathbf{K}_k\mathbf{C}]\mathbf{P}_{k|k-1}[\mathbf{I} - \mathbf{K}_k\mathbf{C}]'\mathbf{A}' + \mathbf{Q} - \mathbf{A}\mathbf{K}_k\mathbf{R}\mathbf{K}_k^{'}\mathbf{A}'$

Note: Vectors and matrices are a compact means to represent nth order systems and signals.

Bottom Line: **To apply a KF,**

1. **Formulate the systems model with random variables as shown**
2. **Specify the parameters A, B, C, Q, *R,* x_0, P_0.**
3. **Implement the KF algorithm**

The constant KF algorithm is very straightforward

4. EXAMPLE APPLICATIONS OF KALMAN FILTER

When sophisticated system theories are correctly and successfully applied to control complex systems behavior, the resulting performance can be very impressive.

4.1. Application of Kalman filter as an Alpha-Beta Tracker

4.1.1. Objective

Apply Kalman filtering to track and filter an incoming noisy signal. The resultant KF-based Alpha-Beta tracker is compared to conventional low-pass filter using Simulink.

4.1.2. Formulation of Alpha-Beta Tracker

A discrete-time $\alpha - \beta$ model is given by

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_k + \begin{bmatrix} T & 0 \\ 0 & T \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}_k$$

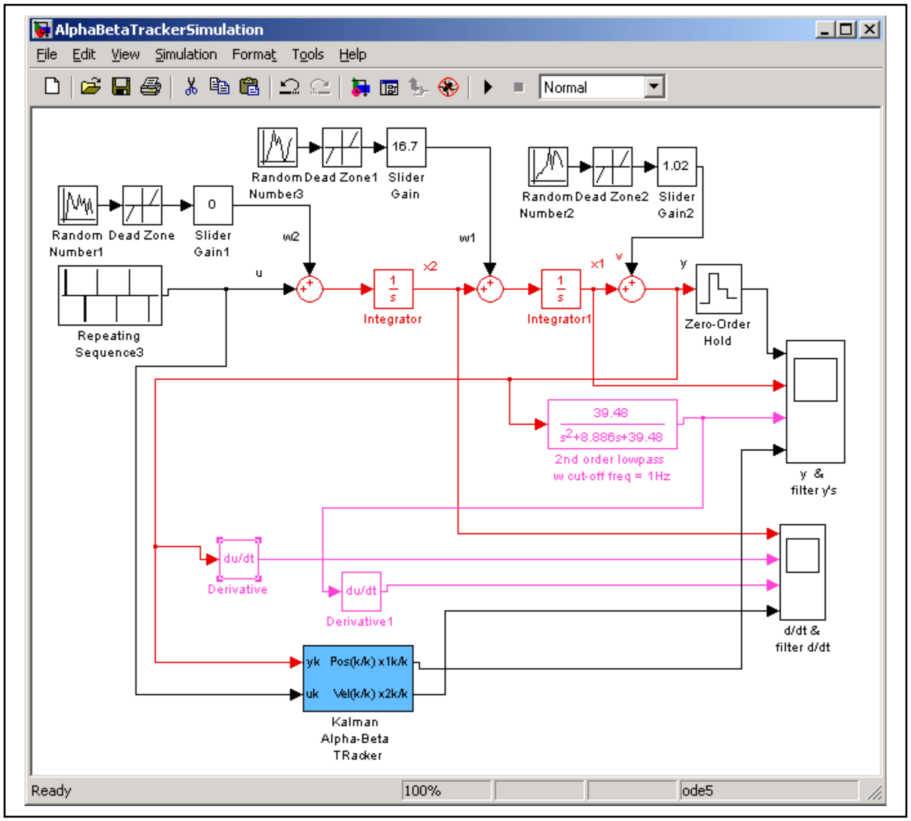
$$y_k = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_k + v_k$$

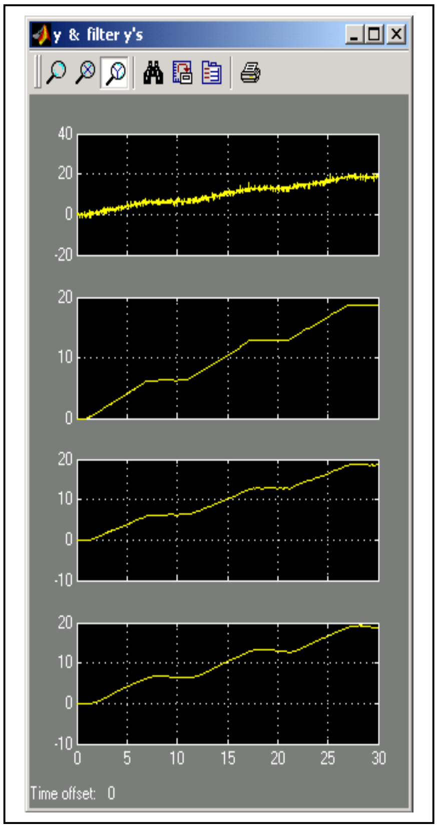
Using the constant gain KF formulation in Section 3, the equations for an $\alpha - \beta$ tracker is given by

$$\boxed{\begin{aligned} \begin{bmatrix} x_{1,k|k} \\ x_{2,k|k} \end{bmatrix} &= \begin{bmatrix} x_{1,k|k-1} \\ x_{2,k|k-1} \end{bmatrix} + \begin{bmatrix} K_1 \\ K_2 \end{bmatrix} \left(y_k - \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_{1,k|k-1} \\ x_{2,k|k-1} \end{bmatrix} \right) \\ \begin{bmatrix} x_{1,k+1|k} \\ x_{2,k+1|k} \end{bmatrix} &= \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{1,k|k} \\ x_{2,k|k} \end{bmatrix} \end{aligned}}$$

where $\begin{bmatrix} K_1 \\ K_2 \end{bmatrix}$ can be solved from the steady state solution of the Kalman gain and presiction error covariance. In Matlab, we can solve the Kalman gain using a function called **dlqe.m.** **>>help dlqe**

4.1.3. Application of Kalman Filter & comparison to signal filter



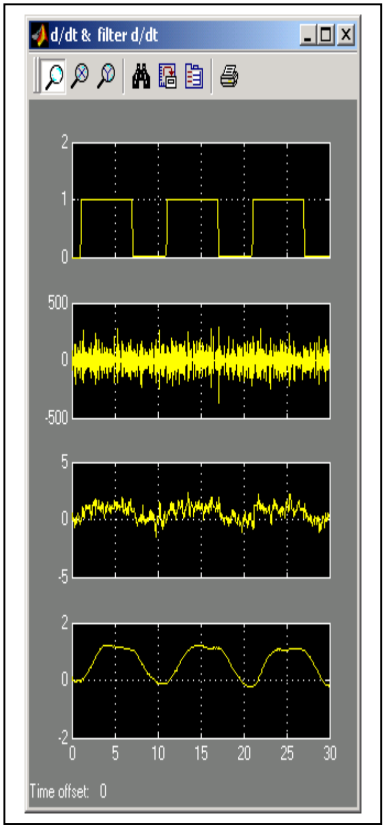


Measured noisy signal

Original clean signal

Lowpass filtered signal

Alpha-beta tracker signal (1st state)



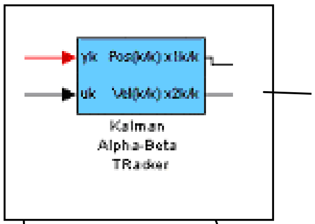
Original signal rate

Rate from differentiating noisy signal

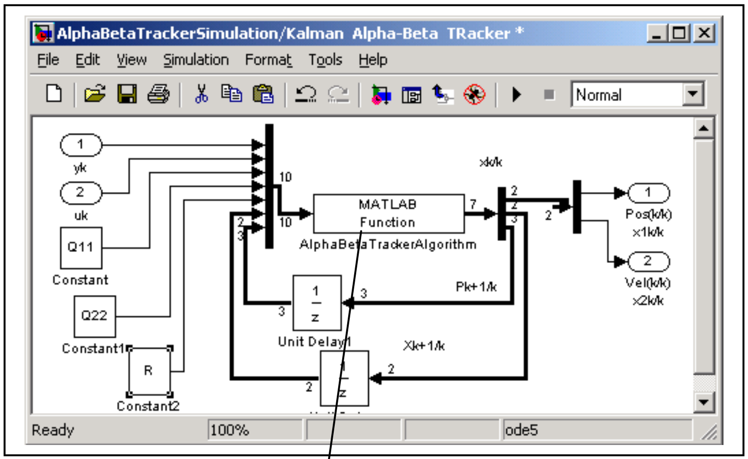
Rate from differenting lowpass filtered signal

Rate produced from Alpha-beta tracker (2nd state)

Implementation of Kalman filter

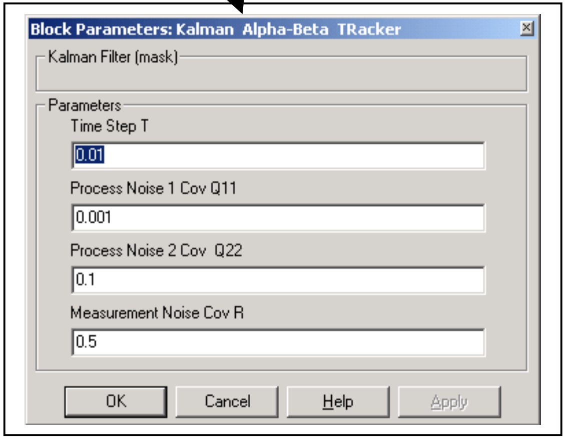


Right click & select Look under mask



Right click & select Edit mask

Double click


Block Parameters: Kalman Alpha-Beta TRacker

Kalman Filter (mask)

Parameters

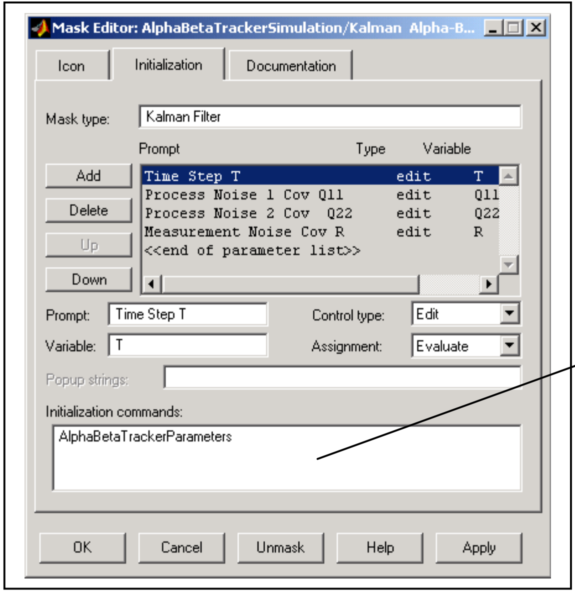
Time Step T: 0.01

Process Noise 1 Cov Q11: 0.001

Process Noise 2 Cov Q22: 0.1

Measurement Noise Cov R: 0.5

```
function [Out] = AlphaBetaTrackerAlgorithm (In);

global A B C D Q %R

Yk = In(1);
Uk = In(2);
q11 = In(3);
q22 =In(4);
R = In(5);
Xkkm1 = [In(6); In(7)];
p11 = In(8);
p12 = In(9);
p22 = In(10);

Q = [q11  0;  0  q22];

Pkkm1 = [p11 p12; p12 p22];
Kk = Pkkm1*C'/(C*Pkkm1*C'+R);
Ykkm1 = C*Xkkm1;
Xkk = Xkkm1 + Kk*(Yk-C*Xkkm1);

Xkp1k = A*Xkk + B*Uk ;
Pkk = (eye(2) - Kk*C )*Pkkm1;
Pkp1k = A'*Pkk*A + Q;

Out = [Xkk ; Xkp1k; Pkp1k(1,1); Pkp1k(1,2); Pkp1k(2,2)];
```


Mask Editor: AlphaBetaTrackerSimulation/Kalman Alpha-B...

Mask type: Kalman Filter

```
Prompt                      Type      Variable
Time Step T                 edit      T
Process Noise 1 Cov Q11     edit      Q11
Process Noise 2 Cov  Q22    edit      Q22
Measurement Noise Cov R     edit      R
<<end of parameter list>>
```

Prompt: Time Step T Control type: Edit

Variable: T Assignment: Evaluate

Popup strings:

Initialization commands: AlphaBetaTrackerParameters

```
% AUTOMATIC INITIALIZATION
% Kalman Filter Matrices for Alpha Beta Tracker

global A B C D Q R T

T = 0.01;
A = [1 T;0 1];
B = [0 0]';
C = [1 0];
D = 0;
Q = [0.01 0;0 0.0001];
R = 10;
```

You may tune the gains for the Kalman filter by adjusting covariance Q & R, to improve its performance, even though they are supposed to represent covariance of the variables.

4.2. Application of KF to a Stabilization Platform

4.2.1. Objective

The configuration of a self-leveling stabilized platform system is shown in Figure 1 below. The control objective is to automatically level the top platform (parallel to the horizontal) and keep it horizontal in the presence of base motion.

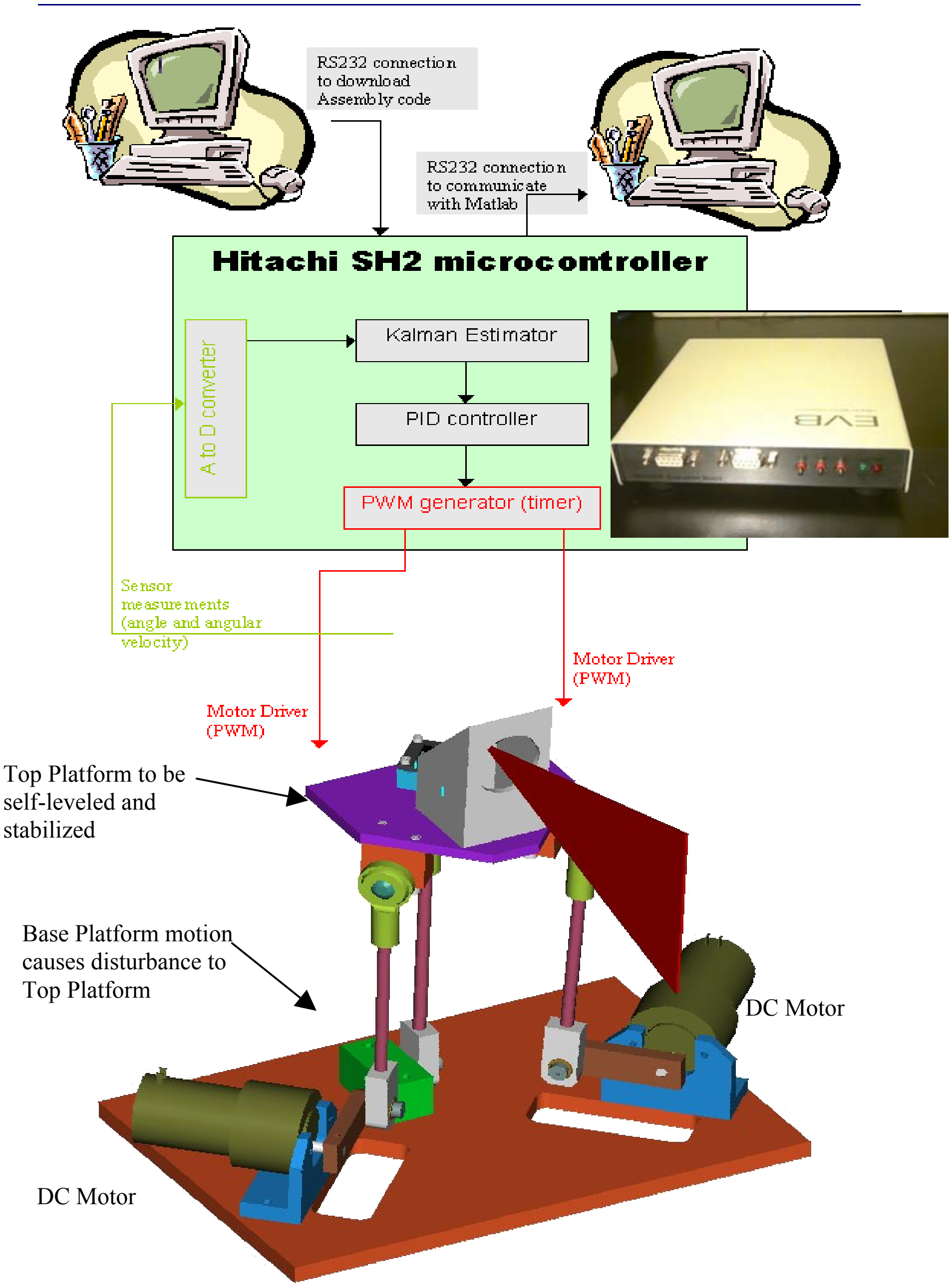

RS232 connection to download Assembly code
RS232 connection to communicate with Matlab
Hitachi SH2 microcontroller
Kalman Estimator
PID controller
PWM generator (timer)
A to D converter
EVB
Sensor measurements (angle and angular velocity)
Motor Driver (PWM)
Motor Driver (PWM)
Top Platform to be self-leveled and stabilized
Base Platform motion causes disturbance to Top Platform
DC Motor
DC Motor

Figure 1. Hardware/Software Development Configuration for Self-Leveling Stabilized Platform System Experiment with Embedded SH2 Microcontroller

4.2.2. Mechatronics Components

Actuation. Referring to Figure 1, the Top Platform (TP) is connected to three posts; one post is attached to the top platform via a universal joint and fixed in length, and the other two posts via ball joints and can be actuated (raised or lowered) by two dc motors (A1 & A2). Hence, the TP can pitch and roll.

Sensing. Two tilt sensors (MEMS accelerometers) are employed to yield information about the roll and pitch angles of the platform. In addition, two rate gyros (MEMS rate transducers) are used to measure roll and pitch angular rates. The accelerators and rate gyros complements each other very well to remove noise and bias that are inherent to the sensors. (See section on Sensor Fusion below)

Input Interface: The accelerators provide TTL PWM (pulse width modulation) signals which are fed directly to the TPU (time processing units) of the SH2 Microcontroller, while the rate gyros generates analog voltage for the A/D channels.

Output Interface. The Microcontroller outputs PWM control signals to H-bridge drivers (LM18200), which, in turn, actuate the dc motors.

Software. Estimation and control scheme for the platform is shown in Figure 2. The tilt and rate sensors are fused using Kalman filtering technique to remove unwanted noise and bias, and estimate the pitch and roll angles, and its rates. The estimates then feed the PID controllers that drive actuators.

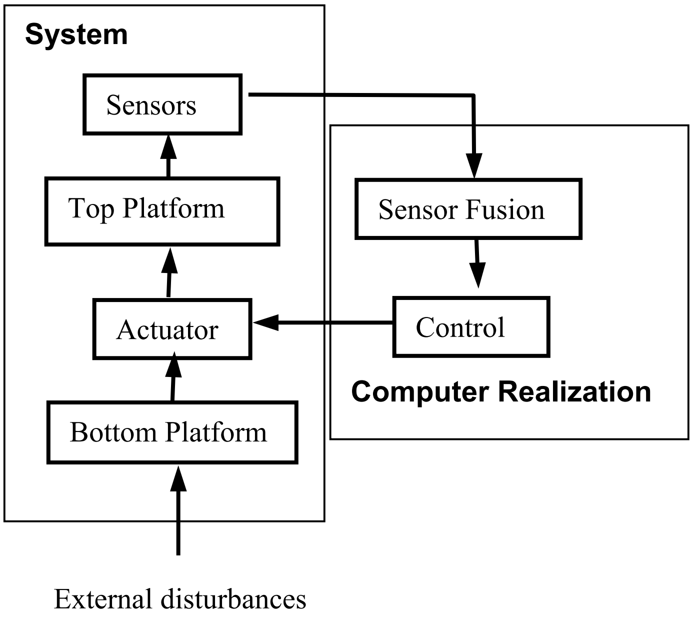


Figure 2. Overview of self-leveling and stabilizing platform control scheme

4.2.3. Modeling

Figure 3 shows the coordinate system and force/torque vectors for the platform. For precise reference to frames, objects, points and operation, we have adopted the following nomenclature for matrices and vectors:

$$
{}^{\text{Frame of Reference}}_{\text{From Point or Of Object}}\left[\mathbf{Vector}\right]^{\text{Operation on the Vector}}_{\text{To Point}} \qquad {}^{\text{Frame of Reference}}_{\text{Of Object}}\left[\mathbf{Matrix}\right]^{\text{Operation}}_{\text{To Frame}}
$$

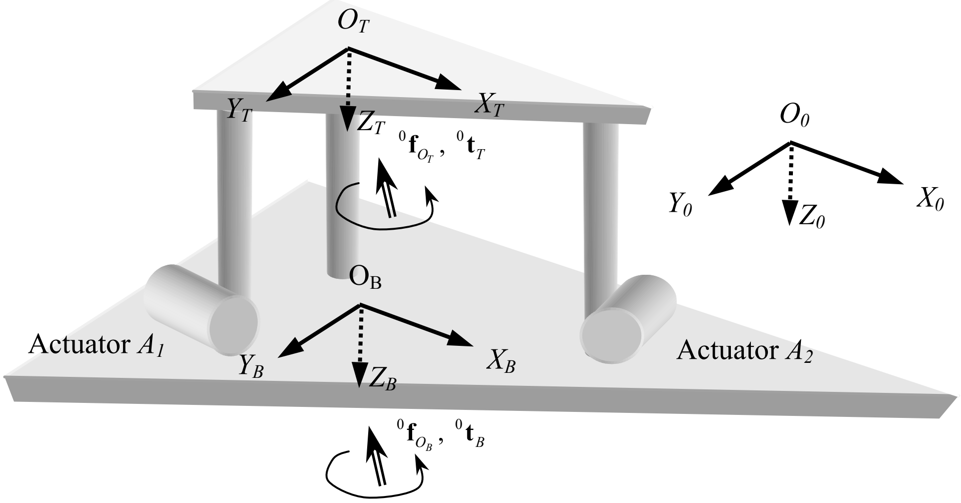


Figure 3. Coordinate system and force/torque vectors for platform

Kinematics and dynamics of Top Platform: Let $O_0X_0Y_0Z_0$ an inertial frame and $O_TX_TY_TZ_T$ represents the coordinate frame attached to the center of gravity of the top platform. The displacement kinematics can be written as [1]

$$
{}^0\mathbf{r}_p = {}^0\mathbf{r}_{O_T} + {}^0\mathbf{R}_T\, {}^T\mathbf{r}_p;\; {}^0\mathbf{r}_{O_T} = {}^0\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{O_T};\; {}^0\mathbf{R}_T = \begin{bmatrix} c\psi\, c\theta & -s\psi\, c\phi + c\psi\, s\theta\, s\phi & s\psi\, s\phi + c\psi\, s\theta\, c\phi \\ s\psi\, c\theta & c\psi\, c\phi + s\psi\, s\theta\, s\phi & -c\psi\, s\phi + s\psi\, s\theta\, c\phi \\ -s\theta & c\theta\, s\phi & c\theta\, c\phi \end{bmatrix};\; {}^T\mathbf{r}_p = {}^T\begin{bmatrix} x \\ y \\ z \end{bmatrix}_p \qquad (1.1)
$$

where x, y & z are translation displacements and ψ, θ & ϕ are yaw, pitch & roll (Euler) angles of the top platform. The angular velocity kinematics is given by

$$
{}^{O_T}\varpi = \begin{bmatrix} {}^{O_T}\omega_x \\ {}^{O_T}\omega_y \\ {}^{O_T}\omega_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \triangleq {}^{O_T}\mathbf{N}_E\, {}^E\varpi \qquad (1.2)
$$

where ${}^E\varpi = \begin{bmatrix} \dot{\phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}'$ are the Euler angle rates and ${}^{O_T}\varpi = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}'$ are the angular rates projected onto the $X_TY_TZ_T$ axis. It can be shown that the translation dynamics of the platform is described by

$$\frac{d\left({}^{0}\mathbf{r}_{O_T}\right)}{dt} = {}^{0}\mathbf{v}_{O_T}\,; \quad \frac{d\left({}^{0}\mathbf{v}_{O_T}\right)}{dt} = \mathbf{M}_T^{-1}\,{}^{0}\mathbf{f}_{O_T}\,; \quad \mathbf{M}_T = \begin{bmatrix} m_T & 0 & 0 \\ 0 & m_T & 0 \\ 0 & 0 & m_T \end{bmatrix}; \quad {}^{0}\mathbf{f}_{O_T} = {}^{0}\begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix}_{O_T} \tag{1.3}$$

where ${}^{0}\mathbf{f}_{O_T}$ is the net forces acting on the center of gravity of the top platform and $\mathbf{M}_T$ is a diagonal matrix of the mass m_T of the platform. Similarly, the rotation dynamics for the platform can be shown to be

$$\begin{gathered}
\frac{d\left({}^{E}\Omega_T\right)}{dt} = {}^{E}\varpi_T\,; \qquad {}^{E}\varpi_T = {}^{E}\mathbf{N}_{O_T}\,{}^{O_T}\varpi_T \\
\frac{d\left({}^{O_T}\varpi_T\right)}{dt} = \left(\mathbf{I}_T\right)^{-1}\left(-\mathbf{S}\,\mathbf{I}_T\,{}^{O_T}\varpi_T + {}^{O_T}\mathbf{t}_T\right); \qquad {}^{O_T}\mathbf{t}_T = {}^{O_T}\mathbf{R}_0\,{}^{0}\mathbf{t}_T \\
{}^{O_T}\mathbf{R}_0 = \begin{bmatrix} c\psi\,c\theta & s\psi\,c\theta & -s\theta \\ -s\psi\,c\phi + c\psi\,s\theta\,s\phi & c\psi\,c\phi + s\psi\,s\theta\,s\phi & c\theta\,s\phi \\ s\psi\,s\phi + c\psi\,s\theta\,c\phi & -c\psi\,s\phi + s\psi\,s\theta\,c\phi & c\theta\,c\phi \end{bmatrix} \qquad \mathbf{I}_B = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \\
{}^{E}\mathbf{N}_{O_T} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix} \qquad \mathbf{S} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}
\end{gathered} \tag{1.4}$$

where ${}^{0}\mathbf{t}_T$ is the net torque acting on the center and $\mathbf{I}_B$ is a diagonal matrix with principal moments of inertia I_{xx}, I_{yy} & I_{zz}.

Dynamics of Bottom Platform: Similarly, let $O_B X_B Y_B Z_B$ represents the coordinate frame attached to the bottom platform. The dynamics of the bottom platform is given by

$$\begin{gathered}
\frac{d\left({}^{0}\mathbf{r}_{O_B}\right)}{dt} = {}^{0}\mathbf{v}_{O_B}\,; \quad \frac{d\left({}^{0}\mathbf{v}_{O_B}\right)}{dt} = \mathbf{M}_B^{-1}\,{}^{0}\mathbf{f}_{O_B}\,; \\
\frac{d\left({}^{E}\Omega_B\right)}{dt} = {}^{E}\varpi_B\,; \quad {}^{E}\varpi_B = {}^{E}\mathbf{N}_{O_B}\,{}^{O_B}\varpi_B; \quad \frac{d\left({}^{O_B}\varpi_B\right)}{dt} = \left(\mathbf{I}_B\right)^{-1}\left(-\mathbf{S}\,\mathbf{I}_B\,{}^{O_B}\varpi_B + {}^{O_B}\mathbf{t}_B\right); \quad {}^{O_B}\mathbf{t}_B = {}^{O_B}\mathbf{R}_0\,{}^{0}\mathbf{t}_B
\end{gathered} \tag{1.5}$$

where the matrices and vectors are define similarly as for the top platform.

Force and torque. The bottom platform is subjected to ${}^{0}\mathbf{f}_{O_B} = {}^{0}_{D}\mathbf{f}_{O_B} + {}^{0}_{A}\mathbf{f}_{O_B}$ & ${}^{0}\mathbf{t}_B = {}^{0}_{D}\mathbf{t}_B + {}^{0}_{A}\mathbf{t}_B$, where ${}^{0}_{D}\mathbf{f}_{O_B}$ & ${}^{0}_{D}\mathbf{t}_B$ are external and internal forces & and torque, and ${}^{0}_{A}\mathbf{f}_{O_B}$ & ${}^{0}_{A}\mathbf{t}_B$ are actuator (dc motor) forces & torques. Similarly, the top platform is subjected to ${}^{0}\mathbf{f}_{O_T} = {}^{0}_{D}\mathbf{f}_{O_T} + {}^{0}_{A}\mathbf{f}_{O_T}$ & ${}^{0}\mathbf{t}_T = {}^{0}_{D}\mathbf{t}_T + {}^{0}_{A}\mathbf{t}_T$, where ${}^{0}_{D}\mathbf{f}_{O_T}$ & ${}^{0}_{D}\mathbf{t}_T$, and ${}^{0}_{A}\mathbf{f}_{O_T}$ & ${}^{0}_{A}\mathbf{t}_T$ are external and actuator forces and torques. Note that the forces and torques due to the actuators are equal and opposite: ${}^{0}_{A}\mathbf{f}_{O_T} = -{}^{0}_{A}\mathbf{f}_{O_B}$ & ${}^{0}_{A}\mathbf{t}_T = -{}^{0}_{A}\mathbf{t}_B$. The actuator forces and torques acting on the platform are given by

$$\begin{aligned}
{}^{0}_{A}\mathbf{f}_{O_T} &= {}^{0}\mathbf{f}_{p_i} \cong \left[0 \quad 0 \quad \left(\alpha_1 f_{A_1} + \alpha_2 f_{A_2}\right)\right]' \\
{}^{0}_{A}\mathbf{t}_T &= \sum\nolimits_{i=1}^{2} \left({}^{0}\mathbf{r}_{p_i} - {}^{0}\mathbf{r}_{O_T}\right) \times {}^{0}\mathbf{f}_{p_i} \cong \left[\left(\beta_1 f_{A_1}\right) \quad \left(\beta_2 f_{A_2}\right) \quad 0\right]'
\end{aligned} \tag{1.6}$$

where point p_i is the location where the post i joins the top platform, ${}^0\mathbf{r}_{p_i}$ points to p_i and ${}^0\mathbf{f}_{p_i}$ is the force vector along the post i acting on p_i, index i =1,2. α_1, α_2, β_1 & β_2 are geometrical dimensions of the platform, and f_{A_1} & f_{A_2} are forces generated by motion of Actuators A_1 and A_2.

Actuator dynamics and joint forces. The two individual identical motor (A_1 & A_2) actuations are accountable by the following equations:

$$\begin{aligned} L_a \frac{di_a}{dt} &= -R_a i_a - K_b N_g \dot{\theta}_{gi} + v_a \\ J_{eq}\ddot{\theta}_{gi} &= -D_{eq}\dot{\theta}_{gi} + N_g K_t \; i_a - \tau_{stickeq} - f_{A_i} \end{aligned} \tag{1.7}$$

where

θ_{gi} = geared output shaft angle for motor i, $i = 1,2$; i_a=armature current; v_a=voltage applied to armature
J_{eq} = equivalent momentof inertia D_{eq} = equivalent viscous damping; N_g =gear ratio
R_a = armature resistance L_a = armature inductance K_b = back emf constant K_t = torque constant
$\tau_{stickeq}$ = equivalent stick friction torque f_{A_i} = external torque acting against the motor i

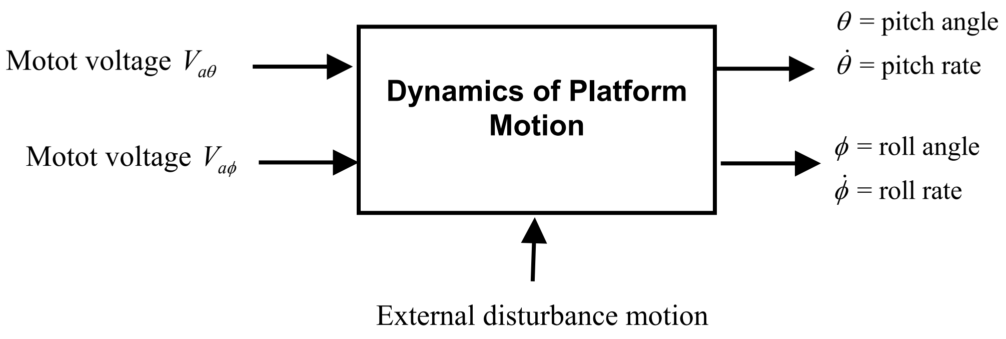
The joint interaction force f_{A_i} between the top platform and the geared motor is given by

$$f_{A_i} = K_s \left\| {}^0\mathbf{r}_{g_i} - {}^0\mathbf{r}_{p_i} \right\| \cong K_s\left(\sin(\theta_{g_i})r_g + L_i - \begin{bmatrix}0 & 0 & 1\end{bmatrix} {}^0\mathbf{r}_{p_i}\right) + D_s\left(\dot{\theta}_{g_i}\cos(\theta_{g_i})r_g - \begin{bmatrix}0 & 0 & 1\end{bmatrix} {}^0\dot{\mathbf{r}}_{p_i}\right) \tag{1.8}$$

for small values of θ. ${}^0\mathbf{r}_{g_i}$ is the tip of the geared arm and L_i is the nominal length for post i, and; i = 1,2. K_s and D_s are the elastic and damping coefficients for the metal posts.

Constraint motion. Because of the fixed rigid post (labeled as post 3), the point p_3 at the top platform is constrained to move with the bottom plate according to

$${}^0\mathbf{r}_{p_3} = {}^0\mathbf{r}_{O_B} + {}^O\mathbf{R}_B\,{}^B\mathbf{r}_{p_3} \tag{1.9}$$


Motot voltage $V_{a\theta}$
Motot voltage $V_{a\phi}$
Dynamics of Platform Motion
θ = pitch angle
$\dot{\theta}$ = pitch rate
ϕ = roll angle
$\dot{\phi}$ = roll rate
External disturbance motion

4.2.4. Formulation of Kalman filter

Sensors. Two tilt angle sensors (MEMS accelerometers) and two angular rate sensors (MEMS gyros) were used; see Figure 4. They provide measurements:

θ_m = pitch angle measurement ϕ_m = roll angle measurement

$\dot{\theta}_m$ = pitch rate measurement $\dot{\phi}_m$ = roll rate measurement

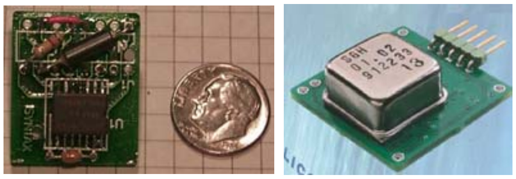


Figure 4. Tilt angle sensor (accelerometer) and angular rate sensor

When the top platform is balanced (leveled and stabilized), the pitch and roll angles can be treated as slow moving small amplitude signals. Hence, the signals' movements can be approximated using α & $\alpha\beta$ shaping dynamics. In the case of pitch movement, the dynamics of the pitch, pitch rate and a bias can be expressed as

$$\dot{\mathbf{x}}_\theta = \mathbf{A}\mathbf{x}_\theta + \mathbf{w}_\theta; \quad \dot{\mathbf{x}}_\theta = \begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \\ \dot{b}_\theta \end{bmatrix}; \mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}; \ \mathbf{x}_\theta = \begin{bmatrix} \theta \\ \dot{\theta} \\ b_\theta \end{bmatrix}; \mathbf{w}_\theta = \begin{bmatrix} w_{\theta 1} \\ w_{\theta 2} \\ w_{\theta 3} \end{bmatrix} = \text{sparse low-amplitude shaping noise} \quad (1.10)$$

The pitch and pitch rate measurements can then be written as

$$y_\theta = \mathbf{C}_\theta \mathbf{x}_\theta + v_\theta; \qquad \mathbf{y}_\theta = \begin{bmatrix} \theta_m \\ \dot{\theta}_m \end{bmatrix}; \ \mathbf{C}_\theta = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}; \ \mathbf{v}_\theta = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \text{measurement noise} \qquad (1.11)$$

A state estimator or observer for filtering the noise is given by [2]

$$\dot{\hat{\mathbf{x}}}_\theta = \mathbf{A}\hat{\mathbf{x}}_\theta + \mathbf{L}\left(\mathbf{y}_\theta - \mathbf{C}_\theta \mathbf{x}_\theta\right) \qquad \hat{\mathbf{x}}_\theta = \begin{bmatrix} \hat{\theta} & \hat{\dot{\theta}} & \hat{b}_\theta \end{bmatrix}' = \text{estimate of sensor states} \qquad (1.12)$$

where **L** is a estimator gain matrix. The 3x2 **L** matrix can be designed using pole-placement technique. In the case of the platform, **L** was determined using Kalman-Bucy filter equation as follows:

$$\mathbf{L} = \mathbf{PC'R}^{-1}; \qquad \mathbf{0} = \mathbf{PA'} + \mathbf{AP} + \mathbf{Q} - \mathbf{PC'R}^{-1}\mathbf{CP}$$

$$\mathbf{L} = \begin{bmatrix} l_{11} & l_{12} \\ l_{21} & l_{22} \\ l_{31} & l_{32} \end{bmatrix} \quad \mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{12} & p_{22} & p_{23} \\ p_{13} & p_{23} & p_{33} \end{bmatrix} \quad \mathbf{Q} = \begin{bmatrix} q_\theta & 0 & 0 \\ 0 & q_{\dot{\theta}} & 0 \\ 0 & 0 & q_b \end{bmatrix} \quad \mathbf{R} = \begin{bmatrix} r_{\theta_m} & 0 \\ 0 & r_{\dot{\theta}_m} \end{bmatrix} \qquad (1.13)$$

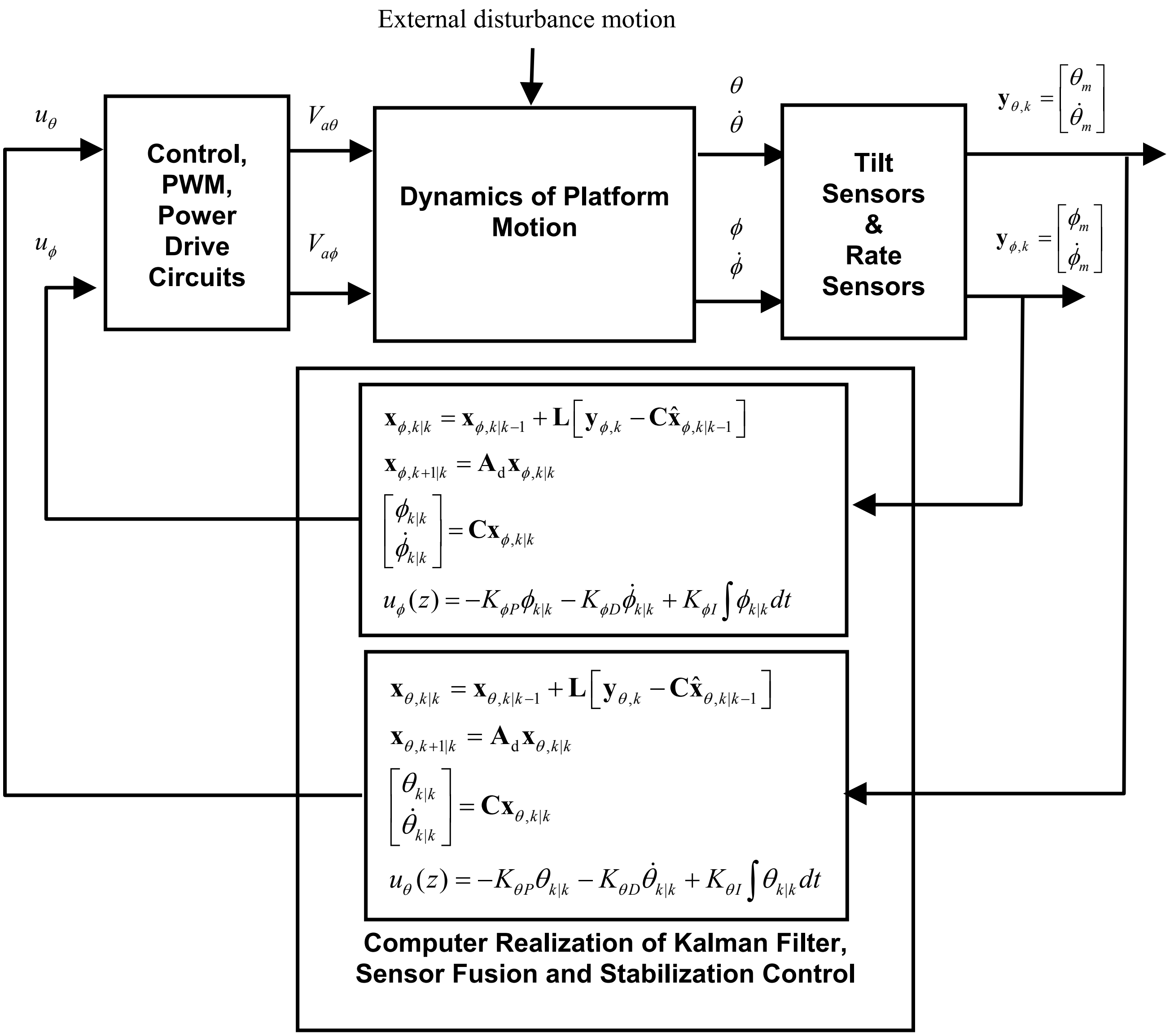
where Q & R are approximation of the covariance of $\mathbf{w}_\theta$ & $\mathbf{v}_\theta$. Using a sampling interval $T = 0.01\text{sec}$, the digital implementation of the pitch estimator is given by

$$\mathbf{x}_{\theta,k|k} = \mathbf{x}_{\theta,k|k-1} + \mathbf{L}\left[\mathbf{y}_{\theta,k} - \mathbf{C}\hat{\mathbf{x}}_{\theta,k|k-1}\right] \text{ (corrector)}; \qquad \mathbf{x}_{\theta,k+1|k} = \mathbf{A}_\text{d}\mathbf{x}_{\theta,k|k} \text{ (predictor)}; \qquad \mathbf{A}_\text{d} = \begin{bmatrix} 1 & T & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (1.14)$$

where $k = 0, 1, 2, ...$ represents sample indices. The discrete-time estimator yields the current state estimate $\mathbf{x}_{\theta,k|k} = \left[\theta_{k|k} \quad \dot{\theta}_{k|k} \quad b_{k|k}\right]'$ and one-step ahead prediction $\mathbf{x}_{\theta,k+1|k} = \left[\theta_{k+1|k} \quad \dot{\theta}_{k+1|k} \quad b_{k+1|k}\right]'$. A estimator for roll, roll rate and roll bias can be similarly synthesized to yield $\mathbf{x}_{\phi,k|k} = \left[\phi_{k|k} \quad \dot{\phi}_{k|k} \quad b_{k|k}\right]'$ and $\mathbf{x}_{\phi,k+1|k} = \left[\phi_{k+1|k} \quad \dot{\phi}_{k+1|k} \quad b_{k+1|k}\right]'$.

Controllers. The self-leveling and stabilizing controllers incorporates PID actions based on the estimated states [3]:

$$u_\theta(z) = -K_{\theta P}\theta_{k|k} - K_{\theta D}\dot{\theta}_{k|k} + K_{\theta I}\int \theta_{k|k}dt \quad \text{and} \quad u_\phi(z) = -K_{\phi P}\phi_{k|k} - K_{\phi D}\dot{\phi}_{k|k} + K_{\phi I}\int \phi_{k|k}dt\,. \qquad (1.15)$$


External disturbance motion
u_θ
u_ϕ
Control, PWM, Power Drive Circuits
$V_{a\theta}$
$V_{a\phi}$
Dynamics of Platform Motion
θ
$\dot{\theta}$
ϕ
$\dot{\phi}$
Tilt Sensors & Rate Sensors
$\mathbf{y}_{\theta,k} = \begin{bmatrix}\theta_m \\ \dot{\theta}_m\end{bmatrix}$
$\mathbf{y}_{\phi,k} = \begin{bmatrix}\phi_m \\ \dot{\phi}_m\end{bmatrix}$
$\mathbf{x}_{\phi,k|k} = \mathbf{x}_{\phi,k|k-1} + \mathbf{L}\left[\mathbf{y}_{\phi,k} - \mathbf{C}\hat{\mathbf{x}}_{\phi,k|k-1}\right]$
$\mathbf{x}_{\phi,k+1|k} = \mathbf{A}_{\mathrm{d}}\mathbf{x}_{\phi,k|k}$
$\begin{bmatrix}\phi_{k|k} \\ \dot{\phi}_{k|k}\end{bmatrix} = \mathbf{C}\mathbf{x}_{\phi,k|k}$
$u_\phi(z) = -K_{\phi P}\phi_{k|k} - K_{\phi D}\dot{\phi}_{k|k} + K_{\phi I}\int \phi_{k|k}dt$
$\mathbf{x}_{\theta,k|k} = \mathbf{x}_{\theta,k|k-1} + \mathbf{L}\left[\mathbf{y}_{\theta,k} - \mathbf{C}\hat{\mathbf{x}}_{\theta,k|k-1}\right]$
$\mathbf{x}_{\theta,k+1|k} = \mathbf{A}_{\mathrm{d}}\mathbf{x}_{\theta,k|k}$
$\begin{bmatrix}\theta_{k|k} \\ \dot{\theta}_{k|k}\end{bmatrix} = \mathbf{C}\mathbf{x}_{\theta,k|k}$
$u_\theta(z) = -K_{\theta P}\theta_{k|k} - K_{\theta D}\dot{\theta}_{k|k} + K_{\theta I}\int \theta_{k|k}dt$
Computer Realization of Kalman Filter, Sensor Fusion and Stabilization Control

4.2.5. Near-Zero Hand-Coding Development Environment for Embedded Controller

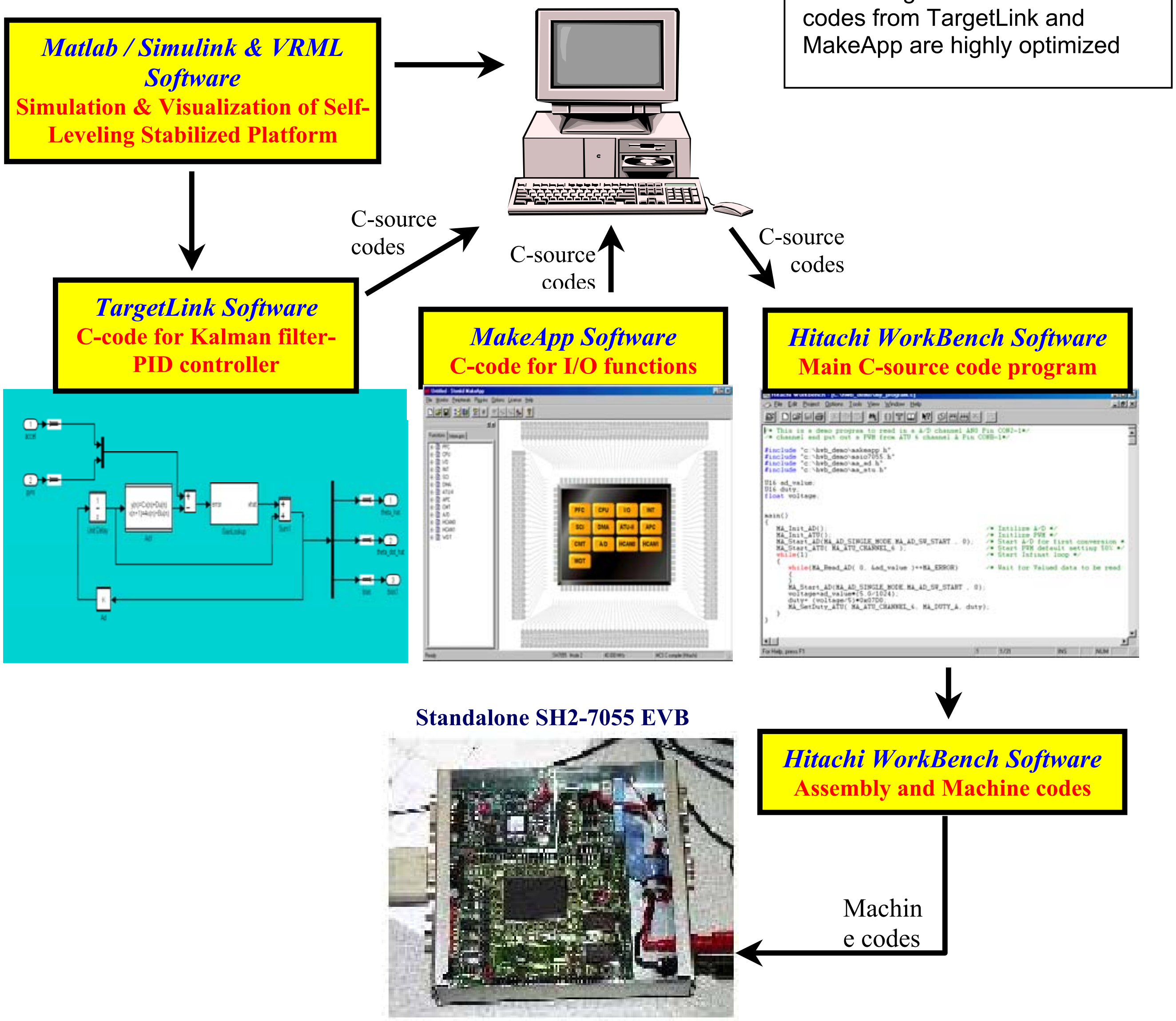
The **dSpace TargetLink** software uses Simulink blocks to automatically code the Kalman filter algorithm & controller

The **MakeApp** software uses a graphical user interface (GUI) to generate automatically C-source code modules for input & output operations of a microcontroller.

The Hitachi WorkBench software combine C-source codes from TargetLink and MakeApp, to form a main C-program.

The microcontroller communicates with a Simulink on PC thru a serial link for monitoring performance of the controlled system.

The auto generated C-source codes from TargetLink and MakeApp are highly optimized


Matlab / Simulink & VRML Software
Simulation & Visualization of Self-Leveling Stabilized Platform
C-source codes
C-source codes
C-source codes
TargetLink Software
C-code for Kalman filter-PID controller
MakeApp Software
C-code for I/O functions
Hitachi WorkBench Software
Main C-source code program
Standalone SH2-7055 EVB
Hitachi WorkBench Software
Assembly and Machine codes
Machin
e codes

```
/* This is a demo program to read in a A/D channel AN0 Pin CON2-1*/
/* channel and put out a PWM from ATU 6 channel A Pin CON8-1*/

#include "c:\hwb_demo\makeapp.h"
#include "c:\hwb_demo\maio7055.h"
#include "c:\hwb_demo\ma_ad.h"
#include "c:\hwb_demo\ma_atu.h"

U16 ad_value;
U16 duty;
float voltage;


main()
{
    MA_Init_AD();                                        /* Intilize A/D */
    MA_Init_ATU();                                       /* Initilize PWM */
    MA_Start_AD(MA_AD_SINGLE_MODE,MA_AD_SW_START , 0);   /* Start A/D for first conversion *
    MA_Start_ATU( MA_ATU_CHANNEL_6 );                    /* Start PWM default setting 50% */
    while(1)                                             /* Start Infinat loop */
    {
        while(MA_Read_AD( 0, &ad_value )==MA_ERROR)      /* Wait for Valued data to be read
        {
        }
        MA_Start_AD(MA_AD_SINGLE_MODE,MA_AD_SW_START , 0);
        voltage=ad_value*(5.0/1024);
        duty= (voltage/5)*0x07D0;
        MA_SetDuty_ATU( MA_ATU_CHANNEL_6, MA_DUTY_A, duty);
    }
}
```

Figure 5. Near-zero hand-coding development environment & process for embedded SH2 microcontroller EVB.

4.2.6. Experimental results

The standalone SH2 EVB microcontroller and the self-leveling stabilized platform were put to test aboard a mobile robot as shown in Figure 6. Figure 7 compares the output from the pitch sensor, which is a noisy accelerometer, with the estimated output from the Kalman filter, as the mobile robot (hence bottom platform) pitches upward and downward. The smoother estimated output was used to level the platform via the PID controller. The experiment demonstrates the effectiveness of the embedded microcontroller. A video on the performance of the self-leveling stabilized platform will be presented at the conference.

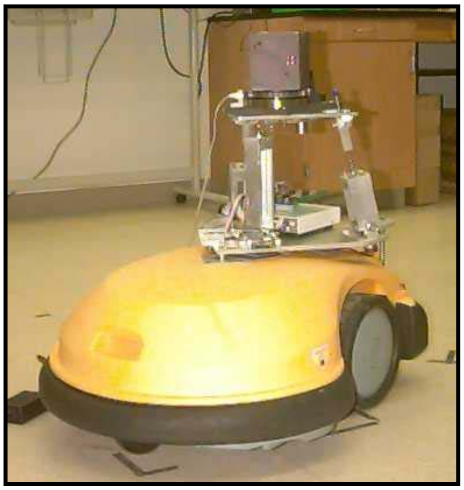


Figure 6. Experiments with the self-leveling stabilized platform aboard a mobile robot

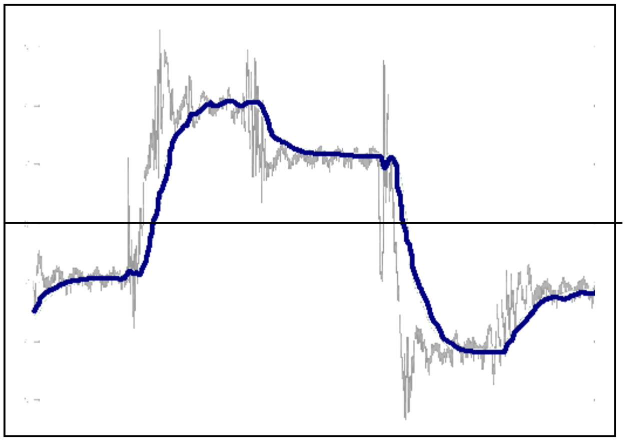


Figure 7. Experimental output from the pitch sensor (noisy accelerometer) versus output from the Kalman filter, as the mobile robot pitches upward

4.2.7. Movie Clip and/or Actual Bench Demonstration

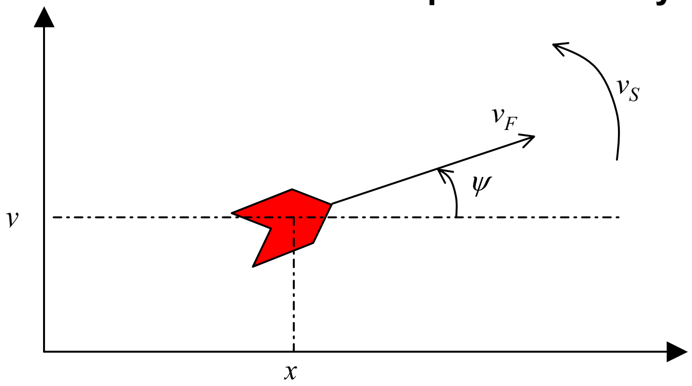
4.3. Application of Kalman filter to Navigation Sensor Fusion

4.3.1. Objective

Data from positioning systems (such as the GPS, laser systems, etc.) often needs to be complemented by other dead reckoning and inertial sensors (such as wheel speed sensor, yaw rate, accelerometers, etc). This section presents a Kalman filter approach to the formulation of a sensor fusion for combining the data from GPS and wheel speed sensors for a mobile robot.

The same technique has been applied to an autonomous vehicle with a laser-based navigation system. An application to precision self-guided lawn mower will be presented.

4.3.2. Kinematics relationship from velocity to position



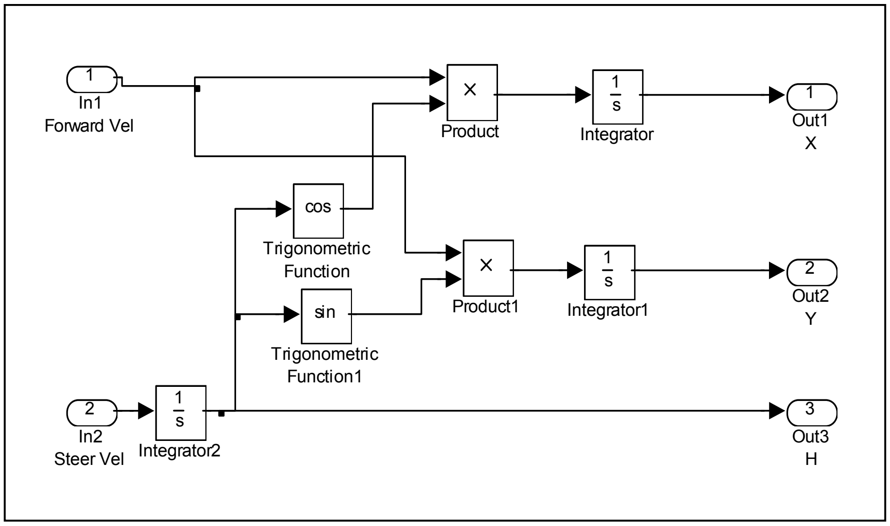
Top view of a mobile robot and its coordinate system

v_F = forward speed of vehicle
v_S = steer rate of vehicle
ψ = yaw heading of vehicle
x = x-coordinate of vehicle
y = y-coordinate of vehicle

Rate to Position Relationship

$\dot{\psi}$ = yaw rate of vehicle
$\dot{x}$ = x-velocity of vehicle
$\dot{y}$ = y-velocity of vehicle

$$\dot{\psi} = v_S$$
$$\dot{x} = v_F \cos\psi$$
$$\dot{y} = v_F \sin\psi$$

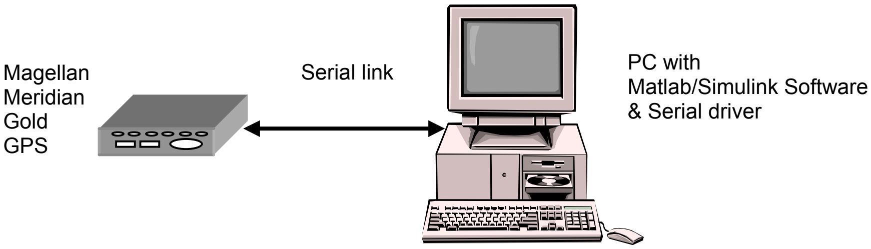


A state space model for kinematic relationship is

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ \psi \end{bmatrix} + \begin{bmatrix} \cos\psi & 0 \\ \sin\psi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_F \\ v_S \end{bmatrix} \qquad \text{continous-time model}$$

4.3.3. GPS Data and Measurement Equation

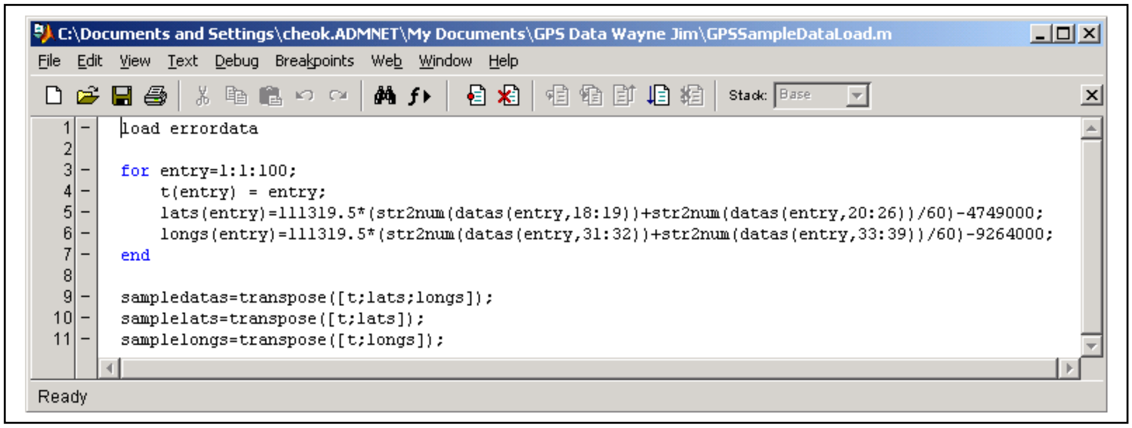
Experimental data



Sample of data received by GPS sitting at a location on the round patio in front of the SEB. Recorded & saved using Simulink.

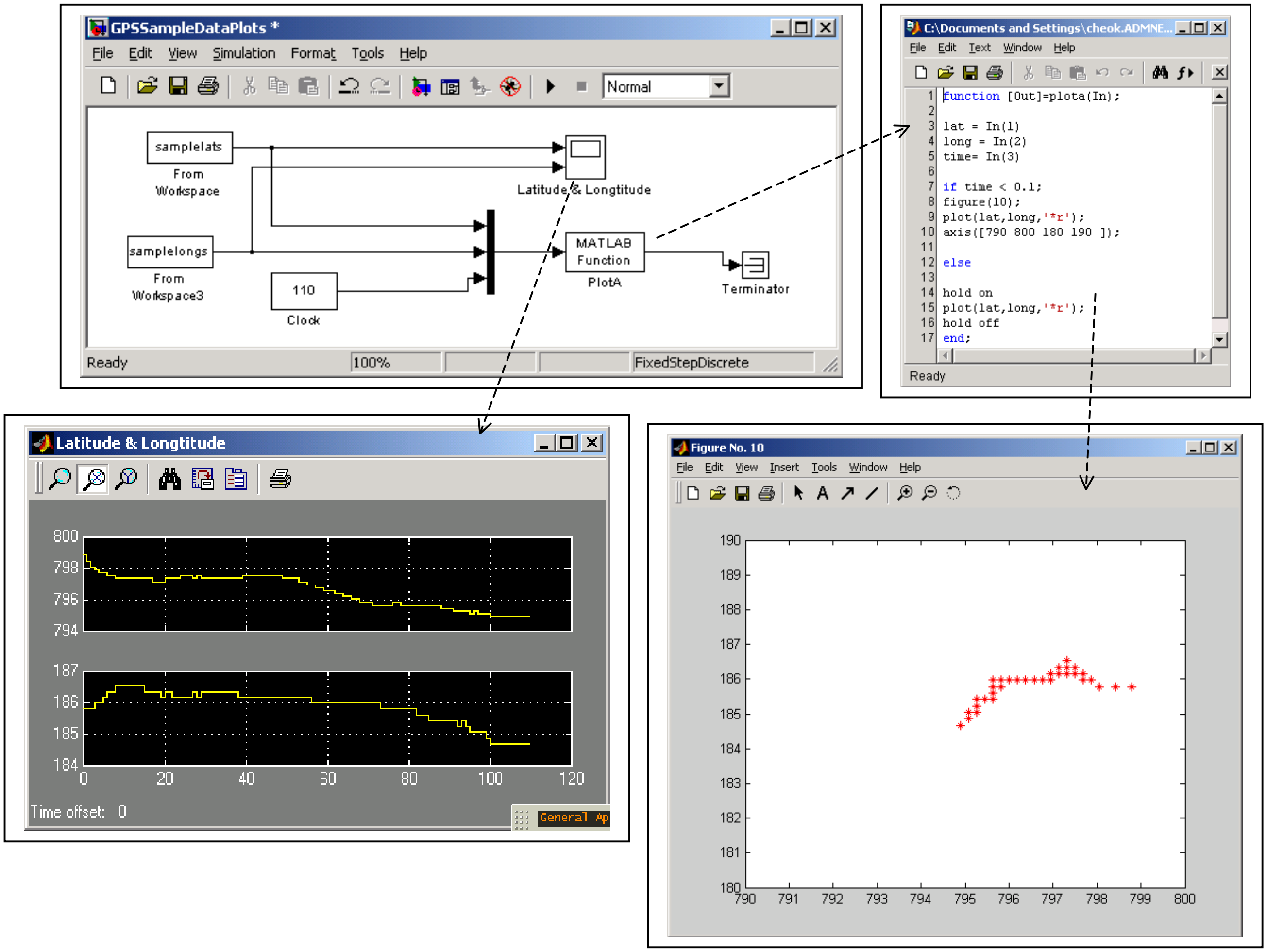
```
$GPGGA,175006.45,4240.0897,N,08313.2954,W,1,07,2.3,00286,M,,
$GPGGA,175007.45,4240.0895,N,08313.2954,W,1,07,2.3,00287,M,,
$GPGGA,175008.46,4240.0894,N,08313.2955,W,1,07,2.4,00288,M,,
$GPGGA,175009.45,4240.0893,N,08313.2955,W,1,07,2.3,00288,M,,
$GPGGA,175010.45,4240.0893,N,08313.2956,W,1,07,2.3,00289,M,,
$GPGGA,175011.46,4240.0892,N,08313.2957,W,1,07,2.3,00290,M,,
$GPGGA,175012.45,4240.0892,N,08313.2957,W,1,07,2.5,00290,M,,
$GPGGA,175013.45,4240.0891,N,08313.2958,W,1,07,2.3,00291,M,,
$GPGGA,175014.46,4240.0891,N,08313.2958,W,1,07,2.3,00292,M,,
$GPGGA,175015.45,4240.0891,N,08313.2958,W,1,07,2.2,00292,M,,
$GPGGA,175016.46,4240.0891,N,08313.2958,W,1,07,2.2,00293,M,,
$GPGGA,175017.46,4240.0891,N,08313.2958,W,1,07,2.2,00293,M,,
$GPGGA,175018.45,4240.0891,N,08313.2958,W,1,07,2.8,00293,M,,
$GPGGA,175019.46,4240.0891,N,08313.2958,W,1,07,2.8,00293,M,,
$GPGGA,175020.46,4240.0891,N,08313.2957,W,1,07,2.8,00293,M,,
$GPGGA,175021.45,4240.0891,N,08313.2957,W,1,07,2.9,00293,M,,
$GPGGA,175022.46,4240.0890,N,08313.2957,W,1,07,2.8,00293,M,,
$GPGGA,175023.46,4240.0890,N,08313.2957,W,1,07,2.8,00293,M,,
$GPGGA,175024.45,4240.0890,N,08313.2956,W,1,07,2.4,00294,M,,
$GPGGA,175025.46,4240.0891,N,08313.2957,W,1,07,2.7,00294,M,,
$GPGGA,175026.46,4240.0891,N,08313.2957,W,1,07,2.8,00294,M,,
$GPGGA,175027.45,4240.0891,N,08313.2956,W,1,07,2.8,00294,M,,
```

A typical Matlab program to convert ASCII strings to useable numerical arrays of latitudes and longitudes:


C:\Documents and Settings\cheok.ADMNET\My Documents\GPS Data Wayne Jim\GPSSampleDataLoad.m
File Edit View Text Debug Breakpoints Web Window Help
Stack: Base
Ready

```
load errordata

for entry=1:1:100;
    t(entry) = entry;
    lats(entry)=111319.5*(str2num(datas(entry,18:19))+str2num(datas(entry,20:26))/60)-4749000;
    longs(entry)=111319.5*(str2num(datas(entry,31:32))+str2num(datas(entry,33:39))/60)-9264000;
end

sampledatas=transpose([t;lats;longs]);
samplelats=transpose([t;lats]);
samplelongs=transpose([t;longs]);
```

A typical Simulink model to display plots of acquired GPS data.



```
function [Out]=plota(In);

lat = In(1)
long = In(2)
time= In(3)

if time < 0.1;
figure(10);
plot(lat,long,'*r');
axis([790 800 180 190 ]);

else

hold on
plot(lat,long,'*r');
hold off
end;
```

The measurement equation for the GPS is $\begin{bmatrix} z_{1k} \\ z_{2k} \end{bmatrix} = \begin{bmatrix} x_k + v_{1k} \\ y_k + v_{2k} \end{bmatrix}$

4.3.4. Wheel Speed measurement

$$v_R = \text{speed of right wheel (in m/s)}$$

$$v_L = \text{speed of left wheel (in m/s)}$$

$$v_F = \frac{v_R + v_L}{2} = \text{forward speed of vehicle}$$

$$v_S = \frac{v_R - v_L}{W} = \text{steer rate of vehicle, W = width of the vehicle}$$

The measurement from the speed sensors is $\begin{bmatrix} v_F \\ v_s \end{bmatrix}$

4.3.5. FUSION OF GPS & WHEEL SPEED

Discretization of state model

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ \psi \end{bmatrix} + \begin{bmatrix} \cos\psi & 0 \\ \sin\psi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_F \\ v_S \end{bmatrix} \qquad \text{continuous-tme model}$$

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \psi_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ y_k \\ \psi_k \end{bmatrix} + \begin{bmatrix} T\cos\psi_k & 0 \\ T\sin\psi_k & 0 \\ 0 & T \end{bmatrix} \begin{bmatrix} v_{F,k} \\ v_{S,k} \end{bmatrix} \qquad \text{discrete-time model, } T = \text{sampling interval}$$

Formulation of Kalman filter

To apply a Kalman filter to the discrete-time model, we formulate the above relationship as astochastic process

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}_k\mathbf{u}_k + \mathbf{w}_k$$

$$\mathbf{x}_k = \begin{bmatrix} x_k \\ y_k \\ \psi_k \end{bmatrix}, \quad \mathbf{u}_k = \begin{bmatrix} v_{F,k} \\ v_{S,k} \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{B}_k = \begin{bmatrix} T\cos\psi_k & 0 \\ T\sin\psi_k & 0 \\ 0 & T \end{bmatrix}$$

where the noise $\mathbf{w}_k$ represents in accuracy in the model including wheel slips and speed measurement noise.

The GPS data is the true position info corrupted with GPS type noise can be modeled as

$$\mathbf{z}_k = \mathbf{C}\mathbf{x}_k + \mathbf{v}_k$$

$$\mathbf{z}_k = \begin{bmatrix} z_{1k} \\ z_{2k} \end{bmatrix} = \begin{bmatrix} x_k + v_{1k} \\ y_k + v_{2k} \end{bmatrix} \qquad \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

We need to specify the mean & covariance of the variables. That is

$$\mathbf{w}_k \sim (0, \mathbf{Q}_k) \qquad \mathbf{v}_k \sim (0, \mathbf{R}_k) \qquad \mathbf{x}_0 \sim (0, \mathbf{P}_0)$$

Knowing the values of these covariance is a key to how well the KF will perform. The values can be found from experiments, derivation and other observations. Note that the covariance $\mathbf{Q}_k$ & $\mathbf{R}_k$ may change with time. It is assumed that the stochastic (random) variables are independent.

Time-Varying Gain Kalman Filter

A synthesis of time-varying gain Kalman filter for the above is given by

$$\begin{aligned}
\mathbf{K}_k &= \mathbf{P}_{k/k-1}\mathbf{C}'\left[\mathbf{C}\ \mathbf{P}_{k/k-1}\mathbf{C}' + \mathbf{R}_k\right]^{-1} \\
\mathbf{x}_{k/k} &= \mathbf{x}_{k/k-1} + \mathbf{K}_k\left[\mathbf{z}_k - \mathbf{C}\ \mathbf{x}_{k/k-1}\right] \qquad \textbf{(corrector update)} \\
\mathbf{P}_{k/k} &= \mathbf{P}_{k/k-1} - \mathbf{K}_k\mathbf{C}\ \mathbf{P}_{k/k-1}
\end{aligned}$$

$$\begin{aligned}
\mathbf{x}_{k+1/k} &= \mathbf{A}\ \mathbf{x}_{k/k} + \mathbf{B}_k\mathbf{u}_k \\
\mathbf{P}_{k+1/k} &= \mathbf{A}\ \mathbf{P}_{k/k}\mathbf{A}' + \mathbf{Q}_k
\end{aligned} \qquad \textbf{(predictor update)}$$

Note that this is an alternative form of the same time-varying KF algorithm presented in Section 3.

Constant Gain Kalman Filter

In the case of time-invariant system, a constant gain version of the Kalman filter is given by

$$\begin{aligned}
\mathbf{x}_{k/k} &= \mathbf{x}_{k/k-1} + \mathbf{K}\left[\mathbf{z}_k - \mathbf{C}\mathbf{x}_{k/k-1}\right] \\
\mathbf{x}_{k+1/k} &= \mathbf{A}_k\mathbf{x}_{k/k} + \mathbf{B}_k\mathbf{u}_k \\
\mathbf{K} &= \mathbf{P}\mathbf{C}'\left[\mathbf{C}\mathbf{P}\mathbf{C}' + \mathbf{R}\right]^{-1} \\
\mathbf{P} &= \mathbf{A}\mathbf{P}\mathbf{A}' - \mathbf{A}\mathbf{P}\mathbf{C}'\left[\mathbf{C}\mathbf{P}\mathbf{C}' + \mathbf{R}\right]^{-1}\mathbf{C}\mathbf{P}\mathbf{A}' + \mathbf{G}_k\mathbf{Q}_k\mathbf{G}_k'
\end{aligned}$$

This is also an alternative form of the same constant gain KF algorithm presented in Section 3.

Sensor Fusion

Notice how data from GPS $\mathbf{z}_k = \begin{bmatrix} z_{1k} \\ z_{2k} \end{bmatrix} = \begin{bmatrix} x_k + v_{1k} \\ y_k + v_{2k} \end{bmatrix}$ are blended with the vehicle speeds $\mathbf{u}_k = \begin{bmatrix} v_F \\ v_S \end{bmatrix} = \begin{bmatrix} \dfrac{v_R + v_L}{2} \\ \dfrac{v_R - v_L}{W} \end{bmatrix}$ by the Kalman filter.

4.3.6. Application to a Precision Self-guided Lawn Mower

The above formulation has been successfully applied to the navigation of an automatic lawn mower. In this project, a laser positioning system (LPS) was used instead of a GPS. The LPS provide better accuracy and faster update rate

4.3.7. Movie Clip