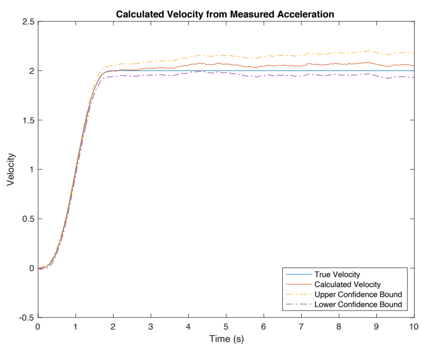
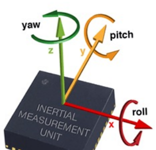
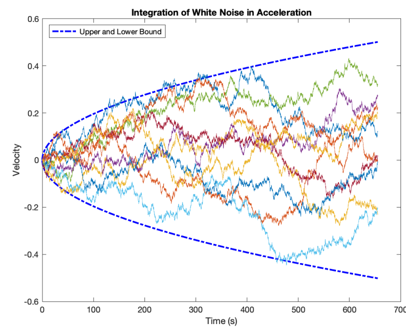
# E80
Experimental Engineering




https://vrtracker.xyz/handling-imu-drift/




## Integral and Derivative Functions of Time
## Part of: Error Analysis

1

# E80
Experimental Engineering

## Motivation

Navigation Measurements

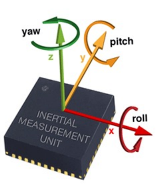
Rate Gyros

https://www.kistler.com/en/product/type-kcd16008/

Accelerometers

https://www.sparkfun.com/products/9755

IMUs (Inertial Measurement Units)




https://vrtracker.xyz/handling-imu-drift/

GPS


https://www.locosystech.com/en/product/GPS-Module/gps-module-hd-1010.html

2

# E80 Experimental Engineering — Sensor output

The output from a sensor can be modeled as

$$x_m(t) = x(t) + \varepsilon_b + \varepsilon_n(t)$$

where $x(t)$ is the true value,

$\varepsilon_b$ is the offset, bias, or baseline error,

and $\varepsilon_n(t)$ is the error due to noise. $\varepsilon_n(t)$ is usually modeled as Gaussian white noise with standard deviation $\sigma$ or $S$.

3

# E80 Experimental Engineering — Relationships in 1-D

Acceleration

$$a = \frac{dv}{dt} = \frac{d^2 r}{dt^2}$$

Velocity

$$v = \int_0^t a dt = \frac{dr}{dt}$$

Position

$$r = \int_0^t v dt = \int_0^t \left( \int_0^t a dt \right) dt$$

Angular Acceleration

$$\alpha = \frac{d\omega}{dt} = \frac{d^2 \theta}{dt^2}$$

Angular Velocity

$$\omega = \int_0^t \alpha dt = \frac{d\theta}{dt}$$

Orientation

$$\theta = \int_0^t \omega dt = \int_0^t \left( \int_0^t \alpha dt \right) dt$$

4

## E80 Experimental Engineering — What Happens to Measurements?

Acceleration

$$a_m(t_i) = a(t_i) + \varepsilon_b + \varepsilon_n(t_i)$$

$\varepsilon_n(t_i)$ is a Gaussian random sequence with standard deviation, $\sigma$.

Velocity

$$v_m(t_i) = v(t_i) + \varepsilon_b t_i + \underbrace{\sigma\sqrt{\Delta t}\sqrt{t_i}}_{\text{Observational bound}}$$

Position

$$r_m(t_i) = r(t_i) + \frac{1}{2}\varepsilon_b t^2 + \underbrace{\frac{2}{3}\sqrt{\Delta t}\sigma t_i^{3/2}}_{\text{Observational bound}}$$

5

## E80 Experimental Engineering — What Happens to Measurements?

Velocity

$$v_m(t_i) = v(t_i) + \varepsilon_b + \varepsilon_n(t_i)$$

$\varepsilon_n(t_i)$ is a Gaussian random sequence with standard deviation, $\sigma$.

Position

$$r_m(t_i) = r(t_i) + \varepsilon_b t_i + \underbrace{\sigma\sqrt{\Delta t}\sqrt{t_i}}_{\text{Observational bound}}$$

Acceleration

$$a_m(t_i) = a(t_i) + \underbrace{A_a\sigma}_{\text{Observational bound}}$$

6

E80
Experimental Engineering

# What Happens to Measurements?

Position

$$r_m(t_i) = r(t_i) + \varepsilon_b + \varepsilon_n(t_i)$$

$\varepsilon_n(t_i)$ is a Gaussian random sequence with standard deviation, $\sigma$.

Velocity

$$v_m(t_i) = v(t_i) + \underbrace{A_v \sigma}_{\text{Observational bound}}$$

Acceleration

$$a_m(t_i) = a(t_i) + \underbrace{A_a \sigma}_{\text{Observational bound}}$$

7

E80
Experimental Engineering

# Observation Bounds Calculation

For Gaussian white noise with standard deviation, $\sigma$,

$$\lambda = z_{cl} \sigma f(t)$$

$$z_{cl} = \sqrt{2}\mathrm{erf}^{-1}(cl)$$

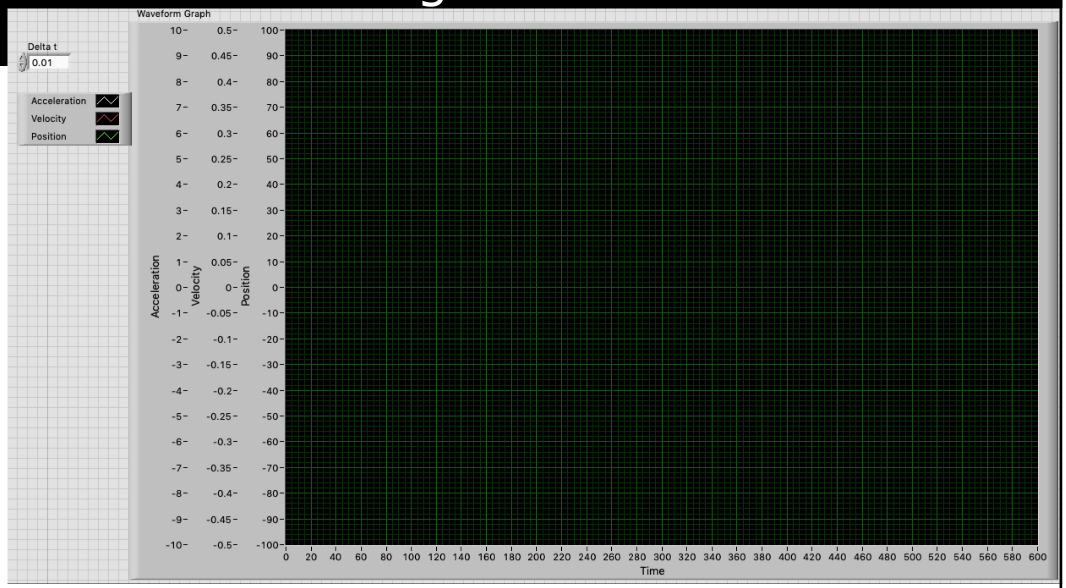
where $\mathrm{erf}^{-1}$ is the inverse error function (`erfinv` in MATLAB) and $cl$ is the confidence level, e.g., 95%

e.g., $\lambda_i = \sqrt{2}\mathrm{erf}^{-1}(cl)\sigma\sqrt{\Delta t}\sqrt{t_i}$ or $\lambda_i = \sqrt{2}\mathrm{erf}^{-1}(cl)\sigma\sqrt{\Delta t}\,\frac{2}{3}t_i^{3/2}$

8

**E80**
Experimental Engineering

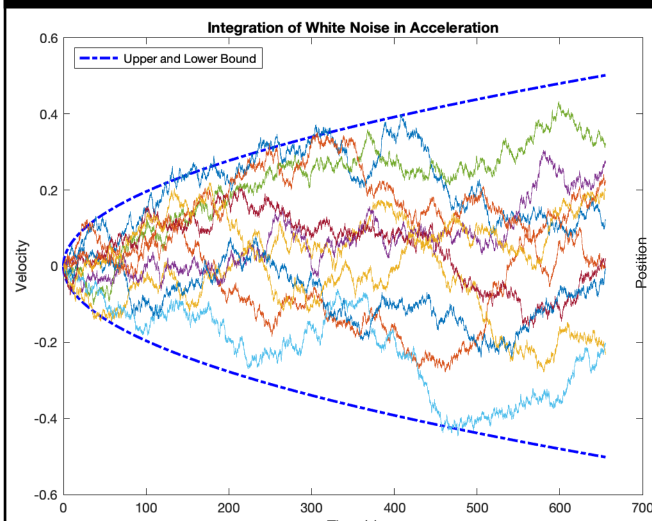
# Numerical Integration of White Noise

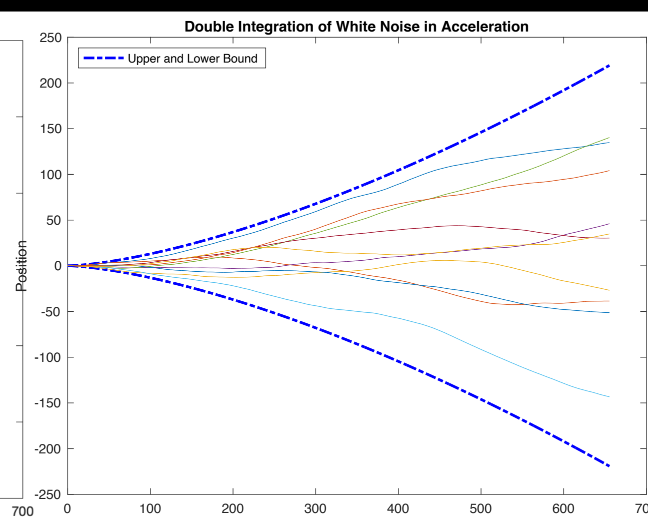



9

**E80**
Experimental Engineering

# Integrals of White Noise with Observation Bounds

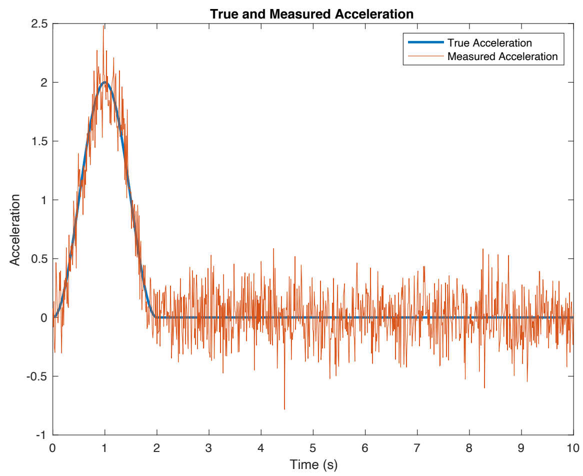


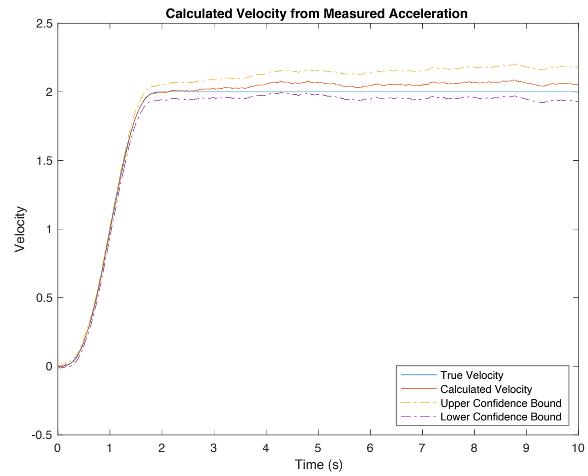




10

E80
Experimental Engineering

# Measured Acceleration and Velocity with Observation Bounds

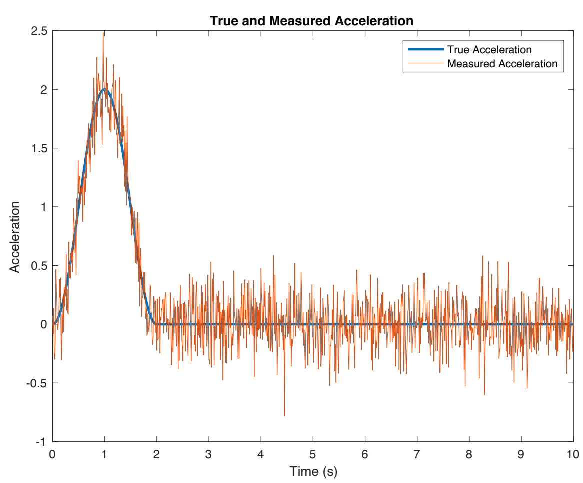


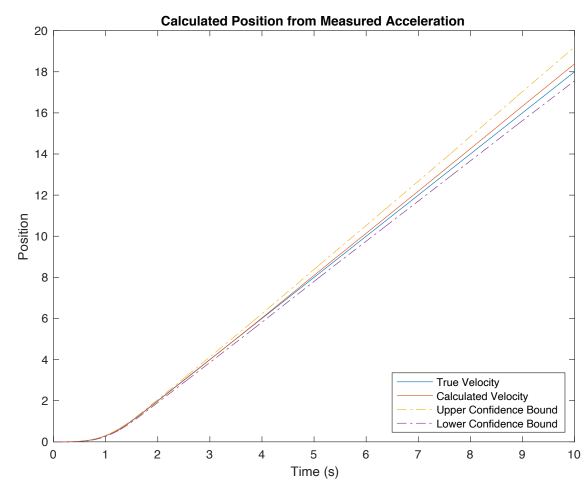




11

E80
Experimental Engineering

# Measured Acceleration and Position with Observation Bounds

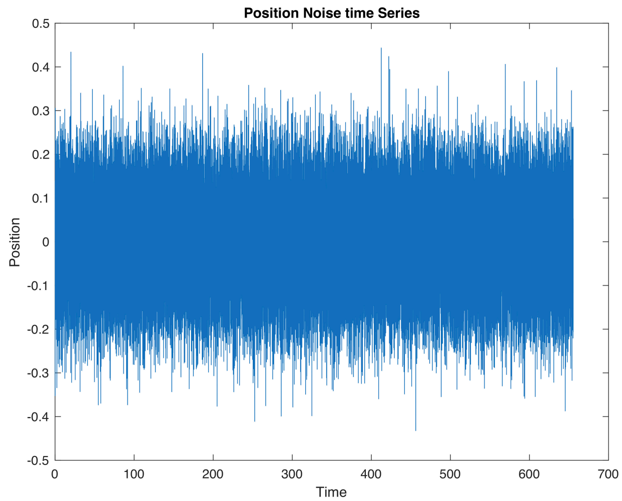



12

E80
Experimental Engineering

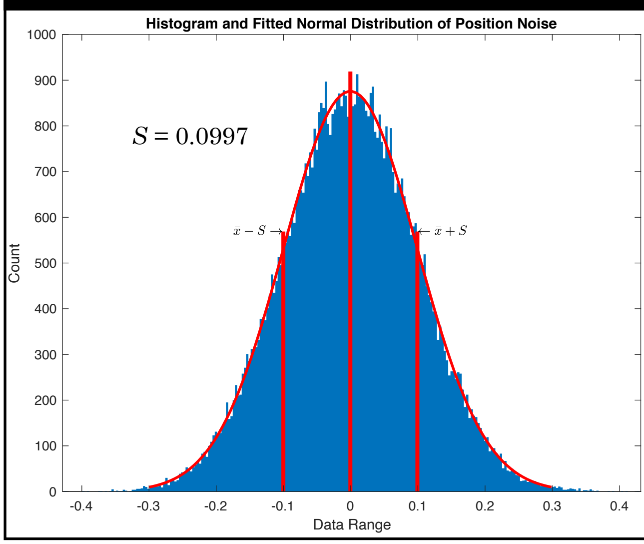
# Zero-Mean Position Time Series

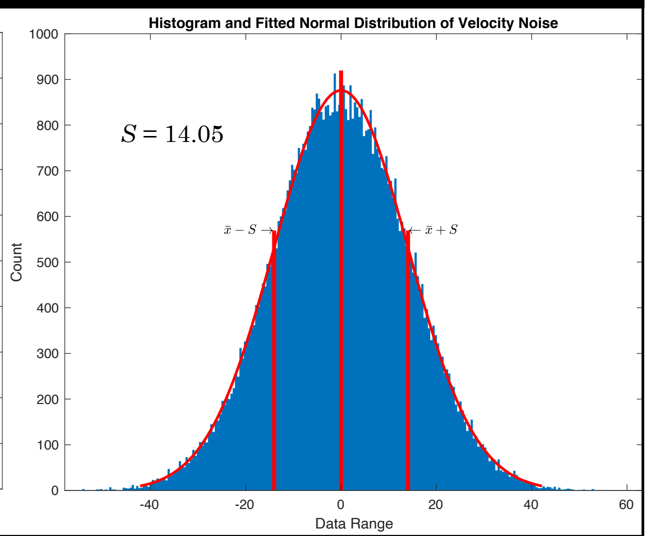



13

E80
Experimental Engineering

# Position Noise and Velocity Noise

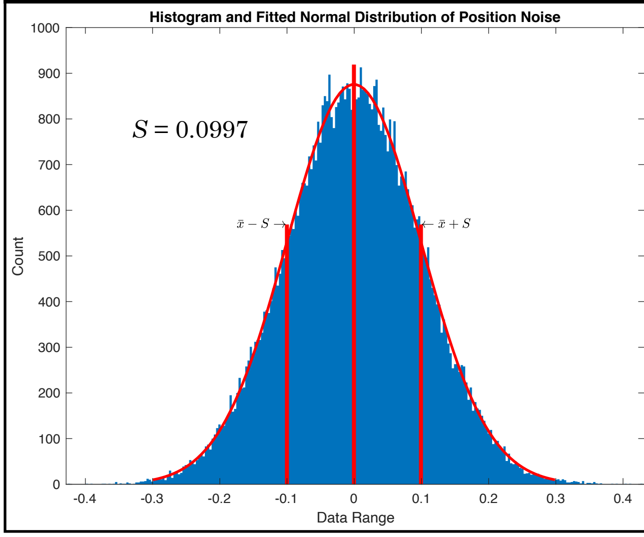


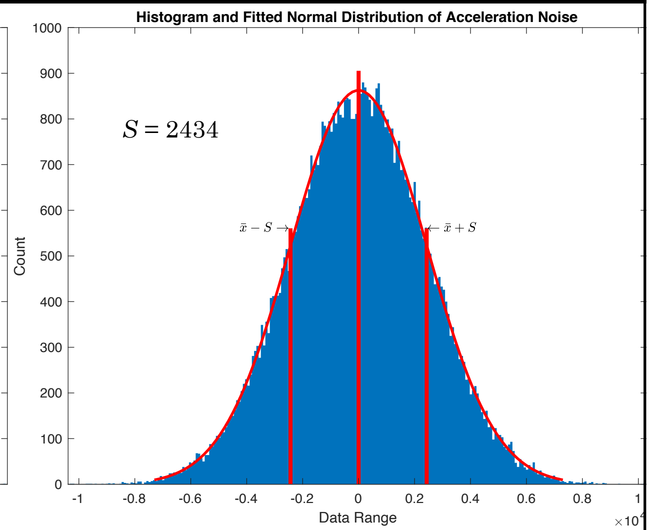




14

# E80
Experimental Engineering

## Position Noise and Acceleration Noise

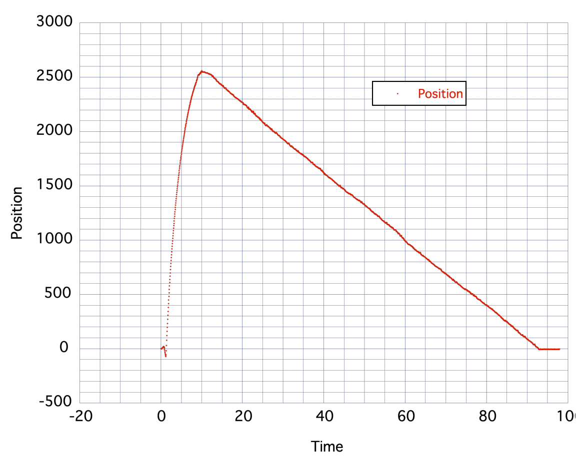



15

# E80
Experimental Engineering

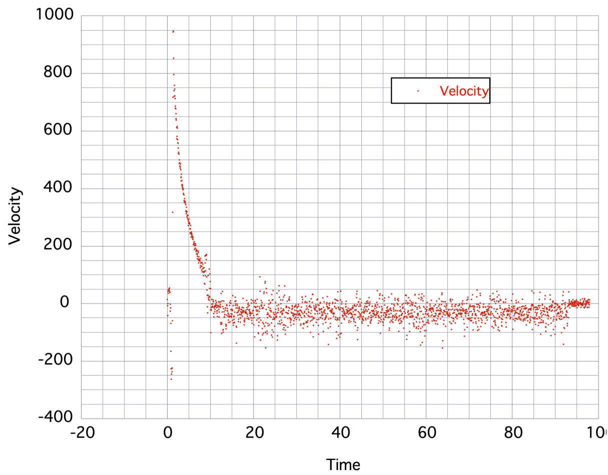
## Experimental Position Data

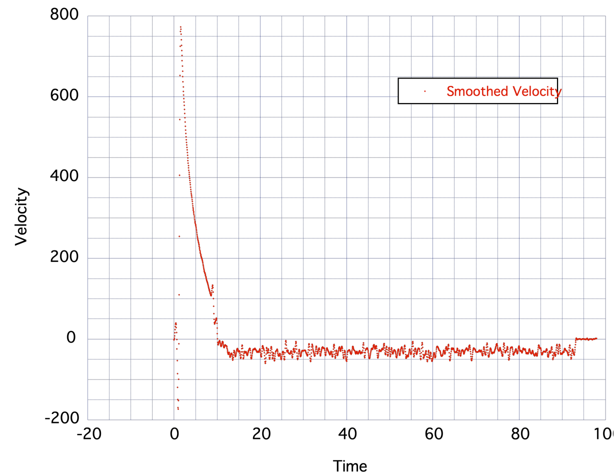



16

E80
Experimental Engineering

# Straight and Smoothed Velocity

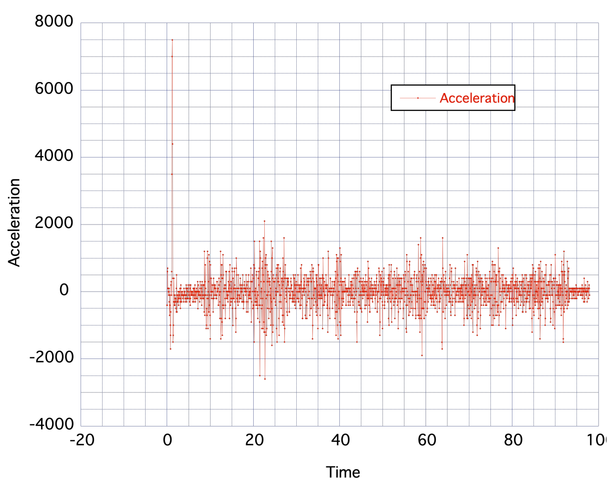


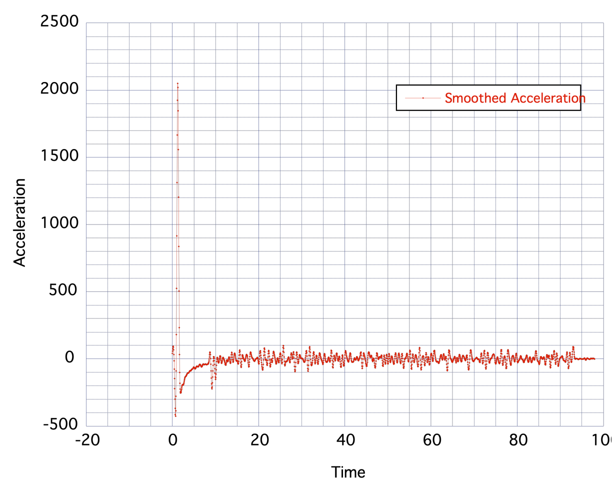




17

E80
Experimental Engineering

# Straight and Smoothed Acceleration




18

E80
Experimental Engineering

# Takeaways

1. Integrating numerical data decreases noise, but is susceptible to bias and random walk errors.
2. Differentiating numerical data increases noise. The increase is most easily determined numerically.
3. Advanced methods for dealing with differentiation exist.
4. When reporting integrated time-series data, plot the data and the confidence interval bounds.
5. When reporting differentiated time-series data, report standard deviation or confidence bounds values. Plotting bounds is likely to be too messy.

19