

**EL7021 Robotics &
Autonomous Systems**

Prof. Martin Adams

Room : 309

Extn : 84209

Email : martin@ing.uchile.cl

Part I

Introduction

Applications that Require Sensors:

1. Nature: Animals & humans navigate and find objects.
2. Closed loop control systems for multivariable process control (eg. controlling chemical plant).
3. Robotics - Autonomous vehicle navigation.
4. Control of an engine (integrate temperature, pressure measurements)

Chapter 1 in the first recommended text book below gives a good introduction of the use of sensor fusion in nature.

Course Contents

My part of “Sensor Fusion” will comprise:

1. Sensors in Autonomous Robotics (Sensor Modelling).
2. Combining Information – Data Fusion
3. System Models – Models for fusing past and present sensor information.

Sensor Fusion examples are many – I will draw many examples from an interesting (my!!!) research field – *Robotics*.

Recommended Text Books

These notes take material from many research papers, too many to mention here.

4 good text books (used here) are:

1. Mongi A. Abidi & Rafael C. Gonzales, “*Data Fusion in Robotics & Machine Intelligence*”, Academic Press 1992.
2. Martin D. Adams, “*Sensor Modelling, Design & Data Processing for Autonomous Navigation*”, World Scientific Publications 1999.
3. S.M. Bozic, “*Digital & Kalman Filtering*”, Edward Arnold Publishers, 1979.
4. J. Borenstein, H.R. Everett & L. Feng, “*Where am I? Sensors and Methods for Mobile Robot Positioning*”,
[http : //www.personal.engin.umich.edu
~johannb/position/htm](http://www.personal.engin.umich.edu/~johannb/position/htm)

Part II: Sensors in Engineering Applications

Sensors used in Tracking and Robotics:

1. Cameras - Use of Vision in Tracking and Robotics.
2. Active Triangulation Sensors.
3. LADAR - Laser Detection and Ranging Sensors.
4. RADAR - Radio Detection and Ranging Sensors.
5. SONAR - Sound Navigation and Ranging Sensors.

Stereo Vision for Range Determination

Range calculated from difference between 2 images.

- Disparity between two images,
- Finding absolute orientation of cameras,
- Computation of depth,
- Research Issue: Finding conjugate points.

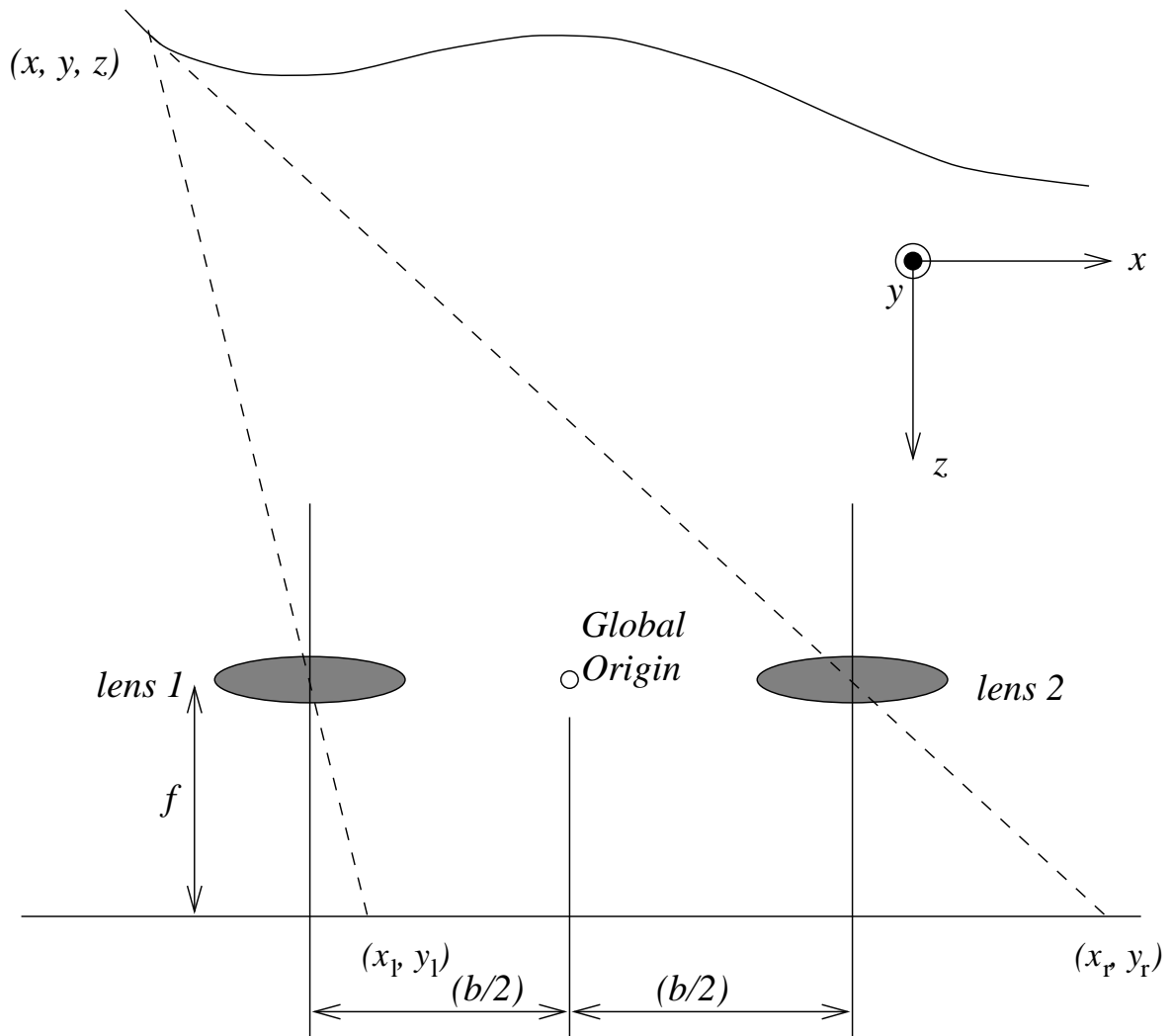
Stereo Vision for Range Determination

Parallel and skewed baseline stereo systems.



Stereo Vision: Disparity

Simple Case: 2 cameras, both optical axes parallel.



Measure x_l , y_l , x_r and y_r w.r.t. centres of left and right lenses respectively.

Stereo Vision: Range Calculation

$$\frac{x_l}{f} = \frac{x + b/2}{z} \quad \text{and} \quad \frac{x_r}{f} = \frac{x - b/2}{z} \quad (0.1)$$

coordinates x , y , z of the point P measured w.r.t. global origin.

Out of the plane of the page:

$$\frac{y_l}{f} = \frac{y_r}{f} = \frac{y}{z} \quad (0.2)$$

where f is the distance of both lenses to the image plane. Note from equations 0.1 that:

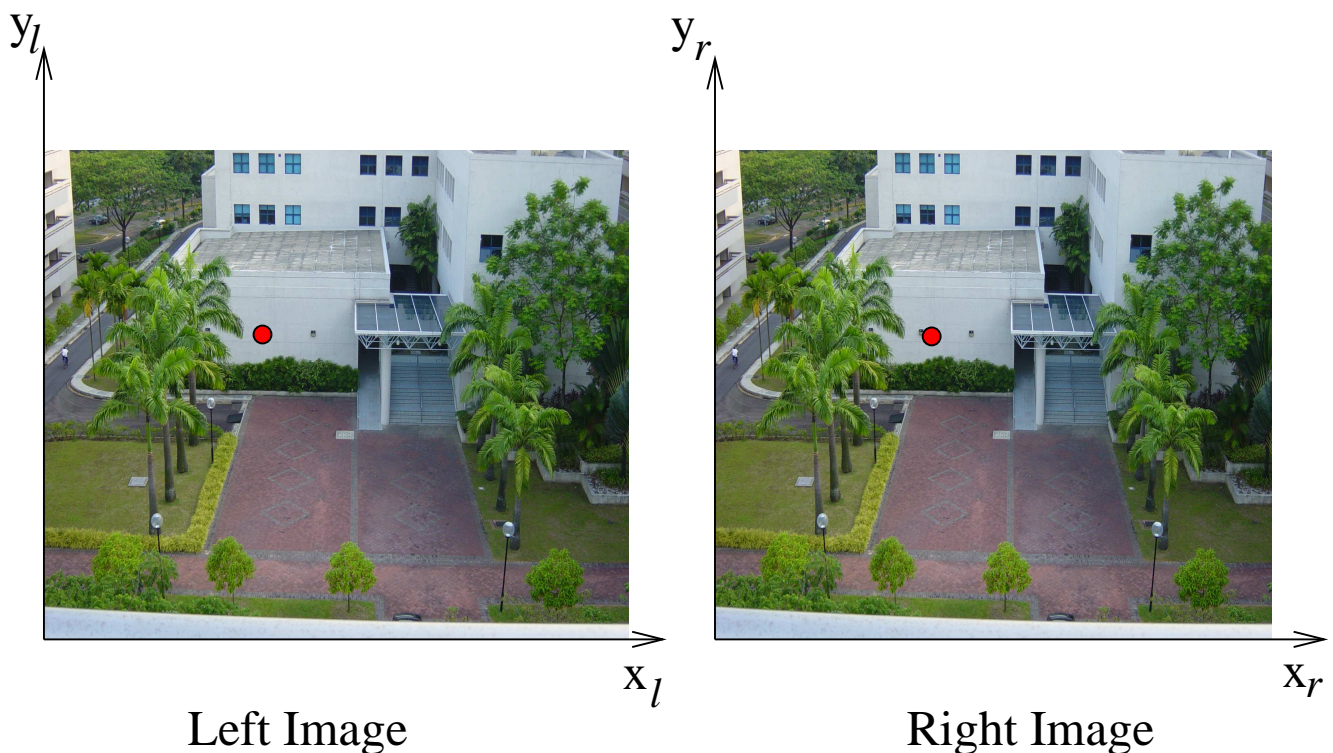
$$\frac{x_l - x_r}{f} = \frac{b}{z} \quad (0.3)$$

this *difference* in the image coordinates, $x_l - x_r =$ *disparity*. Solving (0.1 and 0.2):

$$x = b \frac{(x_l + x_r)/2}{x_l - x_r}, \quad y = b \frac{(y_l + y_r)/2}{x_l - x_r}, \quad z = b \frac{f}{x_l - x_r}$$

Stereo Vision: Range Calculation

Actual coords. calculated from image coords:



$$x = b \frac{(x_l + x_r)/2}{x_l - x_r}, \quad y = b \frac{(y_l + y_r)/2}{x_l - x_r}, \quad z = b \frac{f}{x_l - x_r}$$

Stereo Vision: Range Calculation

Observations from these equations:

1. Distance is inversely proportional to disparity. Distance to near objects can be measured more accurately due to greater number of image pixels between corresponding points.
2. Disparity is proportional to b . For a given disparity, accuracy of depth estimate increases with increasing baseline b .
3. As b is increased, some objects may appear in one camera image, but not in the other.
4. 1 point visible from both cameras produces a pair of image points – a *conjugate pair*. Each image point lies on a line – the *epipolar line*. (Here all epipolar lines parallel to x-axis – in general not true, see below).

Stereo Vision: General Case – Camera Orientation

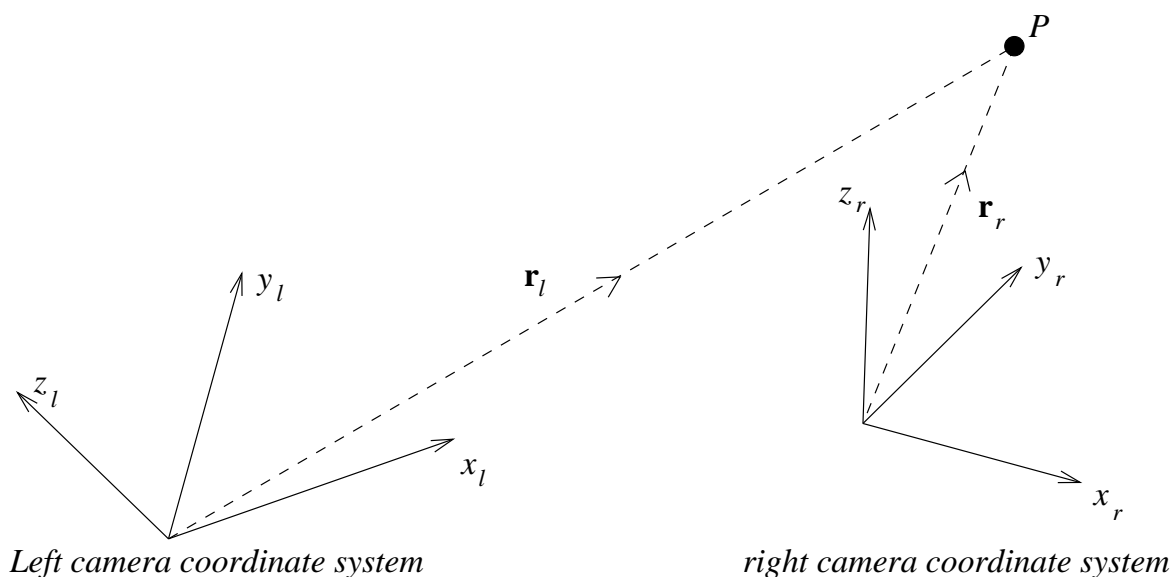
Above idealised case difficult – not easy to align two cameras parallel to each other and perpendicular to their common baseline.

More useful to turn the cameras a little towards each other – their optical axes come closest within region of interest.



Stereo Vision: General Case

Absolute orientation vectors are necessary to solve this.



$\mathbf{r}_l = (x_l, y_l, z_l)$ = position vector of P measured in left image,

$\mathbf{r}_r = (x_r, y_r, z_r)$ = measured in right image,
then:

$$\mathbf{r}_r = \mathbf{R}\mathbf{r}_l + \mathbf{r}_o \quad (0.4)$$

\mathbf{R} is a 3×3 rotation matrix and \mathbf{r}_o = offset translation vector between two cameras.

Stereo Vision: General Case

Expanding equation 0.4 gives:

$$r_{11}x_l + r_{12}y_l + r_{13}z_l + r_{14} = x_r \quad (0.5)$$

$$r_{21}x_l + r_{22}y_l + r_{23}z_l + r_{24} = y_r \quad (0.6)$$

$$r_{31}x_l + r_{32}y_l + r_{33}z_l + r_{34} = z_r \quad (0.7)$$

r_{11} = first element of \mathbf{R} r_{14} = first element of vector \mathbf{r}_o .

Above equations have two uses:

1. \mathbf{r}_r could be found if \mathbf{R} and \mathbf{r}_o and \mathbf{r}_l were known.
2. The system could be calibrated and $r_{11}, r_{12}...$ found, given corresponding values of x_l, y_l, z_l, x_r, y_r and z_r .

To carry out task 2 – 12 unknowns requiring 12 equations!

Hence – for a given scene, 4 conjugate points required for complete calibration.

Stereo Vision: General Case

Definition: *Epipolar Line* – Intersection of plane containing image point (in environment) and centre points of each camera lens, with each image plane produces epipolar lines in each image plane.

An object imaged on the epipolar line in the left image can only be imaged on the corresponding epipolar line in the right image (if it is imaged at all).

Stereo Vision: Calculating Depth

Actual points x_l , y_l etc *not* known, only their projections x'_l , y'_l etc. known.

Given focal lengths of the cameras is f :

$$\frac{x'_l}{f} = \frac{x_l}{z_l} \quad \text{and} \quad \frac{y'_l}{f} = \frac{y_l}{z_l} \quad (0.8)$$

From eqns. 0.5, 0.6, 0.7 and 0.8 depths z_l and z_r can be computed from *any two* of the equations:

$$\left(r_{11} \frac{x'_l}{f} + r_{12} \frac{y'_l}{f} + r_{13} \right) z_l + r_{14} = \frac{x'_r}{f} z_r \quad (0.9)$$

$$\left(r_{21} \frac{x'_l}{f} + r_{22} \frac{y'_l}{f} + r_{23} \right) z_l + r_{24} = \frac{y'_r}{f} z_r \quad (0.10)$$

$$\left(r_{31} \frac{x'_l}{f} + r_{32} \frac{y'_l}{f} + r_{33} \right) z_l + r_{34} = z_r \quad (0.11)$$

\mathbf{r}_l and \mathbf{r}_r can then also be found.

Stereo Vision: Finding Conjugate Points

Key problem in stereo vision:

How to solve correspondence problem – which points *truly correspond* in each image?

Basic principle: Study each image separately – extract features such as:

- Distinctive gray-level patterns that can be matched with confidence (eg. edges).
- Gray-level corners (eg. where brightness surface has nonzero Gaussian curvature).

Stereo Vision: Gray Level Matching

Consider smooth surface: Neighbouring points will map onto neighbouring points in both images.

Therefore match gray-level waveforms on corresponding epipolar lines.

Define *brightness* as *image irradiance* (power of radiant energy per unit area).

Gray-level at given pixel in an image corresponds to the image irradiance $E(x'_l, y'_l)$ at pixel (x'_l, y'_l) .

Gray-level matching: Find a function $z(x, y)$ by searching for points in both images such that:

$$E_l(x', y') = E_r(x', y') \quad (0.12)$$

l and r subscripts are image irradiance functions for left and right images.

Stereo Vision: Gray Level Matching

For simple geometrical configuration of equal base-line stereo, and eqns 0.1 and 0.12:

$$E_l \left(f \frac{x + b/2}{z(x, y)}, y' \right) = E_r \left(f \frac{x - b/2}{z(x, y)}, y' \right) \quad (0.13)$$

Transforming to image coordinates:

$$\frac{x'}{f} = \frac{x}{z} \quad \text{and} \quad d(x', y') = \frac{bf}{z} \quad (0.14)$$

In order to check neighbouring pixels, a *disparity function* $d(x', y')$ must be found, such that:

$$E_l \left(x' + \frac{1}{2}d(x', y'), y' \right) = E_r \left(x' - \frac{1}{2}d(x', y'), y' \right) \quad (0.15)$$

Problem – find pairs of pixels on corresponding epipolar lines where equation 0.15 is approx. obeyed, and $z(x, y)$ and $d(x', y')$ are smooth.

Stereo Vision: Other Matching Methods

Problems with Gray Level matching – If conjugate patches in both images have similar brightness patterns.

Possible solutions:

- **Correlation**

Take a patch from one image, correlate with all patches along the corresponding epipolar line in other image. Point with highest correlation chosen.

See *Horn* for more details.

- **Edge Matching**

First find any edges (sudden brightness discontinuities) in images then correlate the edges between the two images.

See for example *Ayache* for more details.

Optical Sensing – Controlling the Illumination

Stereo vision for range estimation – *difficult!*:

- Correspondence Problem.
- Disparity problems (Occlusions).
- Illumination (shadows).

Possible solution: Control illumination.

Known as *active* sensors.

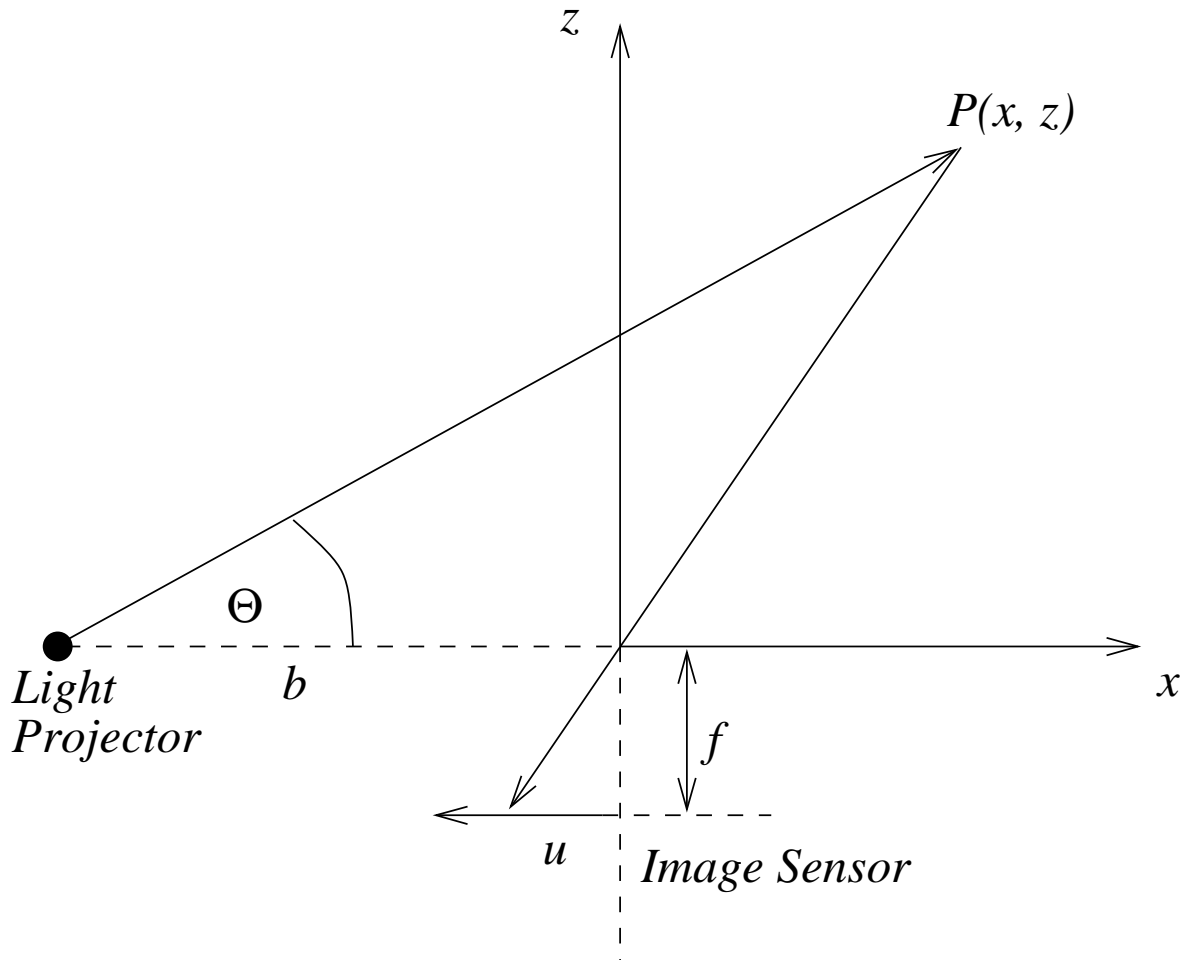
Correspondence/illumination problems eliminated.

Example: Active Triangulation

LASER – Camera Triangulation System:



Example: Active Triangulation



Geometrical method – measure θ and u .

Imaging sensor: Eg: Camera or array of photo-diodes.

$$x = \frac{bu}{f \cot \theta - u}, \quad z = \frac{bf}{f \cot \theta - u} \quad (0.16)$$

Example: Active Triangulation

f = distance of lens to imaging plane.

Measure θ and u – Actual coordinates x and z can be determined

Sensor Performance:

Triangulation Gain G_p = ratio of image resolution to range resolution.

$$\frac{\partial u}{\partial z} = G_p = \frac{bf}{z^2} \quad (0.17)$$

Note that *Range Accuracy* from 0.17:

- for given image resolution is proportional to source detector separation b .
- for given image resolution is proportional to focal length f .
- decreases with square of range z .

Example: Active Triangulation

Hence accuracy in range measurement depends on range itself!

Also necessary to know angle θ of scanning mechanism.

From equation 0.16:

$$\frac{\partial \theta}{\partial z} = G_{\theta} = \frac{b \sin^2 \theta}{z^2} \quad (0.18)$$

ie:- change in scanning angle θ , which must be resolved for given change in range z , dependent on range and scanning angle itself.

Example: Active Triangulation

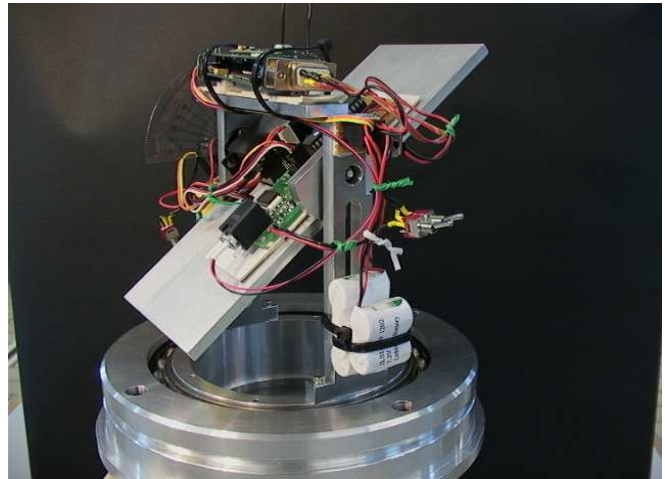
Notes regarding triangulation sensors:

1. *Baseline length b* : Smaller b = more compact sensor. Larger b – better range resolution.

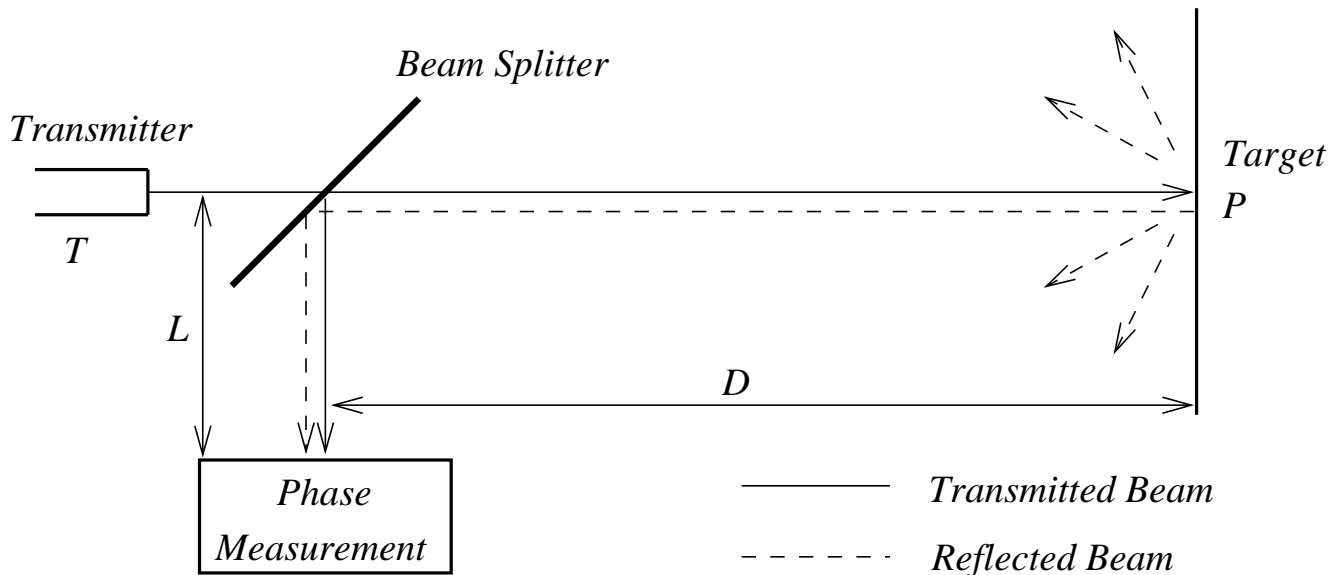
Disparity problem - as baseline b increased, illuminated point may not be visible at receiver.

2. *Detector length and focal length f* : Increasing receiver length can improve range resolution and/or field of view. However, increase in detector length = larger sensor, worse electrical characteristics – increase in noise and reduction of band-width.

LASER Detection and Ranging (LADAR)



LADAR – range finder emits electro-magnetic wave – eliminates disparity problem by keeping transmitted and received beams coaxial.



LASER Detection and Ranging (LADAR)

Transmitter illuminates target with collimated beam.

Receiver detects component reflected coaxially with transmitted beam.

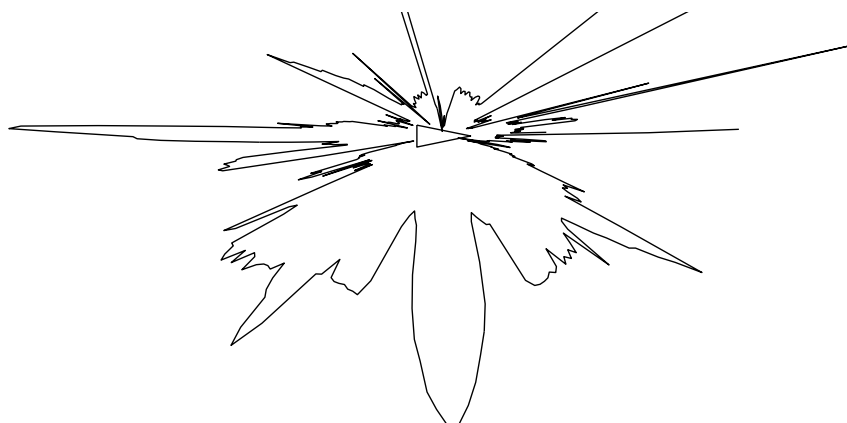
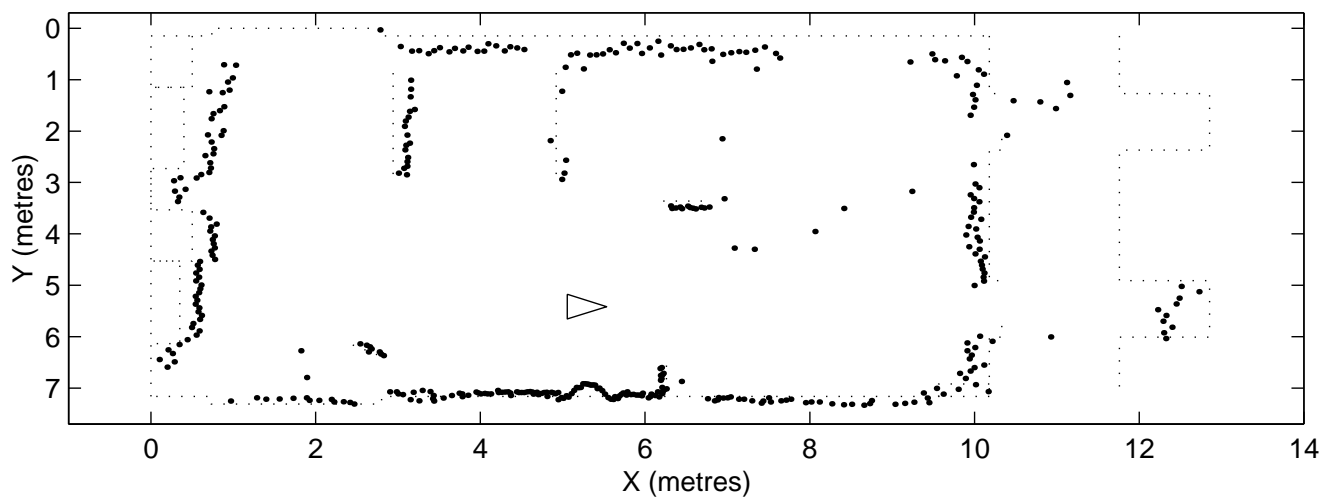
Range estimate from:

- Time of flight (TOF) of transmitted light,
- Amplitude modulated continuous wave (AMCW) phase measurement,
- Frequency modulated continuous wave (FMCW) frequency measurement.

Mechanism sweeps light beam to cover required scene.

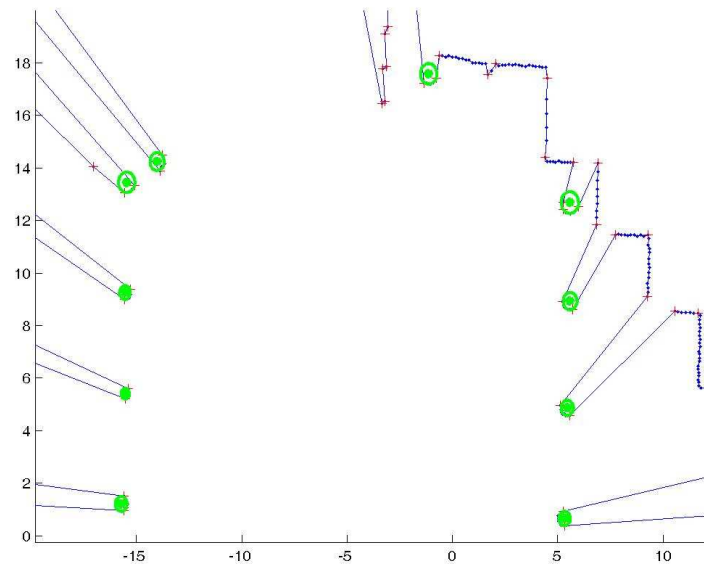
LASER Detection and Ranging (LADAR)

LADAR Indoor Range & Amplitude Data.



LASER Detection and Ranging (LADAR)

LADAR Outdoor Range Data.



LASER Detection and Ranging (LADAR)

Range Measurement Methods include:

- Time of Flight (TOF)
- Amplitude Modulated Continuous Wave (AMCW)
- Frequency Modulated Continuous Wave (FMCW)

LADAR – Range Estimation

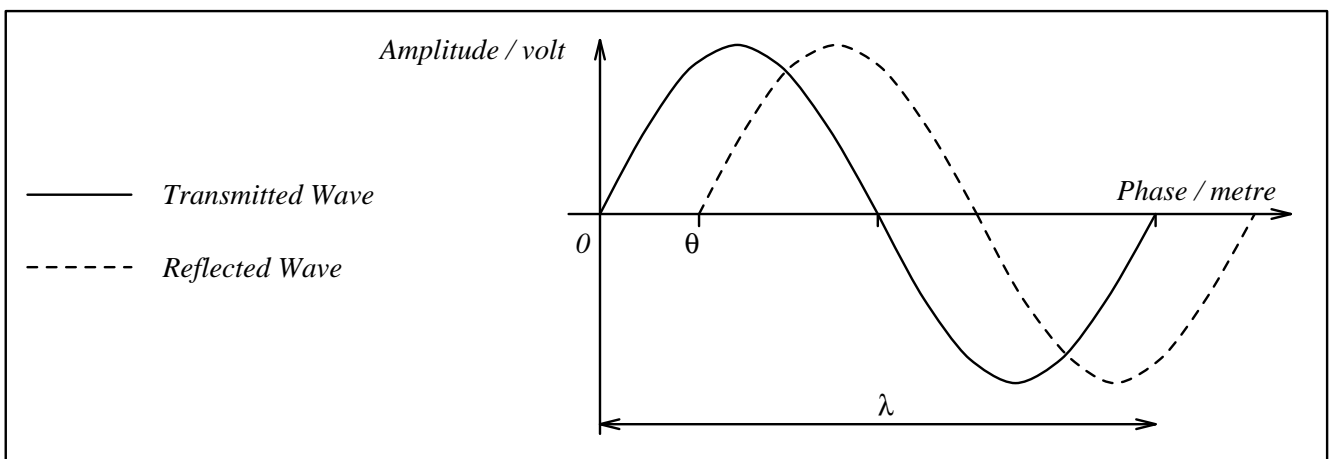
AMCW Range Estimation.

Modulate light wave with a sine wave:

$$c = f\lambda$$

c = speed of light = 3×10^8 ms⁻¹. For modulation frequency 10 MHz, $\lambda = 30$ m.

For $\lambda = 30$ m, max. range = 15m (light travels from sensor to target and back!)



LADAR – Range Estimation

Range $R \propto$ phase shift between transmitted and received light signal.

$$R = \frac{\theta \lambda}{2\pi} = \frac{\theta c}{4\pi f} \quad (0.19)$$

Note: Phase shift repeats every 2π rads.

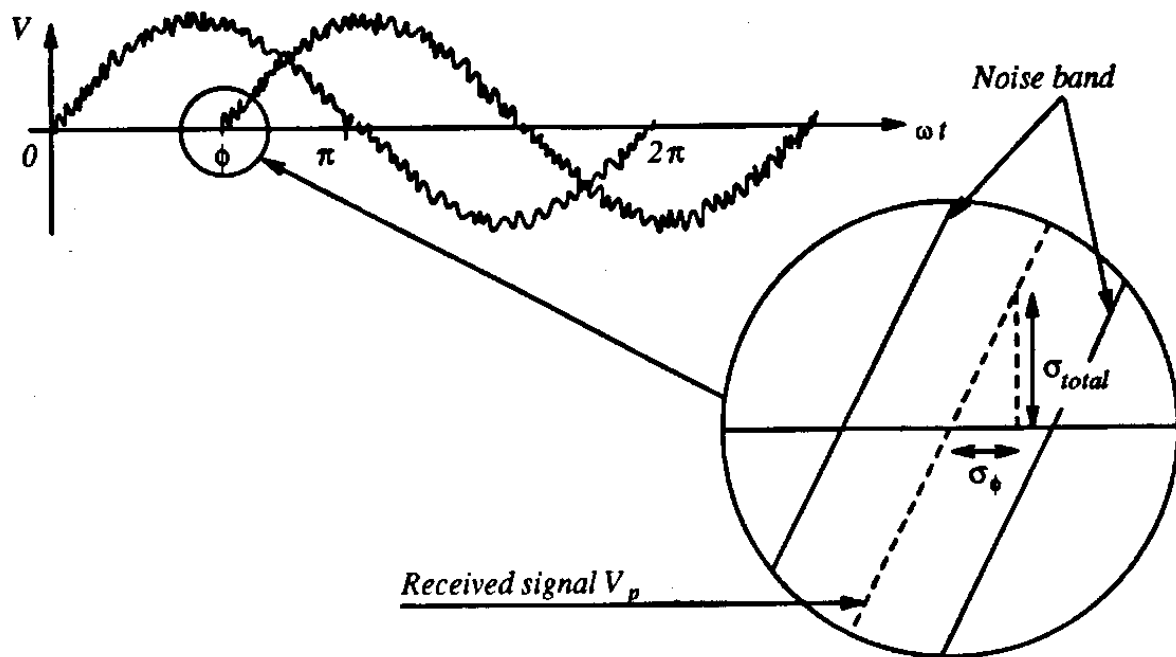
Ambiguity interval – AMCW technique only measures range up to max. $\frac{\lambda}{2} = \frac{c}{2f}$.

Received signal given by:

$$V_{rec} = V_R \sin(\omega t + \theta) \quad (0.20)$$

LADAR – Effect of Noise on Range

Received signal amplitude corrupted with noise with standard deviation σ_{amp} .



Since Range \propto Phase, amplitude noise produces phase noise along ωt axis.

From “Noise Triangle”:

$$\frac{\sigma_{amp}}{\sigma_{phase}} = \left| \frac{\partial V_{rec}}{\partial(\omega t)} \right|_{V_{rec}=0} \quad (0.21)$$

LADAR – Effect of Noise on Range

Substitute equation 0.20 into 0.21:

$$\frac{\sigma_{amp}}{\sigma_{phase}} = |V_R \cos(\omega t + \theta)|_{V_{rec}=0} = V_R$$

Therefore:

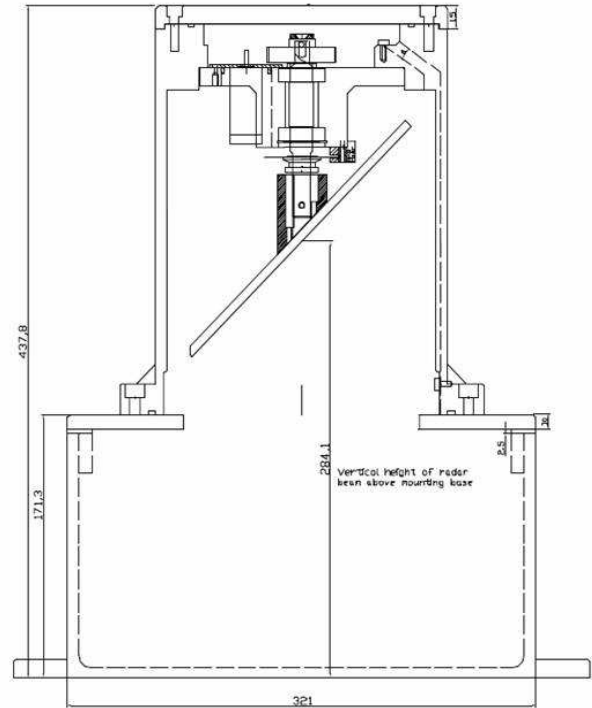
$$\sigma_{range}^2 \propto \sigma_{phase}^2 = \frac{\sigma_{amp}^2}{V_R^2} \quad (0.22)$$

Interpretation: For a constant *amplitude* noise variance, range variance in AMCW LADAR is *inversely proportional* to square of received signal amplitude.

Makes sense: Larger received amplitude, less range noise & vice-versa.

Together, equations 0.19 & 0.22 can be used to derive exact relationship between σ_{range}^2 and V_R .

RADAR = Radio Detection & Ranging

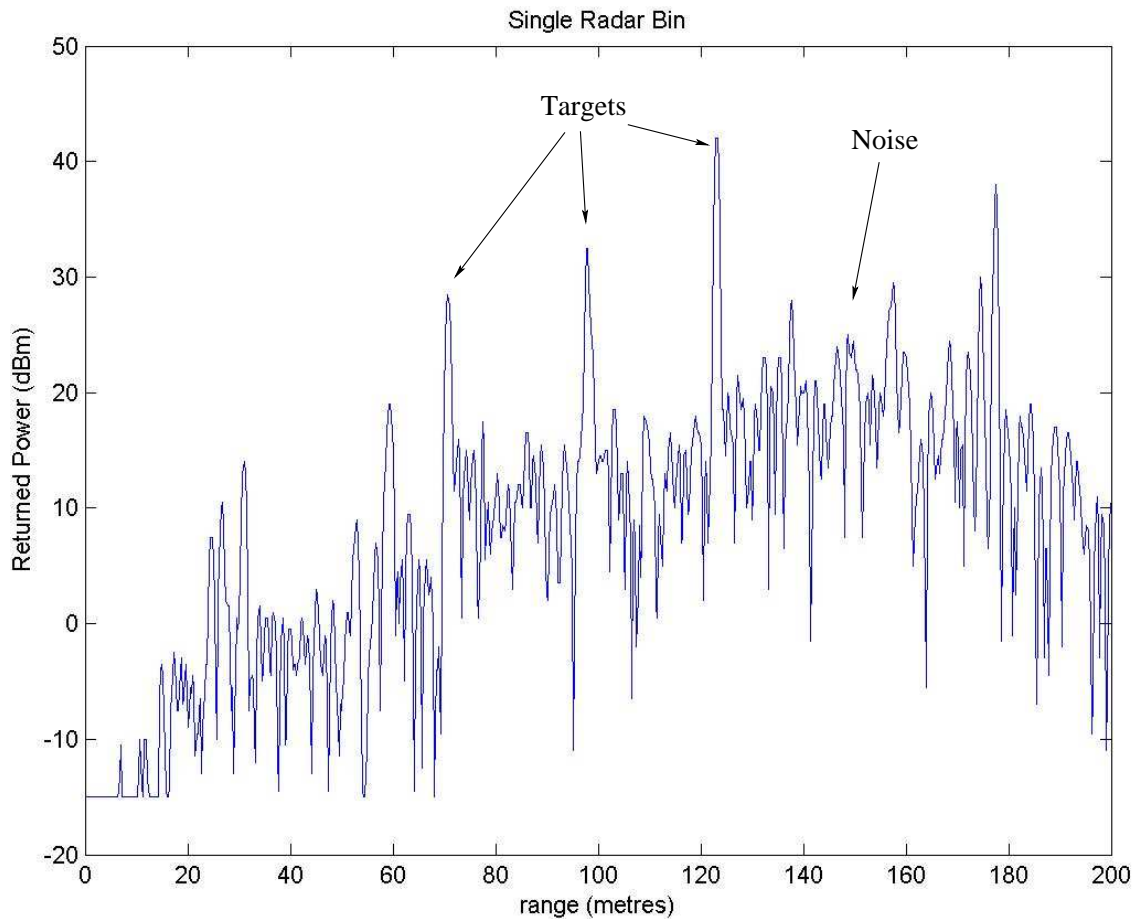


RADAR:

- Excellent ability to penetrate rain, fog, humidity.
- Can penetrate many objects – multiple line of sight targets.
- Expensive!

RADAR = Radio Detection & Ranging

Single Power – Range Spectrum (Range bin):



Problem – Finding targets within noise.

RADAR = Radio Detection & Ranging

Power – Range relation – *RADAR Equation*

$$\text{Received Power } P = \frac{P_T G^2 \lambda^2 \sigma}{(4\pi)^3 R^4 L} \propto \frac{\sigma}{R^4}$$

- P_T = Transmitted power (constant).
- G = RADAR Antennae gain (constant).
- λ = Electromagnetic radiation wavelength (constant).
- σ = Target RADAR cross section (variable)
- R = Target range (variable).
- L = RADAR losses (assumed constant).

RADAR = Radio Detection & Ranging

Particular RADAR here uses FMCW method to estimate range:

RADAR transmits a “Chirp” – signal of increasing frequency in a given time period, and then repeats this.

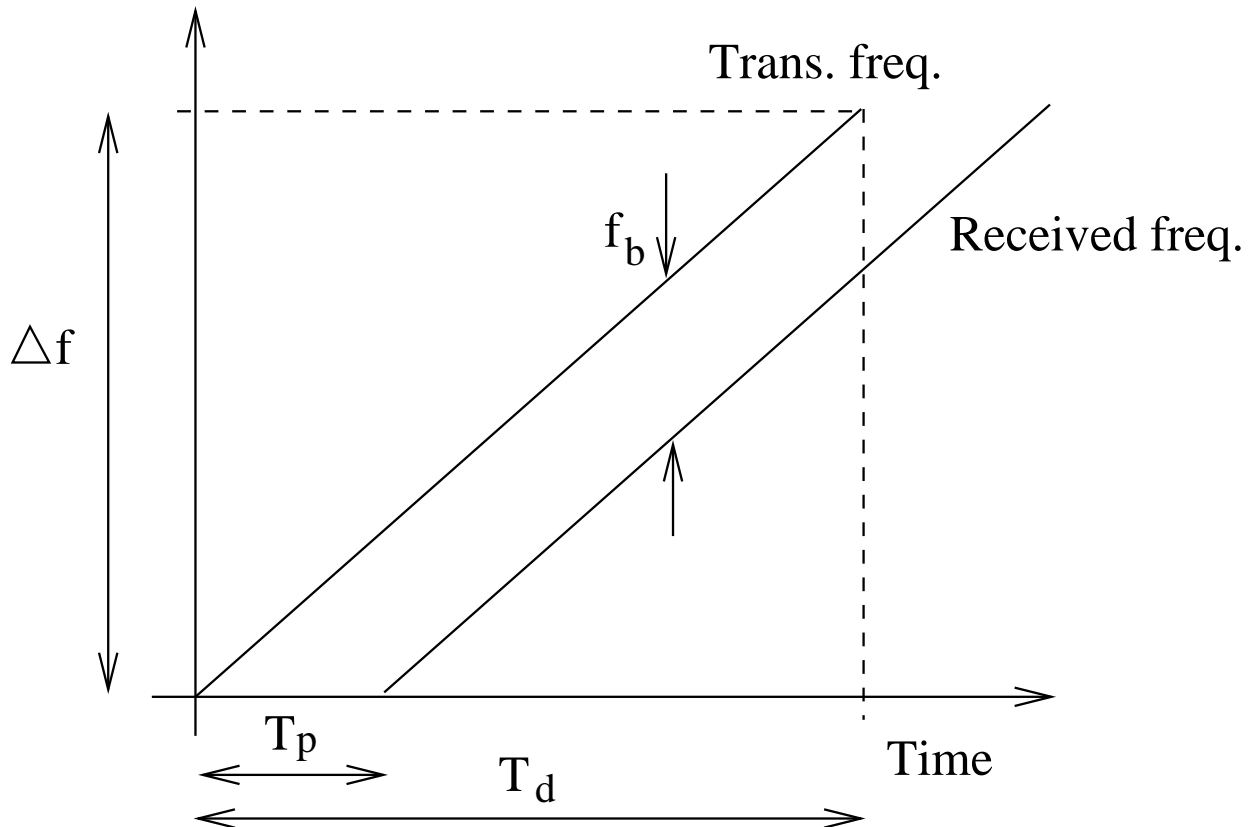
RADAR receiver compares current received frequency with current transmitted frequency. Difference = Beat Frequency \propto Range.

Power P also recorded to give entire Range bin.

RADAR = Radio Detection & Ranging

FMCW Process:

Frequency



$$\text{Beat Frequency } f_b = \frac{\Delta f}{T_d} T_p \quad (0.23)$$

RADAR = Radio Detection & Ranging

But Delay time = dist travelled/speed of radio wave:

$$T_p = \frac{2R}{c} \quad (0.24)$$

where R = range, c = speed of radio wave = 3×10^8 m/s.

Hence sub. eqn. 0.24 into 0.23:

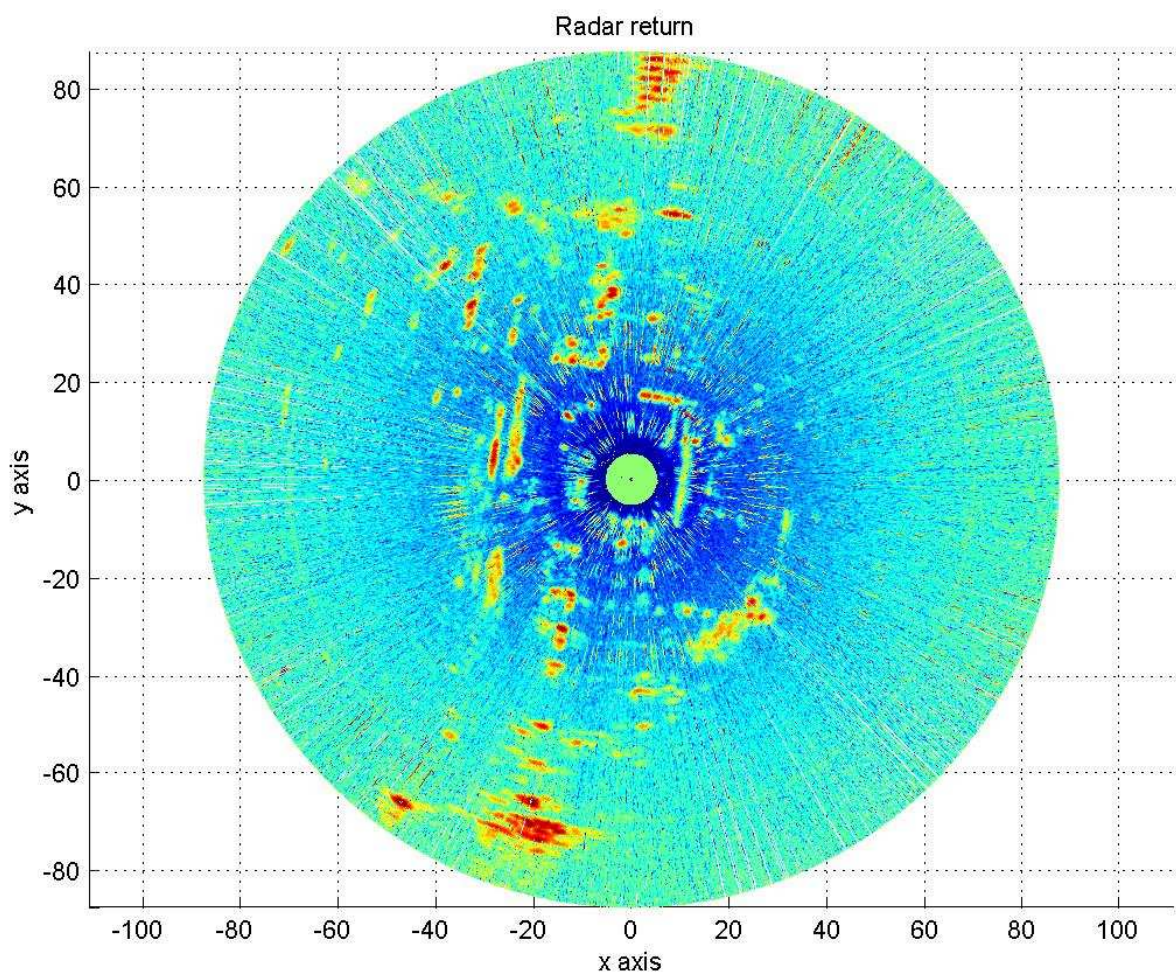
$$f_b = \frac{2R\Delta f}{c T_d} \quad (0.25)$$

and:

$$\text{Range } R = \frac{f_b T_d c}{2\Delta f} \quad (0.26)$$

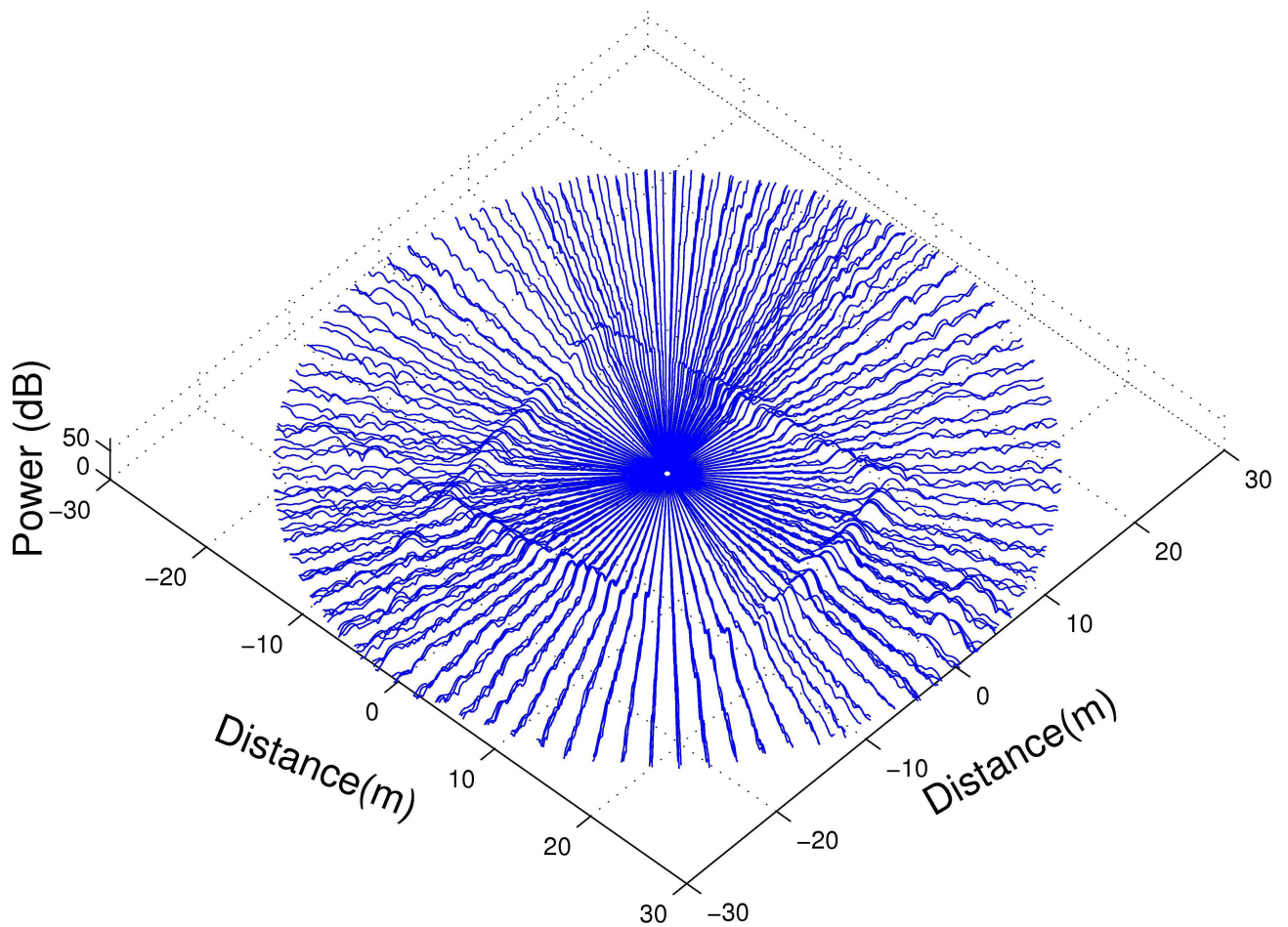
RADAR = Radio Detection & Ranging

2D Scanned RADAR Data.



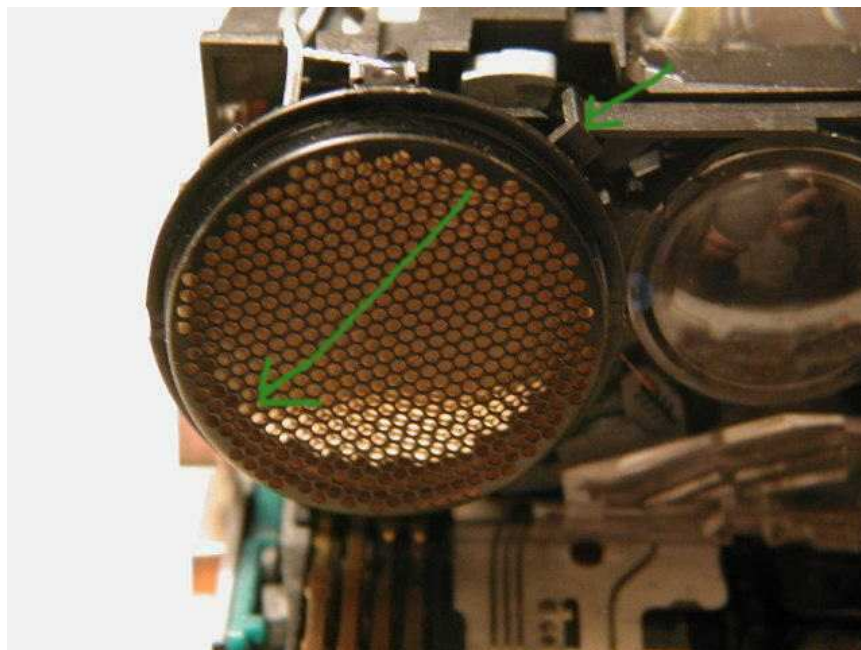
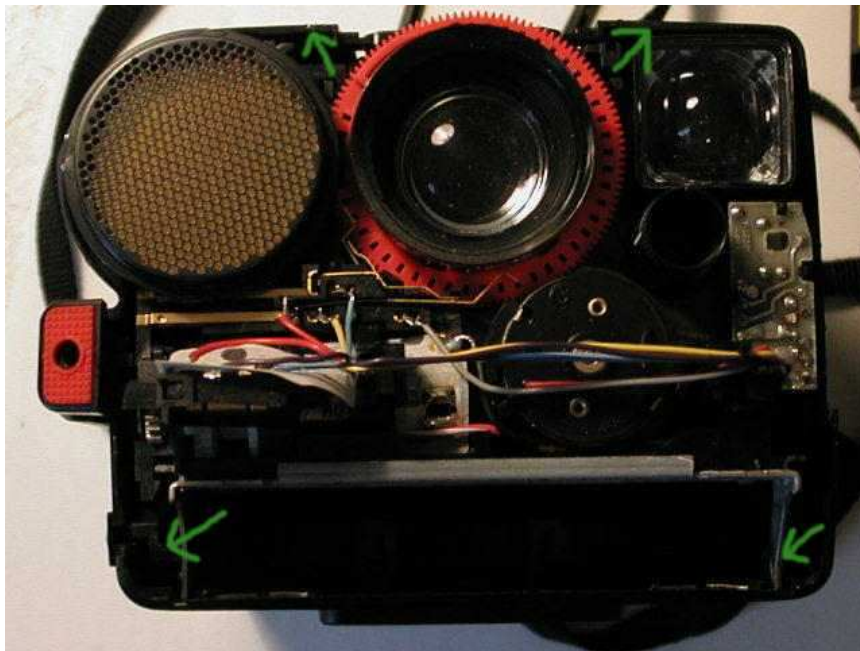
RADAR = Radio Detection & Ranging

2D Scanned RADAR Data.



SONAR = Sound Navigation & Ranging

Eg: Polaroid SONAR uses time of flight (TOF).



SONAR = Sound Navigation & Ranging

Polaroid SONAR – Single transducer.

Initially operates as transmitter – then switches mode to receiver.

After transmission, transducer settling time after pulse emission gives minimum recordable distance.

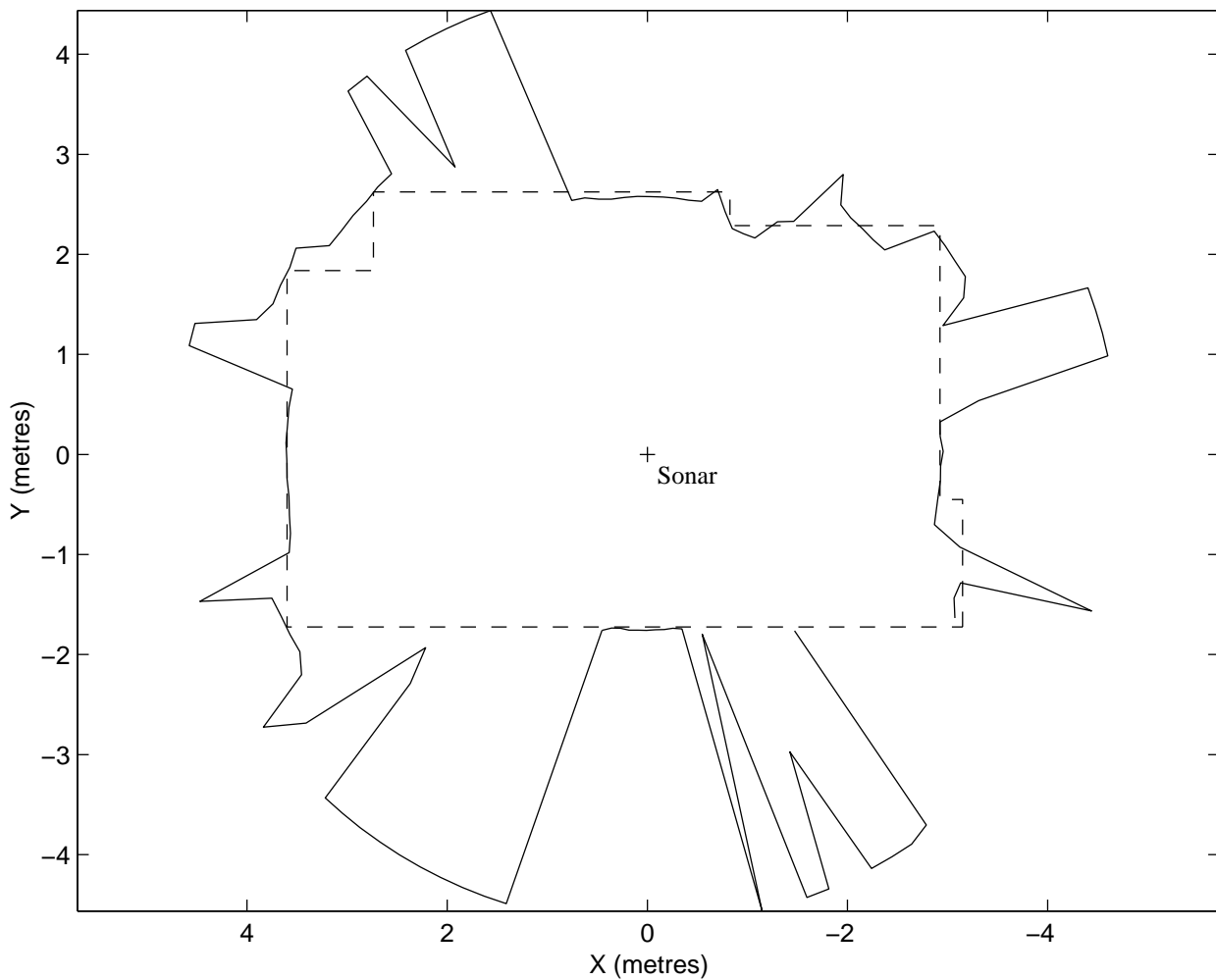
Range reading results when returned echo's amplitude exceeds threshold – time T after transmission.

For TOF SONAR range r is simply:

$$r = \frac{vT}{2} \quad (0.27)$$

v = speed of sound in air.

SONAR = Sound Navigation & Ranging



Dashed lines = simple line model of actual environment,

Solid line = actual range data recorded from scanned SONAR, positioned at cross (+).

Understanding SONAR

SONAR requires correct interpretation (Kuc & Siegel).

Sonar — The Physics of Reflection

SONAR emits longitudinal pressure wave – wavelength λ several millimetres.

Two modes of reflection possible:

1. *specular* or
2. *diffuse*.

In practice, both occur simultaneously, dependent on wavelength λ of incident wave compared with roughness R of target.

In general if:

- $\lambda \ll R$ – *diffuse* reflection occurs.
- $\lambda \gg R$ – *specular* reflection occurs.

Understanding SONAR

- Visible light $310 \text{ nm} < \lambda < 780 \text{ nm}$ incident on wall, since $\lambda \ll R$ – diffuse reflection dominates.
- Visible light incident on extremely smooth (flat) surface (eg: a mirror), $\lambda \gg R$ – specular reflection dominates.
- Polaroid SONAR has $\lambda \approx 7 \text{ mm}$. Most indoor surfaces $\lambda \gg R$ – specular reflection dominates.

SONAR: Indoor environments “Hall of mirrors”.

Large angles of incidence produce over estimates in range.

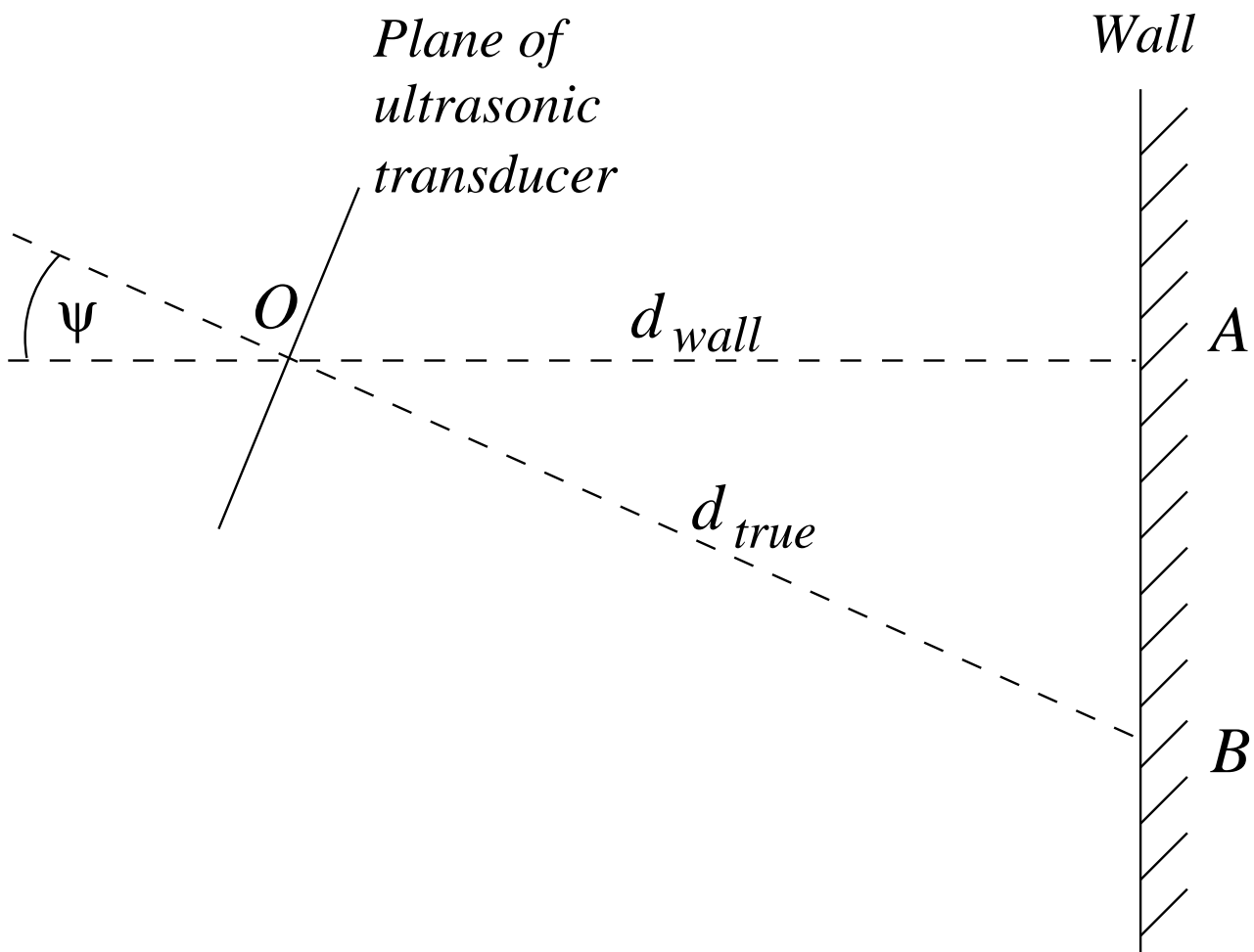
Sound wave undergoes total internal reflection several times before reaching transducer – *Specular reflection*.

Understanding SONAR – Beam Width

SONAR difficult to focus – wide beam width.

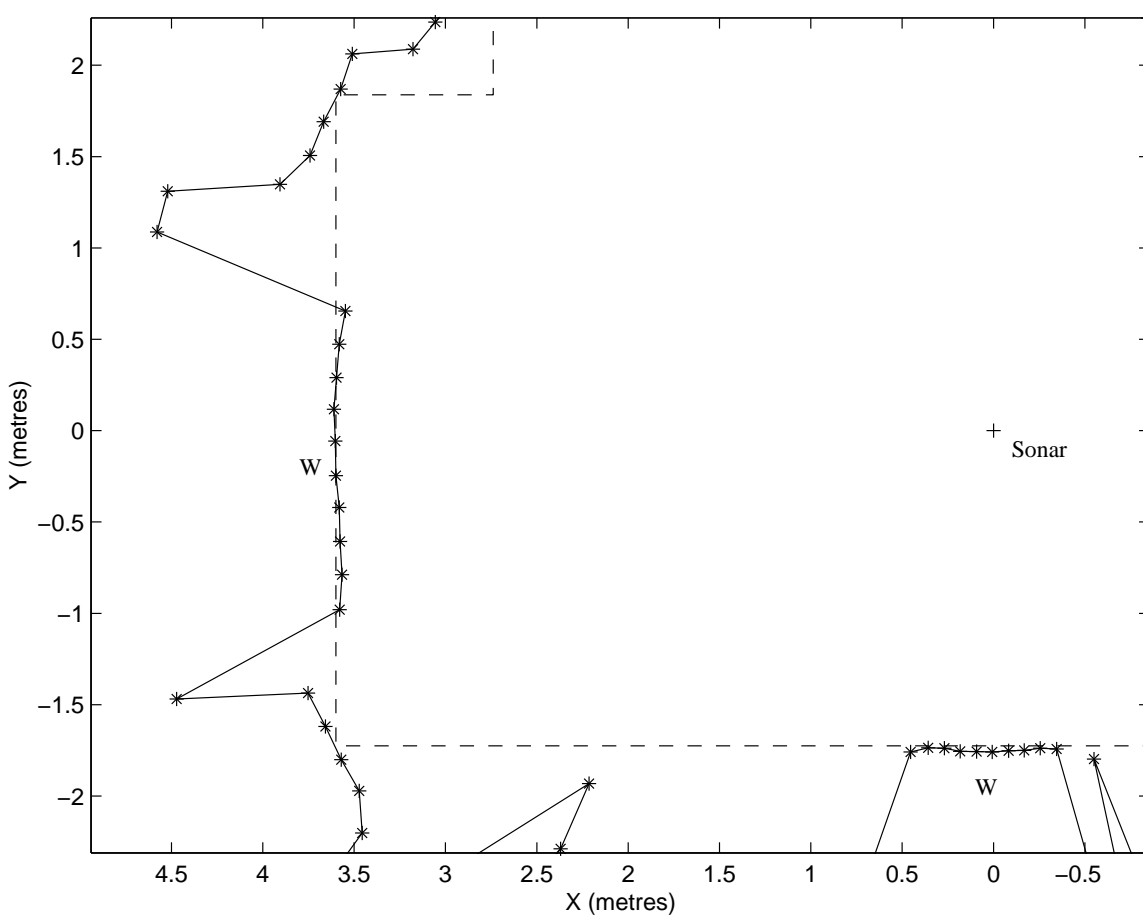
First signal above threshold received anywhere within receiver's beam width (acoustic aperture), gives range reading.

SONAR Reflection from Walls



Understanding SONAR – Beam Width

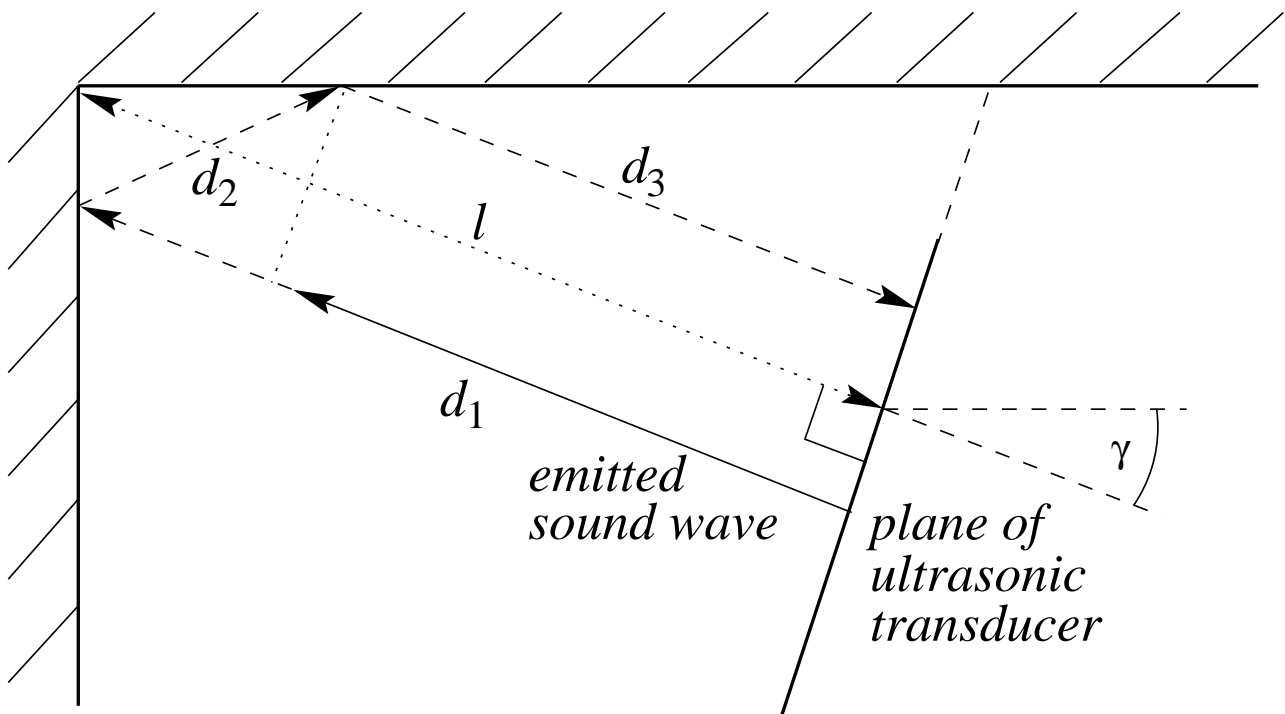
Due to beam width effect, walls appear curved as *Regions of Constant Depth*, (*RCDs*).



“For a wall to be visible, the transmitter/receiver location must have an unobstructed *perpendicular* projection to that wall.”

Understanding SONAR – Corners

SONAR Reflection from Corners



Any transmitted wavefront within sound cone will produce range reading $d_1 + d_2 + d_3$.

For 90° corner, simple geometry shows $d_1 + d_2 + d_3 = 2l$ for all transducer angles γ .

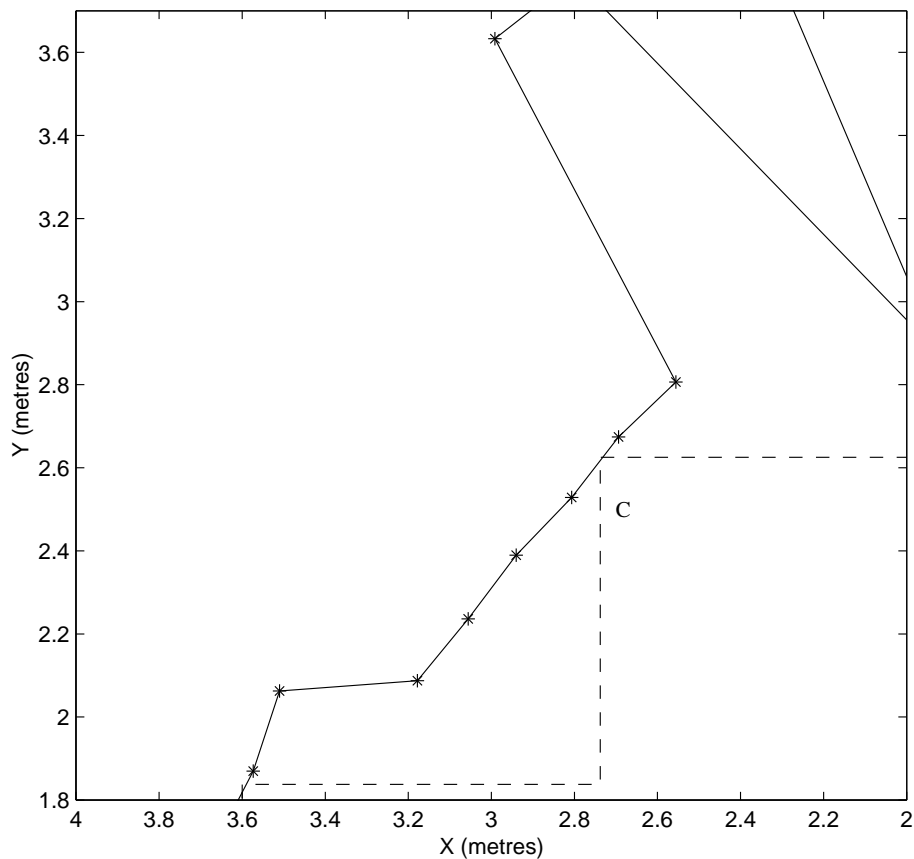
l = actual perp. distance to corner.

Hence RCD also results.

Understanding SONAR – Corners

“For corners to be visible, transducer must have unobstructed line-of-sight to their location.”

Even small 90° corners (eg: door frames), form strong reflectors for SONAR – termed *retro-reflectors*.

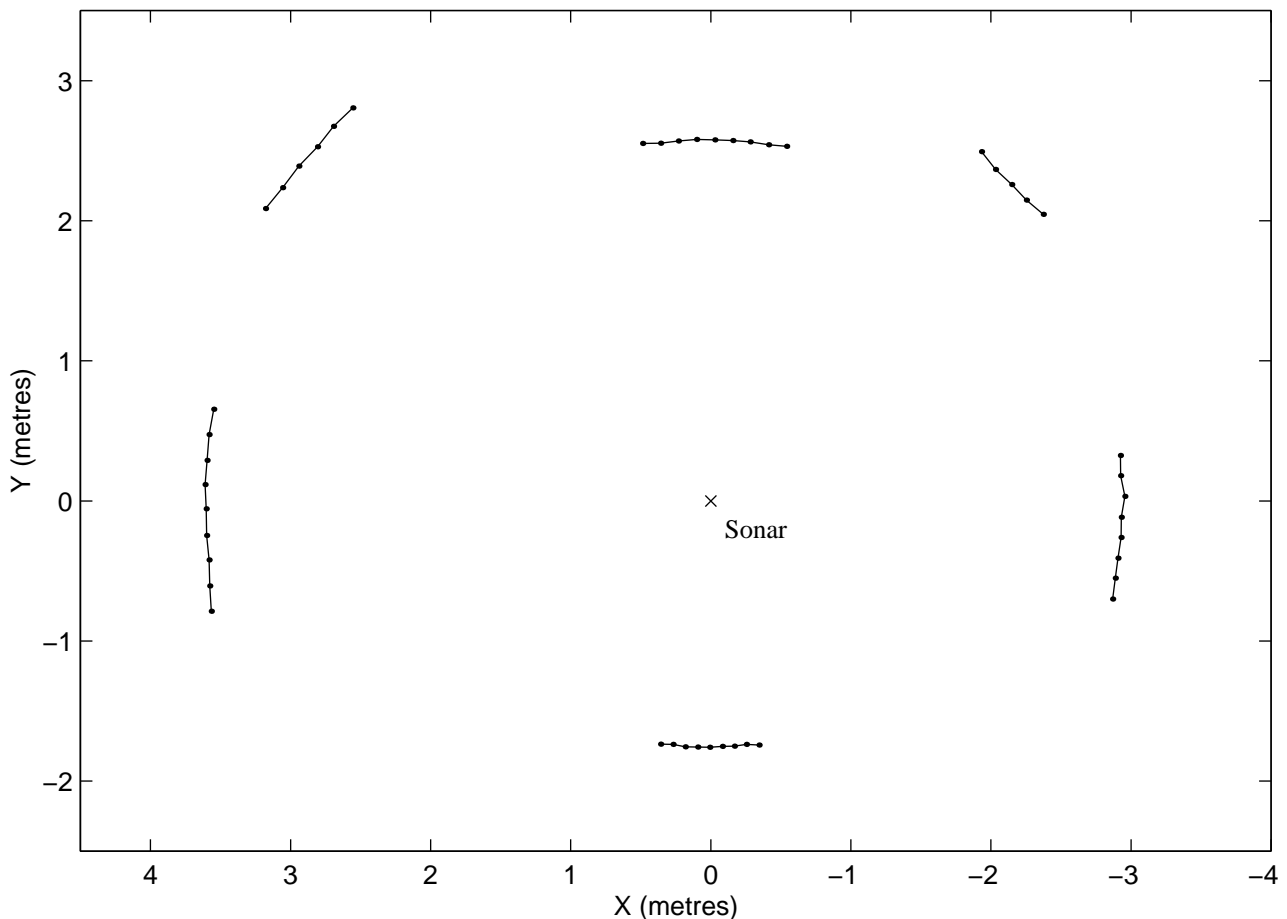


Note that the 6 readings in vicinity of corner ‘C’ have approx. same range and form an RCD.

Understanding SONAR – RCDs

From 1 scan – impossible to determine whether RCD's caused by walls or corners.

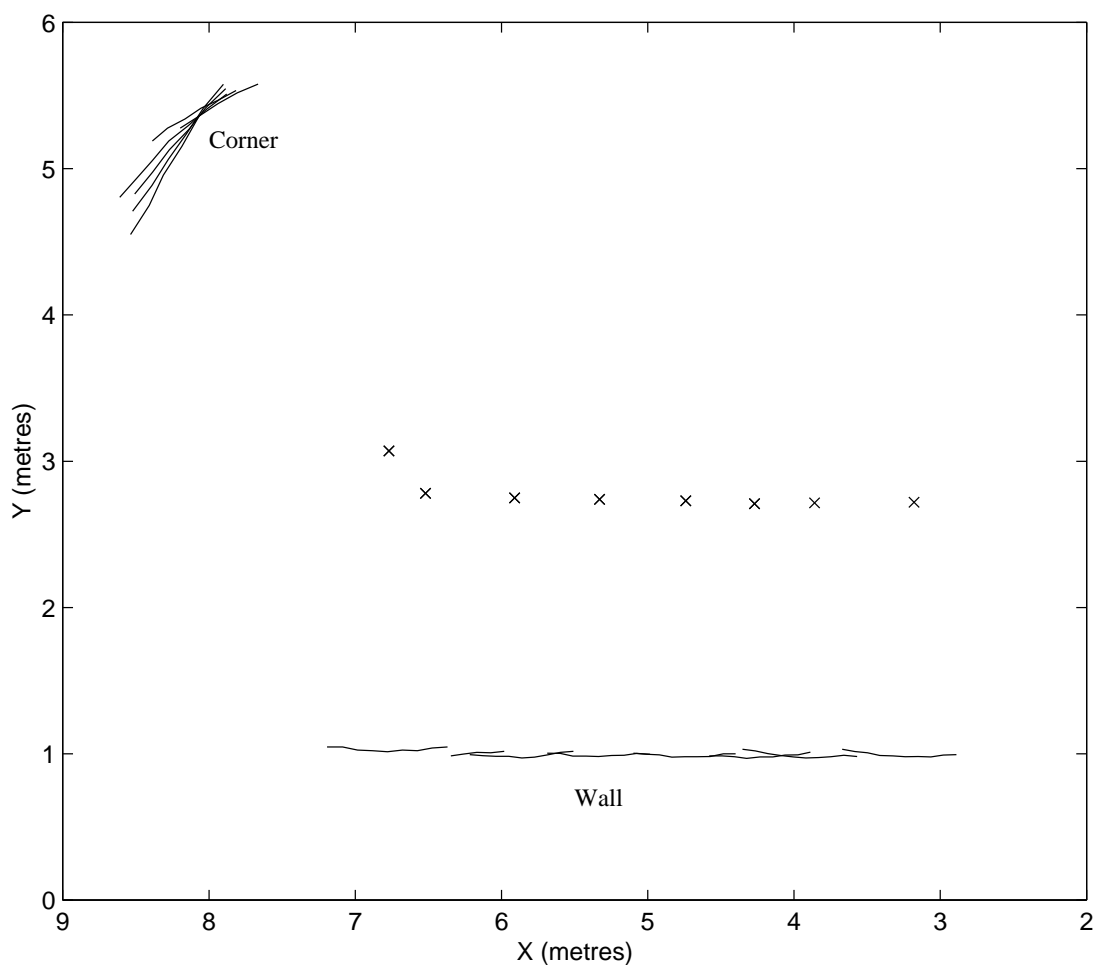
RCD's of angles $> 10^{\circ}$ and range tolerance < 4 cm shown, after extraction from single scan.



Understanding SONAR – RCDs

How to differentiate between walls and corners?

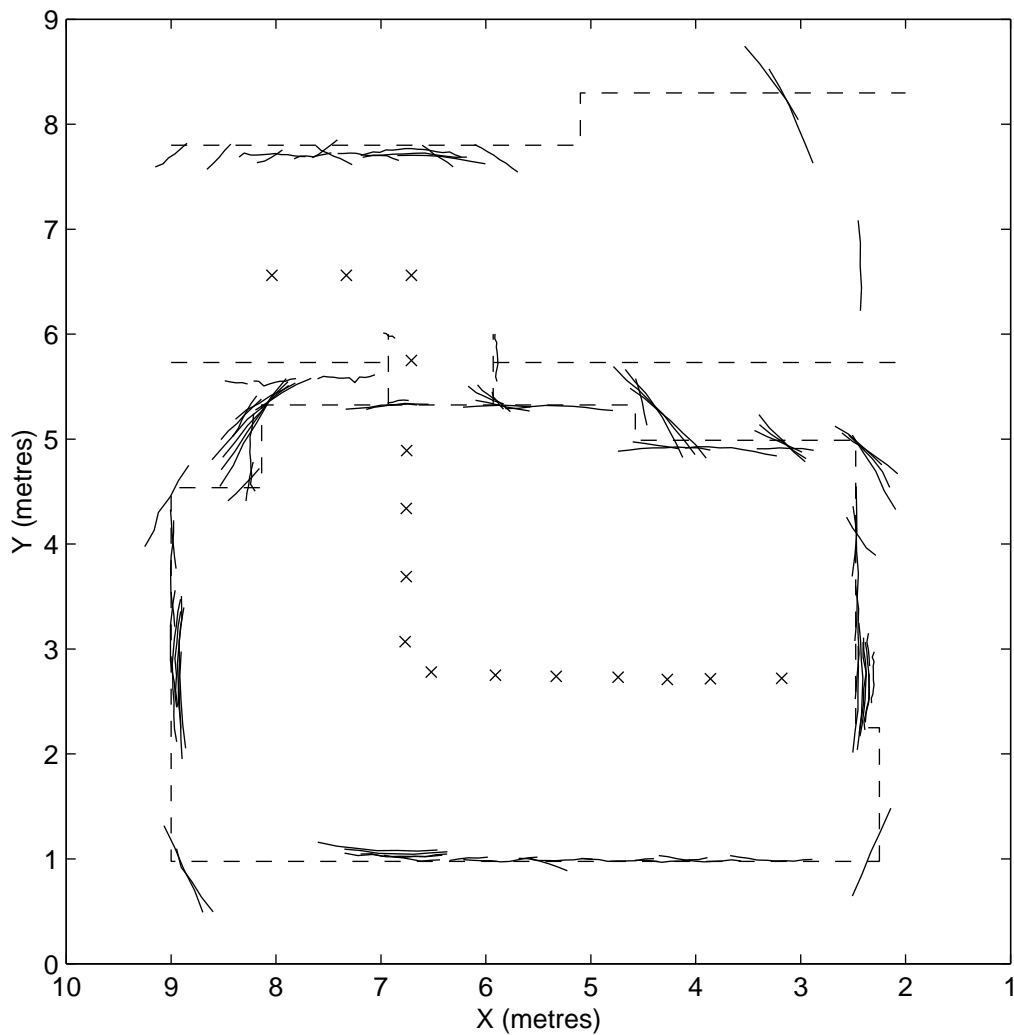
Observe motion of RCDs from different vehicle positions.



- *Wall*: RCDs *translate* tangentially along wall.
- *Corner*: RCDs *rotate* about corner.

Understanding SONAR – RCDs

RCD's extracted from 15 positions superimposed upon each other.



Under proposed model – possible to extract most walls and corners.

Sensing Applications – Summary

- Stereo Vision – Range estimation possible, difficult correspondence problem to solve.
- Active triangulation – Correspondence problem solved, Range estimate improves with disparity, but occlusions possible.
- LADAR – Correspondence & disparity problems solved, expensive scanning system required.
- RADAR – Can produce multiple line of sight targets - expensive.
- SONAR – Cheap, due to large beam-width requires special range interpretation and is slow to use.

Part III: Combining Information – Kalman Filtering

Concepts:

A Kalman Filter finds the optimal estimates of the state of a dynamic system with measurement and prediction uncertainty.

References

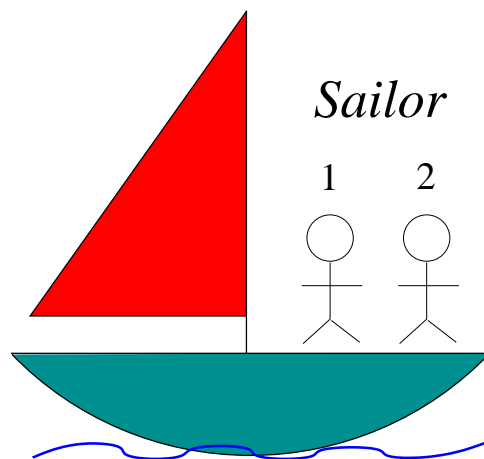
1. S.M. Bozic, “*Digital and Kalman Filtering*”, 4th Edition, Edward Arnold, 1986.
2. Peter S. Maybeck, “*Stochastic Models, Estimation and Control*”, Volume 1, Academic Press, 1979.
3. Y. Bar-Shalom, X. Rong Li, T. Kirubarajan, “*Estimation with Applications to Tracking and Navigation*”, John Wiley & Sons, 2001.

Combining Information: Introduction

Note 2 Useful Websites:

- [http : //www.cs.unc.edu/
~welch/media/pdf /maybeck_ch1.pdf](http://www.cs.unc.edu/~welch/media/pdf/maybeck_ch1.pdf)
- [http : //www.cs.unc.edu/
~tracker/media/pdf /
SIGGRAPH2001_CoursePack_08.pdf](http://www.cs.unc.edu/~tracker/media/pdf/SIGGRAPH2001_CoursePack_08.pdf)

Example: A boat at sea with true position x .



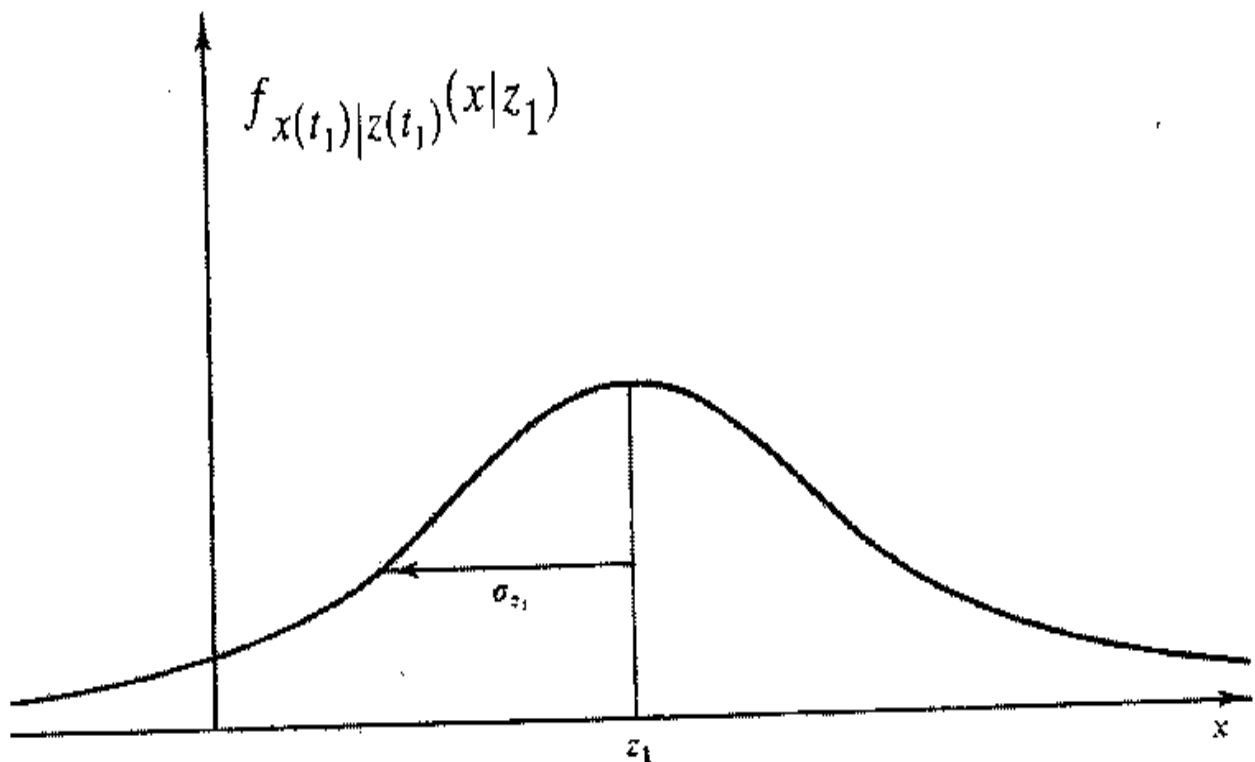
Example: Combining Information

Observer/Sailor 1 - not very sure.

He makes measurement z_1 and estimates his position to be \hat{x}_1 .

Standard deviation in his position estimate = σ_1 .

$$E[\hat{x}_1] = E[x] \quad \text{and} \quad E[(x - \hat{x}_1)^2] = \sigma_1^2$$



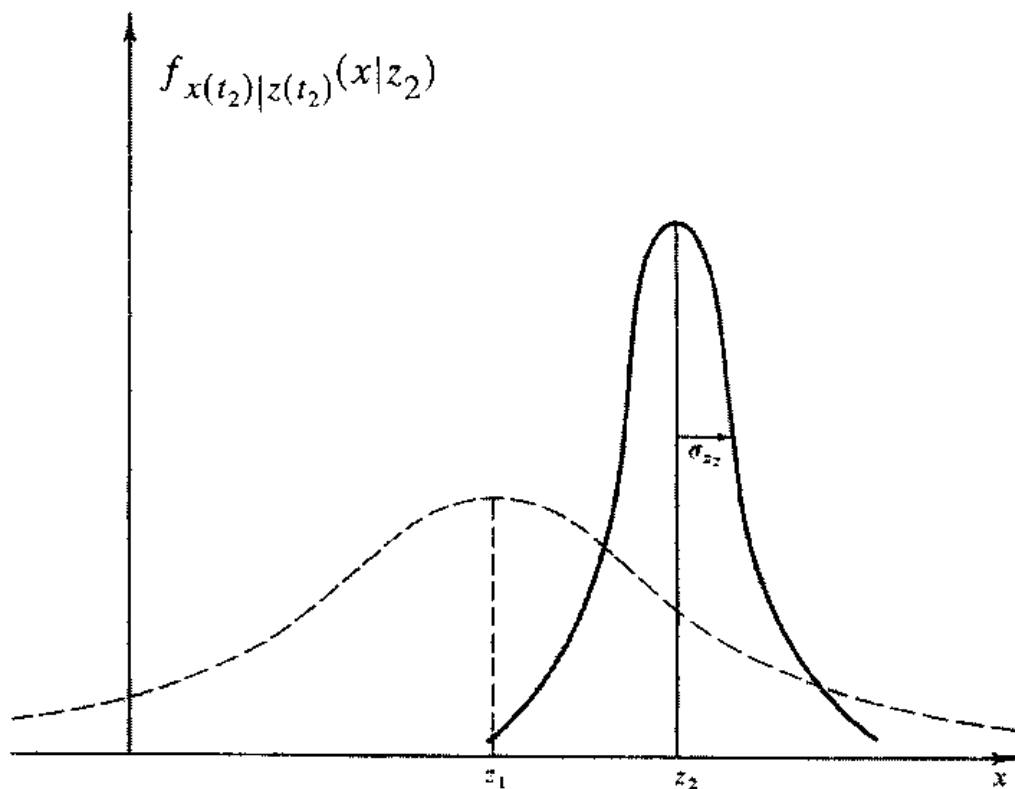
Example: Combining Information

Observer/Sailor 2 - more certain.

He makes measurement z_2 and estimates his position to be \hat{x}_2 .

Standard deviation in his position estimate = σ_2 .

$$E[\hat{x}_2] = E[x] \quad \text{and} \quad E[(x - \hat{x}_2)^2] = \sigma_2^2$$



Example: Combining Information

Both measurements and estimates of position are *independent*, or:

$$E[(x - \hat{x}_1)(x - \hat{x}_2)] = 0$$

Question: How do we combine both estimates?

Form a new estimate \hat{x} :

$$\hat{x} = (1 - W)\hat{x}_1 + W\hat{x}_2 \quad (0.28)$$

W is a weighting coefficient (to be determined).

Taking expectations:

$$\begin{aligned} E[\hat{x}] &= (1 - W)E[\hat{x}_1] + WE[\hat{x}_2] \\ &= (1 - W)E[\hat{x}] + WE[\hat{x}] = E[x] \end{aligned}$$

Note: \hat{x} is an *unbiased* estimate of x .

Example: Combining Information

General Note:

For non-random parameters:

\hat{x} is unbiased iff $E[\hat{x}] = x_0$ - the true value of the parameter.

For random variables:

\hat{x} is unbiased iff $E[\hat{x}] = E[x]$.

Aim: Find W so that the *variance of the new combined estimate is minimised*.

Subtract equation 0.28 from true value of x to give error:

$$x - \hat{x} = (1 - W)(x - \hat{x}_1) + W(x - \hat{x}_2)$$

Example: Combining Information

Variance of the new estimate is:

$$\begin{aligned}\sigma^2 &= E[(x - \hat{x})^2] \\ &= E[((1 - W)(x - \hat{x}_1) + W(x - \hat{x}_2))^2] \\ &= (1 - W)^2\sigma_1^2 + W^2\sigma_2^2\end{aligned}$$

since cross terms $E[(x - \hat{x}_1)(x - \hat{x}_2)] = 0$.

ie. measurements are independent.

$$\begin{aligned}\sigma^2 &= (1 - 2W + W^2)\sigma_1^2 + W^2\sigma_2^2 \\ &= (\sigma_1^2 + \sigma_2^2)W^2 - 2\sigma_1^2W + \sigma_1^2 \\ &= (\sigma_1^2 + \sigma_2^2) \left(W - \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} \right)^2 + \frac{\sigma_1^2\sigma_2^2}{\sigma_1^2 + \sigma_2^2}\end{aligned}$$

Without differentiating, σ^2 minimised if

$$W = \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2}$$

Example: Combining Information

Minimum variance is then

$$\sigma^2 = \frac{\sigma_1^2 \sigma_2^2}{\sigma_1^2 + \sigma_2^2}$$

Points to note:

1.

$$\frac{1}{\sigma^2} = \frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2}$$

ie. $\sigma^2 < \sigma_1^2$ and σ_2^2 .

Even poor quality information improves the overall estimate.

Combining Information: Points to Note

2. From $\hat{x} = (1 - W)\hat{x}_1 + W\hat{x}_2$

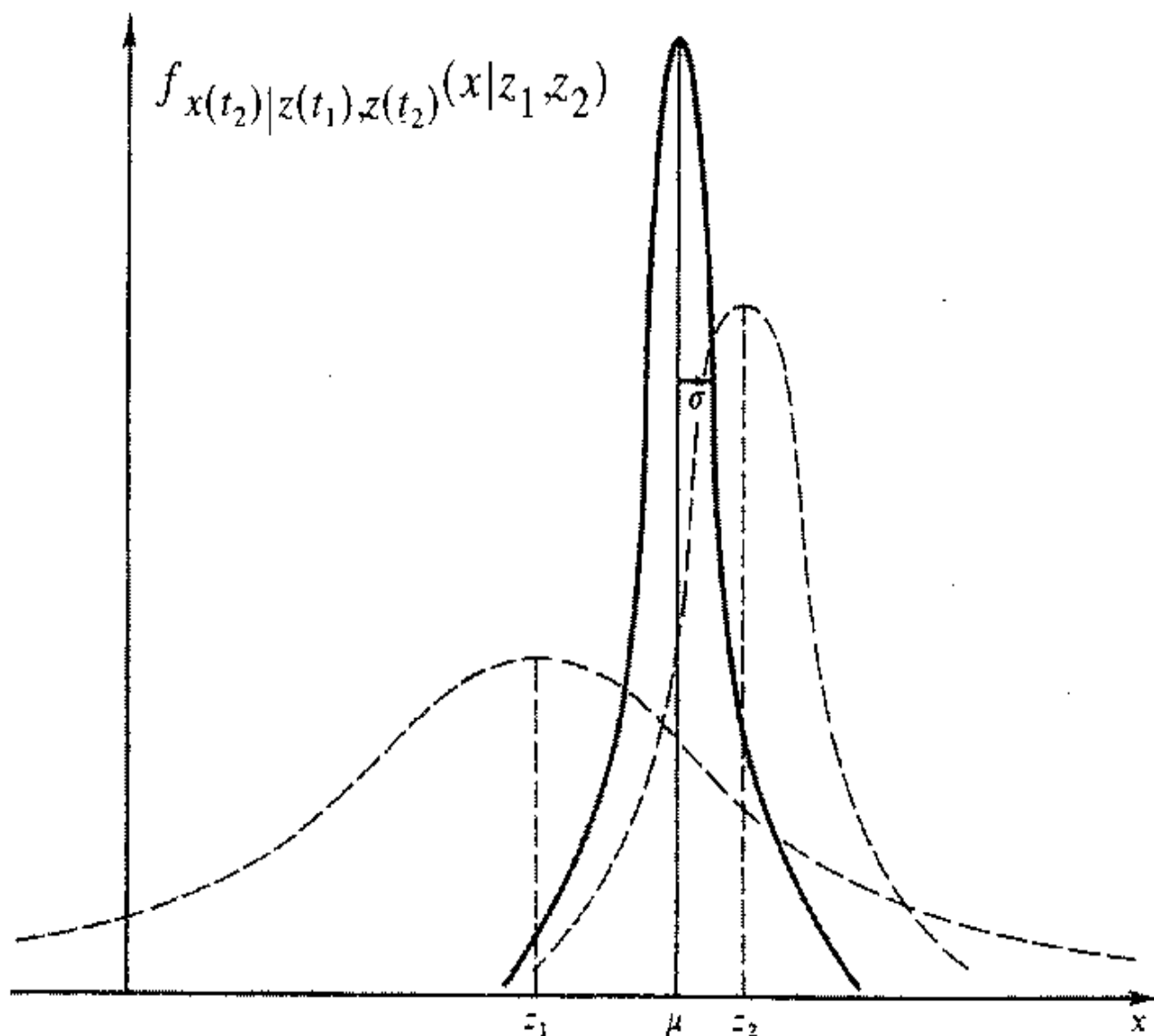
$$\hat{x} = \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2}\hat{x}_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2}\hat{x}_2$$

Mean of new PDF = Weighted average of 2 means.

(a) If $\sigma_1^2 = \sigma_2^2$ (equally “sure”),
 $\hat{x} =$, Average of \hat{x}_1 and \hat{x}_2 .

(b) If $\sigma_1^2 > \sigma_2^2$, (trust measurement 2 more),
 \hat{x}_2 is weighted more than \hat{x}_1

Combining Information: Points to Note



Combining Information: Points to Note

3.

$$\hat{x} = \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2} \hat{x}_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} \hat{x}_2$$

can be rewritten as

$$\hat{x} = \hat{x}_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} [\hat{x}_2 - \hat{x}_1]$$

$$\text{or } \hat{x} = \hat{x}_1 + K[\hat{x}_2 - \hat{x}_1]$$

....the form of a Kalman filter,

ie. **optimal estimate at time t_2**

= best prediction at time t_1

+ (weight \times correction term)

ie. “Predictor” – “Corrector” form.

Example: RADAR - $\alpha - \beta$ Tracker

RADAR = *R*Adio *D*etection and *R*anging.

Aim: Determine Range and Velocity of objects.

Time-of-Flight (TOF) of RADAR pulse = ΔT .

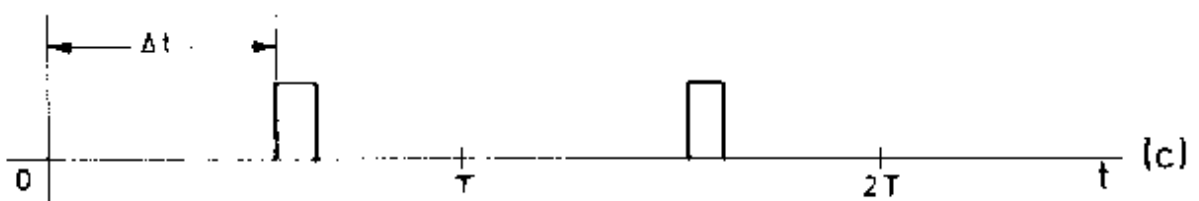
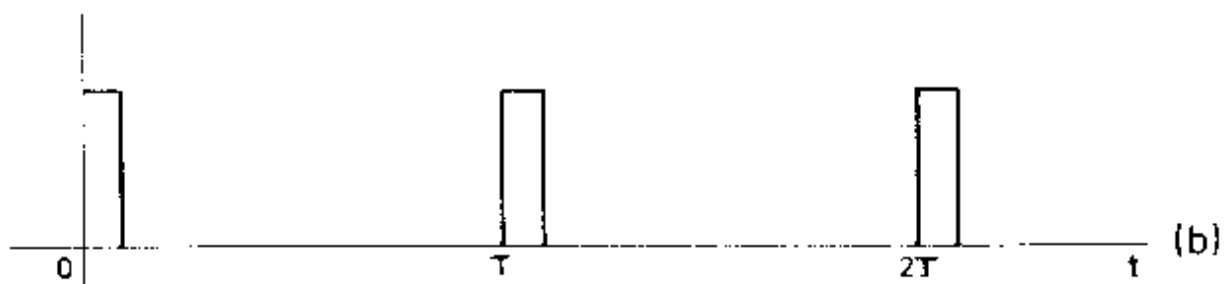
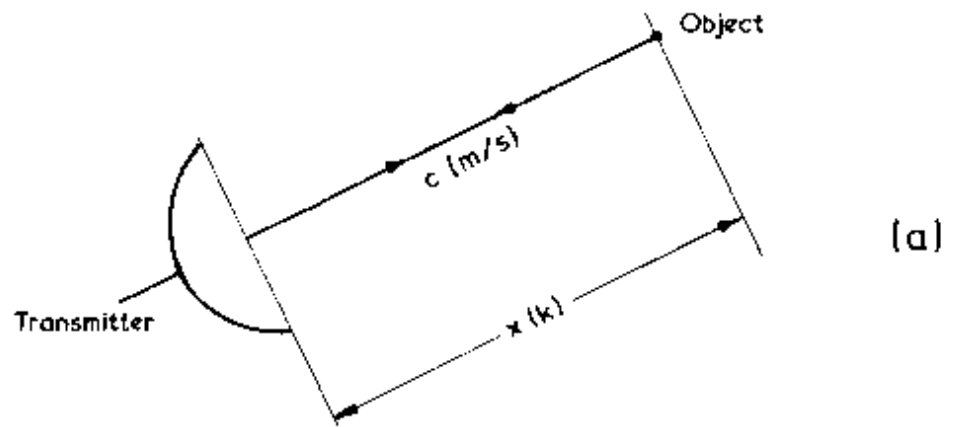
Measured TOF = $\Delta t_1 \neq \Delta T$

Therefore keep transmitting pulses every T seconds, and measure: $x(0)$, $x(1)$, $x(2)$, $x(0)$

Try to estimate *TRUE* range/Velocity.

- $x(k)$ = object's range from k th RADAR pulse return.
- $y(k)$ = estimate of object's range after processing.
- $\dot{y}(k)$ = estimate of object's velocity after processing.
- $y_p(k)$ = prediction of objects range at the k th RADAR pulse, obtained at time $(k - 1)$ ie. *before* $x(k)$ measured.

Example: RADAR - $\alpha - \beta$ Tracker



Example: RADAR - $\alpha - \beta$ Tracker

Prediction:

$$y_p(k) = y(k-1) + T\dot{y}(k-1)$$

Correction (as in previous example):

$$y(k) = y_p(k) + \alpha[x(k) - y_p(k)]$$

where $\alpha > 0$.

Velocity estimate:

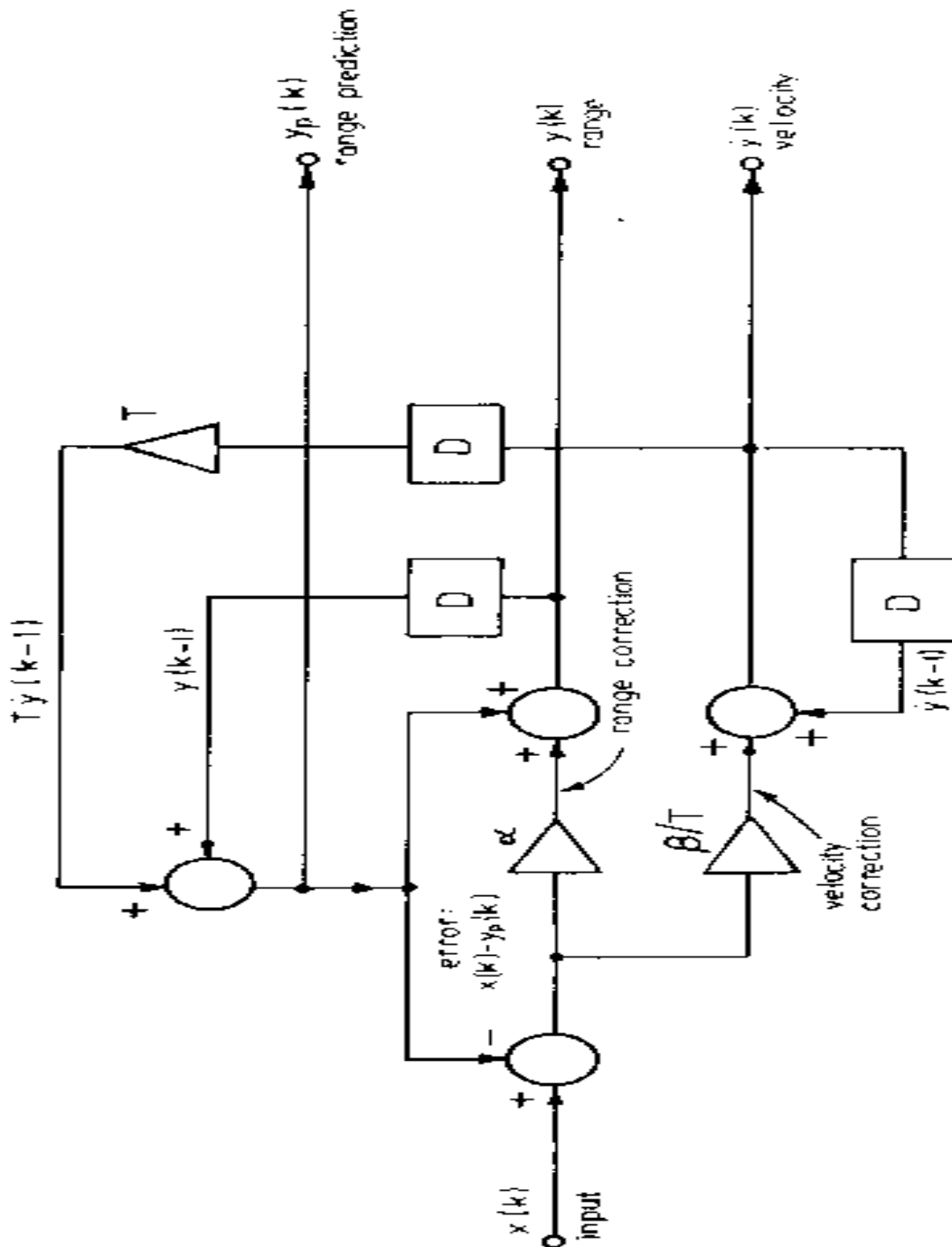
$$\dot{y}(k) = \dot{y}(k-1) + \frac{\beta}{T}[x(k) - y_p(k)]$$

where $\beta > 0$.

These 3 equations form $\alpha - \beta$ tracker.

Tune α and β for good response.

Example: RADAR - $\alpha - \beta$ Tracker



Probabilistic Data Association

- Uncertainty lies at the heart of all descriptions of sensing and data fusion processes.
- Probabilistic models provide a powerful and consistent means of describing uncertainty and lead naturally into ideas of information fusion and decision making.
- Alternative uncertainty measurement methods also described.
- A brief review of probabilistic methods.
- A focus on probabilistic and information theoretic data fusion methods.

Probabilistic Models

- Familiarity with essential probability theory is assumed.
- A probability density function (*pdf*) $P_y(\cdot)$ is defined on a random variable \mathbf{y} ,
- generally written as $P_y(\mathbf{y})$ or simply $P(\mathbf{y})$.
- The random variable may be a scalar or vector quantity, and may be either discrete or continuous in measure.
- The *pdf* is a (probabilistic) model of the quantity \mathbf{y} , observation or state.
- The *pdf* $P(\mathbf{y})$ is considered valid if:
 1. It is positive; $P(\mathbf{y}) > 0$ for all \mathbf{y} , and
 2. it sums (integrates) to a probability of 1;

$$\int_{\mathbf{y}} P(\mathbf{y}) d\mathbf{y} = 1$$

Joint Probabilistic Models

- The joint distribution $P_{xy}(\mathbf{x}, \mathbf{y})$ is defined in a similar manner.
- Integrating the *pdf* $P_{xy}(\mathbf{x}, \mathbf{y})$ over the variable \mathbf{x} gives the marginal *pdf* $P_y(\mathbf{y})$ as

$$P_y(\mathbf{y}) = \int_x P_{xy}(\mathbf{x}, \mathbf{y}) dx$$

- and similarly integrating over \mathbf{y} gives the marginal *pdf* $P_x(\mathbf{x})$.
- The joint *pdf* over n variables, $P(\mathbf{x}_1, \dots, \mathbf{x}_n)$, may also be defined with analogous properties to the joint *pdf* of 2 variables.
- The conditional *pdf* $P(\mathbf{x} | \mathbf{y})$ is defined by

$$P(\mathbf{x} | \mathbf{y}) = \frac{P(\mathbf{x}, \mathbf{y})}{P(\mathbf{y})}$$

- Has the usual properties of a *pdf* with \mathbf{x} the dependent variable given that \mathbf{y} takes on specific fixed values.
-

The Total Probability Theorem

- Chain-rule can be used to expand a joint *pdf* in terms of conditional and marginal distributions

$$P(\mathbf{x}, \mathbf{y}) = P(\mathbf{x} | \mathbf{y})P(\mathbf{y})$$

- The chain rule can be extended to any number of variables

$$P(\mathbf{x}_1, \dots, \mathbf{x}_n) = P(\mathbf{x}_1 | \mathbf{x}_2, \dots, \mathbf{x}_n) \dots \\ \dots P(\mathbf{x}_{n-1} | \mathbf{x}_n)P(\mathbf{x}_n)$$

- Expansion may be taken in any convenient order.

The Total Probability Theorem

- The total probability theorem

$$P_y(\mathbf{y}) = \int_x P_{x|y}(\mathbf{y} | \mathbf{x}) P_x(\mathbf{x}) dx$$

- The total probability in a state \mathbf{y} can be obtained by considering the ways in which \mathbf{y} can occur given that the state \mathbf{x} takes a specific value (this is encoded in $P_{x|y}(\mathbf{y} | \mathbf{x})$), weighted by the probability that each of these values of \mathbf{x} is true (encoded in $P_x(\mathbf{x})$).

Independence and Conditional Independence

- If knowledge of \mathbf{y} provides no information about \mathbf{x} then \mathbf{x} and \mathbf{y} are independent

$$P(\mathbf{x} | \mathbf{y}) = P(\mathbf{x})$$

- Or

$$P(\mathbf{x}, \mathbf{y}) = P(\mathbf{x})P(\mathbf{y})$$

- Conditional independence: Given 3 random variables \mathbf{x} , \mathbf{y} and \mathbf{z} , if knowledge of the value of \mathbf{z} makes the value of \mathbf{x} independent of the value of \mathbf{y} then

$$P(\mathbf{x} | \mathbf{y}, \mathbf{z}) = P(\mathbf{x} | \mathbf{z})$$

Eg: If \mathbf{z} indirectly contains all the information contributed by \mathbf{y} to the value of \mathbf{x} .

- Implies the intuitive result

$$P(\mathbf{x}, \mathbf{y} | \mathbf{z}) = P(\mathbf{x} | \mathbf{z})P(\mathbf{y} | \mathbf{z})$$

Independence and Conditional Independence

- Conditional independence underlies many data fusion algorithms.
- Consider the state of a system \mathbf{x} and 2 observations of this state \mathbf{z}_1 and \mathbf{z}_2 .
- It is clear that the 2 observations are *not* independent

$$P(\mathbf{z}_1, \mathbf{z}_2) \neq P(\mathbf{z}_1)P(\mathbf{z}_2)$$

as they both depend on the common state \mathbf{x} .

- However the observations are usually conditionally independent given the state

$$P(\mathbf{z}_1, \mathbf{z}_2 | \mathbf{x}) = P(\mathbf{z}_1 | \mathbf{x})P(\mathbf{z}_2 | \mathbf{x})$$

- For data fusion purposes this is a good definition of state.

Bayes Theorem

- Consider 2 random variables \mathbf{x} and \mathbf{z} on which a joint *pdf* $P(\mathbf{x}, \mathbf{z})$ is defined.
- The chain rule of conditional probabilities can be used to expand this density function in 2 ways

$$\begin{aligned}P(\mathbf{x}, \mathbf{z}) &= P(\mathbf{x} | \mathbf{z})P(\mathbf{z}) \\ &= P(\mathbf{z} | \mathbf{x})P(\mathbf{x})\end{aligned}$$

- Bayes theorem is obtained as

$$P(\mathbf{x} | \mathbf{z}) = \frac{P(\mathbf{z} | \mathbf{x})P(\mathbf{x})}{P(\mathbf{z})}$$

- Computes posterior $P(\mathbf{x} | \mathbf{z})$ given the prior $P(\mathbf{x})$ and an observation $P(\mathbf{z} | \mathbf{x})$.

Bayes Theorem – Sensor Models

- $P(\mathbf{z} \mid \mathbf{x})$ takes the role of a sensor model:
 - First building a sensor model: Fix $\mathbf{x} = x$ and then ask what *pdf* on \mathbf{z} results.
 - Then use a sensor model: Observe $\mathbf{z} = z$ and then ask what the *pdf* on \mathbf{x} is.
 - Practically $P(\mathbf{z} \mid \mathbf{x})$ is constructed as a function of both variables (or a matrix in discrete form).
 - For each fixed value of \mathbf{x} , a distribution in \mathbf{z} is defined. Therefore as \mathbf{x} varies, a family of distributions in \mathbf{z} is created.

Bayes Theorem – Example I

- Consider a continuous valued state \mathbf{x} , range to target for example.
- Let \mathbf{z} = observation of this state.
- A Gaussian observation model

$$P(\mathbf{z} \mid \mathbf{x}) = \frac{1}{\sqrt{2\pi}\sigma_z^2} \exp\left(-\frac{1}{2} \frac{(\mathbf{z} - \mathbf{x})^2}{\sigma_z^2}\right)$$

- A function of both \mathbf{z} and \mathbf{x} .
- Building model: State is fixed, $\mathbf{x} = x$, and distribution is a function of \mathbf{z} .
- Using model: Observation is made, $\mathbf{z} = z$, and distribution is a function of \mathbf{x} .

- Prior

$$P(\mathbf{x}) = \frac{1}{\sqrt{2\pi}\sigma_x^2} \exp\left(-\frac{1}{2} \frac{(\mathbf{x} - x_p)^2}{\sigma_x^2}\right)$$

Bayes Theorem – Example I

- Posterior after taking an observation

$$\begin{aligned} P(\mathbf{x} \mid \mathbf{z}) &= C \frac{1}{\sqrt{2\pi}\sigma_z^2} \exp\left(-\frac{1(\mathbf{z} - \mathbf{x})^2}{2\sigma_z^2}\right) \\ &\quad \cdot \frac{1}{\sqrt{2\pi}\sigma_x^2} \exp\left(-\frac{1(\mathbf{x} - x_p)^2}{2\sigma_x^2}\right) \\ &= \frac{1}{\sqrt{2\pi}\sigma^2} \exp\left(-\frac{1(\mathbf{x} - \bar{x})^2}{2\sigma^2}\right) \end{aligned}$$

where

$$\bar{x} = \frac{\sigma_x^2}{\sigma_x^2 + \sigma_z^2} z + \frac{\sigma_z^2}{\sigma_x^2 + \sigma_z^2} x_p,$$

and

$$\sigma^2 = \frac{\sigma_z^2 \sigma_x^2}{\sigma_z^2 + \sigma_x^2} \longrightarrow \frac{1}{\sigma^2} = \frac{1}{\sigma_z^2} + \frac{1}{\sigma_x^2}$$

- Note same result as earlier!!!
-

Bayes Theorem – Example IIa

- A single state \mathbf{x} which can take on 1 of 3 values:
 - x_1 : \mathbf{x} is a type 1 target.
 - x_2 : \mathbf{x} is a type 2 target.
 - x_3 : No visible target.

- Single sensor observes \mathbf{x} and returns 3 possible values:
 - z_1 : Observation of a type 1 target.
 - z_2 : Observation of a type 2 target.
 - z_3 : No target observed.

Bayes Theorem – Example IIa

- The sensor model is described by the likelihood matrix $P_1(\mathbf{z} \mid \mathbf{x})$:

	z_1	z_2	z_3
x_1	0.45	0.45	0.1
x_2	0.45	0.45	0.1
x_3	0.1	0.1	0.8

- Likelihood matrix is a function of both \mathbf{x} and \mathbf{z} .
- For a fixed state, it describes the probability of a particular observation being made (the rows of the matrix).
- For an observation it describes a probability over the values of the true state (the columns) and is then the Likelihood Function $\Lambda(\mathbf{x})$.

Bayes Theorem – Example IIb

- The posterior distribution of the true state \mathbf{x} after making an observation $\mathbf{z} = z_i$ is given by

$$P(\mathbf{x} | z_i) = \alpha P_1(z_i | \mathbf{x}) P(\mathbf{x})$$

- α is a normalising constant so that the sum, over \mathbf{x} , of posteriori is 1.

- Assume a non-informative prior:

$$P(\mathbf{x}) = (0.333, 0.333, 0.333)$$

- Observe $\mathbf{z} = z_1$, then posterior is

$$P(\mathbf{x} | z_1) = (0.45, 0.45, 0.1)$$

- The 1st column of the above likelihood matrix (likelihood function given z_1) has been observed.

Bayes Theorem – Example IIb

- Make this posterior the new prior and again observe $\mathbf{z} = z_1$, then

$$\begin{aligned} P(\mathbf{x} | z_1) &= \alpha P_1(z_1 | \mathbf{x}) P(\mathbf{x}) \\ &= \alpha \times (0.45, 0.45, 0.1) \\ &\quad \otimes (0.45, 0.45, 0.1) \\ &= (0.488, 0.488, 0.024) \end{aligned}$$

- Note result is to increase the probability in both type 1 and type 2 targets at the expense of the no-target hypothesis.

Data Fusion using Bayes Theorem

- The effectiveness of fusion relies on the assumption that information obtained from different sources is independent when conditioned on the true underlying state of the world.

- Clearly

$$P(\mathbf{z}_1, \dots, \mathbf{z}_n) \neq P(\mathbf{z}_1) \dots P(\mathbf{z}_n),$$

since each piece of information depends on a *common* underlying state $\mathbf{x} \in \mathcal{X}$.

- Conversely, it is generally quite reasonable to assume that the underlying state is the *only* thing in common between information sources.
- Hence, once the state has been specified, it is reasonable to assume that the information gathered is conditionally independent given this state.

Data Fusion using Bayes Theorem – Eg. I

- A second sensor which makes the same three observations as the first sensor, but whose likelihood matrix $P_2(\mathbf{z}_2 | \mathbf{x})$ is described by

$$\begin{array}{rcc} & z_1 & z_2 & z_3 \\ x_1 & 0.45 & 0.1 & 0.45 \\ x_2 & 0.1 & 0.45 & 0.45 \\ x_3 & 0.45 & 0.45 & 0.1 \end{array}$$

- Whereas the first sensor was good at detecting targets but not at distinguishing between different target types, this second sensor has poor overall detection probabilities, but good target discrimination capabilities.
- With a uniform prior, observe $\mathbf{z} = z_1$ then the posterior is (the first column of the likelihood matrix)

$$P(\mathbf{x} | z_1) = (0.45, 0.1, 0.45)$$

Data Fusion using Bayes Theorem – Eg. I

- Makes sense to combine the information from both sensors to provide good detection and good discrimination capabilities.
- Overall likelihood function for the combined system is

$$P_{12}(\mathbf{z}_1, \mathbf{z}_2 | \mathbf{x}) = P_1(\mathbf{z}_1 | \mathbf{x})P_2(\mathbf{z}_2 | \mathbf{x})$$

- Observe $\mathbf{z}_1 = z_1$ and $\mathbf{z}_2 = z_1$ and assuming a uniform prior, then posterior is

$$\begin{aligned} P(\mathbf{x} | z_1, z_1) &= \alpha P_{12}(z_1, z_1 | \mathbf{x}) \\ &= \alpha P_1(z_1 | \mathbf{x})P_2(z_1 | \mathbf{x}) \\ &= \alpha \times (0.45, 0.45, 0.1) \\ &\quad \otimes (0.45, 0.1, 0.45) \\ &= (0.6924, 0.1538, 0.1538) \end{aligned}$$

- Sensor 2 adds substantial target discrimination power. Cost – slight loss of detection performance for the same number of observations.
-

Data Fusion using Bayes Theorem – Eg II

- Repeating this calculation for each $\mathbf{z}_1, \mathbf{z}_2$ observation pair:

$$\mathbf{z}_1 = z_1$$

$\mathbf{z}_2 =$	z_1	z_2	z_3
x_1	0.6924	0.1538	0.4880
x_2	0.1538	0.6924	0.4880
x_3	0.1538	0.1538	0.0240

Data Fusion using Bayes Theorem – Eg II

$$\mathbf{z}_1 = z_2$$

$$\mathbf{z}_2 = \begin{matrix} z_1 & z_2 & z_3 \end{matrix}$$

$$x_1 \quad 0.6924 \quad 0.1538 \quad 0.4880$$

$$x_2 \quad 0.1538 \quad 0.6924 \quad 0.4880$$

$$x_3 \quad 0.1538 \quad 0.1538 \quad 0.0240$$

$$\mathbf{z}_1 = z_3$$

$$\mathbf{z}_2 = \begin{matrix} z_1 & z_2 & z_3 \end{matrix}$$

$$x_1 \quad 0.1084 \quad 0.0241 \quad 0.2647$$

$$x_2 \quad 0.0241 \quad 0.1084 \quad 0.2647$$

$$x_3 \quad 0.8675 \quad 0.8675 \quad 0.4706$$

Data Fusion with Bayes Theorem – Eg III

- The combined sensor provides substantial improvements in overall system performance.

- Eg. Observe $\mathbf{z}_1 = z_1$ and $\mathbf{z}_2 = z_1$

$$P(\mathbf{x} \mid \mathbf{z}_1, \mathbf{z}_2) = (0.692, 0.154, 0.154)$$

- Target 1 most likely – good discrimination!

- However, observe $\mathbf{z}_1 = z_1$ and $\mathbf{z}_2 = z_2$ we get

$$P(\mathbf{x} \mid \mathbf{z}_1, \mathbf{z}_2) = (0.154, 0.692, 0.154)$$

- Target type 2 has high prob. as sensor 1 good at detection, while sensor 2 good at discrimination.

- If we now observe no target with sensor 2, having detected target type 1 (or 2) with the first sensor, the posterior is (0.488, 0.488, 0.024).

- ie: There is a target (because we know sensor 1 is much better at target detection than sensor 2), but we still have no idea which target (1 or 2) it is, as sensor 2 did not make a valid detection.
-

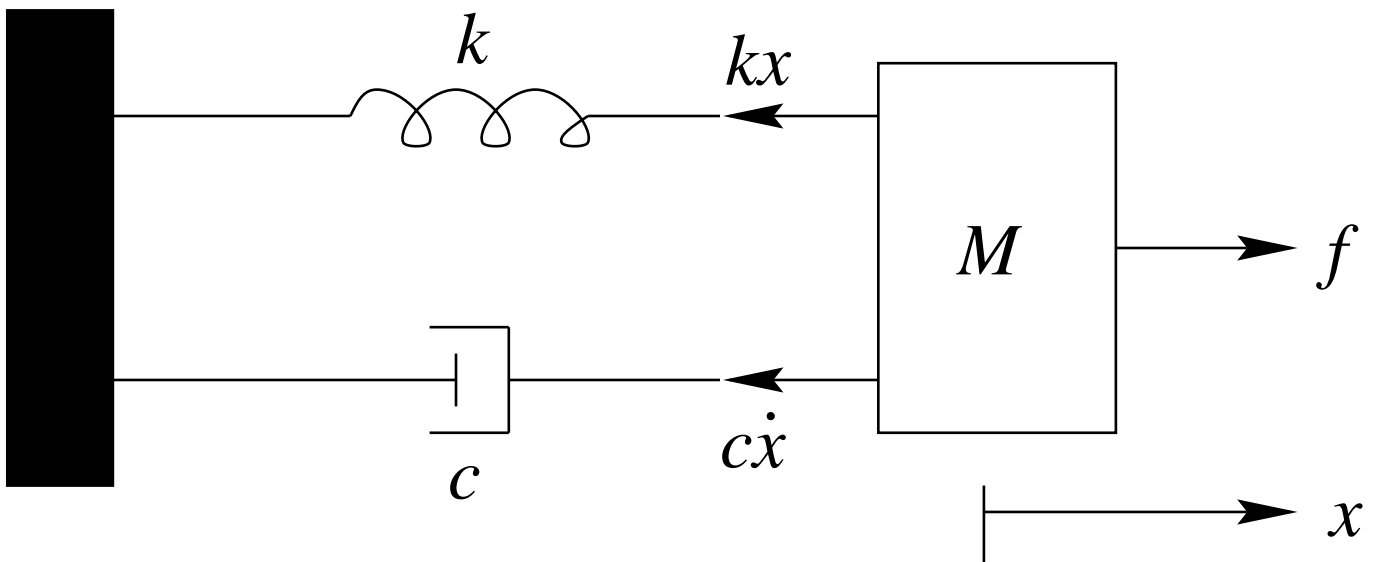
Data Fusion with Bayes Theorem – Eg IV

- Finally, if sensor 1 gets no detection, but sensor 2 detects target type 1, then posterior given by $(0.108, 0.024, 0.868)$. That is we still believe there is no target (sensor 1 is better at providing this information) and, perversely, sensor 2 confirms this.
- Practically, the joint likelihood matrix is never constructed (easy to see why) – it becomes huge!
- Rather, the likelihood matrix is constructed for each sensor and these are only combined when instantiated with an observation.

State Space – Dynamic Systems

N th order differential equation = N , 1st order differential equations.

Mass spring, damper system (Analogous: L-C-R Circuit)



Newton's law:

$$f - kx - c\dot{x} = M\ddot{x}$$

$$f = M\ddot{x} + c\dot{x} + kx \quad (0.29)$$

State Space – Dynamic Systems

2 methods of analysis:

1) Laplace “s”-domain transfer function

$$\frac{x}{f} = \frac{1}{Ms^2 + cs + k}$$

2) State space formulation:

Equation 0.29 is a 2nd order differential equation.

Define 2 states

$$\begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} = \begin{Bmatrix} \text{position} \\ \text{velocity} \end{Bmatrix} = \begin{Bmatrix} x \\ \dot{x} \end{Bmatrix}$$

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{f - kx_1 - cx_2}{M} \end{aligned}$$

State Space – Dynamic Systems

which can be written as:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{M} & -\frac{c}{M} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{M} \end{bmatrix} f$$

ie. $\dot{\mathbf{x}} = \mathbf{ax} + \mathbf{bu}$

where \mathbf{u} is the input control vector.

In discrete time - similar notation:

$$\mathbf{x}(k + 1) = \mathbf{Ax}(k) + \mathbf{Bu}(k)$$

ie.

$$\begin{array}{l} \text{State at} \\ \text{time } k + 1 \end{array} = \begin{array}{l} \text{Function of} \\ \text{state's pre-} \\ \text{vious value} \end{array} + \begin{array}{l} \text{Control} \\ \text{input} \end{array}$$

RADAR Tracking Revisited

Aim: To derive system and observation (measurement) models for RADAR tracking problems.

System Model:

Vehicle range = $R + \rho(k)$ at time k .

T seconds later – range = $R + \rho(k + 1)$ at time $k + 1$.

Vehicle's radial velocity = $\dot{\rho}(k)$.

Assuming small T

$$R + \rho(k + 1) = R + \rho(k) + T\dot{\rho}(k)$$

NOTE: Same as range prediction from $\alpha - \beta$ tracker.

$$\dot{\rho}(k + 1) = \dot{\rho}(k) + u(k)$$

Acceleration on average = 0. i.e. $u(k) \approx N(0, \sigma_u^2)$.

RADAR Tracking Revisited

Acceleration is uncorrelated between intervals:

$$\text{i.e. } E[u(k+1)u(k)] = 0$$

Acceleration has known variance (wind gusts, erratic engine thrusts):

$$E[u^2(k)] = \sigma_u^2.$$

Define:

$$x_1 = \rho(k) \quad \text{and}$$

$$x_2 = \dot{\rho}(k)$$

$$x_1(k+1) = x_1(k) + Tx_2(k)$$

$$x_2(k+1) = x_2(k) + u(k)$$

Matrix form:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0 \\ u(k) \end{bmatrix}$$

i.e.

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{v}(k)$$

RADAR Tracking – Observation Model

In general the \mathbf{A} matrix coefficients can also be time varying $\rightarrow \mathbf{A}(k)$.

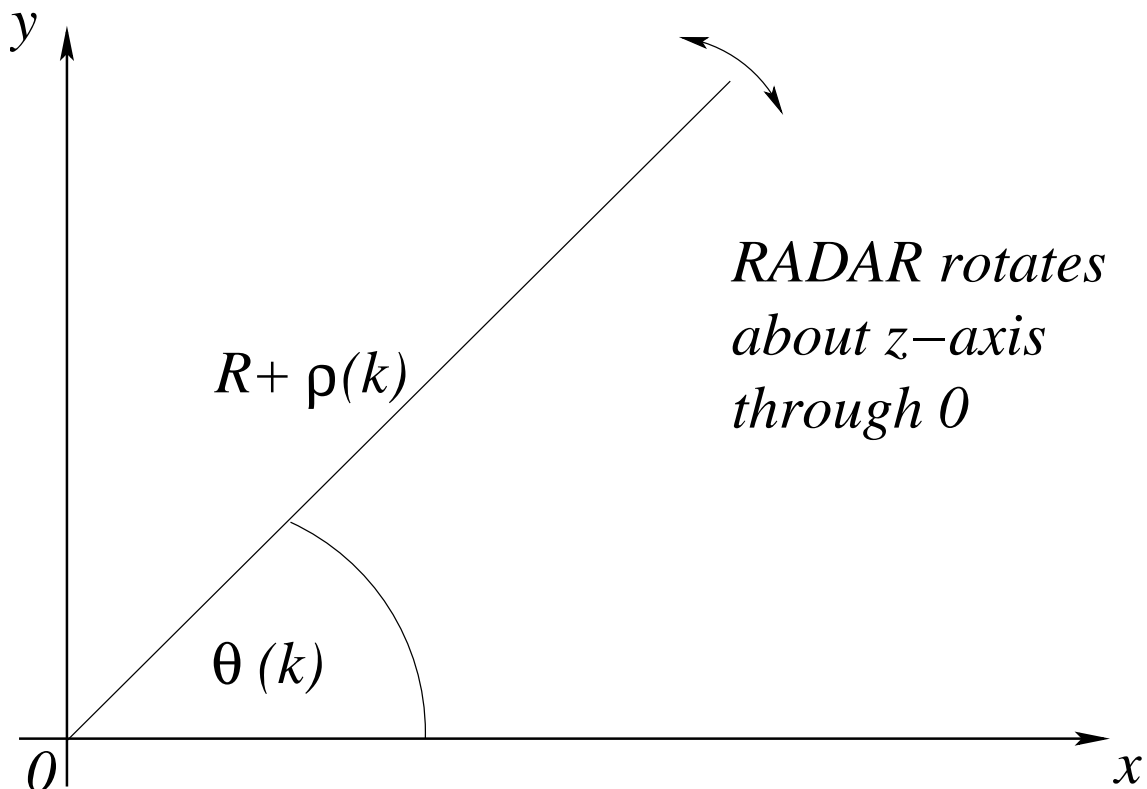
Suppose we now want to estimate:

Range $\rho(k)$,

Radial velocity $\dot{\rho}(k)$,

Bearing $\theta(k)$,

angular velocity $\dot{\theta}(k)$.



RADAR Tracking – Observation Model

$$\text{State vector is: } \mathbf{x}(k) = \begin{bmatrix} \rho(k) \\ \dot{\rho}(k) \\ \theta(k) \\ \dot{\theta}(k) \end{bmatrix}$$

Only measurements of range $y_1(k)$ and bearing $y_2(k)$ are made + noise $w_1(k)$ and $w_2(k)$

$$\begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \rho(k) \\ \dot{\rho}(k) \\ \theta(k) \\ \dot{\theta}(k) \end{bmatrix} + \begin{bmatrix} w_1(k) \\ w_2(k) \end{bmatrix}$$

which is of the form

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{w}(k)$$

where $\mathbf{w}(k)$ is the measurement noise vector at time k .

General Kalman Filter Algorithm

General 1st order dynamic *system model*:

$$\mathbf{x}(k + 1) = \mathbf{F}(k)\mathbf{x}(k) + \mathbf{G}(k)\mathbf{u}(k) + \mathbf{v}(k)$$

Observation (measurement) equation:

$$\mathbf{z}(k) = \mathbf{C}(k)\mathbf{x}(k) + \mathbf{w}(k)$$

Aim: starting with $\hat{\mathbf{x}}(j | j)$ and $\mathbf{P}(j | j)$ find optimal estimate $\hat{\mathbf{x}}(k + 1 | k + 1)$ and $\mathbf{P}(k + 1 | k + 1)$.

General Kalman Filter Algorithm

Kalman filter algorithm:

1. *Filter Initialisation.*

Find $\hat{\mathbf{x}}(j | j)$ and $\mathbf{P}(j | j)$ for some value of j (example later!).

2. *Filter Prediction.*

$$\hat{\mathbf{x}}(k + 1 | k) = \mathbf{F}(k)\hat{\mathbf{x}}(k | k) + \mathbf{G}(k)\mathbf{u}(k)$$

subtract prediction from true state:

$$\begin{aligned}\tilde{\mathbf{x}}(k + 1 | k) &= \mathbf{x}(k + 1) - \hat{\mathbf{x}}(k + 1 | k) \\ &= \mathbf{F}(k)\tilde{\mathbf{x}}(k | k) + \mathbf{v}(k)\end{aligned}$$

$$\begin{aligned}\mathbf{P}(k + 1 | k) &= E[\tilde{\mathbf{x}}(k + 1 | k)\tilde{\mathbf{x}}^T(k + 1 | k)] \\ &= E[(\mathbf{F}(k)\tilde{\mathbf{x}}(k | k) + \mathbf{v}(k))(\mathbf{F}(k)\tilde{\mathbf{x}}(k | k) + \mathbf{v}(k))^T] \\ &= \mathbf{F}(k)E[\tilde{\mathbf{x}}(k | k)\tilde{\mathbf{x}}^T(k | k)]\mathbf{F}^T(k) + E[\mathbf{v}(k)\mathbf{v}^T(k)] \\ &\quad \text{since } \mathbf{v}(k) \text{ and } \tilde{\mathbf{x}}(k | k) \text{ are uncorrelated.} \\ &= \mathbf{F}(k)\mathbf{P}(k | k)\mathbf{F}^T(k) + \mathbf{Q}(k)\end{aligned}$$

where $\mathbf{Q}(k) = E[\mathbf{v}(k)\mathbf{v}^T(k)]$ is the system covariance matrix.

General Kalman Filter Algorithm

Note cancellation of control input $\mathbf{u}(k)$ – it has no effect on prediction accuracy.

3. *Predicted Observation*

$$\hat{\mathbf{z}}(k + 1 | k) = \mathbf{C}(k + 1)\hat{\mathbf{x}}(k + 1 | k)$$

4. *Innovation*. True - predicted observations.

$$\begin{aligned}\tilde{\mathbf{z}}(k + 1 | k) &= \mathbf{z}(k + 1) - \hat{\mathbf{z}}(k + 1 | k) \\ &= \mathbf{C}(k + 1)\tilde{\mathbf{x}}(k + 1 | k) + \mathbf{w}(k + 1)\end{aligned}$$

5. *Measurement Prediction Covariance (Variance of the Innovation)*.

$$\begin{aligned}\mathbf{S}(k + 1) &= E[\tilde{\mathbf{z}}(k + 1 | k)\tilde{\mathbf{z}}^T(k + 1 | k)] \\ &= E[(\mathbf{C}(k + 1)\tilde{\mathbf{x}}(k + 1 | k) + \mathbf{w}(k + 1)) \\ &\quad (\mathbf{C}(k + 1)\tilde{\mathbf{x}}(k + 1 | k) + \mathbf{w}(k + 1))^T] \\ &= \mathbf{C}(k + 1)E[\tilde{\mathbf{x}}(k + 1 | k)\tilde{\mathbf{x}}^T(k + 1 | k)]\mathbf{C}^T(k + 1) \\ &\quad + E[\mathbf{w}(k + 1)\mathbf{w}^T(k + 1)] \\ &= \mathbf{C}(k + 1)\mathbf{P}(k + 1 | k)\mathbf{C}^T(k + 1) + \mathbf{R}(k + 1)\end{aligned}$$

$\mathbf{R}(k + 1)$ = measurement covariance matrix.

General Kalman Filter Algorithm

6. *Kalman Gain*. Weighting Factor. Weighting between prediction & innovation. \rightarrow Derived from *orthogonal* equations.

Definition: $\mathbf{W}(k + 1)$ = Covariance between state and measurement \times inverse of measurement prediction covariance $\mathbf{S}^{-1}(k + 1)$.

Note similarity to $\frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2}$

Covariance between state & measurement:

$$\begin{aligned}\mathbf{P}_{\mathbf{xz}} &= E[\tilde{\mathbf{x}}(k + 1 | k)\tilde{\mathbf{z}}^T(k + 1 | k)] \\ &= E[\tilde{\mathbf{x}}(k + 1 | k)(\mathbf{C}(k + 1)\tilde{\mathbf{x}}(k + 1 | k) + \mathbf{w}(k + 1))^T] \\ &= E[\tilde{\mathbf{x}}(k + 1 | k)\tilde{\mathbf{x}}^T(k + 1 | k)]\mathbf{C}^T(k + 1) \\ &= \mathbf{P}(k + 1 | k)\mathbf{C}^T(k + 1)\end{aligned}$$

The *Kalman gain* is then

$$\mathbf{W}(k + 1) = \mathbf{P}(k + 1 | k)\mathbf{C}^T(k + 1)\mathbf{S}^{-1}(k + 1)$$

General Kalman Filter Algorithm

7. *State vector update.* just as in equation 3

$$\hat{x} = \hat{x}_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2}[\hat{x}_2 - \hat{x}_1]$$

the Kalman filter state vector update equation is

$$\hat{\mathbf{x}}(k+1 | k+1) = \hat{\mathbf{x}}(k+1 | k) + \mathbf{W}(k+1)[\mathbf{z}(k+1) - \mathbf{C}(k+1)\hat{\mathbf{x}}(k+1 | k)]$$

8. *Error covariance matrix update*

$$\mathbf{P}(k+1 | k+1) = \mathbf{P}(k+1 | k) - \mathbf{W}(k+1)\mathbf{S}(k+1)\mathbf{W}^T(k+1)$$

9. *Return to step 2* – Calculate $\hat{\mathbf{x}}(k+2 | k+1)$ and $\mathbf{P}(k+2 | k+1)$ to continue the cycle.

Proof of Kalman Gain

In general $\mathbf{W}(k+1) = \mathbf{P}_{\mathbf{xz}}\mathbf{S}^{-1}(k+1)$ results from solving orthogonality equations.

Consider simpler scalar case – recursive estimator.

$$\hat{x}(k+1 | k+1) = a(k+1)\hat{x}(k+1 | k) + b(k+1)y(k+1)$$

Aim: Find best estimate which minimises:

$$p(k+1 | k+1) = E[\tilde{x}(k+1 | k+1)]$$

where $\tilde{x}(k+1 | k+1) = \hat{x}(k+1 | k+1) - x(k+1)$.

$$p(k+1 | k+1) = E[(a(k+1)\hat{x}(k+1 | k) + b(k+1)y(k+1) - x(k+1))^2]$$

To minimise differentiate w.r.t. $a(k+1)$ and $b(k+1)$ and set to 0.

Proof of Kalman Gain

$$\frac{\partial p(k+1 | k+1)}{\partial a(k+1)} = 2E[a(k+1)\hat{x}(k+1 | k) + b(k+1)y(k+1) - x(k+1)]\hat{x}(k+1 | k) = 0$$

and

$$\frac{\partial p(k+1 | k+1)}{\partial b(k+1)} = 2E[a(k+1)\hat{x}(k+1 | k) + b(k+1)y(k+1) - x(k+1)]y(k+1) = 0$$

Solution in “Digital and Kalman Filtering” by S.M. Bozic - Not examinable.

or alternatively

$$\begin{aligned}\mathbf{W}(k+1) &= \mathbf{P}_{\mathbf{xz}}\mathbf{S}^{-1}(k+1) \\ &= \mathbf{P}(k+1 | k)\mathbf{C}^T(k+1)\mathbf{S}^{-1}(k+1)\end{aligned}$$

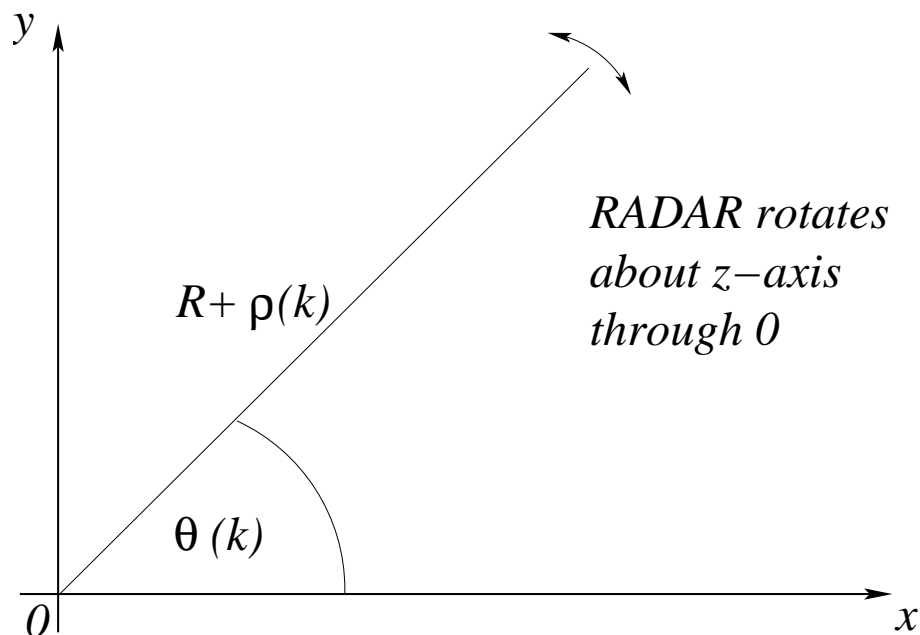
Example – RADAR Tracking

Continue a RADAR tracking example.

Aim: Show how to initialise filter and then run 1 cycle.

Previously had two states $x_1(k) = \rho(k) =$ range and $x_2(k) = \dot{\rho}(k) =$ radial velocity.

Introduce bearing $x_3(k) = \theta(k)$ and bearing rate $x_4(k) = \dot{\theta}(k)$.



Example – RADAR Tracking

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \\ x_4(k+1) \end{bmatrix} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \end{bmatrix} + \begin{bmatrix} 0 \\ u_1(k) \\ 0 \\ u_2(k) \end{bmatrix}$$

i.e.

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{v}(k)$$

The noise terms $u_1(k)$ and $u_2(k)$ represent unmodelled changes in *radial velocity* and *bearing rate* respectively – (erratic engine thrusts, wind gusts).

Observation Model: Measure Range + Noise, Bearing + Noise.

$$\begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \end{bmatrix} + \begin{bmatrix} w_1(k) \\ w_2(k) \end{bmatrix}$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{w}(k)$$

Example – RADAR Tracking

Noise *covariance* matrix for measurement model:

$$\mathbf{R}(k) = E[\mathbf{w}(k)\mathbf{w}^T(k)] = \begin{bmatrix} \sigma_\rho^2 & 0 \\ 0 & \sigma_\theta^2 \end{bmatrix}$$

and the system covariance matrix is given by

$$\mathbf{Q}(k) = E[\mathbf{v}(k)\mathbf{v}^T(k)] = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \sigma_1^2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_2^2 \end{bmatrix}$$

where σ_1^2 and σ_2^2 are the variances in the radial and angular bearing rates respectively.

RADAR Filter Initialisation

Aim: Find $\hat{\mathbf{x}}(2 | 2)$ and $\mathbf{P}(2 | 2)$.

Take two measurements of range, $y_1(k)$ and two measurements of bearing $y_2(k)$ at times $k = 1$ and $k = 2$ (i.e. four measurements in all).

$$y_1(2) = x_1(2) + w_1(2) \rightarrow x_1(2) = y_1(2) - w_1(2)$$

and

$$x_1(2) - \hat{x}_1(2 | 2) = y_1(2) - w_1(2) - y_1(2) = -w_1(2)$$

$$\text{since } \hat{x}_1(2 | 2) = E[x_1(2)] = y_1(2).$$

RADAR Filter Initialisation

To estimate $\hat{\mathbf{x}}(2 | 2)$ use system model.

$$x_1(2) = x_1(1) + T x_2(1)$$

$$\text{so that: } x_2(1) = \frac{x_1(2) - x_1(1)}{T}$$

From the second equation within system model:

$$x_2(2) = x_2(1) + u_1(1) = \frac{x_1(2) - x_1(1)}{T} + u_1(1) \quad (0.30)$$

but from the observation equation:

$$x_1(2) = y_1(2) - w_1(2)$$

$$x_1(1) = y_1(1) - w_1(1)$$

substituting above equations into equation 0.30 gives

$$x_2(2) = \frac{y_1(2) - w_1(2) - (y_1(1) - w_1(1))}{T} + u_1(1)$$

$$\text{so that: } \hat{x}_2(2 | 2) = \frac{y_1(2) - y_1(1)}{T}$$

RADAR Filter Initialisation

and

$$x_2(2) - \hat{x}_2(2 | 2) = +u_1(1) - \frac{w_1(2)}{T} + \frac{w_1(1)}{T}$$

so that the full initialisation is

$$\hat{\mathbf{x}}(2 | 2) = \begin{bmatrix} \hat{x}_1(2 | 2) = y_1(2) \\ \hat{x}_2(2 | 2) = \frac{y_1(2) - y_1(1)}{T} \\ \hat{x}_3(2 | 2) = y_2(2) \\ \hat{x}_4(2 | 2) = \frac{y_2(2) - y_2(1)}{T} \end{bmatrix}$$

$$\mathbf{P}(2 | 2) = E[(\mathbf{x}(2) - \hat{\mathbf{x}}(2 | 2))(\mathbf{x}(2) - \hat{\mathbf{x}}(2 | 2))^T]$$

Use matrix:

$$\mathbf{x}(2) - \hat{\mathbf{x}}(2 | 2) = \begin{bmatrix} -w_1(2) \\ u_1(1) - \frac{w_1(2)}{T} + \frac{w_1(1)}{T} \\ -w_2(2) \\ u_2(1) - \frac{w_2(2)}{T} + \frac{w_2(1)}{T} \end{bmatrix}$$

Example – RADAR Tracking

Now ready to step through Kalman cycle:

1. $\hat{\mathbf{x}}(3 | 2) = \mathbf{A}\hat{\mathbf{x}}(2 | 2)$

2. $\mathbf{P}(3 | 2) = \mathbf{A}\mathbf{P}(2 | 2)\mathbf{A}^T + \mathbf{Q}(2)$

3. $\mathbf{S}(3) = \mathbf{C}\mathbf{P}(3 | 2)\mathbf{C}^T + \mathbf{R}(3)$

4. Filter gain $\mathbf{W}(3) = \mathbf{P}(3 | 2)\mathbf{C}^T\mathbf{S}^{-1}(3)$

5. State vector update $\hat{\mathbf{x}}(3 | 3) = \hat{\mathbf{x}}(3 | 2) + \mathbf{W}(3)[\mathbf{y}(3) - \mathbf{C}\hat{\mathbf{x}}(3 | 2)]$

6. Error covariance matrix update $\mathbf{P}(3 | 3) = \mathbf{P}(3 | 2) - \mathbf{W}(3)\mathbf{S}(3)\mathbf{W}^T(3)$

7. go back to 1.

Kinematic (Polynomial) Models

Simple Constant Velocity Target:

Const. vel., therefore acceleration assumed zero.

$$\ddot{x}(t) = 0$$

In practice, $\ddot{x}(t)$ may undergo random changes
→ model as zero mean Gaussian noise source:

$$\ddot{x}(t) = \tilde{v}(t)$$

Define $\mathbf{x}(k) = [x_1(k), x_2(k)]^T = [\text{position}, \text{velocity}]^T$.
For assumed constant velocity (remember old equations $s = s_0 + ut!$)

$$\begin{aligned}x_1(k+1) &= x_1(k) + x_2(k)T \\x_2(k+1) &= x_2(k) + v(k)\end{aligned}$$

i.e.

$$\mathbf{x}(k+1) = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \mathbf{v}(k)$$

Kinematic (Polynomial) Models

Simple Constant Acceleration Target

If a target moves with constant acceleration

$$\frac{d^3 x(t)}{dt^3} = 0$$

In practice, $\frac{d^3 x(t)}{dt^3}$ may undergo random changes
→ model as zero mean Gaussian noise source:

$$\frac{d^3 x(t)}{dt^3} = \tilde{v}(t)$$

Define $\mathbf{x}(k) = [x_1(k), x_2(k), x_3(k)]^T =$
[position, velocity, acceleration]^T.

For assumed constant acceleration (remember the old equations $s = s_0 + ut + \frac{1}{2}at^2!$)

$$x_1(k+1) = x_1(k) + x_2(k)T + x_3(k)\frac{1}{2}T^2$$

$$x_2(k+1) = x_2(k) + x_3(k)T$$

$$x_3(k+1) = x_3(k) + v(k)$$

Kinematic (Polynomial) Models

i.e.

$$\mathbf{x}(k + 1) = \begin{bmatrix} 1 & T & \frac{1}{2}T^2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \mathbf{v}(k)$$

Kinematic (Polynomial) Models

More “Correct” Kinematic Models:

Discrete White Noise Acceleration Model

If $v(k)$ is the *constant acceleration* during k th sampling period (of length T) then:

Velocity increment = $v(k)T$.

Displacement increment = $\frac{v(k)T^2}{2}$.

which results in a more “correct” const. accel. model:

$$\mathbf{x}(k+1) = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} \frac{1}{2}T^2 \\ T \end{bmatrix} v(k)$$

which is of the form:

$$\mathbf{x}(k+1) = \mathbf{F}\mathbf{x}(k) + \mathbf{\Gamma}v(k)$$

Kinematic (Polynomial) Models

Note that covariance of process noise is:

$$\mathbf{Q} = E[\mathbf{\Gamma}v(k)v(k)\mathbf{\Gamma}^T] = \mathbf{\Gamma}\sigma_v^2\mathbf{\Gamma}^T = \begin{bmatrix} \frac{1}{4}T^4 & \frac{1}{2}T^3 \\ \frac{1}{2}T^3 & T^2 \end{bmatrix} \sigma_v^2$$

Physical dimensions of v and σ_v is [length]/[time]², i.e. that of acceleration.

Example: Falling Body

Deriving correct system models:

Falling body in constant gravitational field.

Model: $\ddot{z} = -g, \quad t \geq 0.$

Let position $z = x_1$ and velocity $\dot{z} = x_2.$

$$\text{Define: } \mathbf{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$$

State space form:

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ -g \end{bmatrix}$$

which is of the form:

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

Solution to auxiliary equation (free response) is:

$$\mathbf{x}(t) = e^{\mathbf{a}t} \mathbf{x}(0)$$

Example: Falling Body

For small t ,

$$e^{\mathbf{a}t} \approx \mathbf{I} + \mathbf{a}t + \frac{(\mathbf{a}t)^2}{2!} + \frac{(\mathbf{a}t)^3}{3!} + \dots$$

so that free response solution is:

$$\mathbf{x}(t) \approx [\mathbf{I} + \mathbf{a}t]\mathbf{x}(0)$$

Hence, full (approximate) forced response solution is:

$$\mathbf{x}(t) = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix} \mathbf{x}(0) + \int_0^t \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ -g \end{bmatrix} dt$$

$$\mathbf{x}(t) = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix} \mathbf{x}(0) + \int_0^t \begin{bmatrix} t \\ 1 \end{bmatrix} (-g) dt$$

$$\mathbf{x}(t) = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix} \mathbf{x}(0) + \begin{bmatrix} \frac{t^2}{2} \\ t \end{bmatrix} (-g)$$

Example: Falling Body

For small sampling time $t = T$, an update equation results:

$$\mathbf{x}(k+1) = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \mathbf{x}(k) + \begin{bmatrix} \frac{T^2}{2} \\ T \end{bmatrix} (-g)$$

or **System Model**:

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}u$$

Able to observe position x_1 at time k with a sensor.

$$y(k) = \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{x}(k) + v(k)$$

or **Observation Model**:

$$y(k) = \mathbf{C}\mathbf{x}(k) + v(k)$$

Assume **initialisation** given:

$$\hat{\mathbf{x}}(0|0) = \begin{bmatrix} 95 \\ 1 \end{bmatrix} \quad \text{and} \quad \mathbf{P}(0|0) = \begin{bmatrix} 10 & 0 \\ 0 & 1 \end{bmatrix}$$

Example: Falling Body

Aim: Run Kalman Filter with sample time $T = 1$ s and g normalised to be 1 and $\sigma_v^2 = R(k) = 1$.

1. Prediction $\hat{\mathbf{x}}(1 | 0) = \mathbf{A}\hat{\mathbf{x}}(0 | 0) + \mathbf{B}u$

$$\hat{\mathbf{x}}(1 | 0) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 95 \\ 1 \end{bmatrix} + \begin{bmatrix} -0.5 \\ -1 \end{bmatrix} = \begin{bmatrix} 95.5 \\ 0 \end{bmatrix}$$

2. Predicted Covariance $\mathbf{P}(1 | 0) = \mathbf{A}\mathbf{P}(0 | 0)\mathbf{A}^T$

$$\mathbf{P}(1 | 0) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 10 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 11 & 1 \\ 1 & 1 \end{bmatrix}$$

3. $\mathbf{S}(1) = \mathbf{C}\mathbf{P}(1 | 0)\mathbf{C}^T + \mathbf{R}(1)$

$$\mathbf{S}(1) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 11 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 1 = 12$$

Example: Falling Body

4. Filter gain $\mathbf{W}(1) = \mathbf{P}(1 | 0)\mathbf{C}^T\mathbf{S}^{-1}(1)$

$$\mathbf{W}(1) = \begin{bmatrix} 11 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \frac{1}{12} = \begin{bmatrix} 11/12 \\ 1/12 \end{bmatrix}$$

5. State vector update $\hat{\mathbf{x}}(1 | 1) = \hat{\mathbf{x}}(1 | 0) + \mathbf{W}(1)[\mathbf{y}(1) - \mathbf{C}\hat{\mathbf{x}}(1 | 0)]$

$$\begin{aligned} \mathbf{x}(1 | 1) &= \begin{bmatrix} 95.5 \\ 0 \end{bmatrix} + \begin{bmatrix} 11/12 \\ 1/12 \end{bmatrix} \\ &\times \left\{ 100 - \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 95.5 \\ 0 \end{bmatrix} \right\} = \begin{bmatrix} 99.6 \\ 0.37 \end{bmatrix} \end{aligned}$$

Example: Falling Body

6. Error covariance matrix update $\mathbf{P}(1 \mid 1) = \mathbf{P}(1 \mid 0) - \mathbf{W}(1)\mathbf{S}(1)\mathbf{W}^T(1)$

$$\begin{aligned} \mathbf{P}(1 \mid 1) &= \begin{bmatrix} 11 & 1 \\ 1 & 1 \end{bmatrix} - \begin{bmatrix} 11/12 \\ 1/12 \end{bmatrix} 12 \begin{bmatrix} 11 & 1 \\ 12 & 12 \end{bmatrix} \\ &= \begin{bmatrix} 11/12 & 1/12 \\ 1/12 & 11/12 \end{bmatrix} \end{aligned}$$

7. go back to 1, find $\hat{\mathbf{x}}(2 \mid 1)$ – Try yourself!

6 iterations of Kalman filter.

	<i>True values</i>		<i>Position Observations</i>	<i>Estimates</i>		<i>Errors in Estimates</i>	
	<i>Position</i>	<i>Velocity</i>		<i>Position</i>	<i>Velocity</i>	<i>Position</i>	<i>Velocity</i>
$t = kT$	$x_1(t)$	$x_2(t)$	$y(k)$	$\hat{x}_1(k)$	$\hat{x}_2(k)$	$p_{11}(k)$	$p_{22}(k)$
0	100.0	0		95.0	1.0	10.0	1.0
1	99.5	-1.0	100.0	99.63	0.38	0.92	0.92
2	98.0	-2.0	97.9	98.43	-1.16	0.67	0.58
3	95.5	-3.0	94.4	95.21	-2.91	0.66	0.30
4	92.0	-4.0	92.7	92.35	-3.70	0.61	0.15
5	87.5	-5.0	87.3	87.68	-4.84	0.55	0.08
6	82.0	-6.0	82.1	82.22	-5.88	0.50	0.05

Interpretation of Results

Note in last example we started with:

$$\hat{\mathbf{x}}(0 | 0) = \begin{bmatrix} 95 \\ 1 \end{bmatrix} \quad \text{and} \quad \mathbf{P}(0 | 0) = \begin{bmatrix} 10 & 0 \\ 0 & 1 \end{bmatrix}$$

and after 1 cycle we obtained:

$$\hat{\mathbf{x}}(1 | 1) = \begin{bmatrix} 99.6 \\ 0.37 \end{bmatrix} \quad \text{and} \quad \mathbf{P}(1 | 1) = \begin{bmatrix} 11/12 & 1/12 \\ 1/12 & 11/12 \end{bmatrix}$$

How do we interpret these results?

In general:

$$\hat{\mathbf{x}}(k | k) = \begin{bmatrix} \text{pos.} \\ \text{vel.} \end{bmatrix} \quad \text{and} \quad \mathbf{P}(k | k) = \begin{bmatrix} \sigma_p^2 & \sigma_p \sigma_v \\ \sigma_v \sigma_p & \sigma_v^2 \end{bmatrix}$$

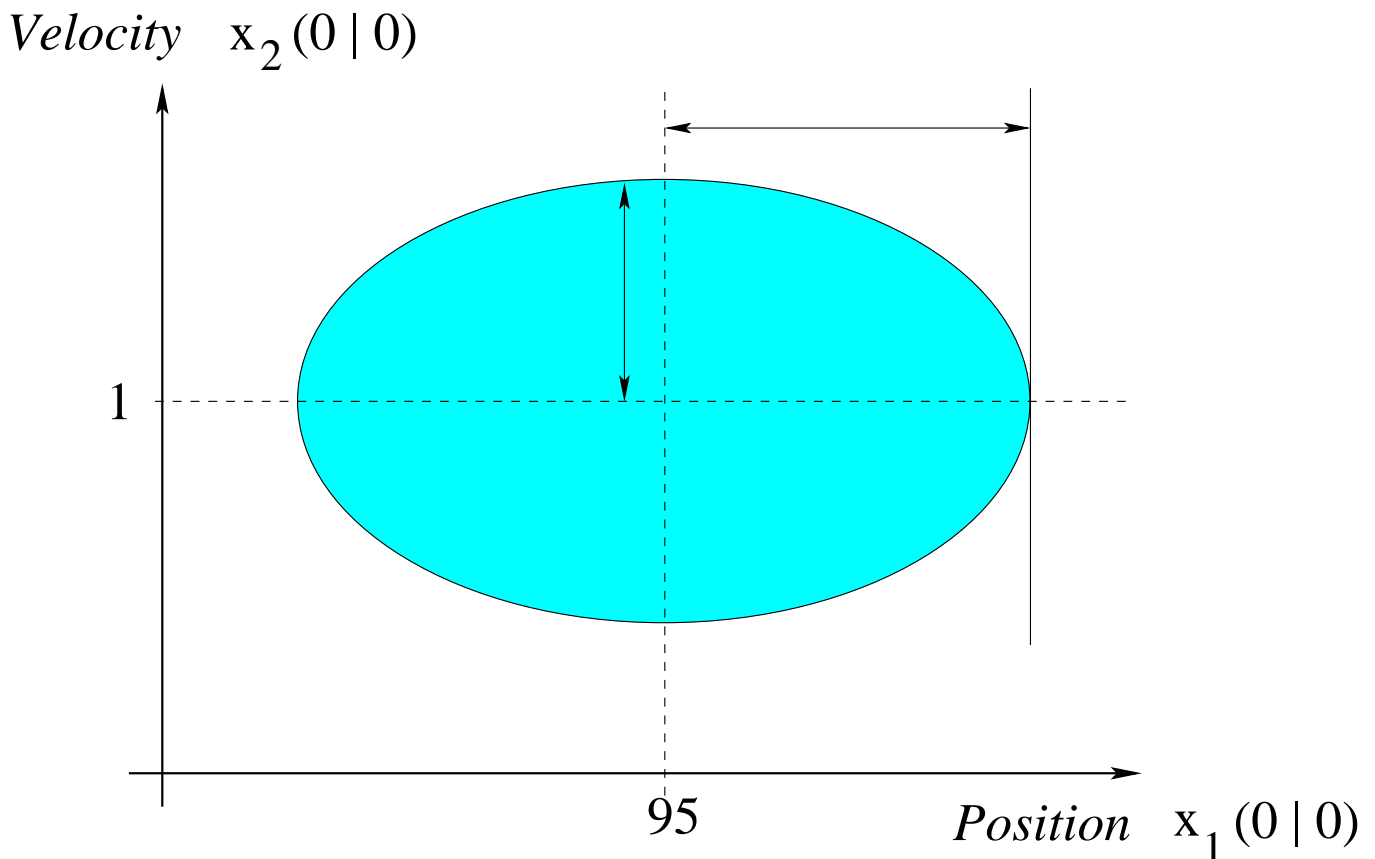
Diagonal terms in $\mathbf{P}(k | k)$ are variances in pos. and vel. respectively.

Off-diagonal terms in $\mathbf{P}(k | k)$ are *correlations* between pos. and vel. Note $\sigma_p \sigma_v = \sigma_v \sigma_p$.

Interpretation of Results

$$\hat{\mathbf{x}}(0 | 0) = \begin{bmatrix} 95 \\ 1 \end{bmatrix} \quad \text{and} \quad \mathbf{P}(0 | 0) = \begin{bmatrix} 10 & 0 \\ 0 & 1 \end{bmatrix}$$

1-Sigma Error Covariance Ellipse:



Region with *minimum area* known to contain state $\hat{\mathbf{x}}(0 | 0)$ with some constant probability.

Interpretation of Results

In general, $\hat{\mathbf{x}}(k | k)$ is a χ^2 random variable.

Probability value obtained from chi-squared distribution tables.

Example: Figure shows “1-sigma limit” since length of semi-major axis of ellipse = $\sigma_p = \sqrt{10}$ and semi-minor axis length = $\sigma_v = \sqrt{1}$.

From χ^2 tables, the 1-sigma limit of a 2-degree of freedom χ^2 random variable, has probability mass 0.393.

Hence - we are almost 40% certain that the state $\hat{\mathbf{x}}(0 | 0)$ is within the ellipse.

Interpretation of Results

How do we plot:

$$\hat{\mathbf{x}}(1 | 1) = \begin{bmatrix} 99.6 \\ 0.37 \end{bmatrix}, \quad \mathbf{P}(1 | 1) = \begin{bmatrix} 11/12 & 1/12 \\ 1/12 & 11/12 \end{bmatrix} ?$$

More difficult, since states are *correlated* – $\mathbf{P}(1 | 1)$ is not diagonal.

Therefore necessary to find a new coordinate system in which state $\hat{\mathbf{x}}(1 | 1)$ can be represented, and in which matrix $\mathbf{P}(1 | 1)$ becomes diagonal – ie. states are the decoupled.

By definition, this coordinate system = *eigenvectors* of matrix $\mathbf{P}(1 | 1)$ and new diagonal elements (“sigma squared” values) are *eigenvalues* of $\mathbf{P}(1 | 1)$.

Interpretation of Results

Finding eigenvalues of $\mathbf{P}(1 | 1)$:

$$|\mathbf{P}(1 | 1) - \lambda \mathbf{I}| = 0$$

$$\begin{vmatrix} \frac{11}{12} - \lambda & \frac{1}{12} \\ \frac{1}{12} & \frac{11}{12} - \lambda \end{vmatrix} = 0$$

Hence 2 solutions:

$$\lambda = \lambda_1 = \frac{5}{6}, \quad \lambda = \lambda_2 = 1$$

Semi-major axis length of “1-sigma” uncertainty ellipse = $\sqrt{1}$.

Semi-minor axis of “1-sigma” uncertainty ellipse = $\sqrt{\frac{5}{6}}$.

Interpretation of Results

Eigenvector associated with λ_1 :

$$\mathbf{P}(1 \mid 1)\mathbf{v}_1 = \lambda_1\mathbf{v}_1$$

Let $\mathbf{v}_1 = (v_{x1}, v_{y1})^T$:

$$\frac{1}{12} \begin{bmatrix} 11 & 1 \\ 1 & 11 \end{bmatrix} \begin{bmatrix} v_{x1} \\ v_{y1} \end{bmatrix} = \frac{5}{6} \begin{bmatrix} v_{x1} \\ v_{y1} \end{bmatrix}$$

ie.

$$11v_{x1} + v_{y1} = 10v_{x1}$$

$$v_{x1} + 11v_{y1} = 10v_{y1}$$

Both equations give:

$$\frac{v_{x1}}{v_{y1}} = \frac{-1}{1}$$

Interpretation of Results

Similarly eigenvector associated with λ_2

$$\mathbf{P}(1 \mid 1)\mathbf{v}_2 = \lambda_2\mathbf{v}_2$$

Let $\mathbf{v}_2 = (v_{x2}, v_{y2})^T$:

$$\frac{1}{12} \begin{bmatrix} 11 & 1 \\ 1 & 11 \end{bmatrix} \begin{bmatrix} v_{x2} \\ v_{y2} \end{bmatrix} = 1 \begin{bmatrix} v_{x2} \\ v_{y2} \end{bmatrix}$$

ie.

$$11v_{x2} + v_{y2} = 12v_{x2}$$

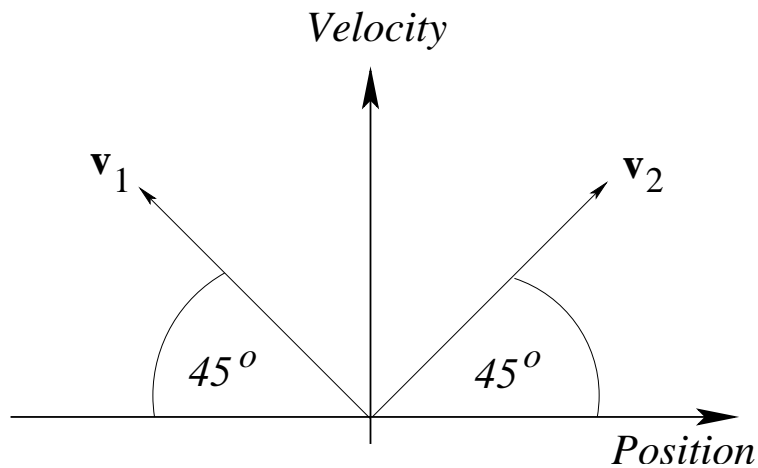
$$v_{x2} + 11v_{y2} = 12v_{y2}$$

Both equations give:

$$\frac{v_{x2}}{v_{y2}} = \frac{1}{1}$$

Interpretation of Results

Hence eigenvectors are:



1-Sigma error covariance ellipse for $\hat{\mathbf{x}}(1 | 1)$

