Chapter 6
State Estimation

Part 1

6.1 Mathematics of Pose Estimation

Carnegie Mellon
THE ROBOTICS INSTITUTE

Outline

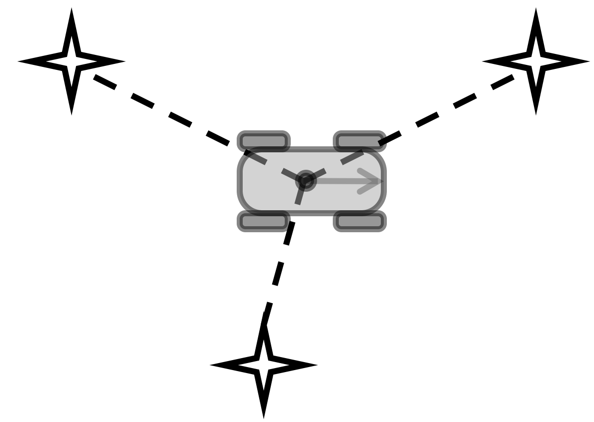
- 6.1 Mathematics of Pose Estimation
  - 6.1.1 Pose Fixing versus Dead Reckoning
  - 6.1.2 Pose Fixing
  - 6.1.3 Error Propagation in Triangulation
  - 6.1.4 Real Pose Fixing Systems
  - 6.1.5 Dead Reckoning
  - 6.1.6 Real Dead Reckoning Systems
  - Summary

Outline

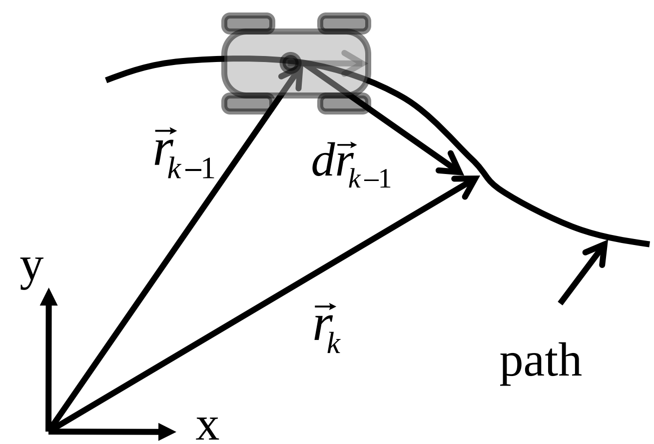
- 6.1 Mathematics of Pose Estimation
  - 6.1.1 Pose Fixing versus Dead Reckoning
  - 6.1.2 Pose Fixing
  - 6.1.3 Error Propagation in Triangulation
  - 6.1.4 Real Pose Fixing Systems
  - 6.1.5 Dead Reckoning
  - 6.1.6 Real Dead Reckoning Systems
  - Summary

Pose Fixing vs Dead Reckoning

- Two alternatives for determining pose of a robot.
- Triangulation
  - Solve nonlinear transcendental or algebraic equations
- Odometry
  - Solve (integrate) differential equations.



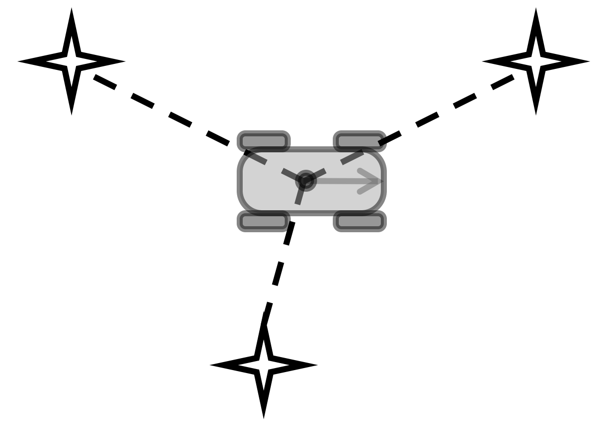
Triangulation



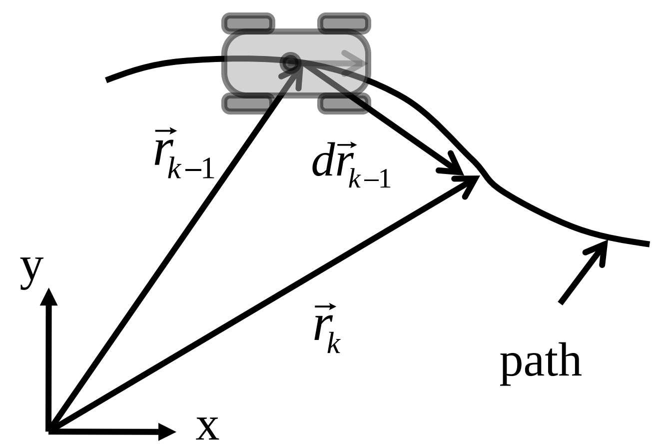
Dead Reckoning

General Points

- Can triangulate position, velocity or angle.
  - GPS triangulates Doppler to get velocity.
- Can dead reckon position, velocity or angle.
  - Can integrate acceleration to get velocity.
  - Can integrate angular velocity to get angle.
- Ultimately, need enough constraints to solve for the unknowns.



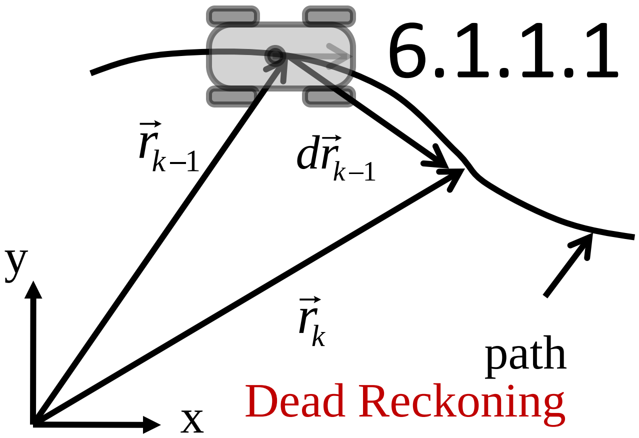
Triangulation

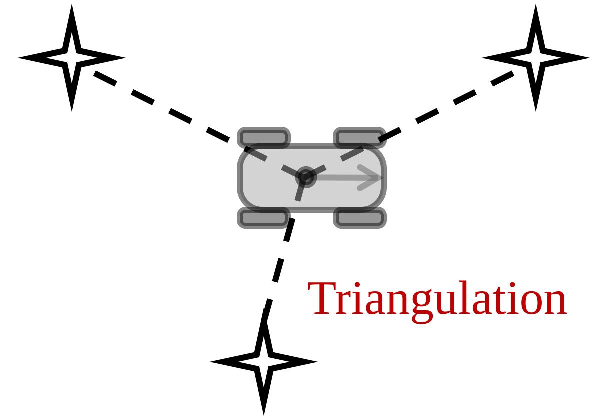


Dead Reckoning

Carnegie Mellon THE ROBOTICS INSTITUTE

6.1.1.1 Complementarity





Attribute	Dead Reckoning	Triangulation Trilateration
Process	Integration	Algebraic
Initial Conditions	Required	Not required
Errors	Time Dependent	Position Dependent
Update Frequency	Determined by required accuracy	Determined by availability
Error Propagation	History Dependent	History Independent
Requires Map	No	Yes

Opposites in Every Respect

Quality of Aiding

Better

Position	Heading	Attitude
Position	Heading	Attitude
Ranging	Bearings	Elevations
Bearings	Δ Position	Gravity
Velocity	Heading Rate	Attitude Rate
Acceleration		
Specific Force		

Worse

IMUs

It Hurts to Lose GPS!

Outline

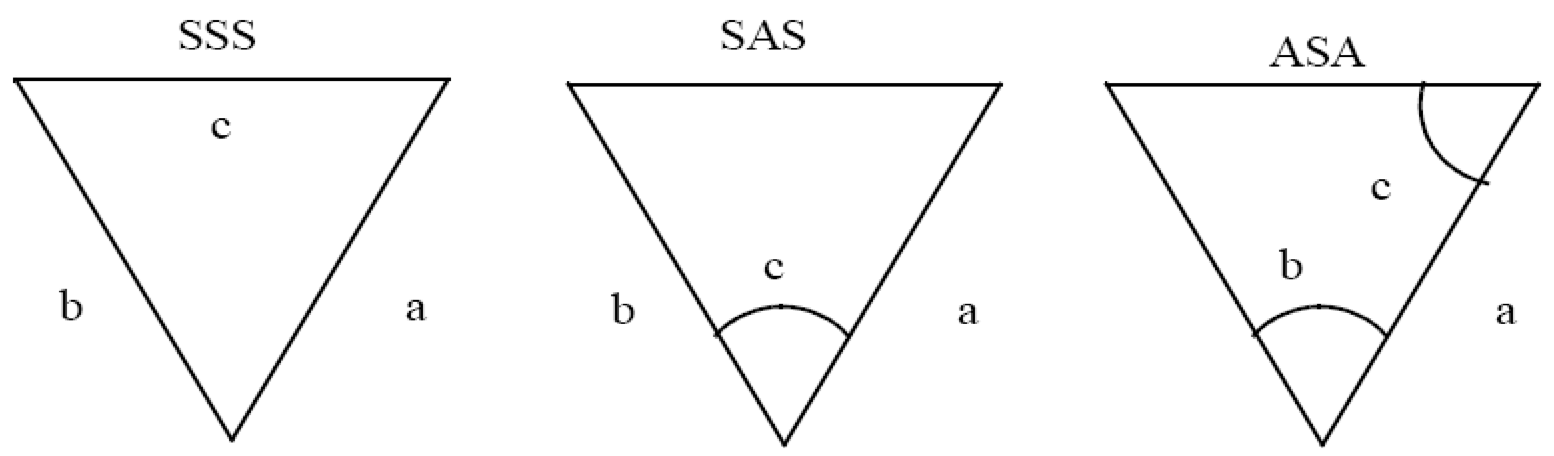
- 6.1 Mathematics of Pose Estimation
  - 6.1.1 Pose Fixing versus Dead Reckoning
  - <u>6.1.2 Pose Fixing</u>
  - 6.1.3 Error Propagation in Triangulation
  - 6.1.4 Real Pose Fixing Systems
  - 6.1.5 Dead Reckoning
  - 6.1.6 Real Dead Reckoning Systems
  - Summary

History

- Roots in survey and cartography.
  - Egyptians
  - Sumarians
- Called “pilotage” in marine applications.
- Economic drivers were the same as those that drove writing and arithmetic.
  - Sound building construction
  - Accurate records of land holdings
  - Accurate records of business transactions

Triangles

- Ancients knew the 3-4-5 triangle was a right angle.
- All 3 parameter triangles but AAA are solveable.

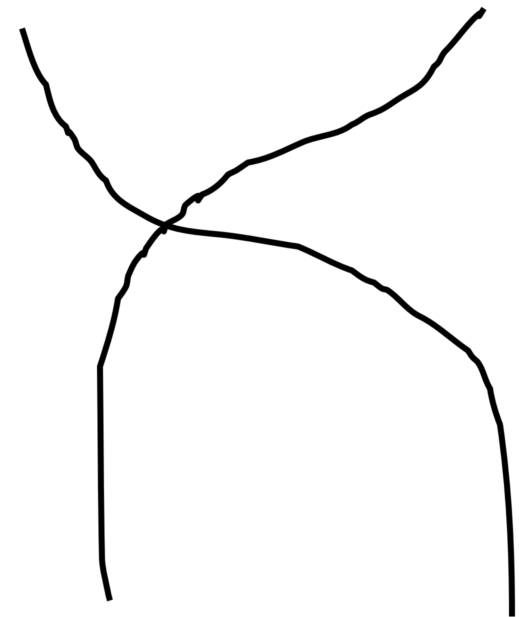


6.1.2.1 Revisiting Nonlinearly Constrained Systems

- It is always a question of satisfying constraints.
  - Navigation variables appear as unknowns.

$$\underline{c}(\underline{x}) = \underline{0}$$

- It is not always a triangle.
- Robot pose is the point where all constraints are satisfied.



6.1.2.1 Revisiting Nonlinearly Constrained Systems

- No existence or uniqueness theorems in general.
- Cannot use approximations in navigational contexts.
- Analogous to manipulator kinematics.
- Many issues:
  - inconsistency of equations (no solution)
  - redundancy (several solutions)
  - dependence of equations (poor conditioning)
  - singularity (poor conditioning)
  - (under/over)constraint (too many/too little)
- 2D answers are clear from geometry but general higher D cases require math to solve.

6.1.2.1.1 Explicit Case

- Rarely, we can write an explicit formula for determining the state $\underline{x} = \begin{bmatrix} x & y & \psi \end{bmatrix}^T$ from the measurements $\underline{z} = \begin{bmatrix} z_1 & z_2 & z_3 \end{bmatrix}^T$

$$\underline{x} = \underline{f}(\underline{z})$$

$$x = f_1(z_1, z_2, z_3)$$

$$y = f_2(z_1, z_2, z_3)$$

$$\psi = f_3(z_1, z_2, z_3)$$

6.1.2.1.2 Implicit Case

- The inverse situation is more common

$$\underline{z} = h(\underline{x})$$

Compare with Kalman Filter Measurement Model

$$z_1 = h_1(x, y, \psi)$$
$$z_2 = h_2(x, y, \psi)$$
$$z_3 = h_3(x, y, \psi)$$

Solving Implicit Case

- Linearize with:

$$\underset{m \times 1}{\Delta \underline{z}} = H \underset{n \times 1}{\Delta \underline{x}}$$

- Solve iteratively with gradient descent, least squares, etc. Consider pseudoinverses:

Left Pseudoinverse

$$\Delta \underline{x} = (H^T H)^{-1} H^T \Delta \underline{z} \quad \text{Overdetermined } (m > n)$$

Right Pseudoinverse

$$\Delta \underline{x} = H^T (H H^T)^{-1} \Delta \underline{z} \quad \text{Underdetermined } (m < n)$$

- Underdetermined is not common here.
  - But its common in a Kalman filter.
  - Helps to use uncertainty as weights.

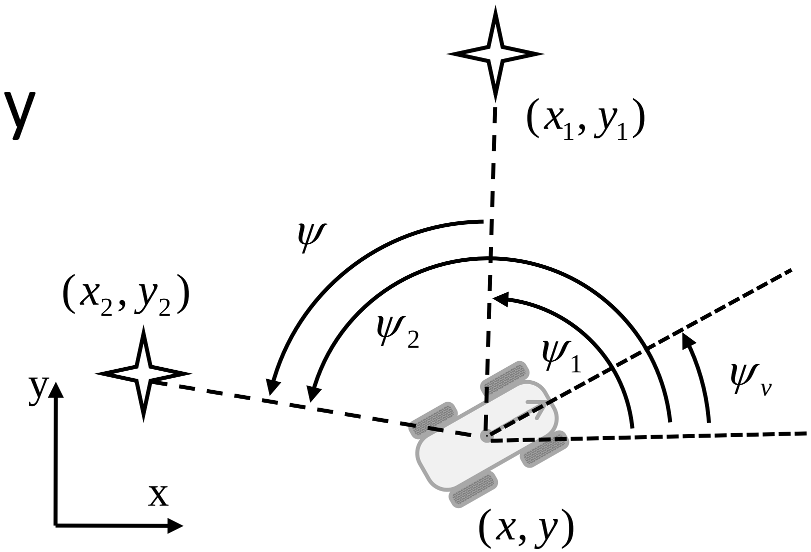
6.1.2.2 Bearing Observations with Known Yaw

- Some sensor gives ψ_v directly
- Relative bearings ψ_1, ψ_2 to landmarks are measured
- Constraints are:

$$\tan\psi_1 = \frac{\sin\psi_1}{\cos\psi_1} = \frac{y_1 - y}{x_1 - x}$$

$$\tan\psi_2 = \frac{\sin\psi_2}{\cos\psi_2} = \frac{y_2 - y}{x_2 - x}$$

$$\begin{bmatrix} -s_1 & c_1 \\ -s_2 & c_2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} -s_1 x_1 + c_1 y_1 \\ -s_2 x_2 + c_2 y_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$



- Solution except when determinant = 0:

$$-s_1 c_2 + s_2 c_1 = 0$$

$$\sin(\psi_2 - \psi_1) = 0$$

$$\psi = n\pi$$

1) Don’t operate on line between landmarks
2) Use 3rd landmark or position 2 appropriately

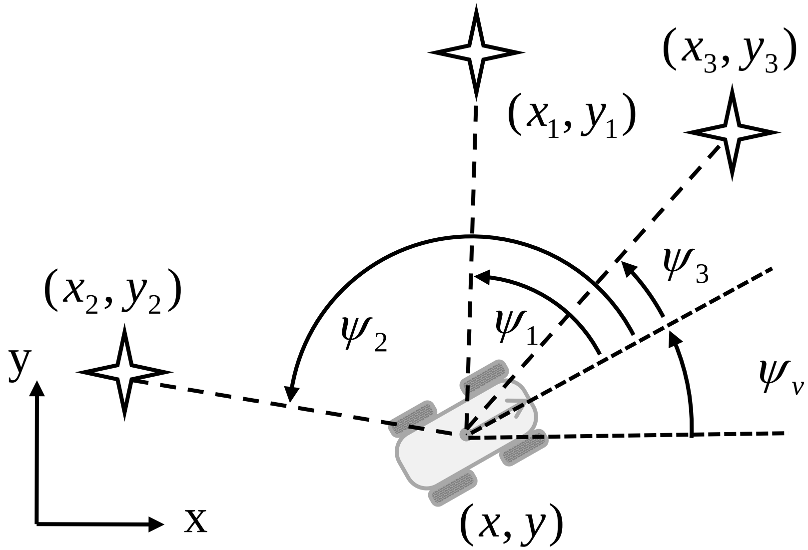
6.1.2.3 Bearing Observations with Unknown Yaw

- No heading sensor so use 3rd landmark.
- Constraints are:

$$\tan(\psi_v + \psi_1) = \frac{\sin(\psi_v + \psi_1)}{\cos(\psi_v + \psi_1)} = \frac{y_1 - y}{x_1 - x}$$

$$\tan(\psi_v + \psi_2) = \frac{\sin(\psi_v + \psi_2)}{\cos(\psi_v + \psi_2)} = \frac{y_2 - y}{x_2 - x}$$

$$\tan(\psi_v + \psi_3) = \frac{\sin(\psi_v + \psi_3)}{\cos(\psi_v + \psi_3)} = \frac{y_3 - y}{x_3 - x}$$

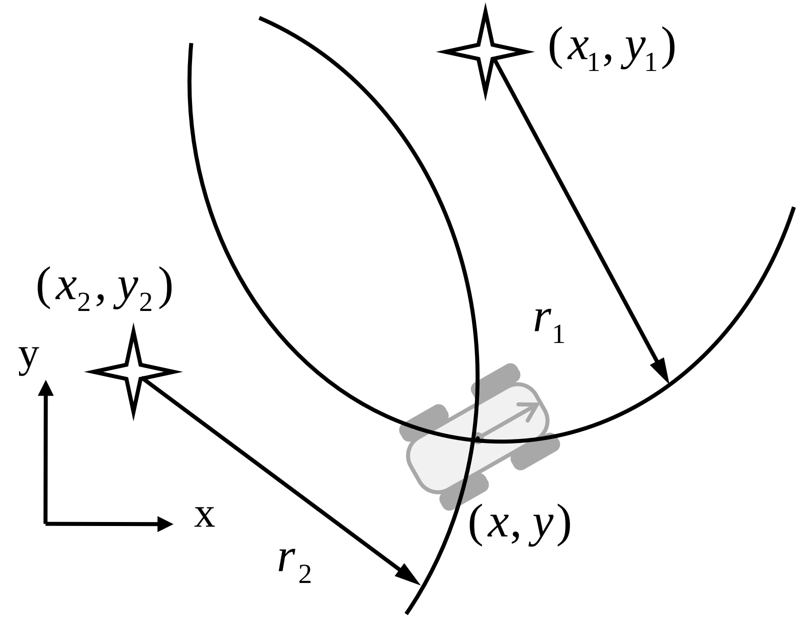


6.1.2.4 Circular Constraints

- Ranges are observables.
- Constraints are:

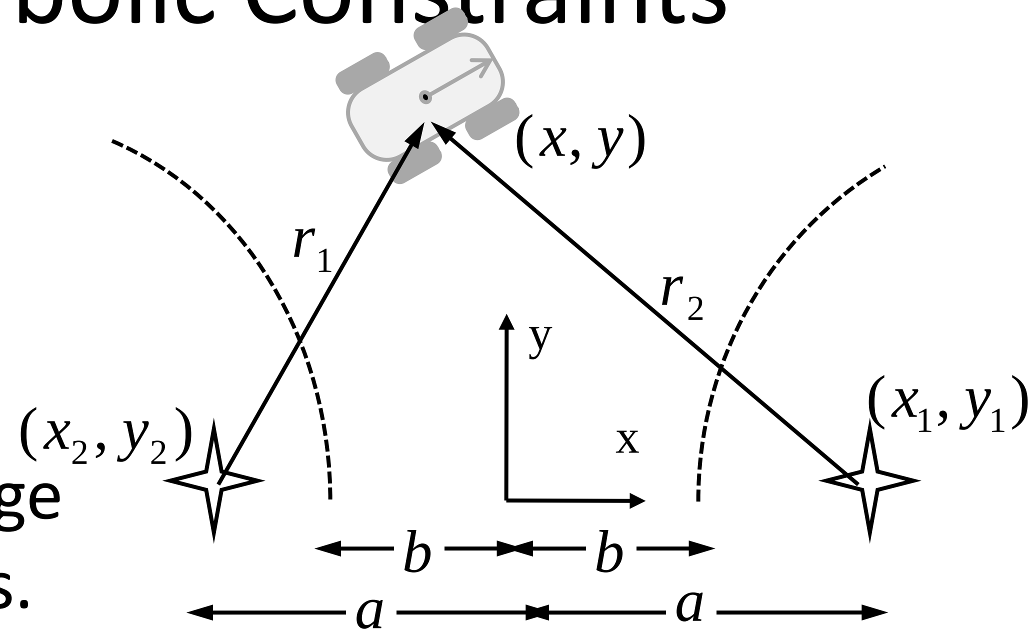
$$r_1 = \sqrt{(x - x_1)^2 + (y - y_1)^2}$$

$$r_2 = \sqrt{(x - x_2)^2 + (y - y_2)^2}$$



- Degeneracy (singularity) not an issue for unique landmarks.
- Redundancy (multiple solutions) is.
  – Use last known position?

6.1.2.5 Hyperbolic Constraints



- Used in marine radio navigation.
- Measure time of flight or phase differences.
- Contours of constant range difference are hyperbolas.
- Why a hyperbola? Put origin between landmarks. Then suppose: $r_1 - r_2 = 2b$

$$\sqrt{(x+a)^2 + y^2} - \sqrt{(x-a)^2 + y^2} = 2b$$

$$\sqrt{(x+a)^2 + y^2} = 2b + \sqrt{(x-a)^2 + y^2}$$

$$b^2 - ax = -b\sqrt{(x-a)^2 + y^2}$$

After squaring both sides.

$$\frac{x^2}{b^2} - \frac{y^2}{(a^2 - b^2)} = 1$$

Outline

- 6.1 Mathematics of Pose Estimation
  - 6.1.1 Pose Fixing versus Dead Reckoning
  - 6.1.2 Pose Fixing
  - <u>6.1.3 Error Propagation in Triangulation</u>
  - 6.1.4 Real Pose Fixing Systems
  - 6.1.5 Dead Reckoning
  - 6.1.6 Real Dead Reckoning Systems
  - Summary

6.1.3.1 First Order Response to Systematic Errors

- Involves the same mathematics used to solve the nonlinear problem by linearization.
  – Linearization evaluates how errors in inputs project onto errors in outputs.
- Analogous to differential kinematics of manipulators.

6.1.3.1 First Order Response to Systematic Errors (Direct Case)

- Recall:

$$\underline{x} = \underline{f}(\underline{z})$$

- Linearize:

$$\delta\underline{x} = \left(\frac{\partial\underline{x}}{\partial\underline{z}}\right)\delta\underline{z} = J\delta\underline{z}$$

- In detail:

$$\begin{bmatrix} \delta x \\ \delta y \\ \delta\theta \end{bmatrix} = \begin{bmatrix} \partial f_1/\partial z_1 & \partial f_1/\partial z_2 & \partial f_1/\partial z_3 \\ \partial f_2/\partial z_1 & \partial f_2/\partial z_2 & \partial f_2/\partial z_3 \\ \partial f_3/\partial z_1 & \partial f_3/\partial z_2 & \partial f_3/\partial z_3 \end{bmatrix} \begin{bmatrix} \delta z_1 \\ \delta z_2 \\ \delta z_3 \end{bmatrix}$$

$\delta\psi$

- Jacobian J normally depends on the state.

6.1.3.1 First Order Response to Systematic Errors (Indirect Case)

- Now:

$$\underline{z} = h(\underline{x})$$

Note: H, not J

- So:

$$\delta\underline{z} = H\delta\underline{x}$$

- If measurements determine or overdetermine the state, then:

$$\delta\underline{x} = (H^T H)^{-1} H^T \delta\underline{z}$$

6.1.3.2 Geometric Dilution of Precision

From Last Slide: $\delta \underline{x} = (H^T H)^{-1} H^T \delta \underline{z}$

- If $R = Exp[\delta \underline{z} \delta \underline{z}^T]$ is the measurement covariance, then the covariance of least square estimate from last slide is:

$$Exp[\delta \underline{x} \delta \underline{x}^T] = (H^T R^{-1} H)^{-1}$$

- So, $(H^T H)^{-1}$ gives the pose error covariance when measurement errors are of unit magnitude.
  - So, it's a measure of the capacity of the pose fixing process to magnify or attenuate error.

6.1.3.2 Geometric Dilution of Precision

- In GPS, the Geometric Dilution of Precision (GDOP) is:

$$GDOP = \sqrt{trace[(H^T H)^{-1}]} = \sqrt{\sigma_{xx} + \sigma_{yy} + \sigma_{\theta\theta}}$$

Relates to Length of Error vector

- By analogy, define the (simpler) *dilution of precision (DOP)*.

$$DOP = \sqrt{det[(H^T H)^{-1}]} = \sqrt{det[H^{-1}]det[H^{-T}]}$$

Relates to Volume of Differential Error region

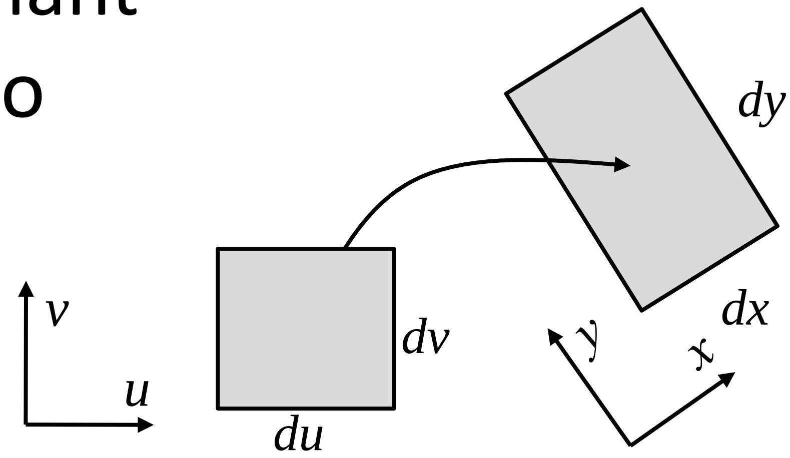
- For square H, this reduces to:

$$DOP = det[H^{-1}] = \frac{1}{det[H]}$$

6.1.3.2.1 Mapping Theory (small bit)

- Recall, the Jacobian determinant relates differential volumes to differential volumes.

$$\|\delta \underline{x}\| = \left|\left(\frac{\partial \underline{x}}{\partial \underline{z}}\right)\right| \|\delta \underline{z}\| = |J| \|\delta \underline{z}\|$$



- Limit of DOP is (forward) Jacobian determinant:

$$\lim_{\delta \underline{z} \to 0} DOP = \frac{\|\delta \underline{x}\|}{\|\delta \underline{z}\|} = |J| = \frac{1}{|H|}$$

Note its J or 1/H

Commentary on DOP

- The DOP is in the range:

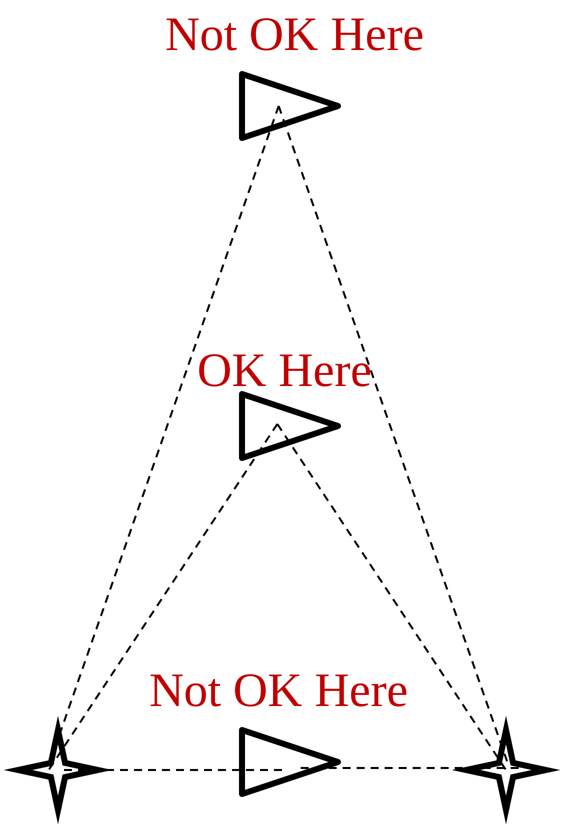
  0 < DOP < infinity

- Infinity is not uncommon, so we must understand it.
- Fix error depends on both measurement error and DOP.
  - Good sensors can overcome poor conditioning in theory.
  - But not in practice when DOP is huge.

$$\|\delta \underline{x}\| = \left|\left(\frac{\partial \underline{x}}{\partial \underline{z}}\right)\right| \|\delta \underline{z}\| = |J| \|\delta \underline{z}\|$$

GDOP "Fields"

- GDOP varies spatially like |J|
  - GDOP varies smoothly with space in real situations
  - GDOP goes to infinity where Jacobian is singular or its inverse has zero determinant



Computing GDOP

- Tricks → using coordinate transforms
- Investigate with contour graphs
- Only |J| is required
  - Not J explicitly

6.1.3.2.2 Implicit GDOP

- Technique has roots in the implicit function theorem. Consider 2 constraints on 4 variables:

$$F(x, y, z, w) = 0$$
$$G(x, y, z, w) = 0$$

2 constraints on 4 variables means 2 free dof are left.

- Arbitrarily choose x & y to be "independent".
- These define two implicit functions :
  - $w(x,y)$ and $z(x,y)$.
- Take total differentials:

$$F_x \delta x + F_y \delta y + F_z \delta z + F_w \delta w = 0$$
$$G_x \delta x + G_y \delta y + G_z \delta z + G_w \delta w = 0$$

6.1.3.2.2 Implicit GDOP

$$F_x\delta x + F_y\delta y + F_z\delta z + F_w\delta w = 0$$
$$G_x\delta x + G_y\delta y + G_z\delta z + G_w\delta w = 0$$

- These define 2 simultaneous equations:

$$\begin{bmatrix} F_x & F_y \\ G_x & G_y \end{bmatrix}\begin{bmatrix} \delta x \\ \delta y \end{bmatrix} = -\begin{bmatrix} F_z & F_w \\ G_z & G_w \end{bmatrix}\begin{bmatrix} \delta z \\ \delta w \end{bmatrix}$$

$$[f_x]\delta \underline{x} = -[f_z]\delta \underline{z}$$

Generates linear behavior without ever solving the equations F,G, explicitly.

- Can also go in opposite direction:

$$\begin{bmatrix} \delta x \\ \delta y \end{bmatrix} = -\begin{bmatrix} F_x & F_y \\ G_x & G_y \end{bmatrix}^{-1}\begin{bmatrix} F_z & F_w \\ G_z & G_w \end{bmatrix}\begin{bmatrix} \delta z \\ \delta w \end{bmatrix}$$

- Using rules for determinants of products:

$$|H| = \begin{vmatrix} F_x & F_y \\ G_x & G_y \end{vmatrix} \Big/ \begin{vmatrix} F_z & F_w \\ G_z & G_w \end{vmatrix} = \frac{\|f_x\|}{\|f_z\|}$$

$$|J| = \begin{vmatrix} F_z & F_w \\ G_z & G_w \end{vmatrix} \Big/ \begin{vmatrix} F_x & F_y \\ G_x & G_y \end{vmatrix} = \frac{\|f_z\|}{\|f_x\|}$$

Can get the GDOP without even computing the explicit Jacobian !!

6.1.3.3 First Order Response to Random Error

- Suppose the input errors are random. For inverse case:

$$\delta \underline{z} = H \delta \underline{x}$$

- The measurement covariance is clearly:

$$C_{\underline{z}} = H C_{\underline{x}} H^T$$

Compare to Innovation Covariance

- If least squares is used to solve for the state, then:

$$C_{\underline{x}} = (H^T C_{\underline{z}}^{-1} H)^{-1}$$

6.1.3.4 Bearing Observations with Known Yaw

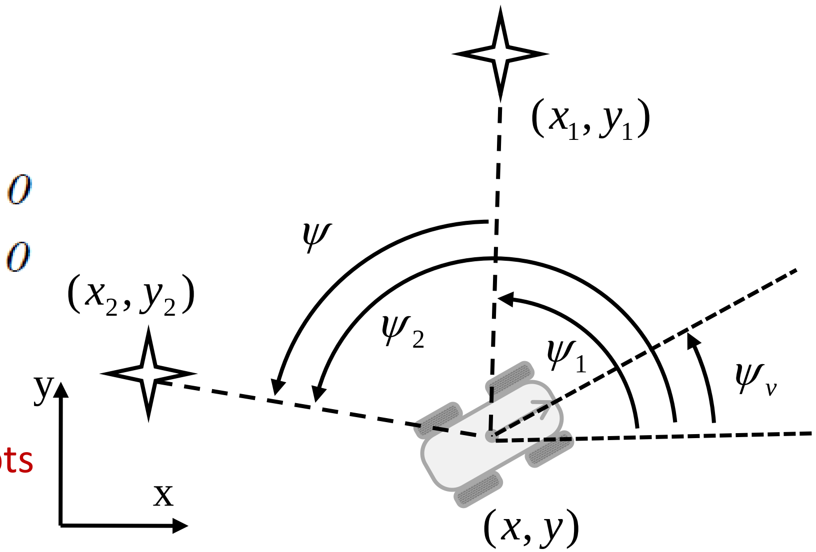
- Write constraints like so:

$$F(x, y, \psi_1, \psi_2) = s_1(x_1 - x) - c_1(y_1 - y) = 0$$
$$G(x, y, \psi_1, \psi_2) = s_2(x_2 - x) - c_2(y_2 - y) = 0$$

- Write total differentials:

$$F_x\delta x + F_y\delta y + F_{\theta_1}\delta\psi_1 + F_{\theta_2}\delta\psi_2 = 0$$
$$G_x\delta x + G_y\delta y + G_{\theta_1}\delta\psi_1 + G_{\theta_2}\delta\psi_2 = 0$$

ψ subscripts not θ



- Jacobian Determinant:

$$|J| = \begin{vmatrix} F_{\psi_1} & F_{\psi_2} \\ G_{\psi_1} & G_{\psi_2} \end{vmatrix} \Big/ \begin{vmatrix} F_x & F_y \\ G_x & G_y \end{vmatrix} \qquad |J| = \begin{vmatrix} c_1\Delta x_1 + s_1\Delta y_1 & 0 \\ 0 & c_2\Delta x_2 + s_2\Delta y_2 \end{vmatrix} \Big/ \begin{vmatrix} -s_1 & c_1 \\ -s_2 & c_2 \end{vmatrix}$$

$$\Delta x_1 = (x_1 - x)$$
$$\Delta y_1 = (y_1 - y)$$

- Written out:

$$|J| = \frac{[c_1\Delta x_1 + s_1\Delta y_1][c_2\Delta x_2 + s_2\Delta y_2]}{sin(\psi_2 - \psi_1)}$$

6.1.3.4 Bearing Observations with Known Yaw

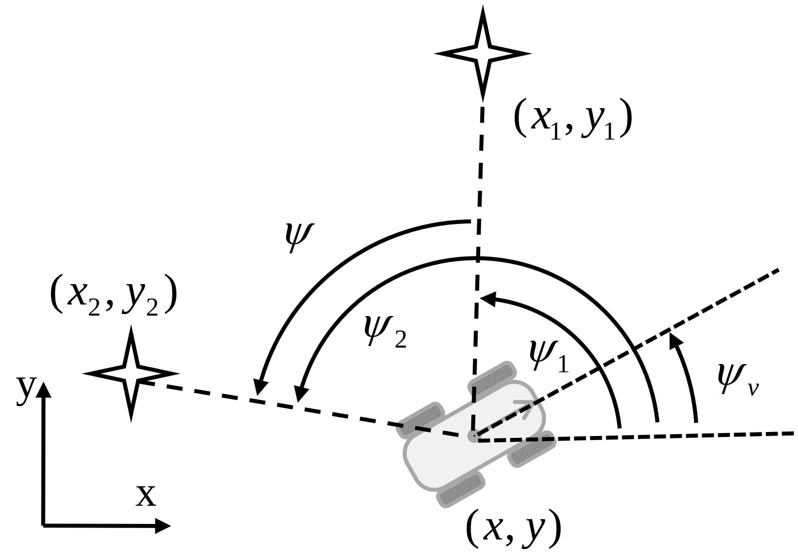
- But this is:

$$|J| = \frac{[c_1\Delta x_1 + s_1\Delta y_1][c_2\Delta x_2 + s_2\Delta y_2]}{\sin(\psi_2 - \psi_1)}$$

$$c_1\Delta x_1 + s_1\Delta y_1 = r_1$$

- Etc., so:

$$|J| = \frac{r_1 r_2}{\sin(\psi)}$$



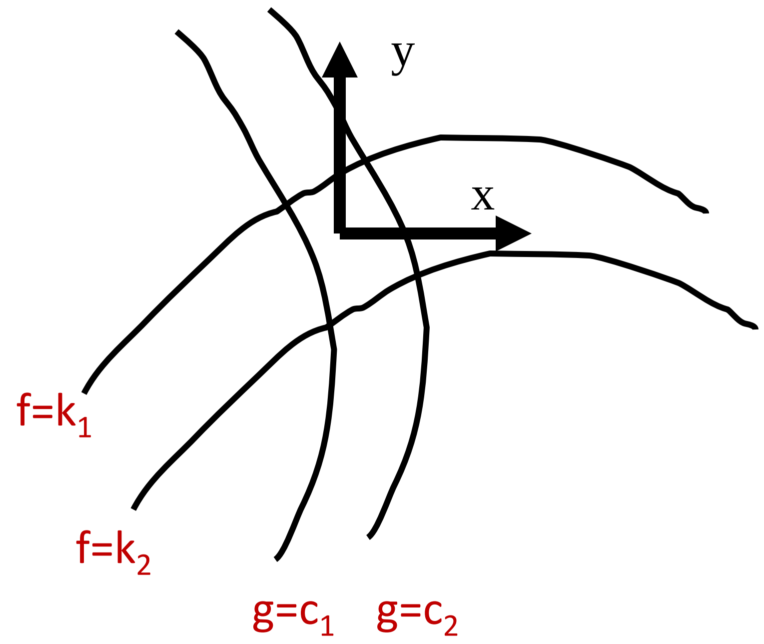
- GDOP:
  - Grows with distance squared.
  - Grows as lines become parallel.
- Both happen at long range.

Contour Diagrams

- Plot “level” curves of constraints:

  $z1 = f(x,y) = k_1, k_2$

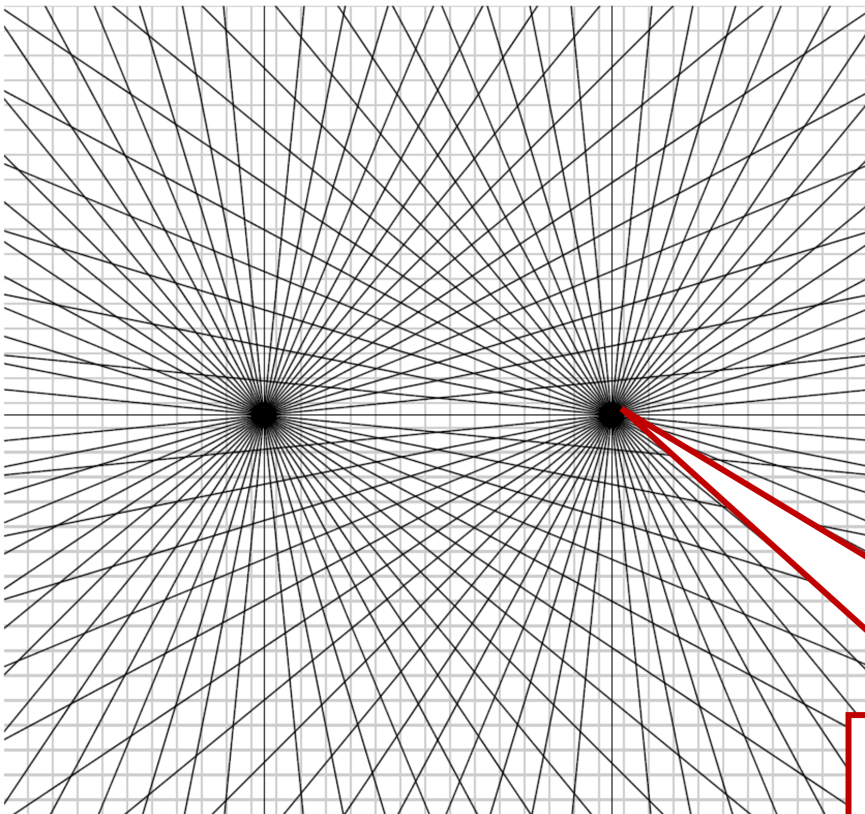
  $z2 = g(x,y) = c_1, c_2$

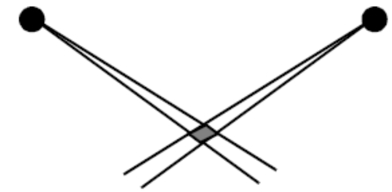


- Relative sizes of enclosed regions are meaningful
  - when spacing of contour values (f,g) is even.
- Size and shape at vehicle position is meaningful.

6.1.3.4 Bearing Observations with Known Yaw

- Contour diagrams explain this case:

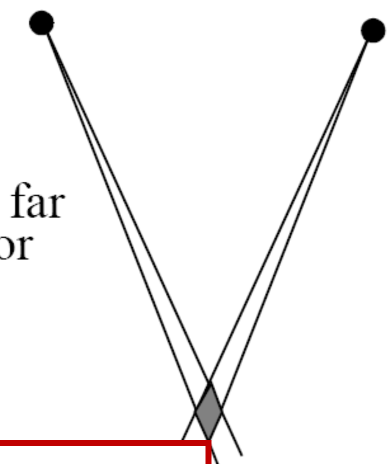




Lines near 90° - good

Lines near parallel - poor

Landmarks far away - poor



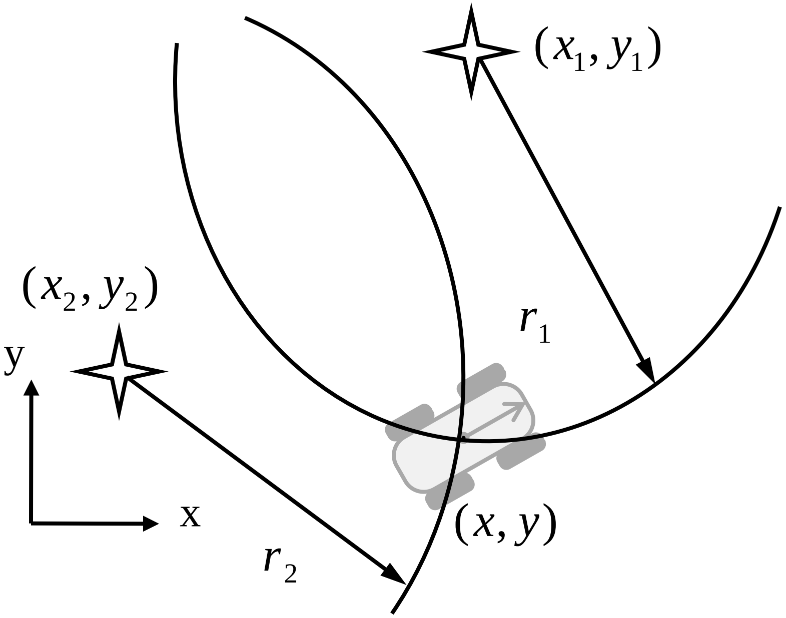
For linear constraints with yaw known. Contours are lines emanating from landmarks.

6.1.3.6 Circular Constraints

- Again, constraints are:

$$r_1 = \sqrt{(x - x_1)^2 + (y - y_1)^2}$$

$$r_2 = \sqrt{(x - x_2)^2 + (y - y_2)^2}$$



- We use a trick. Investigate the inverse Jacobian

$$H = J^{-1}$$

- Take total differentials:

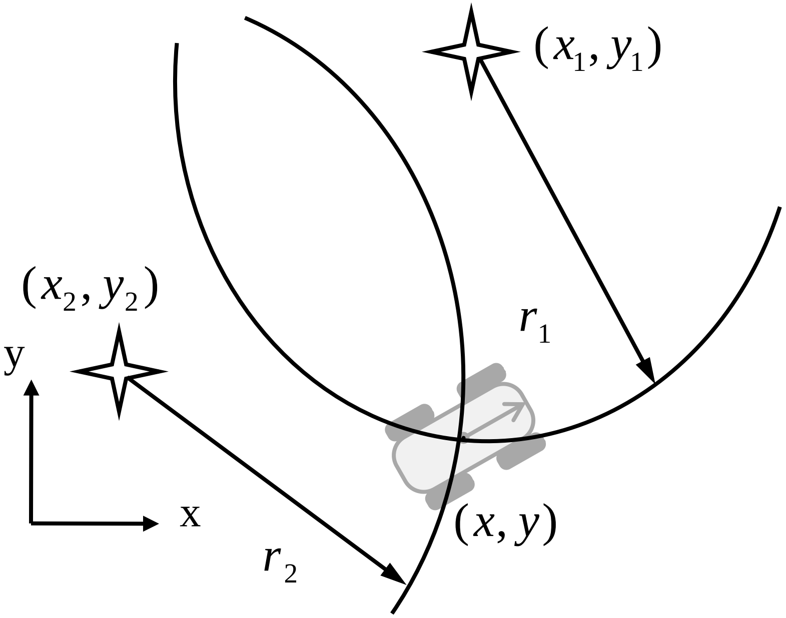
$$\delta r_1 = \frac{(x - x_1)}{r_1}\delta x + \frac{(y - y_1)}{r_1}\delta y$$

$$\delta r_2 = \frac{(x - x_2)}{r_2}\delta x + \frac{(y - y_2)}{r_2}\delta y$$

6.1.3.6 Circular Constraints

- The determinant is:

$$|J^{-1}| = \begin{vmatrix} \frac{x-x_1}{r_1} & \frac{y-y_1}{r_1} \\ \frac{x-x_2}{r_2} & \frac{y-y_2}{r_2} \end{vmatrix}$$



- Vector formulation:

$$|J^{-1}| = \left(\frac{x-x_1}{r_1}\right)\left(\frac{y-y_2}{r_2}\right) - \left(\frac{y-y_1}{r_1}\right)\left(\frac{x-x_2}{r_2}\right)$$

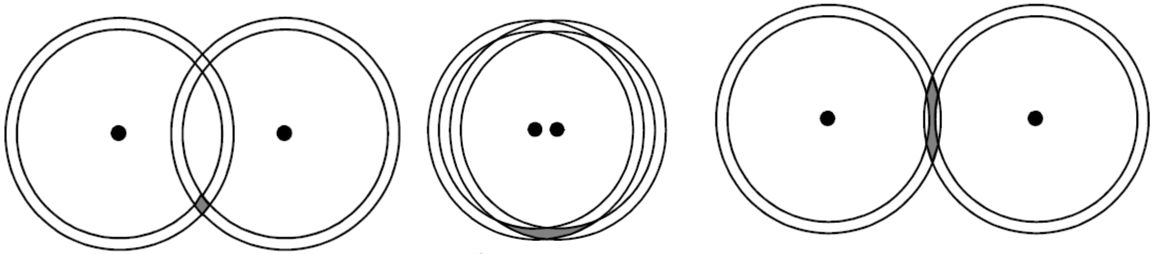
$$|J^{-1}| = \frac{\vec{r}_1 \times \vec{r}_2}{|\vec{r}_1||\vec{r}_2|} = sin(\psi)$$

$$|J| = \frac{1}{sin(\psi)}$$

ψ is the angle between the lines to the landmarks.

Eg: Circular Constraints

$$|J| = \frac{1}{\sin(\psi)}$$

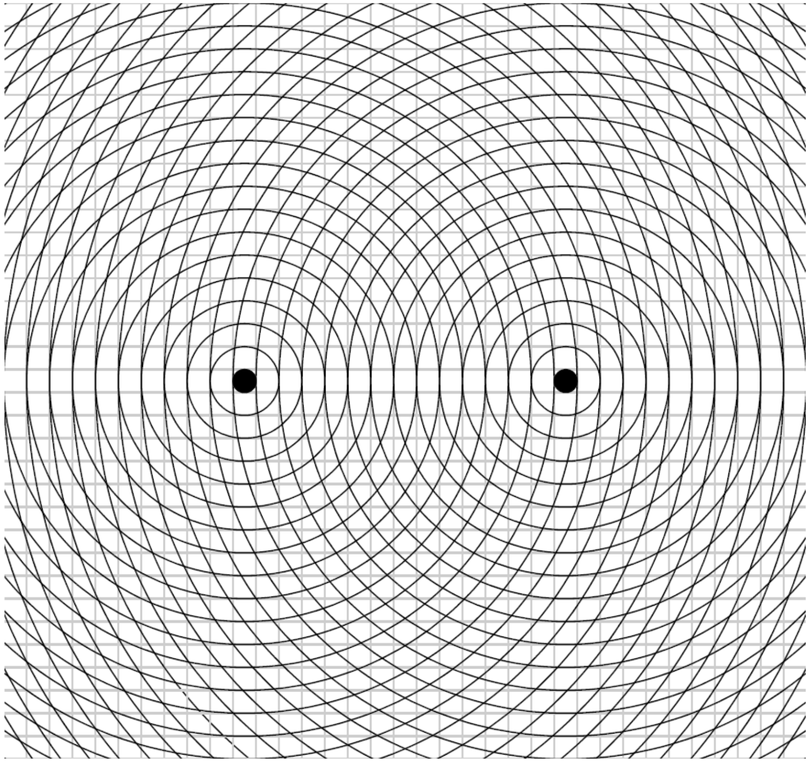


Angle near 90° - good

Angle near 0° - poor

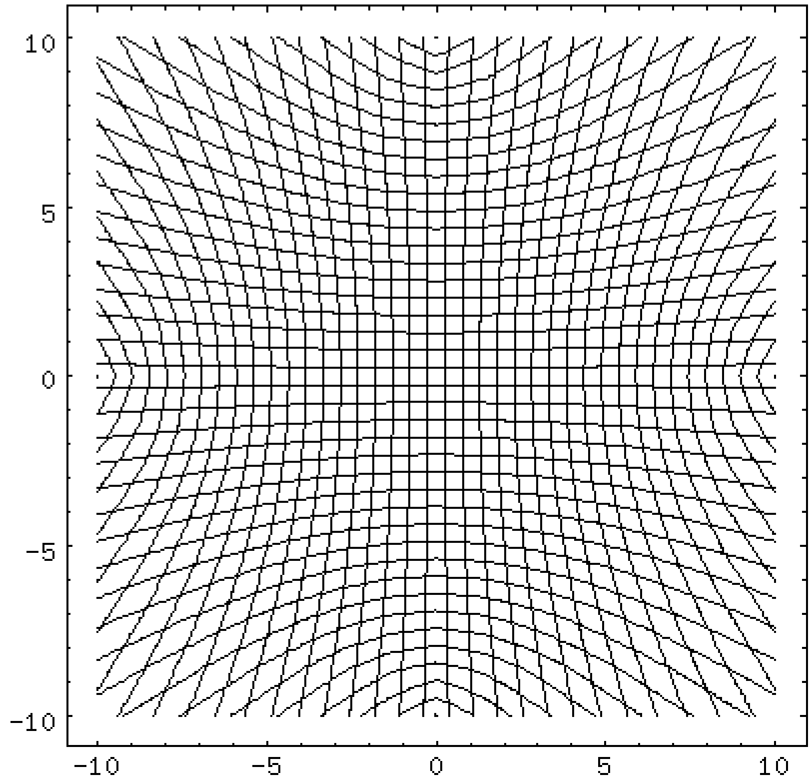
Angle near 180° - poor

- Only an implicit variation with range. q is small when R is large relative to spacing.
- Again, singular on line between landmarks.



6.1.3.7 Hyperbolic Constraints

- GDOP best near origin.
- GDOP increases with distance from either axis.
- No singularities except at infinity.
- Exceptionally well behaved triangulation configuration.

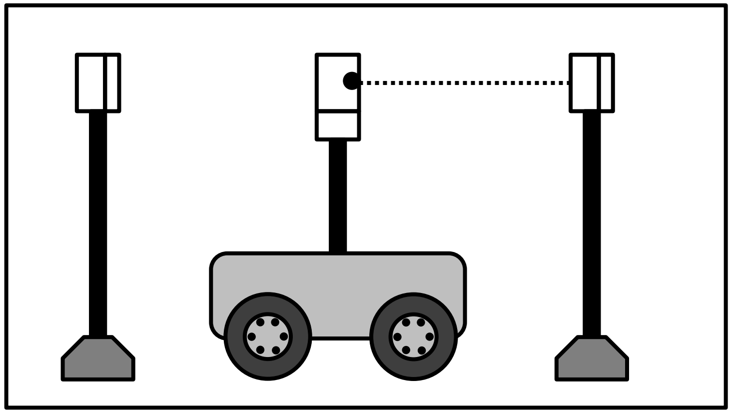


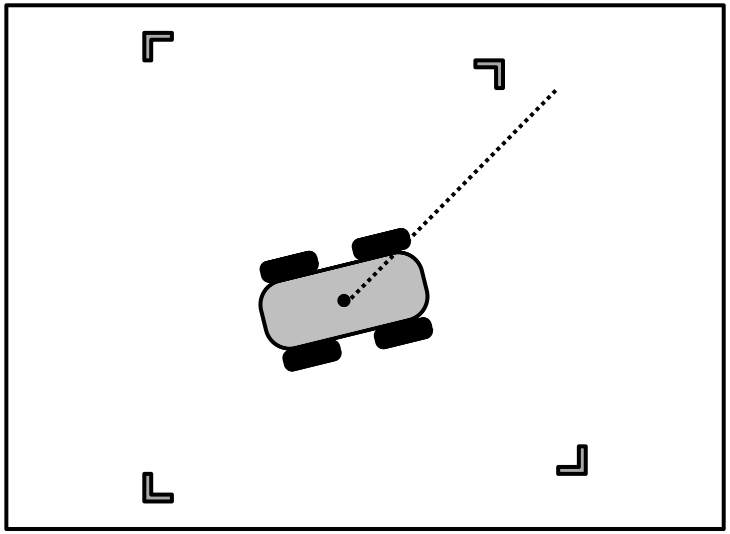
Outline

- 6.1 Mathematics of Pose Estimation
  - 6.1.1 Pose Fixing versus Dead Reckoning
  - 6.1.2 Pose Fixing
  - 6.1.3 Error Propagation in Triangulation
  - <u>6.1.4 Real Pose Fixing Systems</u>
  - 6.1.5 Dead Reckoning
  - 6.1.6 Real Dead Reckoning Systems
  - Summary

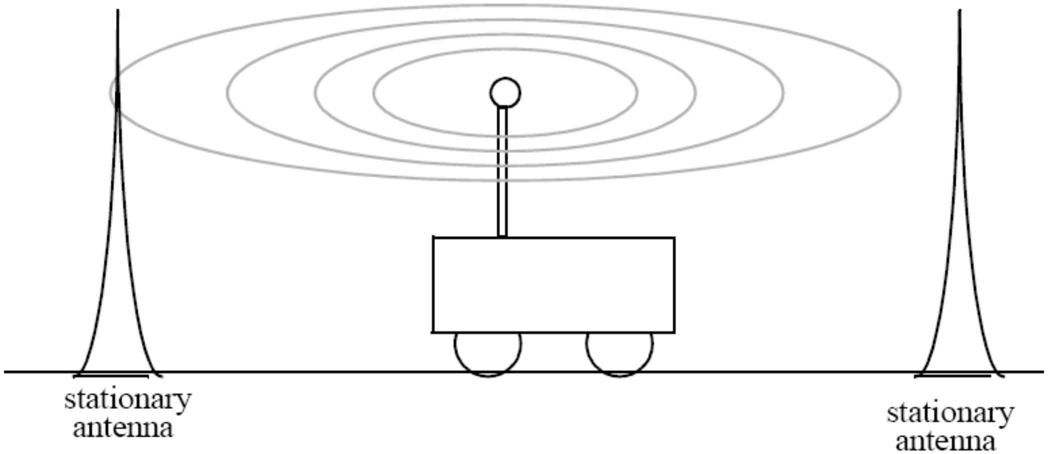
Example: Laser Triangulation

- NDC Automation
- Mount laser emitter and detector on rotary degree of freedom
  - Install retroreflective “artificial landmarks” in work area
  - Measure angles to reflectors and triangulate
  - Math given earlier (need three bearings)
- 50 meter range
- 1 inch accuracy
- Requires line of sight
- Bar coded retroreflectors can permit easy identification



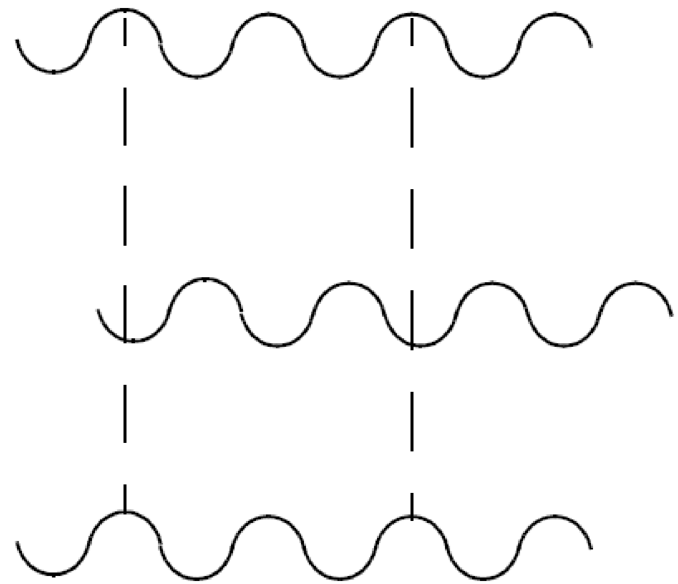


6.1.4.2 Radio Carrier Phase Triangulation



- ARC system
- Identical to inverted Kinematic GPS in concept
- VHF radio (40 MHz) used (wavelength ~ 7.5 meters)
- HF & VHF do not require perfect line of sight
- Carrier phase is direct measure of range
- Remove technology and it is just range triangulation
- Singular between antennae (as always)
- Repeatability 3 cm, accuracy 12 cm
- 100 Hz update
- 5 mile range (limited by FCC regs.)

Principle of Operation – Single Differencing



- Radio wave (at Tx and Rx):

$$I(\vec{r}, t) = I_0 \cos(\omega t + \kappa r)$$

- Antenna signal (at Tx and Rx):

$$v_a(t) = v_0 \cos(\omega t + \kappa r)$$

- Internal oscillator (at Rx):

$$v_o(t) = v_1 \cos(\omega t + \boxed{const})$$

- Phase difference:

$$\Delta\Phi(t) = \Phi_a - \Phi_o = (\boxed{\omega}t + \kappa r) - (\boxed{\omega}t + const) = \kappa r - const$$

Differential phase measurements eliminate time but the Tx and Rx frequences must be identical.

Phase difference is a constant plus an amount **proportional to range** from Tx antenna to vehicle.

Principle of Operation – Double Differencing

- Phase difference at two times:

$$\Delta\Phi(t_1) = \Phi_a(t_1) - \Phi_o(t_1)$$
$$\Delta\Phi(t_2) = \Phi_a(t_2) - \Phi_o(t_2)$$

- Again:

$$\Delta\Phi(t_1) = (\omega_a t_1 + \kappa r_1) - (\omega_o t_1 + \text{const}) = (\omega_a - \omega_o)t_1 + \kappa r_1 - \text{const}$$
$$\Delta\Phi(t_2) = (\omega_a t_2 + \kappa r_2) - (\omega_o t_2 + \text{const}) = (\omega_a - \omega_o)t_2 + \kappa r_2 - \text{const}$$

Very short time.

- Double difference:

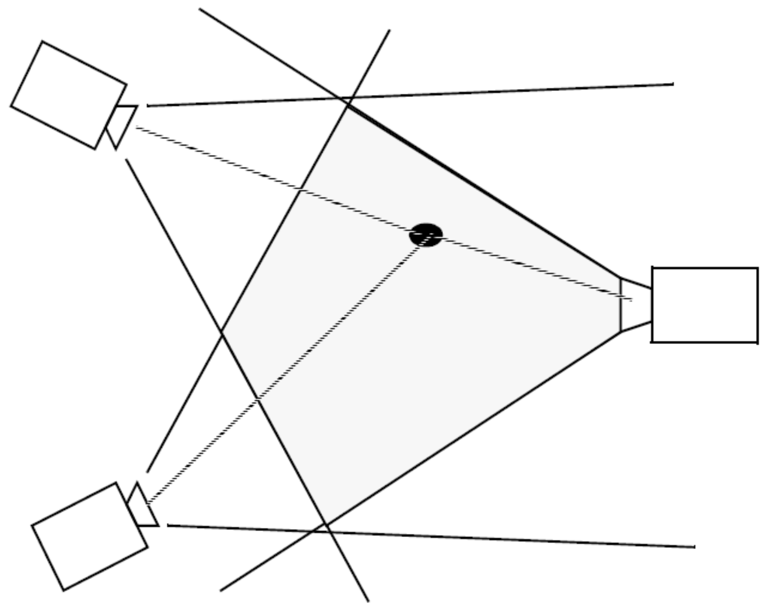
$$\Delta^2\Phi(t_2) = \Delta\Phi(t_1) - \Delta\Phi(t_2) = (\omega_a - \omega_o)\Delta t + \kappa(r_1 - r_2)$$

$$\Delta^2\Phi(t_2) \approx \kappa(r_1 - r_2)$$

Double difference proportional to range difference (range rate) and immune to frequency drift.
If two different transmitters are used, you can similarly do hyperbolic navigation.

Video Triangulation

- Workhorse of motion capture in animated films.
- Innovision Systems Reflex
  - 4 to 7 CCD cameras
  - Internal LED flashers
  - Specially sensitive to IR
  - Tape patches attached to subject (IR retroreflectors. Perfect 1 inch circles
- Real time video processor determines centroids of patches
- Angular resolution of 0.005% of camera field of view (1/200)
- 30 meter range
- 50 Hz sampling rate

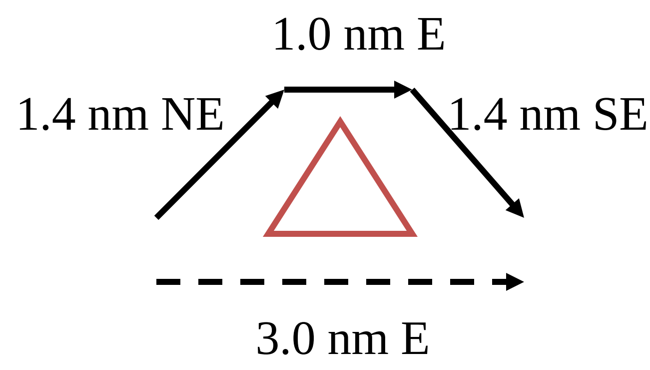


Outline

- 6.1 Mathematics of Pose Estimation
  - 6.1.1 Pose Fixing versus Dead Reckoning
  - 6.1.2 Pose Fixing
  - 6.1.3 Error Propagation in Triangulation
  - 6.1.4 Real Pose Fixing Systems
  - 6.1.5 Dead Reckoning
  - 6.1.6 Real Dead Reckoning Systems
  - Summary

6.1.5 Dead Reckoning

- Roots in ancient marine course & speed “chart”, when mariners first strayed from sight of land.
- Governed by mathematics of quadrature (basic integration - as distinct from solving DEs).
- Can integrate differential position, or velocity or acceleration.
- Can integrate differential angles or angular velocity to get attitude and/or heading.



$$\vec{r} = \vec{r}(0) + \int_0^t d\vec{r} = \vec{r}(0) + \int_0^t \vec{v}\,dt$$

$$\vec{r} = \vec{r}(0) + \int_0^t \left[\vec{v}(0) + \int_0^t \vec{a}\,dt \right] dt$$

$$\vec{r} = \vec{r}(0) + \vec{v}(0)t + \int_0^t \int_0^t \vec{a}\,dt\,dt$$

General Case

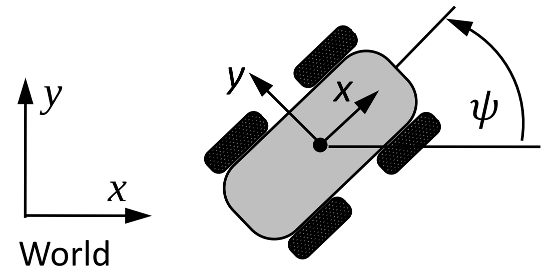
- Suppose a state x(t) depends on some inputs u(t) and parameters p:

$$\dot{\underline{x}}(t) = \underline{f}[\underline{x}(t), \underline{u}(t), \underline{p}]$$

- We might also have some measurements:

$$\underline{z}(t) = h[\underline{x}(t), \underline{u}(t), \underline{p}]$$

- Three such cases follow:



Parameters involve kinematic mapping from inputs onto state rates. Think wheelbase, wheelradius, etc.

6.1.5.1.1 Direct Heading

- Have a compass.

$$\frac{d}{dt}\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} v(t)\cos\psi(t) \\ v(t)\sin\psi(t) \end{bmatrix}$$

- State and inputs:

$$\underline{x}(t) = \begin{bmatrix} x(t) & y(t) \end{bmatrix}^T$$

$$\underline{u}(t) = \begin{bmatrix} v(t) & \psi(t) \end{bmatrix}^T$$

You can make ψ a state if you like, but its more work for no gain.

- Observer is trivial:

$$\underline{z}(t) = \underline{u}(t) = \begin{bmatrix} V(t) & \theta(t) \end{bmatrix}^T$$

6.1.5.1.2 Integrated Heading

- Have a heading state.

$$\frac{d}{dt}\begin{bmatrix} x(t) \\ y(t) \\ \psi(t) \end{bmatrix} = \begin{bmatrix} v(t)\cos\psi(t) \\ v(t)\sin\psi(t) \\ \omega(t) \end{bmatrix}$$

- State and inputs:

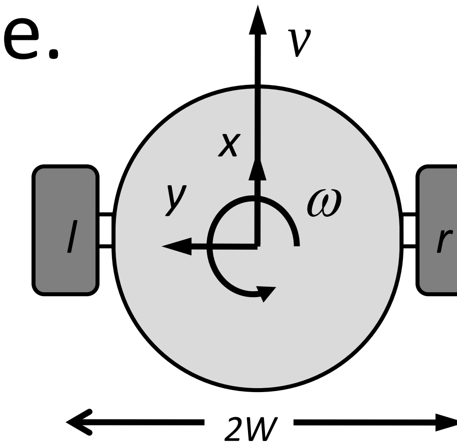
$$\underline{x}(t) = \left[x(t)\ y(t)\ \psi(t)\right]^T$$

$$\underline{u}(t) = \left[v(t)\ \omega(t)\right]^T$$

- Observer is trivial again.

6.1.5.1.3 Differential Heading

- Two wheel rates determine yaw rate.
- Same dynamics



$$\frac{d}{dt}\begin{bmatrix} x(t) \\ y(t) \\ \psi(t) \end{bmatrix} = \begin{bmatrix} v(t)\cos\psi(t) \\ v(t)\sin\psi(t) \\ \omega(t) \end{bmatrix}$$

- State and inputs:

$$\underline{x}(t) = \begin{bmatrix} x(t) & y(t) & \psi(t) \end{bmatrix}^T$$

$$\underline{u}(t) = \begin{bmatrix} v(t) & \omega(t) \end{bmatrix}^T$$

- Observer is not trivial:

$$\begin{bmatrix} v_r(t) \\ v_l(t) \end{bmatrix} = \begin{bmatrix} 1 & W \\ 1 & -W \end{bmatrix}\begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix} \quad \text{or} \quad \underline{z}(t) = H\underline{u}(t)$$

Recall: Solution Integrals

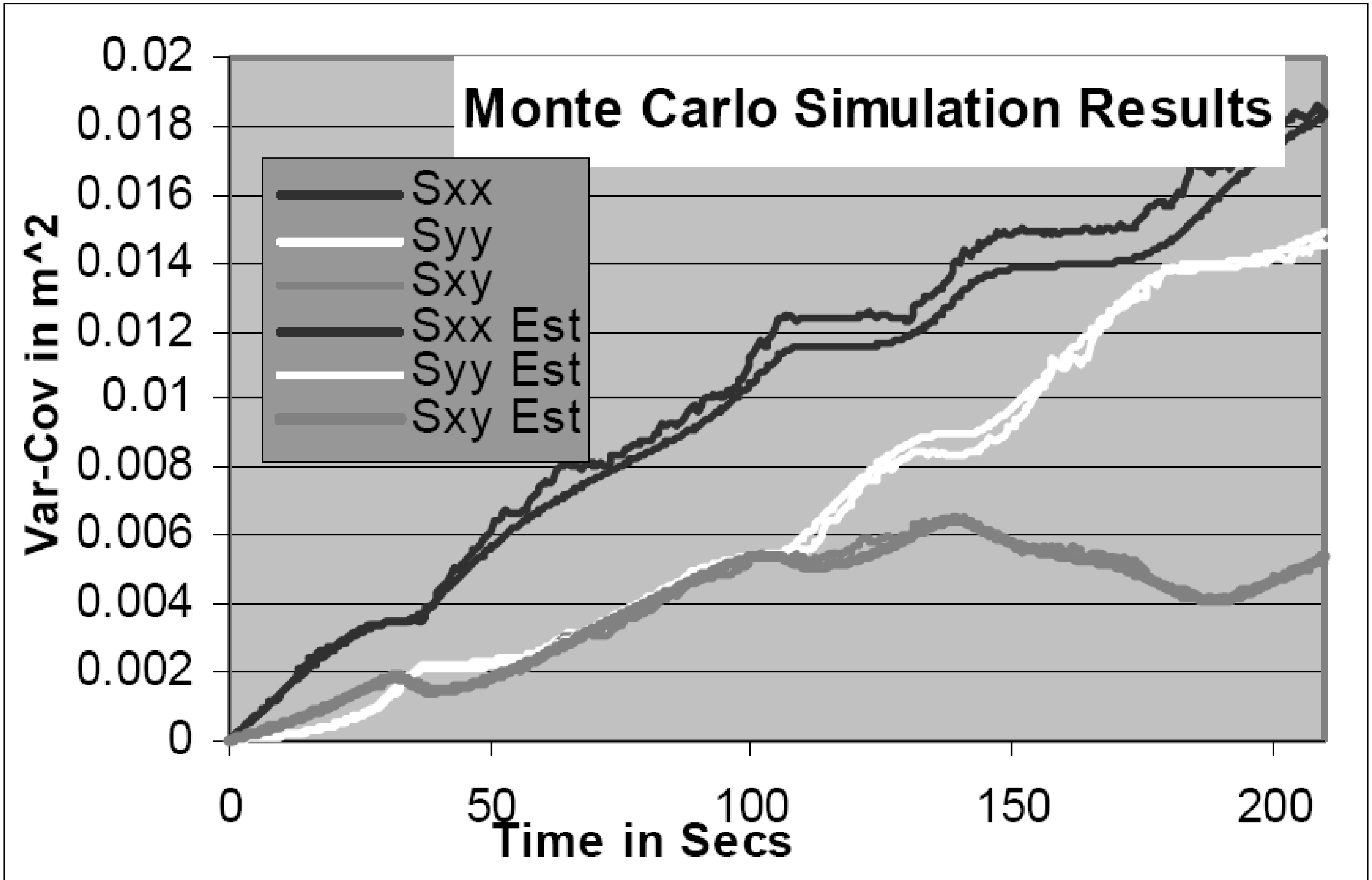
- Systematic Error:

$$\delta \underline{x}(t) = \Phi(t, t_0)\delta \underline{x}(t_0) + \int_{t_0}^{t} \Gamma(t, \tau)\delta \underline{u}(\tau) d\tau$$

- Random Error:

$$P(t) = \Phi(t, t_0)P(t_0)\Phi^T(t, t_0) + \int_{t_0}^{t} \Gamma(t, \tau)Q(\tau)\Gamma^T(t, \tau) d\tau$$

Validation



Monte Carlo Simulation Results
Sxx
Syy
Sxy
Sxx Est
Syy Est
Sxy Est
Var-Cov in m^2
0.02
0.018
0.016
0.014
0.012
0.01
0.008
0.006
0.004
0.002
0
0
50
100
150
200
Time in Secs

6.1.5.2.2 Input Transition Matrix

- The product of the transition matrix and the input Jacobian is:

$$\Gamma(t, \tau) = \Phi(t, \tau)G(\tau)$$
systematic

$$\Gamma(t, \tau) = \Phi(t, \tau)L(\tau)$$
stochastic

G and L are two different conventional names for same matrix in the system dynamics.

- Governs propagation of both systematic and random error in odometry.
- Integrals of …
  – a) its columns and of
  – b) outer products of its columns
- … are the canonical error propagation modes.

6.1.5.2.3 Moments of Error (Systematic)

- The vector convolution integral can be written:

$$\delta \underline{x}(t) = \Phi(t, t_0)\delta \underline{x}(t_0) + \sum_i \left[\int_{t_0}^{t} \underline{\gamma}_i \delta u_i d\tau \right]$$

ith column

Individual error sources

- Hence, the error in pose:
  - is the sum of the contributions of each input error source.
  - where each contribution is an integral or moment which depends on the trajectory followed.
- Integrals mean errors are generally path dependent.

6.1.5.2.3 Moments of Error (Stochastic)

- The matrix convolution integral can be written:

$$P(t) = \Phi(t, t_0)P(t_0)\Phi^T(t, t_0) + \sum_i \sum_j \left[\int_{t_0}^{t} (\underline{\gamma}_i \underline{\gamma}_j^T q_{ij}) d\tau \right]$$

Outer product of two columns.

Individual error covariances

- The same sum of moments interpretation applies but now the moments are matrix-valued.

6.1.5.3 Integrated Heading Odometry (Linearized Dynamics)

- Dynamics are:

$$\frac{d}{dt}\begin{bmatrix} x(t) \\ y(t) \\ \theta(t) \end{bmatrix} = \begin{bmatrix} V(t)\cos\theta(t) \\ V(t)\sin\theta(t) \\ \omega(t) \end{bmatrix}$$

Change θ to ψ Everywhere on This slide

- System and Input Jacobians:

$$F(t) = \begin{bmatrix} 0 & 0 & -Vs\theta \\ 0 & 0 & Vc\theta \\ 0 & 0 & 0 \end{bmatrix} \qquad G(t) = \begin{bmatrix} c\theta(t) & 0 \\ s\theta(t) & 0 \\ 0 & 1 \end{bmatrix}$$

- Hence, linearized dynamics are:

$$\frac{d}{dt}\begin{bmatrix} \delta x(t) \\ \delta y(t) \\ \delta\theta(t) \end{bmatrix} = \begin{bmatrix} 0 & 0 & -Vs\theta \\ 0 & 0 & Vc\theta \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} \delta x(t) \\ \delta y(t) \\ \delta\theta(t) \end{bmatrix} + \begin{bmatrix} c\theta(t) & 0 \\ s\theta(t) & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} \delta V(t) \\ \delta\omega(t) \end{bmatrix}$$

6.1.5.3 Integrated Heading Odometry

(Transition Matrix)

- If you can find a matrix Ψ such that:

$$\Psi(t,\tau)F(t) = F(t)\Psi(t,\tau)$$

- Then Ψ is the transition matrix!
- A good candidate is:

$$\Psi(t,\tau) = \exp\left(\int_{\tau}^{t} F(\zeta)d\zeta\right) = \exp[R(t,\tau)]$$

Matrix R is defined here.

6.1.5.3 Integrated Heading Odometry (Transition Matrix)

- This case will satisfy: $\Psi(t,\tau)F(t) = F(t)\Psi(t,\tau)$
- So $\Psi(t,\tau)$ is the transition matrix.
- To get it, form:

$$R(t,\tau) = \int_{\tau}^{t} \begin{bmatrix} 0 & 0 & -Vs\theta \\ 0 & 0 & Vc\theta \\ 0 & 0 & 0 \end{bmatrix} d\zeta = \begin{bmatrix} 0 & 0 & -\Delta y(t,\tau) \\ 0 & 0 & \Delta x(t,\tau) \\ 0 & 0 & 0 \end{bmatrix}$$

Change θ to ψ

- Where:

$$\Delta x(t,\tau) = [x(t) - x(\tau)]$$

$$\Delta y(t,\tau) = [y(t) - y(\tau)]$$

- The transition matrix is then:

$$\Psi(t,\tau) = \exp[R(t,\tau)] = I + R = \begin{bmatrix} 1 & 0 & -\Delta y(t,\tau) \\ 0 & 1 & \Delta x(t,\tau) \\ 0 & 0 & 1 \end{bmatrix}$$

6.1.5.3 Integrated Heading Odometry (Systematic Error Result)

- Hence, the input transition matrix is:

$$\Gamma(t,\tau) = \Phi(t,\tau)G(\tau) = \begin{bmatrix} c\psi(t) & -s\psi(t) & -\Delta y(t,\tau) \\ s\psi(t) & c\psi(t) & \Delta x(t,\tau) \\ 0 & 0 & 1 \end{bmatrix}$$

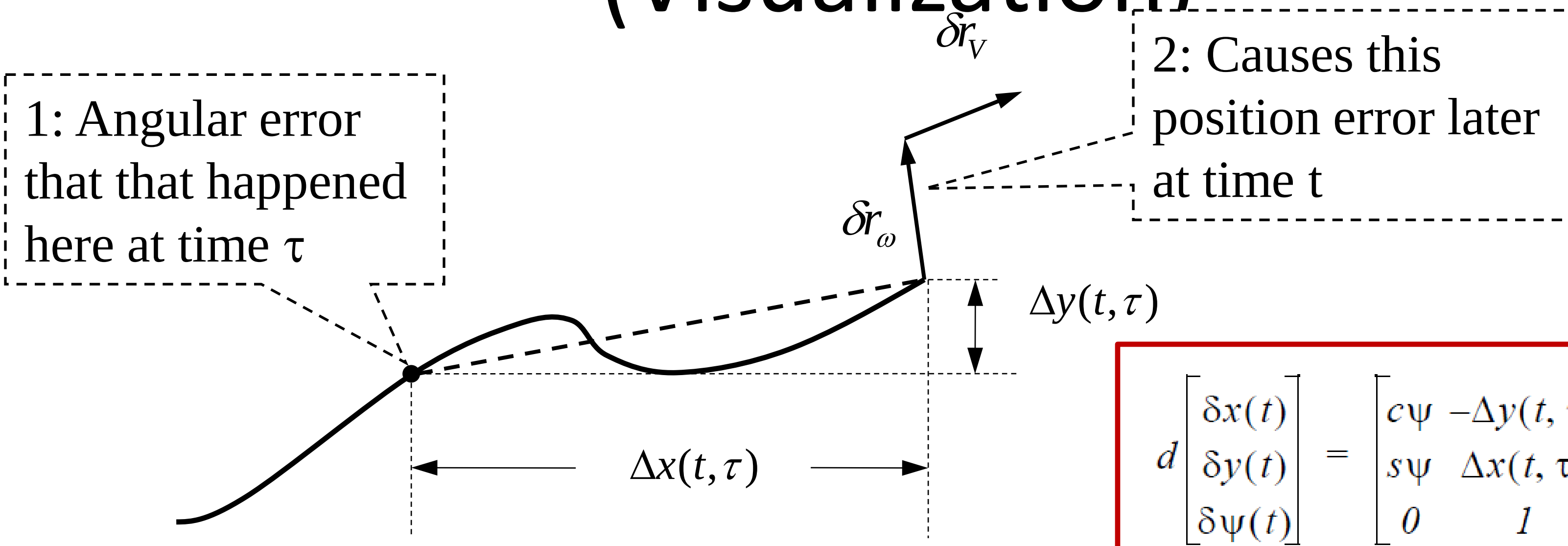
- So, the general solution for systematic error propagation in integrated heading odometry is:

$$\delta \underline{x}(t) = \begin{bmatrix} 1 & 0 & -y(t) \\ 0 & 1 & x(t) \\ 0 & 0 & 1 \end{bmatrix} \delta \underline{x}(0) + \int_0^t \begin{bmatrix} c\psi & -\Delta y(t,\tau) \\ s\psi & \Delta x(t,\tau) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \delta v(\tau) \\ \delta\omega(\tau) \end{bmatrix} d\tau$$

Projection of δV onto world x, y axes

Moment arm multiplies δω.

6.1.5.3 Integrated Heading Odometry (Visualization)



$$d\begin{bmatrix}\delta x(t)\\ \delta y(t)\\ \delta\psi(t)\end{bmatrix} = \begin{bmatrix}c\psi & -\Delta y(t,\tau)\\ s\psi & \Delta x(t,\tau)\\ 0 & 1\end{bmatrix}\begin{bmatrix}\delta v(\tau)\\ \delta\omega(\tau)\end{bmatrix}d\tau$$

- Solution simply adds up the impact of every historical error to produce the present error.
- Dead reckoning never forgets an error.

6.1.5.3 Integrated Heading Odometry (Moment Form)

- In moment form, the solution is:

$$\delta \underline{x}(t) = \underline{IC}_d + \int_0^t \underline{\gamma}_v(\tau)\delta v d\tau + \int_0^t \underline{\gamma}_\omega(\tau)\delta\omega d\tau$$

Contribution of δV to state error

Contribution of δω to state error.

- Where:

$$\underline{IC}_d = \begin{bmatrix} 1 & 0 & -y(t) \\ 0 & 1 & x(t) \\ 0 & 0 & 1 \end{bmatrix} \delta \underline{x}(0)$$

$$\underline{\gamma}_v(\tau) = \begin{bmatrix} c\psi & s\psi & 0 \end{bmatrix}^T$$

$$\underline{\gamma}_\omega(\tau) = \begin{bmatrix} -\Delta y(t,\tau) & \Delta x(t,\tau) & 1 \end{bmatrix}^T$$

6.1.5.3 Integrated Heading Odometry (Stochastic)

- Likewise, the general solution for stochastic error is:

$$P(t) = IC_s + \int_0^t \begin{bmatrix} c\psi & -\Delta y(t,\tau) \\ s\psi & \Delta x(t,\tau) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \sigma_{vv} & \sigma_{v\omega} \\ \sigma_{v\omega} & \sigma_{\omega\omega} \end{bmatrix} \begin{bmatrix} c\psi & -\Delta y(t,\tau) \\ s\psi & \Delta x(t,\tau) \\ 0 & 1 \end{bmatrix}^T d\tau$$

- Where:

$$IC_s = \begin{bmatrix} 1 & 0 & -y(t) \\ 0 & 1 & x(t) \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \sigma_{xx}(0) & \sigma_{xy}(0) & \sigma_{x\theta}(0) \\ \sigma_{xy}(0) & \sigma_{yy}(0) & \sigma_{y\theta}(0) \\ \sigma_{x\theta}(0) & \sigma_{y\theta}(0) & \sigma_{\theta\theta}(0) \end{bmatrix} \begin{bmatrix} 1 & 0 & -y(t) \\ 0 & 1 & x(t) \\ 0 & 0 & 1 \end{bmatrix}^T$$

Change ψ to θ

- Or:

$$P(t) = IC_s + \int_0^t [\Gamma_{v\omega}(\tau) + \Gamma_{\omega v}(\tau)]\sigma_{v\omega} d\tau + \int_0^t \Gamma_{vv}(\tau)\sigma_{vv} d\tau + \int_0^t \Gamma_{\omega\omega}(\tau)\sigma_{\omega\omega} d\tau$$

6.1.5.3 Integrated Heading Odometry (Stochastic)

- Where:

$$\Gamma_{vv}(\tau) = \begin{bmatrix} c\psi \\ s\psi \\ 0 \end{bmatrix} \begin{bmatrix} c\psi \\ s\psi \\ 0 \end{bmatrix}^T = \begin{bmatrix} c^2\psi & c\psi s\psi & 0 \\ c\psi s\psi & s^2\psi & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\Gamma_{v\omega}(\tau) = \Gamma^T_{\omega v}(\tau) = \begin{bmatrix} c\psi \\ s\psi \\ 0 \end{bmatrix} \begin{bmatrix} -\Delta y \\ \Delta x \\ 1 \end{bmatrix}^T = \begin{bmatrix} -c\psi\Delta y & c\psi\Delta x & c\psi \\ -s\psi\Delta y & s\psi\Delta x & s\psi \\ 0 & 0 & 0 \end{bmatrix}$$

$$\Gamma_{\omega\omega}(\tau) = \begin{bmatrix} -\Delta y \\ \Delta x \\ 1 \end{bmatrix} \begin{bmatrix} -\Delta y \\ \Delta x \\ 1 \end{bmatrix}^T = \begin{bmatrix} \Delta y^2 & -\Delta x\Delta y & -\Delta y \\ -\Delta x\Delta y & \Delta x^2 & \Delta x \\ -\Delta y & \Delta x & 1 \end{bmatrix}$$

6.1.5.3.2 Error Models

- Getting specific results requires:
  - Specific assumed input errors
  - Specific (reference) trajectories
- For systematic error, assume:

$$\delta V = \delta V_v \times V$$
$$\delta\omega = \text{const}$$

Notation

$$\delta v_v = \frac{\partial}{\partial v}(\delta v)$$

- For random error, assume:

$$\sigma_{vv} = \sigma_{vv}^{(v)}|V|$$

Distance dependent random walk

$$\sigma_{\omega\omega} = \text{const} \qquad \sigma_{v\omega} = 0$$

6.1.5.3.2 Error Models

- Now the general solution on any trajectory is:

$$\delta \underline{x}(t) = \underline{IC}_d + \delta v_v \int_0^s \begin{bmatrix} c\psi \\ s\psi \\ 0 \end{bmatrix} ds + \delta\omega \int_0^t \begin{bmatrix} -\Delta y \\ \Delta x \\ 1 \end{bmatrix} d\tau$$

$$P(t) = IC_s + \sigma_{vv}^{(v)} \int_0^s \begin{bmatrix} c^2\psi & c\psi s\psi & 0 \\ c\psi s\psi & s^2\psi & 0 \\ 0 & 0 & 0 \end{bmatrix} ds + \sigma_{\omega\omega} \int_0^t \begin{bmatrix} \Delta y^2 & -\Delta x \Delta y & -\Delta y \\ -\Delta x \Delta y & \Delta x^2 & \Delta x \\ -\Delta y & \Delta x & 1 \end{bmatrix} d\tau$$

6.1.5.3.2 Error Models

Trajectories

- Finally, must select trajectories because everything is path dependent.
- For a straight line: $\omega(t) = 0 \quad V(t) = \text{arbitrary}$
- Trajectory: $x(t) = s(t) \; y(t) = 0 \quad \theta(t) = 0$
- Systematic error:

Linear in distance

Not linear in distance

$$\delta\underline{x}(t) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & s \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \delta x(0) \\ \delta y(0) \\ \delta\theta(0) \end{bmatrix} + \delta V_v \begin{bmatrix} s \\ 0 \\ 0 \end{bmatrix} + \delta\omega \begin{bmatrix} 0 \\ st/2 \\ t \end{bmatrix}$$

- Random error:

$$P(t) = IC_s + \sigma_{vv}^{(v)} \begin{bmatrix} s & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \sigma_{\omega\omega} \begin{bmatrix} 0 & 0 & 0 \\ 0 & (s^2 t)/3 & (st)/2 \\ 0 & (st)/2 & t \end{bmatrix}$$

6.1.5.3.2 Integrated Heading

Arbitrary Trajectory

Box 6.5: Error Propagation in Integrated Heading Odometry

Integrated heading odometry is defined as the process of dead reckoning position and orientation from measurements of linear and angular velocity. For such a system, if velocity measurements have a scale error and angular velocity measurements have a bias, errors propagate as follows.

$$\delta \underline{r}(t) = \delta v_v \underline{r}(t)$$

The effect of velocity scale error is proportional to the position vector and is independent of path shape.

$$\delta\psi(t) = \delta\omega t$$

Heading error due to gyro bias grows linearly in time.

$$\delta \underline{r}(t) = \delta \underline{\omega} t \times [\underline{r}(t) - \bar{\rho}(t)]$$

Position error caused by gyro bias is proportional to time and radius from the dwell centroid, directed normally.

If velocity measurements contain random errors proportional to distance and gyro measurements contain random noise, errors propagate as follows:

$$\sigma_{rr} = \sigma_{vv}^{(v)} s$$

Total position covariance is proportional to distance travelled and is independent of path shape.

$$\sigma_{\psi\psi} = \sigma_{\omega\omega} t$$

Heading variance due to gyro noise grows linearly in time.

$$\sigma_{rr} = \sigma_{\omega\omega} t [r_{\bar{\rho}}^{2}(t) + \rho_{\bar{\rho}}^{2}(t)]$$

Position variance caused by gyro noise is proportional to time and both instantaneous and average squared radius from the dwell centroid.

Notes

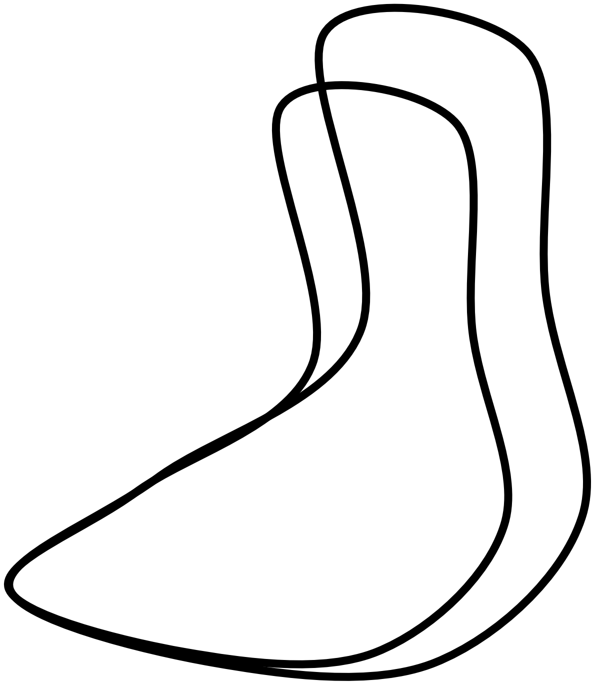
- Maybe next year add the derivation of box 6.5 and perhaps delete the following content on insights.

6.1.5.3.3 Insights

Path Independence

- Errors which propagate with the first Fourier Excursion Moment (e.g. velocity scale errors) vanish on closed trajectories.

$$S_c = \int_0^s \cos\theta \, ds = x(s)$$



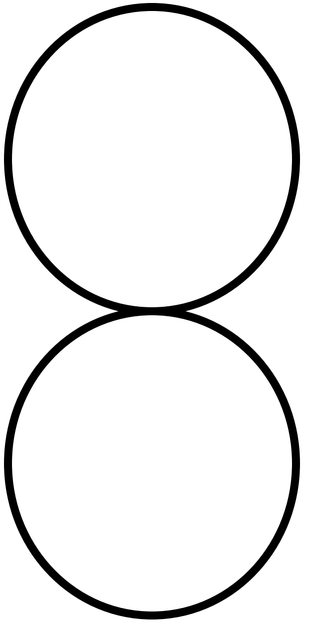
- The wrong way to check your encoder scale factor !!!

6.1.5.3.3 Insights

Symmetry

- Errors which propagate with the first Spatial Excursion Moment (e.g. gyro bias) vanish at the centroid of the trajectory.

$$S_x = sx(s) - \int_0^s x(\xi)d\xi = s[x(s) - \bar{x}(s)]$$



- The wrong way to check your gyro bias

6.1.5.3.3 Insights

Monotonicity

- Many stochastic error behaviors are monotone.

$$\frac{d}{ds}(S_{cc}) = \frac{d}{ds}\left(\int_0^s [\cos\theta^2]ds\right) = \cos\theta^2 \geq 0$$

- However, some (gyro bias effects) are NOT!!.

$$T_{xx} = \int_0^t \Delta x^2 d\tau = \int_0^t [x(t) - x(\tau)]^2 d\tau$$

<Is not monotone for an excursion to (1,0) and back>

6.1.5.3.3 Insights

Further Insights

- Superposition:
  - Response to input errors is always the (path dependent) sum of one moment for each error source.
- Path Independence:
  - Response to initial conditions (initial pose errors) is always path independent.

Outline

- 6.1 Mathematics of Pose Estimation
  - 6.1.1 Pose Fixing versus Dead Reckoning
  - 6.1.2 Pose Fixing
  - 6.1.3 Error Propagation in Triangulation
  - 6.1.4 Real Pose Fixing Systems
  - 6.1.5 Dead Reckoning
  - 6.1.6 Real Dead Reckoning Systems - Skip
  - Summary

Outline

- 6.1 Mathematics of Pose Estimation
  - 6.1.1 Pose Fixing versus Dead Reckoning
  - 6.1.2 Pose Fixing
  - 6.1.3 Error Propagation in Triangulation
  - 6.1.4 Real Pose Fixing Systems
  - 6.1.5 Dead Reckoning
  - 6.1.6 Real Dead Reckoning Systems
  - Summary

Summary

- Dead reckoning and triangulation behave very differently.
- Linearization provides the basic mapping between systematic and random input and output error.
- The Geometric Dilution of Precision is often illuminating and easily computed.
- Different forms of triangulation have different sensitivity behaviors.

Summary

- Odometry uses integration to generate pose.
- Errors in odometry propagate according to integrals. If we linearize (perturb) the equations, a general solution can be found.
- Error propagation can be reduced to computing moments of arc on the trajectory.
- Many unusual error behaviors result from the dynamic behavior of odometry.
  - They include path independence, response to symmetric inputs, reversibility, monotonicity, etc.