1 Triangulation vs Dead Reckoning

• There are ultimately only two alternative ways to estimate position:
  • triangulation (aka. “fixing” position)
  • deduced reckoning (aka. “dead” or “ded.” reckoning)

• In triangulation, navigation variables are related to observables by algebraic/
transcendental equations. Hence, we solve nonlinear systems in order to navigate.

- In **dead reckoning**, navigation variables are related to observables by differential equations. Hence, integrate in order to navigate.

- In both cases, relationships must be kinematic (i.e. involving only linear and angular displacements and their derivatives, not mass, force, etc.).

### 1.1 General Points

- Navigation variables can be linear or angular position or any of their derivatives.
- e.g. can do Doppler triangulation of velocity.
- e.g. can do ded. reckoning of velocity from acceleration.
- e.g. can integrate gyro output to get angle.
- Ultimately need N simultaneous constraint equations of any form relating observables to variables of interest.

### 1.2 Differences Between Both Techniques

- DR and Triangulation can complement each other:

**Table 1: Differences Between Techniques**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Dead Reckoning</th>
<th>Triangulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Process</td>
<td>Integration</td>
<td>Nonlinear Systems</td>
</tr>
<tr>
<td>Initial Conditions</td>
<td>Required</td>
<td>Not required</td>
</tr>
<tr>
<td>Errors</td>
<td>(Often) Time Dependent</td>
<td>Position Dependent</td>
</tr>
<tr>
<td>Update Frequency</td>
<td>Determined by Required Accuracy</td>
<td>Determined only by the update requirements</td>
</tr>
<tr>
<td>Error Propagation</td>
<td>Dependent on Previous Estimates</td>
<td>Independent of Previous Estimates</td>
</tr>
<tr>
<td>Requires a map.</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
2 Triangulation

- Historical roots in survey and cartography of Egyptians and earlier civilizations, even Sumarians. Development driven by same needs that drove the invention of writing and arithmetic:
  - need for sound building construction
  - need for accurate records of land holdings
  - need for accurate records of business transactions
- Also used in marine “pilotage” where shore landmarks are used to guide the vessel.

2.1 Solving Triangles

- Favorite trick of the ancients was the 3-4-5 triangle for right angles

- Some solvable triangles:

```
SSS
b c a

SAS
b c a

ASA
b c a
```

- All 3 parameter triangles but AAA are solvable, provided the parameters are consistent. (ASS and SSA have existence and uniqueness issues respectively)

2.2 Constrained Systems

- There is not always a triangle, but name “triangulation” has endured.
- General idea is solution of simultaneous algebraic constraint equations in which navigation variables appear as unknowns. The robot position is a point where all constraints are satisfied.
- Observables are usually “bearings” or ranges to landmarks
2.3 Nonlinearly Constrained Systems

- No existence or uniqueness theorems
- Cannot use approximations in most navigation situations.
- Analogy to kinematics of manipulators.
- Many concerns:
  - (in)consistency of equations (no solution)
  - (in)dependence of equations (poor conditioning)
  - (under/over)constraint (too many/too little)
  - singularity (poor conditioning)
  - redundancy (several solutions)
- 2D illustrates most problems that occur
- 2D answers are obvious from geometry, but you need math, not geometry in higher dimensions, so you get the math too.

2.3.1 Explicit Case

- Suppose we have the minimum number of measurements required to determine 2D pose - three.
- Sometimes, (rarely) we can write an explicit formula to determine the state $x = \begin{bmatrix} x \ y \ \theta \end{bmatrix}$ from the measurements $z = \begin{bmatrix} z_1 \ z_2 \ z_3 \end{bmatrix}$:
  
  \[ x = g_1(z_1, z_2, z_3) \]
  \[ y = g_2(z_1, z_2, z_3) \]
  \[ \theta = g_3(z_1, z_2, z_3) \]

- Which is of the form:
  
  \[ x = g(z) \]
2.3.2 Implicit Case

- The implicit case is the (more common, more general) inverse situation:

\[
\begin{align*}
  z_1 &= h_1(x, y, \theta) \\
  z_2 &= h_2(x, y, \theta) \\
  z_3 &= h_3(x, y, \theta)
\end{align*}
\]

- In the Kalman filter, this is just the measurement equation:

\[
\bar{z} = h(\bar{x})
\]

- We can solve this with iteration, gradient descent, least squares, etc. For example, we can linearize:

\[
\Delta z = H \Delta x
\]

and solve iteratively with the pseudoinverse:

\[
\begin{align*}
  \Delta x &= (H^T H)^{-1} H^T \Delta z & \text{overdetermined} \\
  \Delta x &= H^T (HH^T)^{-1} \Delta z & \text{underdetermined}
\end{align*}
\]

- The underdetermined case does not arise often in explicit triangulation - but it is the rule in the Kalman Filter.
- The weighted case is more general again. Often, the measurement covariance \( R \) is used as the weight matrix.

2.4 Example 1: Linear Constraints With Known Heading

- Example arises when vehicle measures bearings of landmarks and has sensor for measuring vehicle heading \( \theta_v \)
- Commercial systems based on
  - collimated laser beam and retroreflectors
  - IR flash and retroreflective tape
  - plane of laser light and detector
- Heading constrains orientation, bearings constrain position
• By adding the observed bearings of the landmarks to the heading, the landmark heading angles $\theta_1$ and $\theta_2$ are determined.

• Constraint equations are:

\[
\begin{align*}
\tan \theta_1 &= \frac{\sin \theta_1}{\cos \theta_1} = \frac{y_1 - y}{x_1 - x} \\
\tan \theta_2 &= \frac{\sin \theta_2}{\cos \theta_2} = \frac{y_2 - y}{x_2 - x}
\end{align*}
\]

• These are two (basically) linear simultaneous equations

\[
\begin{align*}
(x_1 - x)s_1 &= (y_1 - y)c_1 \\
(x_2 - x)s_2 &= (y_2 - y)c_2
\end{align*}
\]

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

\[
\begin{bmatrix}
-s_1 & c_1 \\
-s_2 & c_2
\end{bmatrix}
\begin{bmatrix}
x \\
y
\end{bmatrix} =
\begin{bmatrix}
b_1 \\
b_2
\end{bmatrix}
\]

• Which always has a solution except when

\[
-s_1 c_2 + s_2 c_1 = 0 \quad \text{or} \quad \theta = n\pi
\]

• This is an example of degeneracy (no solution because constraints are not independent).

• There is always a line between two landmarks so

• Don’t operate near the line
• Position landmarks appropriately outside excursion
• Use third landmark
• Note: you don’t need to be between landmarks for this to happen.
2.5 Example 2: Linear Constraints With Unknown Heading

- No heading sensor, use third landmark
- 3 bearings

\[
\begin{align*}
\tan(\theta + \theta_1) &= \frac{\sin(\theta + \theta_1)}{\cos(\theta + \theta_1)} = \frac{y_1 - y}{x_1 - x} \\
\tan(\theta + \theta_2) &= \frac{\sin(\theta + \theta_2)}{\cos(\theta + \theta_2)} = \frac{y_2 - y}{x_2 - x} \\
\tan(\theta + \theta_3) &= \frac{\sin(\theta + \theta_3)}{\cos(\theta + \theta_3)} = \frac{y_3 - y}{x_3 - x}
\end{align*}
\]

2.6 Example 3: Circular Constraints

- Arise when landmark ranges are observables
- Cannot triangulate heading unless
  - a) measure bearings, or
  - b) measure ranges to two points on vehicle

- Degeneracy not an issue for unique landmarks.
- **Inconsistency** (no solution) is an issue
- **Redundancy** (multiple solutions) is an issue.
• Can use a rough estimate to actively eliminate wrong answer.
• Can use rough estimate and iterate solution. Convergence to closest solution will occur.

• Constraints are:

\[ r_1 = \sqrt{(x-x_1)^2 + (y-y_1)^2} \]
\[ r_2 = \sqrt{(x-x_2)^2 + (y-y_2)^2} \]

• Solve using favorite nonlinear equation solver
• Easier when vehicle is moving because last position is good starting point for the next iteration

2.7 Example 4: Hyperbolic Constraints

• Receivers measure range differences directly
• Basis of marine radionavigation (Loran, Omega) for many decades.
• Measure time of flight difference or phase difference

• Contours of constant range difference are hyperbolae:

\[ r_1 - r_2 = 2b \]  \hspace{1cm} (Property of hyperbolae)
\[ \sqrt{(x + a)^2 + y^2} - \sqrt{(x - a)^2 + y^2} = 2b \]
\[ \sqrt{(x + a)^2 + y^2} = 2b + \sqrt{(x - a)^2 + y^2} \]
\[ b^2 - ax = -b \sqrt{(x - a)^2 + y^2} \]
\[ \frac{x^2}{a^2} - \frac{y^2}{b^2} = 1 \]
3 Error Propagation in Triangulation

3.1 Response to Systematic Measurement Error - Direct Case

• The response to systematic error involves the same mathematics used to solve the nonlinear problem by linearization.

• We can investigate the response to perturbations (solution error caused by measurement error) with our friend the Jacobian matrix.

\[
\delta x = \left( \frac{\partial x}{\partial z} \right) \delta z = J \delta z
\]

• In the fully determined explicit case, that looks like this:

\[
\begin{bmatrix}
\delta x \\
\delta y \\
\delta \theta
\end{bmatrix} =
\begin{bmatrix}
\frac{\partial f_1}{\partial z_1} & \frac{\partial f_1}{\partial z_2} & \frac{\partial f_1}{\partial z_3} \\
\frac{\partial f_2}{\partial z_1} & \frac{\partial f_2}{\partial z_2} & \frac{\partial f_2}{\partial z_3} \\
\frac{\partial f_3}{\partial z_1} & \frac{\partial f_3}{\partial z_2} & \frac{\partial f_3}{\partial z_3}
\end{bmatrix}
\begin{bmatrix}
\delta z_1 \\
\delta z_2 \\
\delta z_3
\end{bmatrix}
\]

• Interpretations of the Jacobian Matrix:
  • matrix equation for the total differential
  • truncated multidimensional Taylor series
  • sensitivity coefficients
  • multidimensional first derivative
  • first order approximate behavior

• First order behavior is always linear in measurements by definition.

• Usually J is a function of vehicle position.

• Analogy to differential kinematics of manipulators.
3.2 Response to Systematic Measurement Error - Indirect Case

- In the indirect case, we have:\n\[ \delta z = H \delta x \]

- Then, if the measurements determine or overdetermine the state, the best fit error in the state is given by the left pseudoinverse:\n\[ \delta x = (H^T H)^{-1} H^T \delta z \]

3.3 Geometric Dilution of Precision

- If you want one number which gives a sense for error magnitude, it would be the volume or area of the region encoded in \( \delta x \).
- If you want one number which characterizes whether the geometry is favorable, a good number is the geometric dilution of precision - the ratio of state error “magnitude” to measurement area “magnitude”:\n\[ GDOP = \frac{\| \delta x \|}{\| \delta z \|} \]

\[ \| \delta x \| = \left| \frac{\partial x}{\partial z} \right| \| \delta z \| = |J| \| \delta z \| \]

where \( \| \delta x \| = \delta x_1 \delta x_2 \delta x_3 \) and \( \| \delta z \| = \delta z_1 \delta z_2 \delta z_3 \).
- Thus, the precision of the “fix” \( \delta x \) is the precision of the measurements \( \delta z \) multiplied by this factor.

3.3.1 Mapping Theory (small piece of it)

- Ratio of areas (volumes) in the domain (input) and range (output) of a multidimensional mapping is given by the Jacobian determinant:
• Limit of GDOP is the (forward) Jacobian determinant:

\[
\lim_{\Delta z \to 0} \text{GDOP} = \frac{\|\delta x\|}{\|\delta z\|} = |J|
\]

3.3.2 Commentary

• Fix error depends on both measurement error and GDOP, so large GDOP is not necessarily bad. Good sensors can overcome poor conditioning in theory.
• The GDOP is in the range: \(0 < \text{GDOP} < \infty\).
• Infinity is not uncommon, so we must understand it.
• 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
• Can be smart about computing it using coordinate transforms
• Can also investigate with contour graphs

• Need not compute \(J\) explicitly, only \(|J|\) is required
• NOTE: fix error also depends on errors in landmark positions - not considered here.

3.3.3 Implicit GDOP

• This technique has its roots in the implicit function theorem of calculus. Often real constraint equations are implicit.
• Consider two nonlinear constraints on 4 variables

\[
F(x, y, z, w) = 0 \\
G(x, y, z, w) = 0
\]

• These define two implicit functions \(z(x, y)\) and \(w(x, y)\). Take total differentials

\[
F_x \delta x + F_y \delta y + F_z \delta z + F_w \delta w = 0 \\
G_x \delta x + G_y \delta y + G_z \delta z + G_w \delta w = 0
\]
• These are simultaneous linear equations for \( \delta z \) and \( \delta w \) in terms of \( \delta x \) and \( \delta y \):

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

• So:

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

• Using the rules for determinant of product and inverse:

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

• Alterately, we could solve in the other direction:

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

• To give:

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

• So, we can get the GDOP without even explicitly computing the total Jacobian.

### 3.4 Response to Random Error

• When the input errors are random, we can use our uncertainty transformation formulas to compute the covariance in the state. For the inverse case:

\[
\delta z = H \delta x
\]
• The measurement covariance is clearly:

\[ C_z = HC_xH^T \]

• And, this can be inverted to produce:

\[ C_x = (H^T C_z^{-1} H)^{-1} \]

### 3.5 Example 1: Linear Constraints With Known Heading

• Again the situation is:

\[
\begin{align*}
\landmark{1} & : (x_1, y_1) \\
\landmark{2} & : (x_2, y_2) \\
(x, y) & : \text{robot position}
\end{align*}
\]

• The constraint equations are:

\[
\begin{align*}
(x_1 - x)s_1 &= (y_1 - y)c_1 \\
(x_2 - x)s_2 &= (y_2 - y)c_2
\end{align*}
\]

• We can get the GDOP using our implicit formula for implicit functions \( x(\theta_1, \theta_2) \) and \( y(\theta_1, \theta_2) \):

\[
\begin{align*}
F(x, y, \theta_1, \theta_2) &= s_1(x_1 - x) - c_1(y_1 - y) = 0 \\
G(x, y, \theta_1, \theta_2) &= s_2(x_2 - x) - c_2(y_2 - y) = 0
\end{align*}
\]

• Write total differentials:

\[
\begin{align*}
F_x \delta x + F_y \delta y + F_{\theta_1} \delta \theta_1 + F_{\theta_2} \delta \theta_2 &= 0 \\
G_x \delta x + G_y \delta y + G_{\theta_1} \delta \theta_1 + G_{\theta_2} \delta \theta_2 &= 0
\end{align*}
\]

• Calculate Jacobian determinant:

\[
\left| J \right| = \begin{vmatrix}
F_{\theta_1} & F_{\theta_2} \\
G_{\theta_1} & G_{\theta_2}
\end{vmatrix}
= \begin{vmatrix}
F_x & F_y \\
G_x & G_y
\end{vmatrix}
\]

Let:

\[
\begin{align*}
\Delta x_1 &= (x_1 - x) \\
\Delta y_1 &= (y_1 - y)
\end{align*}
\]

\[
\left| J \right| = \begin{vmatrix}
c_1 \Delta x_1 + s_1 \Delta y_1 & -s_1 c_1 \\
0 & c_2 \Delta x_2 + s_2 \Delta y_2
\end{vmatrix}
\]

\[
= \begin{vmatrix}
c_1 & -s_1 c_1 \\
0 & c_2
\end{vmatrix}
\]
- Multiplied out, this is:

\[
|J| = \frac{[c_1 \Delta x_1 + s_1 \Delta y_1][c_2 \Delta x_2 + s_2 \Delta y_2]}{\sin(\theta_2 - \theta_1)}
\]

- Its easier to see what this means using a vector formulation:

\[
|J| = \frac{\hat{r}_1 \cdot \hat{r}_1}{|\hat{r}_1|} \cdot \frac{\hat{r}_2 \cdot \hat{r}_2}{|\hat{r}_2|} \cdot \frac{\hat{r}_1 \times \hat{r}_1}{|\hat{r}_2|}
\]

\[
|J| = \frac{r_1 r_2}{\sin(\theta)}
\]

- Conclusions:
  - GDOP infinite at points of degeneracy
  - GDOP grows with distance squared
  - GDOP grows as lines become parallel
  - both happen at large distance so it grows faster in this case.

- A contour diagram represents level curves of the constraints - in the 2D case where they are easy to draw. Contours are most useful visualizations when:
  - are equally spaced in both observables
  - pass through the vehicle position
Here is a summary of the contour diagram for this case:

- Lines near parallel - poor
- Lines near 90° - good
- Landmarks far away - poor

The size of the shaded region is an indication of the GDOP at its center.

A complete contour diagram:
### 3.6 Example 2: Linear Constraints With Unknown Heading

- Again, the situation is:

![Diagram of linear constraints with unknown heading]

\[
\begin{align*}
\tan(\theta + \theta_1) &= \frac{\sin(\theta + \theta_1)}{\cos(\theta + \theta_1)} = \frac{y_1 - y}{x_1 - x} \\
\tan(\theta + \theta_2) &= \frac{\sin(\theta + \theta_2)}{\cos(\theta + \theta_2)} = \frac{y_2 - y}{x_2 - x} \\
\tan(\theta + \theta_3) &= \frac{\sin(\theta + \theta_3)}{\cos(\theta + \theta_3)} = \frac{y_3 - y}{x_3 - x}
\end{align*}
\]

- Same conclusions
  - GDOP -> infinity as vehicle approaches either of the three lines between landmarks, inside or outside landmark triangle
  - GDOP increases with range

### 3.7 Example 3: Circular Constraints

- Again, the situation is:

![Diagram of circular constraints]

- Constraints are:

\[
\begin{align*}
r_1 &= \sqrt{(x - x_1)^2 + (y - y_1)^2} \\
r_2 &= \sqrt{(x - x_2)^2 + (y - y_2)^2}
\end{align*}
\]

- To get the GDOP, we use a good trick. We investigate the inverse Jacobian $H = J^{-1}$ instead of the Jacobian because the constraint equations are in indirect form.
• Take total differentials

\[
\delta r_1 = \frac{(x-x_1)}{r_1} \delta x + \frac{(y-y_1)}{r_1} \delta y \\
\delta r_2 = \frac{(x-x_2)}{r_2} \delta x + \frac{(y-y_2)}{r_2} \delta y
\]

• 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:

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

\[
|J^{-1}| = \frac{\hat{r}_1 \times \hat{r}_2}{||\hat{r}_1|| ||\hat{r}_2||} = \sin(\theta)
\]

\[
|J| = \frac{1}{\sin(\theta)}
\]

• No explicit variation with range, but there is an implicit one (\(\theta\) decreases with range).
• Again, singular on line between landmarks.
• The robot is “far away” when the angle is small which is at many many times the landmark separation distance.
• Complete contour diagram:
3.8 Example 4: Hyperbolic Constraints

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

4.1 Laser Retroreflector Triangulation

- Cegelek (UK), FMC, Denning (Mass)
- Does not give z, or heading
- 50 meter range
- 1 inch accuracy
- Requires line of sight
- 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)

• Bar coded retroreflectors permit easy identification
4.2 Radio Carrier Phase Triangulation

- ARC system. Use four (or six) stationary reference antennae

- 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 vanilla range triangulation
- Singular between antennae (as always)
- Repeatability 3 cm, accuracy 12 cm

- 100 Hz update
- 5 mile range (limited by FCC reg)

4.2.1 Principle of Operation

- Oscillator provides reference for phase measurement
- “Differential” phase removes time dependence (single difference)

\[
\begin{align*}
\text{radio wave:} & \quad I(\hat{r}, t) = I_0 \cos(\omega t + \kappa r) \\
\text{antenna signal:} & \quad v_a(t) = v_0 \cos(\omega t + \kappa r) \\
\text{internal oscillator:} & \quad v_o(t) = v_1 \cos(\omega t + \text{const}) \\
\text{phase difference:} & \quad \Delta \Phi(t) = \Phi_a - \Phi_o = \\
& \quad (\omega t + \kappa r) - (\omega t + \text{const}) = \kappa r - \text{const}
\end{align*}
\]
• Thus the phase difference observed at the stationary antenna is proportional to the range to the vehicle.
• Relative mode does not require synchronization of references or rover.
• Frequency stability is critical (else get \((w_1-w_2)t\) error)
• Frequency drift is disastrous

“Double differencing” removes frequency drift:

\[
\Delta \Phi(t_1) = \Phi_a(t_1) - \Phi_o(t_1)
\]
\[
\Delta \Phi(t_2) = \Phi_a(t_2) - \Phi_o(t_2)
\]

\[
\Delta \Phi(t_1) = (\omega_a t_1 + \kappa r_1) - (\omega_o t_1 + \text{const}) = (\omega_a - \omega_o)t_1 + \kappa r_1 - \text{const}
\]

\[
\Delta \Phi(t_2) = (\omega_a t_2 + \kappa r_2) - (\omega_o t_2 + \text{const}) = (\omega_a - \omega_o)t_2 + \kappa r_2 - \text{const}
\]

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

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

• So, hyperbolic navigation becomes feasible.
4.3 Video Triangulation

- Innovision Systems Reflex
- Designed for biological motion analysis
- 4 to 7 CCD cameras
  - Internal LED flashers
  - Specially sensitive to IR
- Tape patches attached to subject
  - Perfect 1 inch circles
  - IR retroreflectors
- Real time video processor determines centroids of patches
- Angular resolution of 0.005% of camera field of view
- 30 meter range

- 50 Hz sampling rate
5 Deduced 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).
• Error in new estimate incorporates errors from old estimates
• Numerical errors play a larger role
• Requires initial conditions
• Use a regular fix to “damp” errors (mariners used pilotage or celestial observations)
• Can integrate differential position, or velocity or acceleration.
• Can integrate differential angles or angular velocity to get attitude and/or heading.

\[
\theta = \theta(0) + \int_0^t d\theta = \theta(0) + \int_0^t \dot{\theta} dt
\]

\[
\hat{r} = \dot{r}(0) + \int_0^t d\hat{r}
\]

\[
= \dot{r}(0) + \int_0^t \dot{\hat{r}} dt
\]

\[
= \dot{r}(0) + \int_0^t \left[ \dot{\hat{r}}(0) + \int_0^t \ddot{\hat{r}} dt \right] dt
\]

\[
= \dot{r}(0) + \dot{\hat{r}}(0) t + \int_0^t \int_0^t \ddot{\hat{r}} dt dt
\]

• In practice, the continuous integrals become discrete sums.
5.1 Canonical Cases

- A few generic cases are useful to distinguish.
- For now, adopt the (nonholonomic) assumption that the vehicle moves in the direction in which it is pointed. Not all vehicles move this way.

We will model the moving vehicle in terms of a set of state equations for the state $x(t)$ which depend on some inputs $u(t)$ and parameters $p$. Thus, the general situation is:

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

- We also might not be able to directly measure the inputs - in which case some sort of observer equation generating measurements $z(t)$ would also apply:

$$z(t) = h[x(t), u(t), p]$$

- We will restrict our attention mostly to planar motion - because that is already difficult enough. [see Kalman Filter notes for 3D dead reckoning].
- Following are three important cases.

5.1.1 Direct Heading

- If you have a direct measurement of heading from, for example, a compass, the vehicle dynamics can be expressed as:

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

---

1. Heading is not considered a state here because it has no dynamics associated with it. You can make it a state along the equation $\frac{d}{dt} \theta(t) = 0$ but it serves little purpose to do so.
Where the state and input vectors are:

\[
\begin{align*}
\dot{x}(t) &= \begin{bmatrix} x(t) & y(t) \end{bmatrix}^T \\
u(t) &= \begin{bmatrix} V(t) & \theta(t) \end{bmatrix}^T
\end{align*}
\]

and the observer is trivial:

\[
\bar{z}(t) = \bar{u}(t) = \begin{bmatrix} V(t) & \theta(t) \end{bmatrix}^T
\]

In this trivial observer case, the input can be considered to be a command or feedback depending on the context.

5.1.2 Integrated Heading

If you have a measurement of the rate of heading from, for example, a gyro, the vehicle dynamics can be expressed as:

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

Where the state and input vectors are:

\[
\begin{align*}
\dot{x}(t) &= \begin{bmatrix} x(t) & y(t) & \theta(t) \end{bmatrix}^T \\
u(t) &= \begin{bmatrix} V(t) & \omega(t) \end{bmatrix}^T
\end{align*}
\]

and the observer is, again, trivial.

5.1.3 Differential Heading

An important special case of integrated heading is the case where there is no gyro but the differential speeds of two wheels determine heading rate.

\[\text{Figure 4 Differential Heading}\]
• The state equation is the same but the observer equation is:

\[
\begin{bmatrix}
r(t) \\
l(t)
\end{bmatrix} = \begin{bmatrix} 1 & W/2 \\ 1 & -W/2 \end{bmatrix} \begin{bmatrix} V(t) \\ \omega(t) \end{bmatrix}
\]

or \( z(t) = Mu(t) \)

5.2 General Solution of the Perturbative Dynamics

• Figuring out error propagation in odometry is significantly harder than in triangulation. Yet, there is a solution.
• Recall all that stuff on the transition matrix. This is what its for.

5.2.1 Solution Integrals

• Systematic error propagates according to the vector convolution integral:

\[
\delta \hat{x}(t) = \Phi(t, t_0)\delta \hat{x}(t_0) + \int_{t_0}^{t} \Phi(t, \tau)G(\tau)\delta u(\tau)d\tau
\]

• Random error propagates according to the matrix convolution integral:

\[
P(t) = \Phi(t, t_0)P(t_0)\Phi^T(t, t_0)
\]

\[
+ \int_{t_0}^{t} \Phi(t, \tau)L(\tau)Q(\tau)L^T(\tau)\Phi^T(t, \tau)d\tau
\]

• The following figure validates these linearized models against a fully nonlinear Monte Carlo simulation:

5.2.2 Input Transition Matrix

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

\[
\tilde{\Phi}(t, \tau) = \Phi(t, \tau)G(\tau) = \Phi(t, \tau)L(\tau)
\]

because \( G(\tau) \) and \( L(\tau) \) are two (different, conventional) names for the same thing.
• In general, this matrix governs the propagation of both systematic and random error in odometry.
• The vector convolution integral can be written:

\[
\delta x(t) = \Phi(t, t_0)\delta x(t_0) + \sum_{i}^{t} \left[ \Phi_i \delta u_i d\tau \right]
\]

• The elements \( \Phi_{ij} \) are the \( ij \)th column of the input transition matrix. The elements \( \delta u_i \) are the individual input error sources.

• The result says that 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.

• So, odometry errors are usually path dependent.

• The matrix convolution integral can be written:

\[
P(t) = \Phi(t, t_0)P(t_0)\Phi^T(t, t_0) + \sum_{i}^{t} \sum_{j}^{t} \int_{t_0}^{t} q_{ij} \Phi_i \Phi_j d\tau
\]

Figure 5 Monte Carlo Simulation Compared to Linearized Theory:

- In fact, integrals of its columns and of outer products of its columns are the canonical error propagation modes.

5.2.3 Moments of Error

- We can isolate the contribution of each input error on the output with some (omitted) algebra.
• The elements $\Phi_i \Phi_j$ are matrices formed from the outer product of two columns of the input transition matrix. The elements $q_{ij}$ are the individual covariances of the input error sources.
• The same sum of moments interpretation applies but now the moments are matrix-valued.

5.3 Example: Integrated Heading Odometry
• The use of the general solution will be illustrated on this case.
• Consider again the integrated heading case:

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

5.3.1 Linearization
• The system and input Jacobians for this case are:

\[
F(t) = \begin{bmatrix} 0 & 0 & -V_s \theta \\ 0 & 0 & V_c \theta \\ 0 & 0 & 0 \end{bmatrix}, \quad G(t) = \begin{bmatrix} c \theta(t) & 0 \\ s \theta(t) & 0 \\ 0 & 1 \end{bmatrix}
\]

• So, the linearized dynamics are:

\[
\frac{d}{dt} \begin{bmatrix} \delta x(t) \\ \delta y(t) \\ \delta \theta(t) \end{bmatrix} = \begin{bmatrix} 0 & 0 & -V_s \theta \\ 0 & 0 & V_c \theta \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta x(t) \\ \delta y(t) \\ \delta \theta(t) \end{bmatrix} + \begin{bmatrix} c \theta(t) & 0 \\ s \theta(t) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \delta V(t) \\ \delta \omega(t) \end{bmatrix}
\]

5.3.2 Transition Matrix
• This case will turn out to satisfy the commutative dynamics condition:

\[
\Psi(t, \tau)F(t) = F(t)\Psi(t, \tau)
\]

• so $\Psi(t, \tau)$ is the transition matrix.
• The steps to get it are:

\[
R(t, \tau) = \int_{\tau}^{t} \begin{bmatrix} 0 & 0 & -V_s \theta \\ 0 & 0 & V_c \theta \\ 0 & 0 & 0 \end{bmatrix} d\zeta = \begin{bmatrix} 0 & 0 & -\Delta y(t, \tau) \\ 0 & 0 & \Delta x(t, \tau) \\ 0 & 0 & 0 \end{bmatrix}
\]

where:

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

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

• The matrix exponential is then:

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

• Then, the input transition matrix is:

\[
\Phi(t, \tau) = \Phi(t, \tau)G(\tau) = \begin{bmatrix} c\theta(\tau) & -\Delta y(t, \tau) \\ s\theta(\tau) & \Delta x(t, \tau) \\ 0 & 1 \end{bmatrix}
\]

5.3.3 General Solution

• Substituting gives the elegant expression for the general solution for the propagation of systematic error in integrated heading odometry:

\[
\delta\chi(t) = \begin{bmatrix} 1 & 0 & -y(t) \\ 0 & 1 & x(t) \\ 0 & 0 & 1 \end{bmatrix} \delta\chi(0) + \int_{\tau}^{t} \begin{bmatrix} c\theta & -\Delta y(t, \tau) \\ s\theta & \Delta x(t, \tau) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \delta V(\tau) \\ \delta\omega(\tau) \end{bmatrix} d\tau
\]

• This has the following intuitive interpretation:

![Figure 6 Convolution Integral.](image)

• The matrix relating input systematic errors occurring at time \( \tau \) to their later effect at time \( t \) is:

\[
d \begin{bmatrix} \delta x(t) \\ \delta y(t) \\ \delta \theta(t) \end{bmatrix} = \begin{bmatrix} c\theta & -\Delta y(t, \tau) \\ s\theta & \Delta x(t, \tau) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \delta V(\tau) \\ \delta\omega(\tau) \end{bmatrix} d\tau
\]
• So, the solution is just adding up the effect of the entire history of input errors on the present pose error.

• In moment form, the vector convolution integral is:

\[
\delta \dot{\mathbf{x}}(t) = I\mathbf{C}_d + \int_{0}^{t} \Phi_v(\tau) \delta V d\tau + \int_{0}^{t} \Phi_\omega(\tau) \delta \omega d\tau
\]

• where:

\[
I\mathbf{C}_d = \begin{bmatrix} 1 & 0 & -y(t) \\ 0 & 1 & x(t) \\ 0 & 0 & 1 \end{bmatrix} \delta \mathbf{x}(0) \quad \Phi_v(\tau) = \begin{bmatrix} c \theta & s \theta & 0 \end{bmatrix}^T \\
\Phi_\omega(\tau) = \begin{bmatrix} -c \Delta y(t, \tau) & \Delta x(t, \tau) & 1 \end{bmatrix}^T
\]

• Likewise, the general solution for the propagation of random error in integrated heading odometry is:

• In moment form, this is:

\[
P(t) = I\mathbf{C}_s + \int_{0}^{t} \tilde{\Phi}_{\omega\omega}(\tau) \sigma_{\omega\omega} d\tau
\]

\[
+ \int_{0}^{t} \tilde{\Phi}_{\omega v}(\tau) \sigma_{\omega v} d\tau + \int_{0}^{t} \tilde{\Phi}_{v v}(\tau) \sigma_{v v} d\tau
\]
• where:

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

\[
\Phi_{vv}(\tau) = \begin{bmatrix}
c\theta \\
s\theta \\
0
\end{bmatrix}
T
\begin{bmatrix}
c^2 \theta & c\theta s\theta & 0 \\
s\theta s\theta & s^2 \theta & 0 \\
0 & 0 & 0
\end{bmatrix}
\]

\[
\Phi_{v\omega}(\tau) = \Phi_{\omega v}(\tau) = \begin{bmatrix}
c\theta \\
s\theta \\
0
\end{bmatrix}
T
\begin{bmatrix}
-\Delta y \\
-\Delta y \\
-\Delta \theta
\end{bmatrix}
= \begin{bmatrix}
-c\theta \Delta y & c\theta \Delta x & c\theta \\
-s\theta \Delta y & s\theta \Delta x & s\theta \\
0 & 0 & 0
\end{bmatrix}
\]

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

5.3.4 Error Models

• Getting concrete results still requires both the input errors and trajectory followed to be specified.

• Let the systematic error model be a constant scale error on the velocity (encoder) and a constant bias error on the angular velocity (gyro):

\[
\delta V = \delta V_v \times V
\]

\[
\delta \omega = const
\]

• where the notation means:

\[
\delta V_v = \frac{\partial}{\partial \delta V}(\delta V)
\]

• Let the random error model be a distance-dependent random walk for linear velocity and a constant covariance for angular velocity.

\[
\sigma_{vv} = \sigma^{(v)}_{vv} |V|
\]

\[
\sigma_{\omega\omega} = const
\]

\[
\sigma_{v\omega} = 0
\]

• Now the general solution for this systematic error model on any trajectory is:

\[
\delta \chi(t) = \begin{bmatrix}
1 & 0 & -y(t) \\
0 & 1 & x(t) \\
0 & 0 & 1
\end{bmatrix}
\delta \chi(0) + \delta V_v \int_s^t \begin{bmatrix}
c\theta \\
s\theta \\
0
\end{bmatrix}
ds + \delta \omega \int_s^t \begin{bmatrix}
-\Delta y \\
-\Delta \theta
\end{bmatrix} d\tau
\]
5.3 Example: Integrated Heading Odometry

- The general solution for this random error model on any trajectory is:

\[
P(t) = IC_s + \sigma_v^{(v)} \int_{s}^{t} \begin{bmatrix} c^2 \theta & c \theta s \theta & 0 \\ c \theta s \theta & s^2 \theta & 0 \\ 0 & 0 & 0 \end{bmatrix} ds + \sigma_{\omega \omega} \int_{s}^{t} \begin{bmatrix} \Delta y^2 & -\Delta x \Delta y & -\Delta y \\ -\Delta x \Delta y & \Delta x^2 & \Delta x \\ -\Delta y & \Delta x & 1 \end{bmatrix} d\tau
\]

- Note especially that the remaining integrals are functions of the shape of the trajectory followed.

5.3.5 Straight Line

- The inputs which generate a straight line are:

\[
\omega(t) = 0 \quad V(t) = \text{arbitrary}
\]

- The associated solution to the integrated heading odometry equations are:

\[
x(t) = s(t) \quad y(t) = 0 \quad \theta(t) = 0
\]

- Then, the systematic errors in pose are:

\[
\delta \tilde{x}(t) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & s \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \delta x(0) \\ \delta y(0) \\ \delta \theta(0) \end{bmatrix} + \delta V \begin{bmatrix} s \\ s t/2 \\ t \end{bmatrix} + \delta \omega \begin{bmatrix} 0 \\ 0 \\ st/2 \end{bmatrix}
\]

Constant velocity was assumed in getting the deterministic error term \(st/2\). Alongtrack error is linear in distance while heading error is linear in time. Crosstrack error includes a term linear in distance and another term which is quadratic in time or distance (for constant velocity).

- The random errors in pose are:

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

Heading variance is linear in time as was intended. Heading covariance with crosstrack is linear in distance and time (or quadratic in either for constant velocity). Notice that the alongtrack variance is linear in distance rather than time whereas crosstrack variance is cubic in time (or distance for constant velocity).

---

1. Of course, for constant velocity, \(st = vt^2 = s^2 / v\). Even in this simple case, the common assumption that odometry error is linear in distance is wrong. Gyro error depends neatly on the product \(st\).
5.4 Other Techniques

- If you can’t find the transition matrix to solve the dynamics in closed form, the dynamics can be integrated numerically.
- For systematic error, you can either:
  - Integrate the nonlinear dynamics with a perturbative input and subtract the unperturbed input. This is a full nonlinear solution.
  - Integrate the linearized dynamics.
- For random error:
  - A nonlinear solution requires Monte Carlo techniques which are tedious but doable.
  - You can integrate the linear variance equation numerically - which is just as good for almost all practical purposes.

5.5 Insights

Further analysis of odometry shows that:

- Because odometry is homogeneous in $V$, the effects of scale errors vanish on a closed trajectory. The path is the wrong size but the right shape - so it still closes - so there is no error at the point of closure.
- The effects of gyro bias errors vanish at the centroid of the trajectory. Every path has a centroid but symmetric paths have an obvious one.
- Response to input errors is always the (path dependent) sum of one moment for each error source.
- Response to initial conditions (initial pose errors) is always path independent.
- Even systematic odometry error is not linear in distance.
- The response to random gyro bias error is not monotone.
Here is a nifty table of error characteristics:

**Table 2: Odometry Error Propagation**

<table>
<thead>
<tr>
<th></th>
<th>Locally</th>
<th>Globally</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Velocity</strong></td>
<td>$\delta x \sim s$</td>
<td>$\delta x \sim \int_0^s c\theta ds \sim x$</td>
<td>Path Independent</td>
</tr>
<tr>
<td>Scale</td>
<td>$\delta y \sim 0$</td>
<td>$\delta y \sim \int_0^s s\theta ds \sim y$</td>
<td>Path Independent</td>
</tr>
<tr>
<td></td>
<td>$\sigma_{xx} \sim s$</td>
<td>$\sigma_{xx} \sim \int_0^s c^2 \theta ds$</td>
<td>Motion Dependent</td>
</tr>
<tr>
<td><strong>Error</strong></td>
<td>$\sigma_{yy} \sim 0$</td>
<td>$\sigma_{yy} \sim \int_0^s s^2 \theta ds$</td>
<td>Motion Dependent</td>
</tr>
<tr>
<td><strong>Gyro</strong></td>
<td>$\delta x \sim s$</td>
<td>$\delta x \sim \int_0^t \Delta y d\tau$</td>
<td>Time Dependent</td>
</tr>
<tr>
<td></td>
<td>$\delta y \sim Vt^2 \sim st$</td>
<td>$\delta y \sim \int_0^t \Delta x d\tau$</td>
<td>Time Dependent</td>
</tr>
<tr>
<td><strong>Bias</strong></td>
<td>$\sigma_{xx} \sim s$</td>
<td>$\sigma_{xx} \sim \int_0^t \Delta y^2 d\tau$</td>
<td>Time Dependent</td>
</tr>
<tr>
<td></td>
<td>$\sigma_{yy} \sim V^2 t^3$</td>
<td>$\sigma_{yy} \sim \int_0^t \Delta x^2 d\tau$</td>
<td>Time Dependent</td>
</tr>
</tbody>
</table>
6 Deduced Reckoning Systems

6.1 Wheeled Robot

- Often do “odometry” or ded. reckoning from linear or angular position measurements
- Integration is easy. The hard issue is the kinematic relationships between wheel motions and vehicle motion.
- Wheels may be passive or active.
- Active wheels may be steerable, driven, or both.
- Wheel slip occurs when wheel/floor interface friction saturates.
- Body motion may be over determined function of wheel speeds unless there is wheel slip.
- Wheel slip cannot usually be modelled effectively and is major source of error.
- Minimum requirements for 2D motion are two driven and steered simple wheels.

Consider the following differential odometry case:

\[
\dot{v}_1 = \dot{\omega}_1 \times \dot{r}_1 = r \omega_1 [\cos \theta \hat{i} + \sin \theta \hat{j}]
\]

\[
\dot{v}_2 = \dot{\omega}_2 \times \dot{r}_2 = r \omega_2 [\cos \theta \hat{i} + \sin \theta \hat{j}]
\]

NOTE: We can dead reckon 3 outputs from only 2 inputs.
6.2 Legged Robot

- Separate body moves from leg moves
- Assumes no ballistics (running)
- Most legged vehicles are overdetermined
  - Each leg is a manipulator
  - Just one leg determines body trajectory

\[
\begin{align*}
\dot{\mathbf{v}} &= \frac{\mathbf{v}_1 + \mathbf{v}_2}{2} = \frac{r}{2} \left[ (\omega_1 + \omega_2) \cos \theta \mathbf{i} + (\omega_1 + \omega_2) \sin \theta \mathbf{j} \right] \\
\omega &= \frac{v_2 - v_1}{L} = \frac{r}{L} [\omega_2 - \omega_1]
\end{align*}
\]

- This is simplest possible example, others are much harder.
- Can use external measurement of heading to improve performance.

\[
\begin{align*}
\dot{x} &= \frac{r}{2} (d\theta_1 + d\theta_2) \cos \theta \\
\dot{y} &= \frac{r}{2} (d\theta_1 + d\theta_2) \sin \theta \\
\dot{\theta} &= \frac{r}{L} (d\theta_2 - d\theta_1)
\end{align*}
\]

body velocity:
\[
\begin{align*}
\dot{\mathbf{v}} &= \hat{\omega} \times \dot{\mathbf{r}} \\
\dot{\mathbf{r}} &= r \left[ -\sin \theta \mathbf{i} + \cos \theta \mathbf{j} \right] \\
\hat{\omega} &= \omega \hat{k} \\
v &= r\omega \left[ -\cos \theta \mathbf{i} - \sin \theta \mathbf{j} \right]
\end{align*}
\]

ded. reckoning equations:
\[
\begin{align*}
dx &= -r\omega \cos \theta dt \\
dx &= -r\omega \sin \theta dt
\end{align*}
\]
7 Summary

- Dead reckoning and triangulation are two distinct, complementary methods of determining position and attitude.

7.1 Triangulation

- Triangulation problems can be solved in general with techniques for solving simultaneous nonlinear equations.
- If you linearize triangulation constraints, you can find the relationships between systematic and random input and output error.
- The Geometric Dilution of Precision is the multiplicative factor that converts total sensor error into total position error. It can be obtained from the Jacobian determinant. Several tricks were presented to help get it easily.
- Heading cannot be triangulated unless bearings are observed or two points on the vehicle are located independently.
- Different forms of triangulation have different sensitivity behaviors.

7.2 Odometry

- Odometry is a form of dead reckoning where, terrain-relative velocity indications are integrated to generate position.
- Errors in odometry propagate according to nonlinear differential equations whose solutions are integrals. If we linearize (perturb) the equations, a general solution can be found.
- Many problems can be reduced to computing moments of arc on the trajectory. These moments are the equivalent of the moment of inertia in mechanics or the Laplace Transform in electrical engineering.
- Many unusual error behaviors result from the dynamic behavior of odometry. They include path independence, response to symmetric inputs, reversibility, monotonicity, etc.
- Odometry can be calibrated from a number of trajectories to a single point. This cuts down on the ground truth info required.
8 Notes
Perhaps cover calibration in systems engineering section.