(1) a) when a Global positioning System (GPS) tries to Estimate a localization the result consists of:

$$P = \text{True Position} + \text{Systematic Error (bias)} + \text{white noise}$$

during the calculation of speed we getting two position in time $t_1$ and $t_2$ ($t_1 < t_2$) in order to measure Speed of vehicle. So this is the formula for Speed:

$$(P_2 - P_2)/(t_2 - t_1)$$

$$\Big(\text{True value}(P_2) + \text{bias}(P_2) + \text{white Noise}(P_2) - \big(\text{True Value}(P_1) + \text{bias}(P_1) + \text{white Noise}(P_1)\big)\Big) / (t_2 - t_1) =$$

$$= \Big[\text{True Value}(P_2) - \text{True Value}(P_1) + \text{bias}(P_2) - \text{bias}(P_1) + \text{whitenosis}(P_2 - P_1)\Big] =$$

when we know bias is the same for points in close range

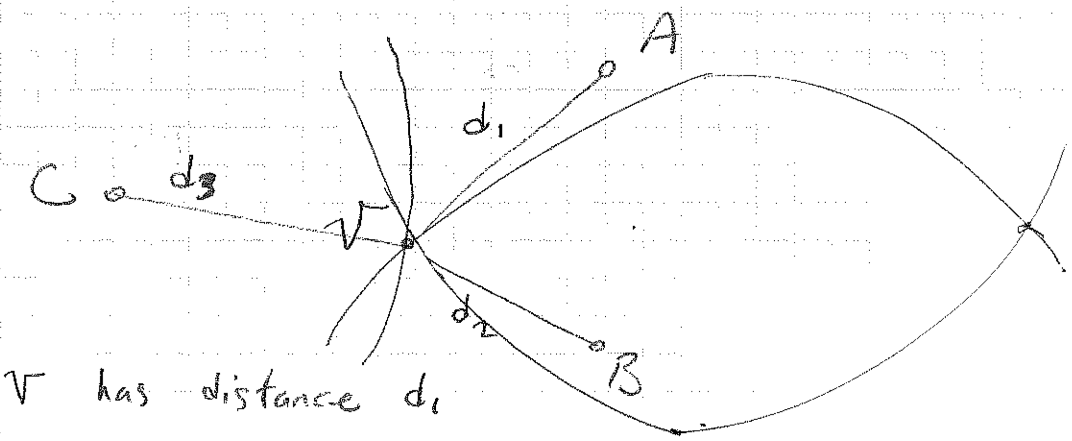
$$\text{bias}(P_2) = \text{bias}(P_1)$$

$$\rightarrow \text{bias}(P_2) - \text{bias}(P_1) = 0$$

we see this bias (Systematic Error) easily vanished durring Calcalation of Speed! So in Speed calculation by GPS the answers are more accurate than position estimation.

b) Total satellites needed are 4. we need 3 Sat. for doing Triangulation to find position and need one Sat for

(2) Triangulation is one of the ~~relative~~ absolute localizations methods. in this method we need to ~~comman~~ know the distances of vechicle from 3 different specific ~~pos~~ known position in our envirunment. Assume we have 3 known position and in each ~~in position~~ position placed a beam that vehicle can detect them and measure the distance from them. A, B, C are our beams. V is our vehicle.




if we know V has distance $d_1$ from A So V should be one point on a circle with radius $d_1$ and center of A. if then we know V has distance $d_2$ from B so V should be on circle with radius $= d_2$ and center (B) _ this two intersect in two (2) point. So V should be One of them. by knowing distance V from C as $d_3$ only One True point from 2 possible point should be our position.

3p

(3) a) rate-gyro : is a gyro so provide data about vehicle direction in 3 axes (x, y, z) and since this is a rate-gyro we will have the (~~acceleration~~) and angular Speed of changing the ...

b) as we know position and direction of our vehicle robot is important if robot is moving in environment. ~~the inverse~~

if error in estimating direction (angle) reduces. So the error in whole estimation (Uncertainty) will be less than before.

In most of robots direction of movement is mentioned and applied in position calculations. using rate-gyro give us more accurate measurement about current direction of robot. 2

c)

the Speed of measurement of orientation and direction is not high. ~~Second~~ drift.

other limitation comes up since rate-gyro has a mechanical Structure and is also heavy and needs Space to place in robot.

1/2

(4)

$$\begin{cases} X_{k+1} = X_k + \Delta X_k \Rightarrow \Delta X_k = \Delta s_k \cos\left(\theta_k + \frac{\Delta\theta_k}{2}\right) \\ Y_{k+1} = Y_k + \Delta Y_k \Rightarrow \Delta Y_k = \Delta S_k \sin\left(\theta_k + \frac{\Delta\theta_k}{2}\right) \\ \theta_{k+1} = \theta_k + \Delta\theta_k \Rightarrow \Delta\theta_k = \Delta\theta_k \end{cases}$$

$$\Sigma_{XY\theta_{k+1}} = J_{X_k, Y_k, \theta_k} * \Sigma_{XY\theta_k} * J^T_{XY\theta_{kkk}} + J_{\Delta S_k \Delta\theta_k} * \Sigma_{\Delta S \Delta\theta} * J^T_{\Delta S \Delta\theta}$$ 1

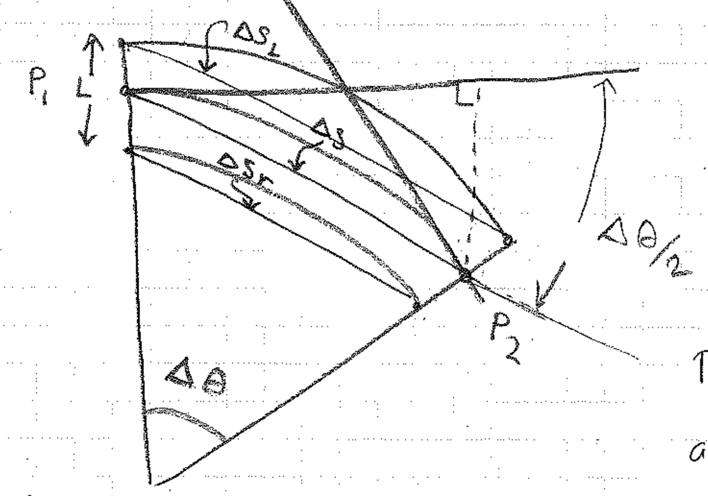
$$\Sigma_{XY\theta(k+1)} = J_{XY\theta(k)} = \begin{bmatrix} 1 & 0 & \left[-\sin\left(\theta_k + \frac{\Delta\theta_k}{2}\right)\right]\Delta S \\ 0 & 1 & \left[\cos\left(\theta_k + \frac{\Delta\theta_k}{2}\right)\right]\Delta S \\ 0 & 0 & 1 \end{bmatrix}$$ 2

$$J_{\Delta S \Delta\theta} = \begin{array}{c} \begin{bmatrix} \cos\left(\theta_k + \frac{\Delta\theta_k}{2}\right) & -\frac{\Delta S}{2}\sin\left(\theta_k + \frac{\Delta\theta}{2}\right) \\ \sin\left(\theta_k + \frac{\Delta\theta_k}{2}\right) & \frac{\Delta S}{2}\cos\left(\theta_k + \frac{\Delta\theta}{2}\right) \\ 0 & 1 \end{bmatrix} & 2 \\ 3\times 2 & \end{array}$$

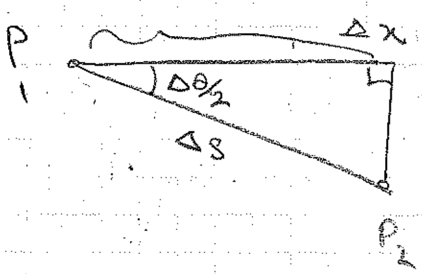
$$\Sigma_{\Delta S \Delta\theta} = \begin{bmatrix} \mathrm{cov}(\Delta S, \Delta S) & \mathrm{cov}(\Delta S, \Delta\theta) \\ \mathrm{cov}(\Delta\theta, \Delta S) & \mathrm{cov}(\Delta\theta, \Delta\theta) \end{bmatrix}_{2\times 2} = \begin{bmatrix} \mathrm{Var}(\Delta S) & 0 \\ 0 & \mathrm{Var}(\Delta\theta) \end{bmatrix}_{2\times 2}$$

$$= \begin{array}{c} \begin{bmatrix} (\sigma_r^2 + \sigma_l^2)/4 & 0 \\ 0 & (\sigma_r^2 + \sigma_l^2)/L^2 \end{bmatrix} & 2 \\ 2\times 2 & \end{array}$$

(b)




$\frac{\Delta\theta}{2}$ is comming from the angle that is made by the Direct line from $P_1$ to $P_2$ (straigh line) and the heading of Robot in $P_1$.




here for simplify $\theta = 0$ assumed:

$\Delta x = \Delta S \cos \frac{\Delta\theta}{2}$

$\Delta y = \Delta S \cos \frac{\Delta\theta}{2}$

if $\theta \neq 0$ this triangle just rotate constantly so the $\left(\theta + \frac{\Delta\theta}{2}\right)$ is still working.

2

(5) a)

$$\Sigma_{X_{k+1}} = J_X \Sigma_{X_k} J_X^T + \underbrace{J_{V\alpha T} \Sigma_{V\alpha T} J_{V\alpha T}^T}_{\text{Cov}(\Delta x, \Delta y, \Delta\theta)}$$

$$\text{Cov}(\Delta x, \Delta y, \Delta\theta) = J_{V,\alpha,T} \Sigma_{V,\alpha,T} J_{V,\alpha,T}^T$$

$$J_{V,\alpha,T} = \begin{bmatrix} P'_V & 0 & 0 \\ Q'_V & 0 & 0 \\ \frac{\partial \Delta\theta}{\partial V} & 0 & 0 \end{bmatrix}$$

we just need to have $\underline{P'_V \text{ and } Q'_V}$ - we doont need other derivations since the $\sigma_\alpha^2 = 0$ and $\sigma_T^2 = 0$ So they will vanish during the matrix Multiplication process.

$$\underbrace{V(k) * \cos(\alpha(k)) * T}_{A} * \underbrace{\cos\left(\theta(k) + \frac{V(k)\sin(\alpha(k)) * T}{2L}\right)}_{B} = P$$

$$P'_V = A'_V B + A B'_V = \cos(\alpha(k)) * T * \cos\left(\theta(k) + \frac{V(k)\sin(\alpha(k)) * T}{2L}\right) +$$

$$V(k)\cos(\alpha(k)) T * \left[-\frac{\sin(\alpha(k)) * T}{2L} * \sin\left(\theta(k) + \frac{V(k)\sin(\alpha(k)) * T}{2L}\right)\right].$$

$$\underbrace{V(k)\cos(\alpha(k)) T}_{A} \underbrace{\sin\left(\theta(k) + \frac{V(k)\sin(\alpha(k)) T}{2L}\right)}_{B} = Q$$

$$Q'_V = A'_V B + A B'_V$$

$$Q'_V = \cos(\alpha(k)) T \sin\left(\theta(k) + \frac{V(k)\sin(\alpha(k)) T}{2L}\right) +$$

$$J_{v,\alpha,T} = \begin{bmatrix} \cos(\alpha(k))T\cos\left(\theta(k)+\frac{v(k)\sin(\alpha(k))T}{2L}\right) - v(k)\cos(\alpha(k))T\,\frac{\sin(\alpha(k))T}{2L} \times \sin\left(\theta(k)+\frac{v(k)\sin(\alpha(k))T}{2L}\right) & 0 & \cdots \\ & 1 & \\ & 0 & \\ \cos(\alpha(k))T\,\sin\left(\theta(k)+\frac{v(k)\sin(\alpha(k))T}{2L}\right) + v(k)\cos(\alpha(k))T\cos\left(\theta(k)+\frac{v(k)\sin(\alpha(k))T}{2L}\right)\cdot\frac{\sin\alpha T}{2L} & & \end{bmatrix}$$

$$\sum_{v\alpha T} = \begin{bmatrix} \sigma_v^2 & 0 & 0 \\ 0 & \sigma_\alpha^2 & 0 \\ 0 & 0 & \sigma_T^2 \end{bmatrix} = \begin{bmatrix} \sigma_v^2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

b) Systematic error (bias) is a error that comes from shape or physics of our robor or envirunment. it has a certain value that we can measure it and always have a same direction.

bias vector $= \vec{V}(x, y, z)$ ↙ bias

bias or systematic Error in <u>not Symmetric</u>.

Random Error is a errors That is because or sensors or devices can not be 100% accurate and they are ~~...~~ changing in every direction. Random error is called noise (white noise) is happening in Random direction and amplitude each time. most of time noise has symmetric shape of distribution and we can say it with a statistic probability distribution like $N \sim (\mu, \sigma^2)$ ~~...~~

$\uparrow$ noise

2/4p

Example: left encoder always counts more in every direction

(6)

a) $X = aX_1 + (1-a)X_2$ ← this is a weighted average with parametric weights $\underline{a}$ and $\underline{(1-a)}$ and is linear combination also.

$Var(X) = Var\left[aX_1 + (1-a)X_2\right] =$

$= a^2 Var(X_1) + (1-a)^2 Var(X_2)$

$\sigma^2 = a^2\sigma_1^2 + (1-a)^2\sigma_2^2$

$(\sigma^2)'_a = 0 \Rightarrow 2a\sigma_1^2 - 2(1-a)\sigma_2^2 = 0 \qquad a\sigma_1^2 - (1-a)\sigma_2^2 = 0$

$\Rightarrow a\sigma_1^2 - \sigma_2^2 + a\sigma_2^2 = 0 \qquad a(\sigma_1^2 + \sigma_2^2) = \sigma_2^2 \qquad \boxed{a = \dfrac{\sigma_2^2}{\sigma_2^2 + \sigma_1^2}}$

So best linear combination with least variance is:

$$X = \frac{\sigma_2^2}{\sigma_2^2 + \sigma_1^2} X_1 + \frac{\cancel{\sigma_2^2} + \sigma_1^2 - \cancel{\sigma_2^2}}{\sigma_2^2 + \sigma_1^2} X_2 = \frac{\sigma_2^2}{\sigma_2^2 + \sigma_1^2} X_1 + \frac{\sigma_1^2}{\sigma_2^2 + \sigma_1^2} X_2$$

$$\boxed{\underset{\text{estimation}}{\hat{X}} = \frac{\sigma_2^2}{\sigma_2^2 + \sigma_1^2}\hat{X}_1 + \frac{\sigma_1^2}{\sigma_2^2 + \sigma_1^2}\hat{X}_2} \Leftrightarrow \text{means}$$

4

b) $\sigma_2^2 = a^2\sigma_1^2 + (1-a)^2\sigma_2^2$

$$\sigma_2^2 = \left(\frac{\sigma_2^2}{\sigma_2^2 + \sigma_1^2}\right)^2 \times \sigma_1^2 + \left(\frac{\sigma_1^2}{\sigma_2^2 + \sigma_1^2}\right)^2 \sigma_2^2 = \frac{\sigma_2^4}{(\sigma_2^2 + \sigma_1^2)^2}\sigma_1^2 + \frac{\sigma_1^4}{(\sigma_2^2 + \sigma_1^2)^2}\sigma_2^2 =$$

$$= \frac{\sigma_2^2\sigma_1^2\cancel{(\sigma_1^2 + \sigma_2^2)}}{\ldots} = \frac{\sigma_1^2\sigma_2^2}{\ldots} \Rightarrow \frac{1}{\ldots} = \frac{1}{\ldots} + \frac{1}{\ldots}$$

(7) As we see in odometry that Angle of heading is exist in $\Delta x$ $\Delta y$ $\Delta \theta$ So this is better to have low angular uncertainty because the error will ~~be~~ raise up very fast in angular uncertainty and our Robot will be lost soon. also uncertainty in position $x, y$ some times kills each others error when moving in opposite directions but angular error always grows so fast. 2p

(8) This grid is used to be like our map and by Ultra Sonic range finder sensors we will find Obstacles in field and the corresponding obstacle position inside grid will be valued as 1. This is How VFF and VFH methods work to avoid collision to obstacles. This grid will be used to make an angular Histogram. in VFF in this step we will measure repulsive forces from obstacles and attractive forces from destination Target in grid. So Total of these 2 vector will be our safe movement direction. in VFH second step is making a polar 1-D histogram and with a suitable Threshold on polar Histogram collision avoidance is working. 2p

(9) → in 3rd sheet →

→ 9 in cox Algorithem we have known our MAP.

Robot has odometry encoders on wheel (front wheel)-

also there is a Ultra Sonic Range finder (laser range finder) place in front of Robot. can scan 180° to find point around as specially to find points of envirunmant walls.

we will use Odometry to estimate position and Uncertainty each step - in every k step (k= 8) range finder start to scan eara. we transfer the points from sensor coordination System to world coordination System in order to find a transform that transfer these points to their corresponding Target walls that they belong. 1#

at first we need to calculate each image point distance to every walls on map. Shortest distance shows the proper right Target wall of that point. by knowing the distance of each point to it's corresponding Target wall we need to solve a equation to find best Transform matrix. we have to do this step iteratively to reach the closest answer by measuring LS (Least Squard method) for each iteration.

after finding best ($\Delta x$ $\Delta y$ $\Delta\theta$) Transform we can correct our fuse old Data by Odometry with new data from range finder Sensor.

4p