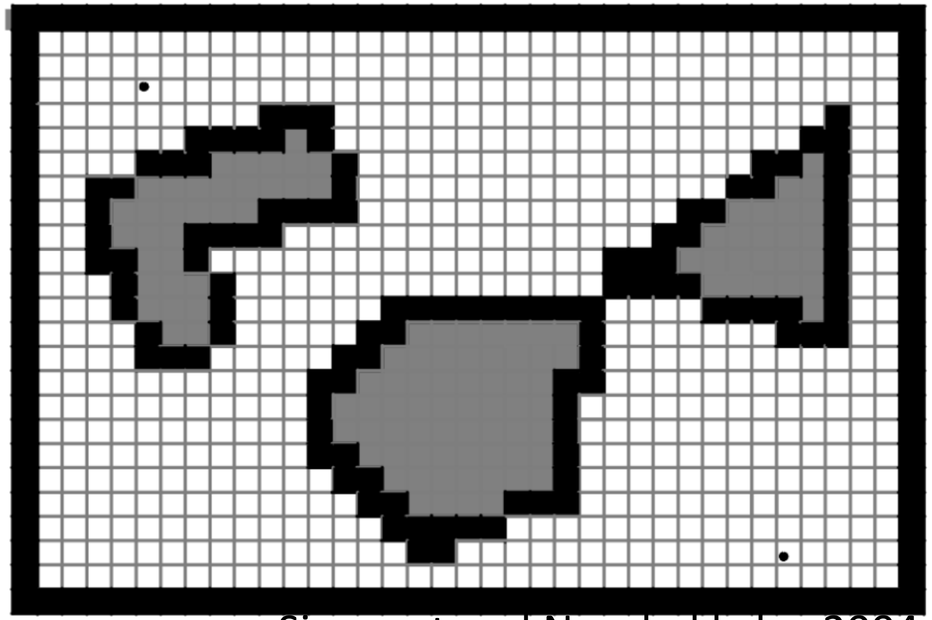
EKF SLAM

CMPT 882

Mar. 29

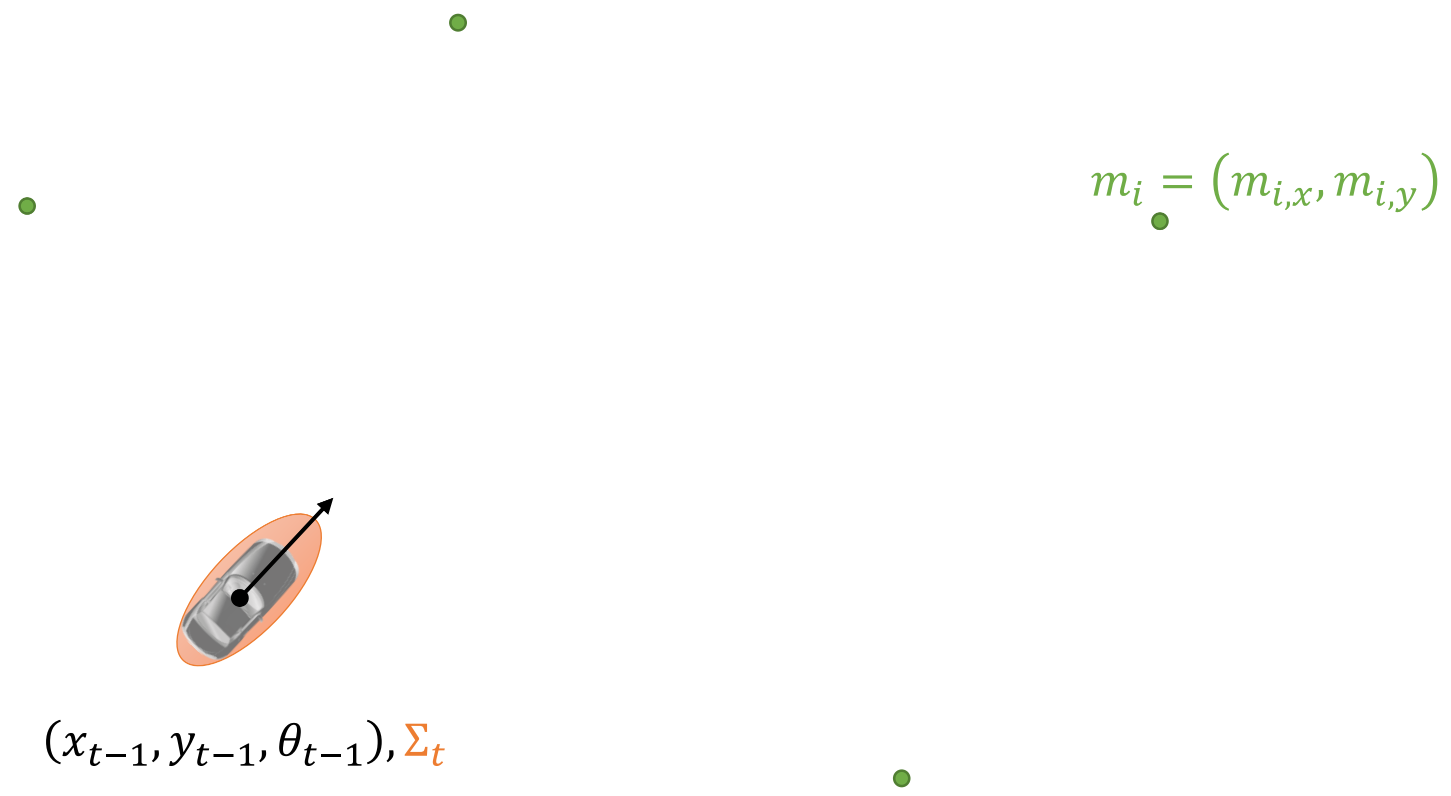
Localization: Problem Setup

- Assume a map is given: $m = \{m_1, m_2, \dots, m_N\}$
  - Location based: each m_i represents a specific location and whether it's occupied
  - **Feature based:** each m_i contains the location of the ith land mark
- Robot maintains and updates its belief about where it is with respect to the map
  - Position belief is updated based on sensor data
  - Position belief is a probability distribution

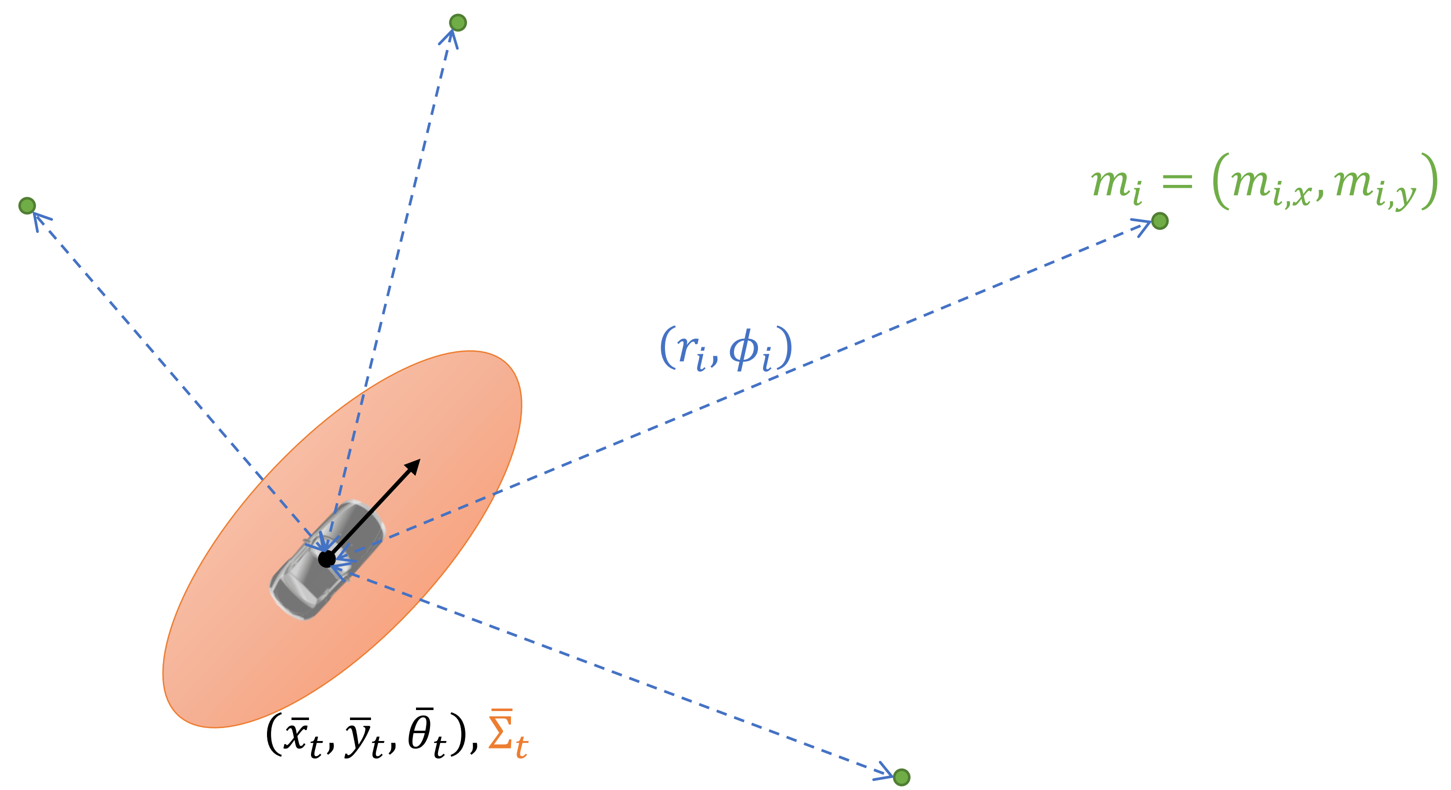


Siegwart and Nourbakhshs, 2004

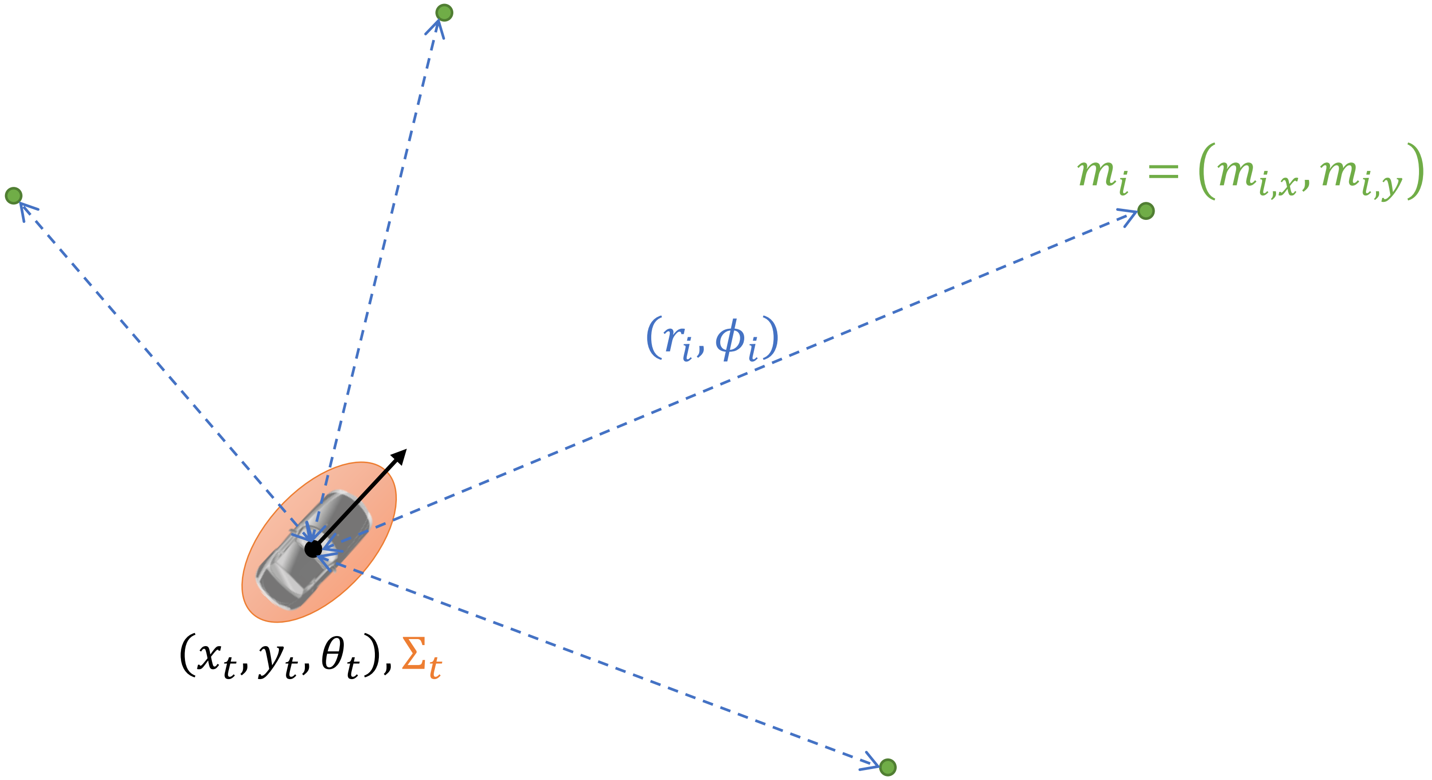
Localization



Localization



Localization

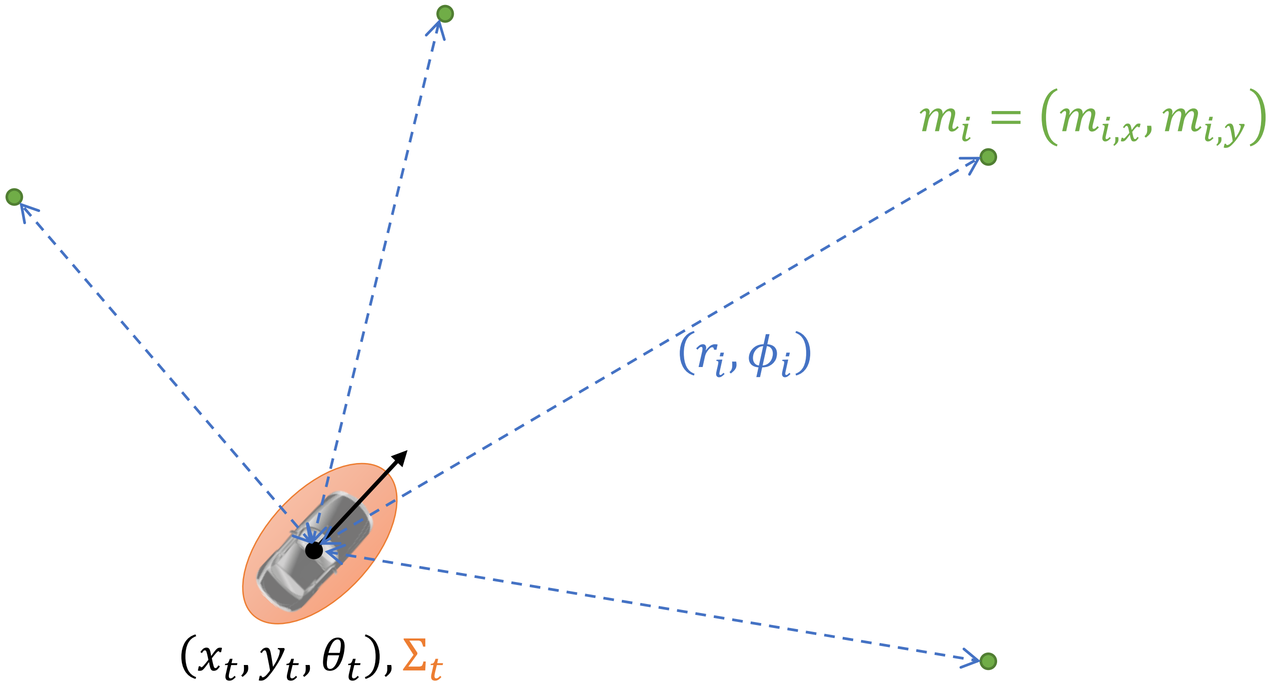


Simultaneous Localization and Mapping (SLAM)

- Land marks m are unknown, and must be estimated at the same time as internal state estimation

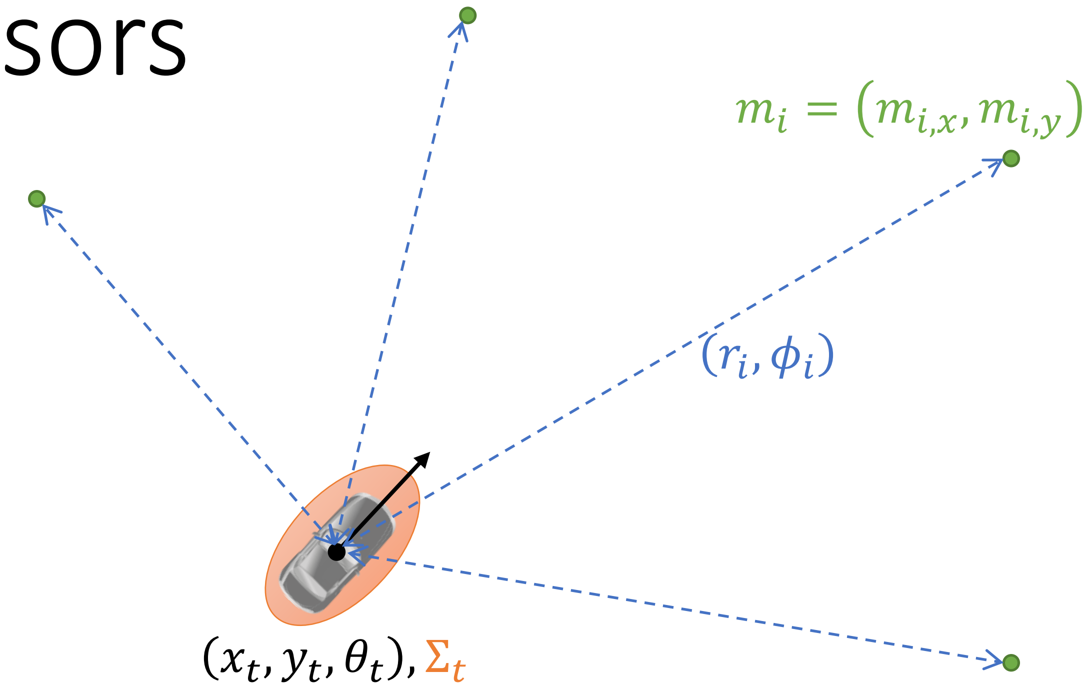
- Define combined state vector
  - $y \coloneqq \begin{bmatrix} x \\ m \end{bmatrix}$

- Calculate $p(y_t | z_{1:t}, u_{1:t-1})$
  - Previously, $p(x_t | z_{1:t}, u_{1:t-1}, m)$



- Strategy: define dynamics for y_t, and apply EKF from last class

Simple Car with Range Sensors



- Internal state dynamics (Forward Euler)
  - $x_{1,t} = x_{1,t-1} + \Delta t \cdot v \cos x_{3,t-1}$
  - $x_{2,t} = x_{2,t-1} + \Delta t \cdot v \sin x_{3,t-1}$
  - $x_{3,t} = x_{3,t-1} + \Delta t \cdot u_{t-1}$
- Environment dynamics
  - State: $m_i = (m_{i,x}, m_{i,y}), i = 1, \ldots, N$ (note that in general we may not know how many land marks are present)
  - $m = I_{2N \times 2N} m$ (identity dynamics, since land marks don't move)
- Car measures (with noise) range and bearing of each land mark

Simple Car Dynamics

- Put dynamics in the form $y_t = g(y_{t-1}, u_{t-1}) + \epsilon_t, \epsilon_t \sim N(0, R_t)$:
  - y_t and $g(y_{t-1}, u_{t-1})$ have $3 + 2N$ components
  - First three components of $g(y_{t-1}, u_{t-1})$: Forward Euler from ODE model of car
  - Remaining components of $g(y_{t-1}, u_{t-1})$: identity
  - R_t has zero entries except for top left 3×3 block
- Jacobian $G_t = \frac{\partial g}{\partial y_{t-1}}(y_{t-1}, u_{t-1}) = \begin{bmatrix} \frac{\partial g_1}{\partial y_{1,t-1}} & \cdots & \frac{\partial g_1}{\partial y_{3+2N,t-1}} \\ \vdots & \ddots & \vdots \\ \frac{\partial g_{3+2N}}{\partial y_{1,t-1}} & \cdots & \frac{\partial g_{3+2N}}{\partial y_{3+2N,t-1}} \end{bmatrix}$
  - Mostly zeros… only $y_{1,t-1}, y_{2,t-1} y_{3,t-1}$ appears in $g(y_{t-1}, u_{t-1})$

EKF SLAM: Prediction Step

- Extended Kalman filter algorithm:
  - $y_t = g(y_{t-1}, u_{t-1}) + \epsilon_t, \epsilon_t \sim N(0, R_t)$
  - $z_t = h(y_t) + \delta_t, \delta_t \sim N(0, Q_t)$
  - Linearization: $G_t = \nabla g(\mu_{t-1}, u_{t-1}), H_t = \nabla h(\bar{\mu}_t)$

Input: $\mu_{t-1}, \Sigma_{t-1}, u_{t-1}, z_t$

Output: μ_t, Σ_t

Perform prediction:

$$\bar{\mu}_t = g(\mu_{t-1}, u_{t-1})$$
$$\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^\top + R_t$$

Perform measurement update:

$$K_t = \bar{\Sigma}_t H_t^\top (H_t \bar{\Sigma}_t H_t^\top + Q_t)^{-1}$$
$$\mu_t = \bar{\mu}_t + K_t\big(z_t - h(\bar{\mu}_t)\big)$$
$$\Sigma_t = (I - K_t H_t)\bar{\Sigma}_t$$

Return μ_t, Σ_t

EKF SLAM prediction step details:

- $G_t \in \mathbb{R}^{(3+2N)\times(3+2N)}$; plug in μ_{t-1}
  - μ now refers to mean of y, which includes estimates of land mark positions
- $\Sigma_t \in \mathbb{R}^{(3+2N)\times(3+2N)}$;
  - Initialize upper left 3×3 block with zeros if initial internal state is known exactly
  - Initialize lower right $2N \times 2N$ block with $\infty \times I_{2N\times 2N}$ if there is no knowledge about land marks
  - now refers to covariance of y
- $R_t \in \mathbb{R}^{(3+2N)\times(3+2N)}$;
  - Zeros except for upper left 3×3 block

Simple Car with Range Sensors

- Measurements
  - $z_t = \{z_t^1, z_t^2, \dots\} = \{(r_t^1, \phi_t^1), (r_t^2, \phi_t^2), \dots\}$
- Measurement model
  - Assume ith measurement at time t corresponds to jth land mark
  - $\begin{bmatrix} r_t^i \\ \phi_t^i \end{bmatrix} = \begin{bmatrix} \sqrt{\left(m_{j,x} - x_{1,t}\right)^2 + \left(m_{j,y} - x_{2,t}\right)^2} \\ \text{atan2}\left(m_{j,y} - x_{2,t}, m_{j,x} - x_{1,t}\right) - x_{3,t} \end{bmatrix} + \delta_t,\ \ \delta_t \sim N(0, Q_t)$

Function in most programming languages and returns any possible angle

Data Association

- Define correspondence variable $c_t^i \in \{1, \ldots, N+1\}$
  - $c_t^i = j \leq N$ means ith measurement at time t corresponds to jth land mark
  - $c_t^i = N+1$ means measurement does not correspond to any land mark
- This class: assume c_t^i are known
- More advanced (and practical): estimate c_t^i using maximum likelihood

Simple Car Measurement Model

- Measurement from a single land mark:
  - $z_t^i = \begin{bmatrix} r_t^i \\ \phi_t^i \end{bmatrix} = \begin{bmatrix} \sqrt{\left(m_{j,x} - y_{1,t}\right)^2 + \left(m_{j,y} - y_{2,t}\right)^2} \\ \operatorname{atan2}\left(m_{j,y} - y_{2,t}, m_{j,x} - y_{1,t}\right) - y_{3,t} \end{bmatrix} + \delta_t = h^i(y_t),\ \ \delta_t \sim N(0, Q_t)$

- Jacobian: Mostly zeros. Let $r_t^i = \sqrt{\left(m_{j,x} - y_{1,t}\right)^2 + \left(m_{j,y} - y_{2,t}\right)^2}$ (remember to plug in estimates)

$$\frac{\partial h^i}{\partial y_t} = \begin{bmatrix} \frac{\partial h_1^i}{\partial y_{1,t}} & \cdots & \frac{\partial h_1^i}{\partial y_{3+2N,t}} \\ \frac{\partial h_2^i}{\partial y_{1,t}} & \cdots & \frac{\partial h_2^i}{\partial y_{3+2N,t}} \end{bmatrix}$$

Algebra… First Row

$$\frac{\partial}{\partial y_{1,t}}\sqrt{\left(m_{j,x}-y_{1,t}\right)^2+\left(m_{j,y}-y_{2,t}\right)^2}$$

$$=\frac{1}{2\sqrt{\left(m_{j,x}-y_{1,t}\right)^2+\left(m_{j,y}-y_{2,t}\right)^2}}\times 2\left(m_{j,x}-y_{1,t}\right)\times(-1)$$

$$=\frac{-\left(m_{j,x}-y_{1,t}\right)}{\sqrt{\left(m_{j,x}-y_{1,t}\right)^2+\left(m_{j,y}-y_{2,t}\right)^2}}$$

Simple Car Measurement Model

- Measurement from a single land mark:
  - $z_t^i = \begin{bmatrix} r_t^i \\ \phi_t^i \end{bmatrix} = \begin{bmatrix} \sqrt{\left(m_{j,x} - y_{1,t}\right)^2 + \left(m_{j,y} - y_{2,t}\right)^2} \\ \text{atan2}\left(m_{j,y} - y_{2,t}, m_{j,x} - y_{1,t}\right) - y_{3,t} \end{bmatrix} + \delta_t = h^i(y_t),\ \ \delta_t \sim N(0, Q_t)$
- Jacobian: Mostly zeros. Let $r_t^i = \sqrt{\left(m_{j,x} - y_{1,t}\right)^2 + \left(m_{j,y} - y_{2,t}\right)^2}$ (remember to plug in estimates)

$$\frac{\partial h^i}{\partial y_t} = \begin{bmatrix} \frac{\partial h_1^i}{\partial y_{1,t}} & \cdots & \frac{\partial h_1^i}{\partial y_{3+2N,t}} \\ \frac{\partial h_2^i}{\partial y_{1,t}} & \cdots & \frac{\partial h_2^i}{\partial y_{3+2N,t}} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{-\left(m_{j,x} - y_{1,t}\right)}{r_t^i} & & \\ & & \end{bmatrix}$$

Simple Car Measurement Model

- Measurement from a single land mark:
  - $z_t^i = \begin{bmatrix} r_t^i \\ \phi_t^i \end{bmatrix} = \begin{bmatrix} \sqrt{\left(m_{j,x} - y_{1,t}\right)^2 + \left(m_{j,y} - y_{2,t}\right)^2} \\ \operatorname{atan2}\left(m_{j,y} - y_{2,t}, m_{j,x} - y_{1,t}\right) - y_{3,t} \end{bmatrix} + \delta_t = h^i(y_t),\ \ \delta_t \sim N(0, Q_t)$

- Jacobian: Mostly zeros. Let $r_t^i = \sqrt{\left(m_{j,x} - y_{1,t}\right)^2 + \left(m_{j,y} - y_{2,t}\right)^2}$ (remember to plug in estimates)

$$\frac{\partial h^i}{\partial y_t} = \begin{bmatrix} \frac{\partial h_1^i}{\partial y_{1,t}} & \cdots & \frac{\partial h_1^i}{\partial y_{3+2N,t}} \\ \frac{\partial h_2^i}{\partial y_{1,t}} & \cdots & \frac{\partial h_2^i}{\partial y_{3+2N,t}} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{-(m_{j,x}-y_{1,t})}{r_t^i} & \frac{-(m_{j,y}-y_{2,t})}{r_t^i} & 0 & 0 & \cdots & 0 & \frac{m_{j,x}-y_{1,t}}{r_t^i} & \frac{m_{j,y}-y_{2,t}}{r_t^i} & 0 & \cdots & 0 \\ & & & & & & & & & & \end{bmatrix}$$

Algebra… Second Row

$$\operatorname{atan2}(m_{j,y} - y_{2,t}, m_{j,x} - y_{1,t}) = \arctan\left(\frac{m_{j,y}-y_{2,t}}{m_{j,x}-y_{1,t}}\right)$$

$$\frac{\partial}{\partial y_{1,t}} \arctan\left(\frac{m_{j,y}-y_{2,t}}{m_{j,x}-y_{1,t}}\right)$$

$$= \frac{1}{1+\left(\frac{m_{j,y}-y_{2,t}}{m_{j,x}-y_{1,t}}\right)^2} \times \frac{-(m_{j,y}-y_{2,t})}{(m_{j,x}-y_{1,t})^2} \times (-1)$$

$$= \frac{m_{j,y}-y_{2,t}}{(m_{j,x}-y_{1,t})^2+(m_{j,y}-y_{2,t})^2}$$

Simple Car Measurement Model

- Measurement from a single land mark:
  - $z_t^i = \begin{bmatrix} r_t^i \\ \phi_t^i \end{bmatrix} = \begin{bmatrix} \sqrt{\left(m_{j,x} - y_{1,t}\right)^2 + \left(m_{j,y} - y_{2,t}\right)^2} \\ \operatorname{atan2}\left(m_{j,y} - y_{2,t}, m_{j,x} - y_{1,t}\right) - y_{3,t} \end{bmatrix} + \delta_t = h^i(y_t),\ \ \delta_t \sim N(0, Q_t)$

- Jacobian: Mostly zeros. Let $r_t^i = \sqrt{\left(m_{j,x} - y_{1,t}\right)^2 + \left(m_{j,y} - y_{2,t}\right)^2}$ (remember to plug in estimates)

$$\frac{\partial h^i}{\partial y_t} = \begin{bmatrix} \frac{\partial h_1^i}{\partial y_{1,t}} & \cdots & \frac{\partial h_1^i}{\partial y_{3+2N,t}} \\ \frac{\partial h_2^i}{\partial y_{1,t}} & \cdots & \frac{\partial h_2^i}{\partial y_{3+2N,t}} \end{bmatrix}$$

$$\begin{array}{cccccccccccc} & & & & & & & & \text{Column } 2+2j & & & \\ = & \left[\begin{array}{ccccccccccc} \frac{-(m_{j,x}-y_{1,t})}{r_t^i} & \frac{-(m_{j,y}-y_{2,t})}{r_t^i} & 0 & 0 & \cdots & 0 & \frac{m_{j,x}-y_{1,t}}{r_t^i} & \frac{m_{j,y}-y_{2,t}}{r_t^i} & 0 & \cdots & 0 \\ \frac{m_{j,y}-y_{2,t}}{\left(r_t^i\right)^2} & & & & & & & & & & \end{array}\right] & & & & & & & & & & \end{array}$$

Simple Car Measurement Model

- Measurement from a single land mark:
  - $z_t^i = \begin{bmatrix} r_t^i \\ \phi_t^i \end{bmatrix} = \begin{bmatrix} \sqrt{\left(m_{j,x} - y_{1,t}\right)^2 + \left(m_{j,y} - y_{2,t}\right)^2} \\ \text{atan2}\left(m_{j,y} - y_{2,t}, m_{j,x} - y_{1,t}\right) - y_{3,t} \end{bmatrix} + \delta_t = h^i(y_t),\ \ \delta_t \sim N(0, Q_t)$

- Jacobian: Mostly zeros. Let $r_t^i = \sqrt{\left(m_{j,x} - y_{1,t}\right)^2 + \left(m_{j,y} - y_{2,t}\right)^2}$ (remember to plug in estimates)

$$\frac{\partial h^i}{\partial y_t} = \begin{bmatrix} \frac{\partial h_1^i}{\partial y_{1,t}} & \cdots & \frac{\partial h_1^i}{\partial y_{3+2N,t}} \\ \frac{\partial h_2^i}{\partial y_{1,t}} & \cdots & \frac{\partial h_2^i}{\partial y_{3+2N,t}} \end{bmatrix} \in \mathbb{R}^{2\times(3+2N)}$$

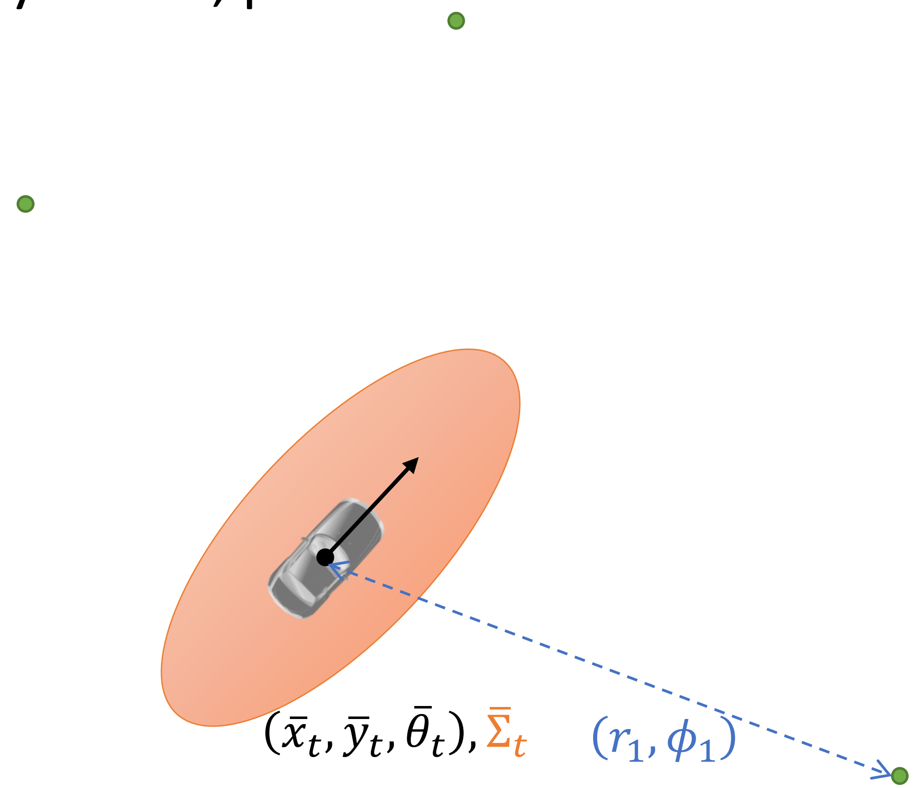
$$= \begin{array}{c} \begin{array}{ccccccccccc} &&&&&&& \text{Column } 2+2j &&& \\ &&&&&&& \downarrow &&& \end{array} \\ \begin{bmatrix} \frac{-\left(m_{j,x}-y_{1,t}\right)}{r_t^i} & \frac{-\left(m_{j,y}-y_{2,t}\right)}{r_t^i} & 0 & 0 & \cdots & 0 & \frac{m_{j,x}-y_{1,t}}{r_t^i} & \frac{m_{j,y}-y_{2,t}}{r_t^i} & 0 & \cdots & 0 \\ \frac{m_{j,y}-y_{2,t}}{\left(r_t^i\right)^2} & \frac{-\left(m_{j,x}-y_{1,t}\right)}{\left(r_t^i\right)^2} & -1 & 0 & \cdots & 0 & \frac{-\left(m_{j,y}-y_{2,t}\right)}{\left(r_t^i\right)^2} & \frac{m_{j,x}-y_{1,t}}{\left(r_t^i\right)^2} & 0 & \cdots & 0 \end{bmatrix} \end{array}$$

Alternate Form For Measurement Model Jacobian

- $\frac{\partial h^i}{\partial y_t} = \begin{bmatrix} \frac{-(m_{j,x}-y_{1,t})}{r_t^i} & \frac{-(m_{j,y}-y_{2,t})}{r_t^i} & 0 & 0 & \cdots & 0 & \frac{m_{j,x}-y_{1,t}}{r_t^i} & \frac{m_{j,y}-y_{2,t}}{r_t^i} & 0 & \cdots & 0 \\ \frac{m_{j,y}-y_{2,t}}{\left(r_t^i\right)^2} & \frac{-(m_{j,x}-y_{1,t})}{\left(r_t^i\right)^2} & -1 & 0 & \cdots & 0 & \frac{-(m_{j,y}-y_{2,t})}{\left(r_t^i\right)^2} & \frac{m_{j,x}-y_{1,t}}{\left(r_t^i\right)^2} & 0 & \cdots & 0 \end{bmatrix}$
- Rewrite: $\frac{\partial h^i}{\partial y_t} = \bar{h}_t^i F_j$
  - $\bar{h}_t^i = \begin{bmatrix} \frac{-(m_{j,x}-y_{1,t})}{r_t^i} & \frac{-(m_{j,y}-y_{2,t})}{r_t^i} & 0 & \frac{m_{j,x}-y_{1,t}}{r_t^i} & \frac{m_{j,y}-y_{2,t}}{r_t^i} \\ \frac{m_{j,y}-y_{2,t}}{\left(r_t^i\right)^2} & \frac{-(m_{j,x}-y_{1,t})}{\left(r_t^i\right)^2} & -1 & \frac{-(m_{j,y}-y_{2,t})}{\left(r_t^i\right)^2} & \frac{m_{j,x}-y_{1,t}}{\left(r_t^i\right)^2} \end{bmatrix}$
  - $F_j = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & 0 & \cdots & 0 \end{bmatrix}$

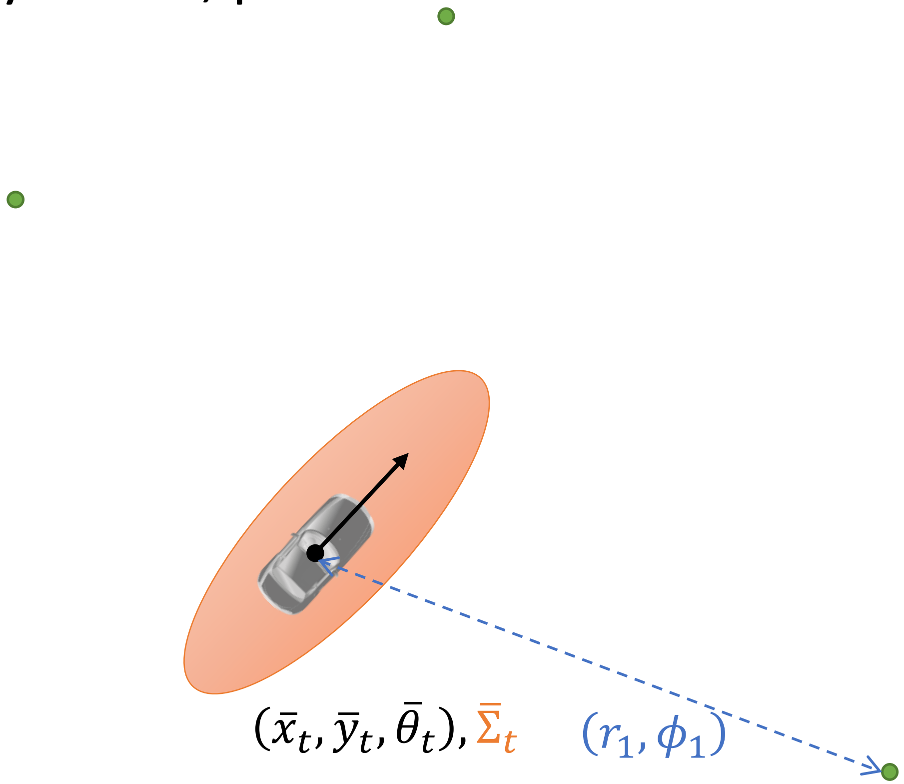
Simple Car Measurement Model

- Measurement model needs to be in the form $p(z_t|y_t, c_t)$
  - Assume independent measurements, $p(z_t|y_t) = \prod_i p\left(z_t^i \middle| y_t, c_t^i\right)$
  - At every time t, process each measurement separately/sequentially



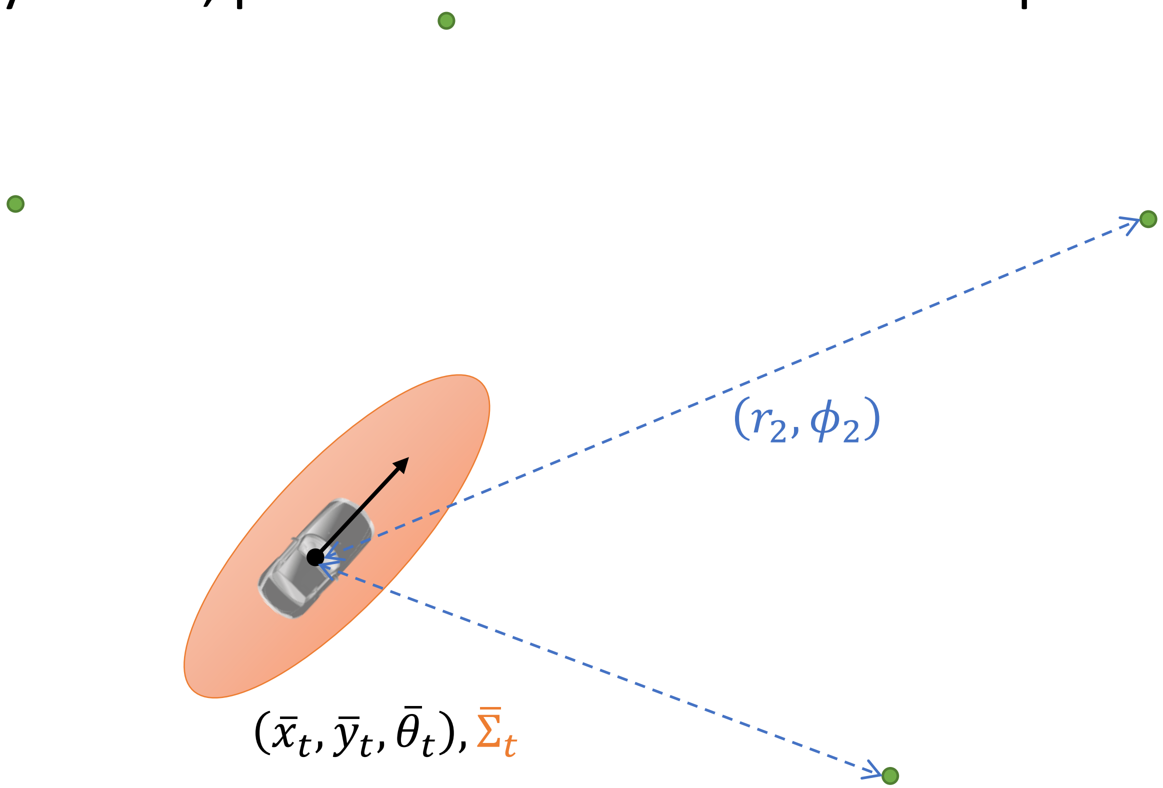
Simple Car Measurement Model

- Measurement model needs to be in the form $p(z_t|y_t, c_t)$
  - Assume independent measurements, $p(z_t|y_t) = \prod_i p\left(z_t^i|y_t, c_t^i\right)$
  - At every time t, process each measurement separately/sequentially



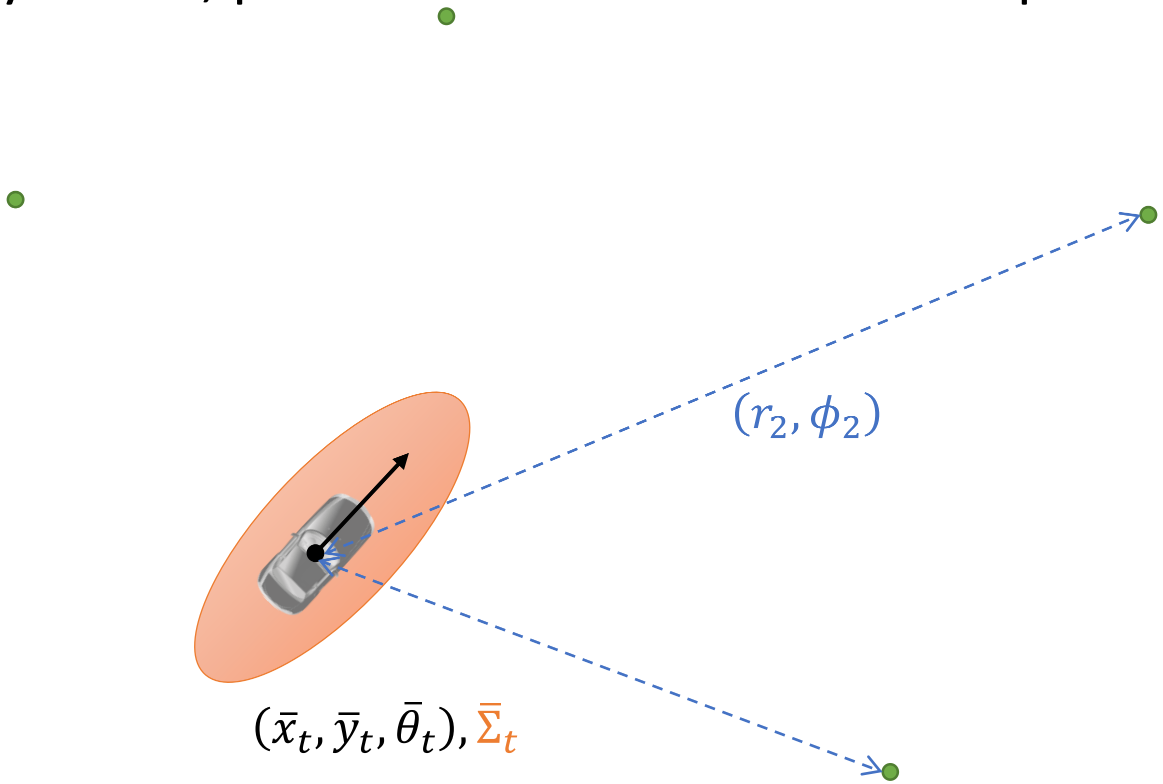
Simple Car Measurement Model

- Measurement model needs to be in the form $p(z_t|y_t, c_t)$
  - Assume independent measurements, $p(z_t|y_t) = \prod_i p\left(z_t^i \middle| y_t, c_t^i\right)$
  - At every time t, process each measurement separately/sequentially



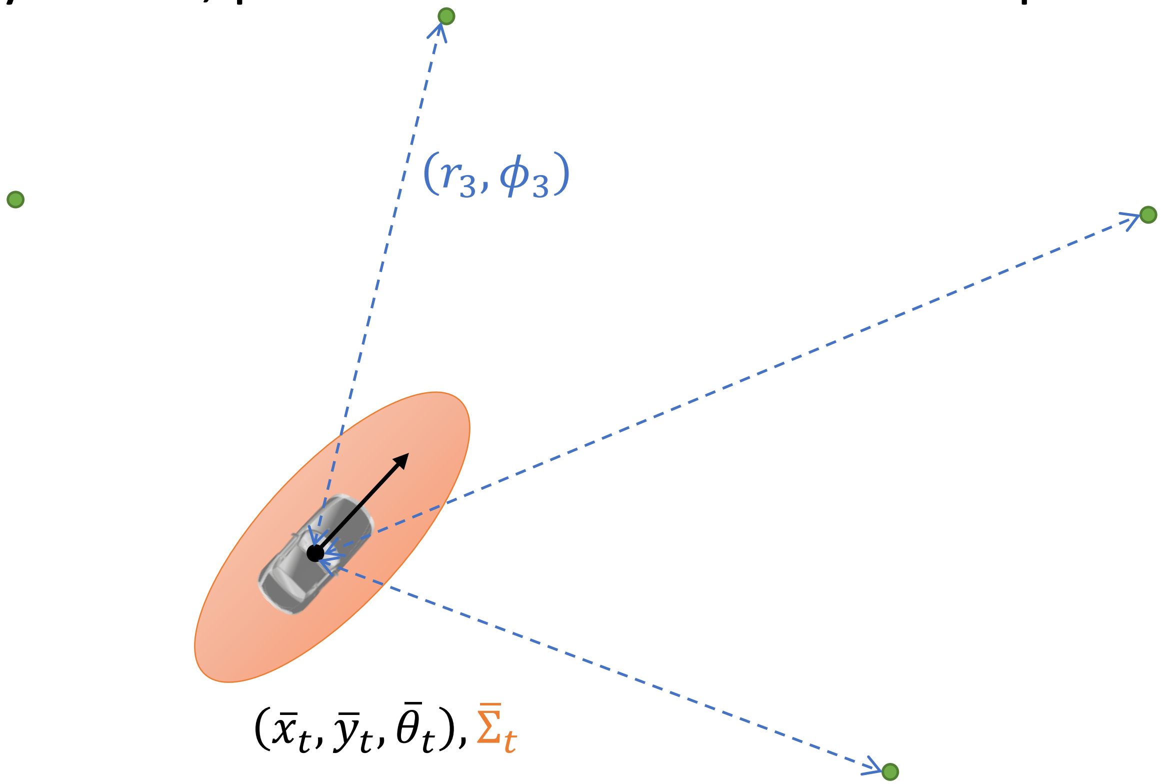
Simple Car Measurement Model

- Measurement model needs to be in the form $p(z_t|y_t, c_t)$
  - Assume independent measurements, $p(z_t|y_t) = \prod_i p\left(z_t^i \middle| y_t, c_t^i\right)$
  - At every time t, process each measurement separately/sequentially



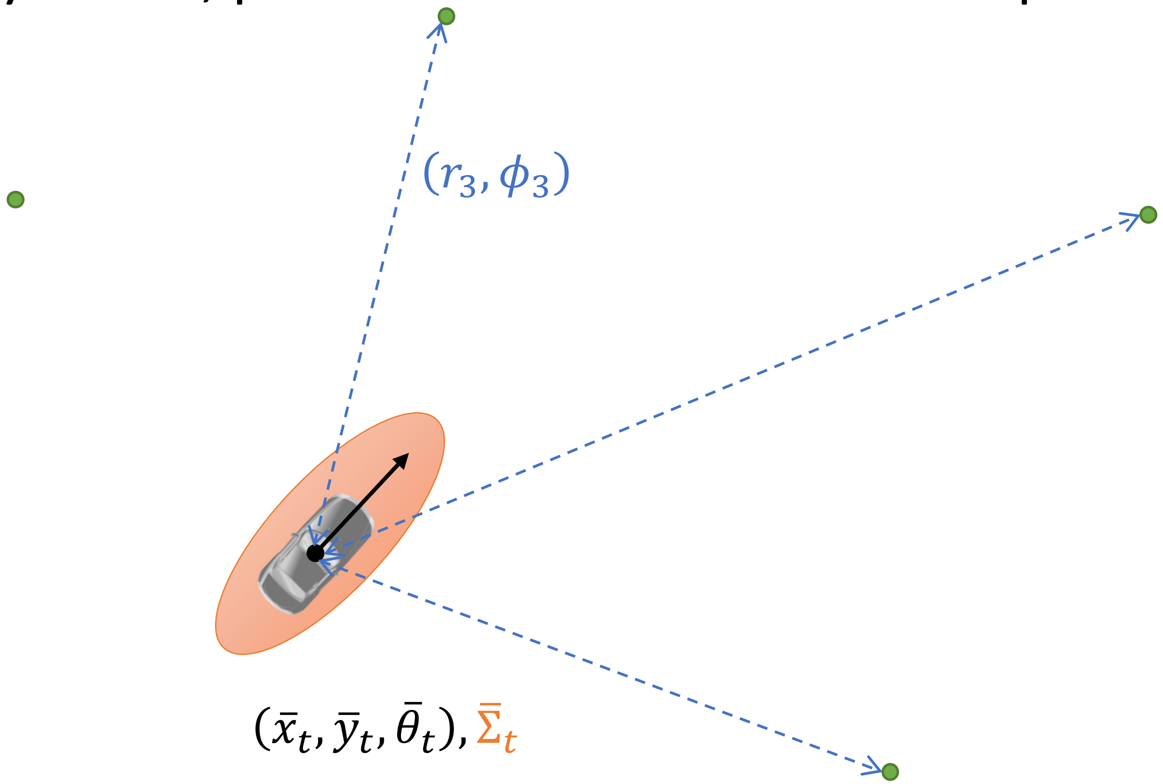
Simple Car Measurement Model

- Measurement model needs to be in the form $p(z_t|y_t, c_t)$
  - Assume independent measurements, $p(z_t|y_t) = \prod_i p\left(z_t^i \middle| y_t, c_t^i\right)$
  - At every time t, process each measurement separately/sequentially



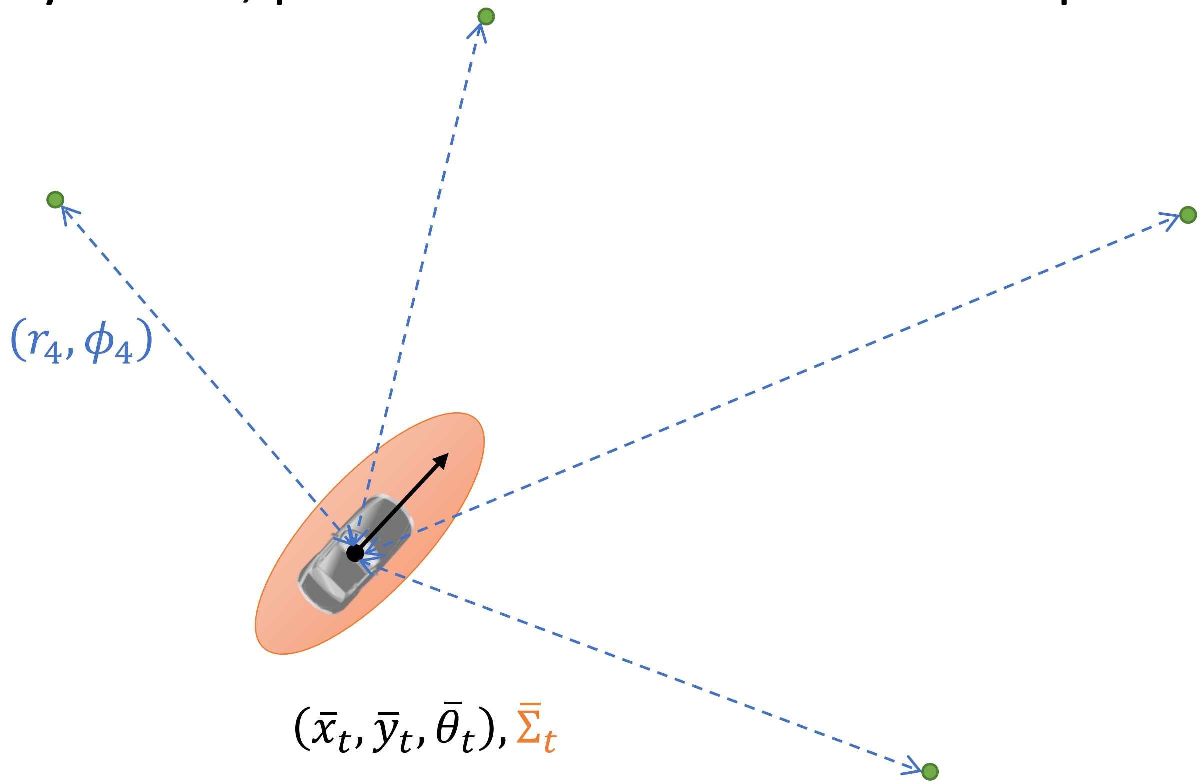
Simple Car Measurement Model

- Measurement model needs to be in the form $p(z_t|y_t, c_t)$
  - Assume independent measurements, $p(z_t|y_t) = \prod_i p(z_t^i|y_t, c_t^i)$
  - At every time t, process each measurement separately/sequentially



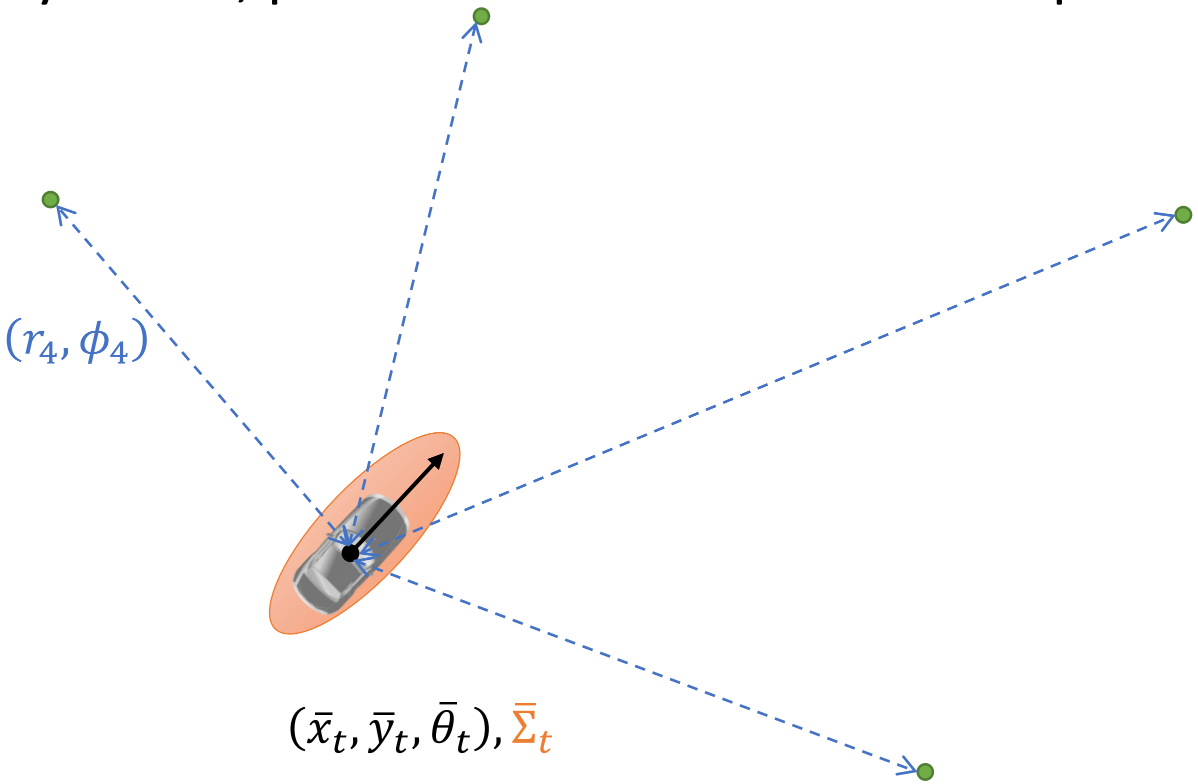
Simple Car Measurement Model

- Measurement model needs to be in the form $p(z_t|y_t, c_t)$
  - Assume independent measurements, $p(z_t|y_t) = \prod_i p\left(z_t^i \middle| y_t, c_t^i\right)$
  - At every time t, process each measurement separately/sequentially



Simple Car Measurement Model

- Measurement model needs to be in the form $p(z_t|y_t, c_t)$
  - Assume independent measurements, $p(z_t|y_t) = \prod_i p\left(z_t^i \middle| y_t, c_t^i\right)$
  - At every time t, process each measurement separately/sequentially



Extended Kalman Filter

- Extended Kalman filter algorithm:
  - $y_t = g(y_{t-1}, u_{t-1}) + \epsilon_t, \epsilon_t \sim N(0, R_t)$
  - $z_t = h(y_t) + \delta_t, \delta_t \sim N(0, Q_t)$
  - Linearization: $G_t = \nabla g(\mu_{t-1}, u_{t-1}), H_t = \nabla h(\bar{\mu}_t)$

Input: $\mu_{t-1}, \Sigma_{t-1}, u_{t-1}, z_t$

Output: μ_t, Σ_t

Perform prediction:

$$\bar{\mu}_t = g(\mu_{t-1}, u_{t-1})$$
$$\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^\top + R_t$$

Perform measurement update:

$$K_t = \bar{\Sigma}_t H_t^\top (H_t \bar{\Sigma}_t H_t^\top + Q_t)^{-1}$$
$$\mu_t = \bar{\mu}_t + K_t\big(z_t - h(\bar{\mu}_t)\big)$$
$$\Sigma_t = (I - K_t H_t)\bar{\Sigma}_t$$

Return μ_t, Σ_t

EKF SLAM prediction step details:

- For each measurement,
$$H_t^i \in \mathbb{R}^{2\times(3+2N)}$$
- Process each measurement separately
  - For each $z_t^i, i = 1, \dots, N$,

1. $K_t^i = \bar{\Sigma}_t \left(H_t^i\right)^\top \left(H_t^i \bar{\Sigma}_t \left(H_t^i\right)^\top + Q_t\right)^{-1}$
2. $\bar{\mu}_t \leftarrow \bar{\mu}_t + K_t^i\left(z_t^i - h^i(\bar{\mu}_t)\right)$
3. $\bar{\Sigma}_t \leftarrow \left(I - K_t^i H_t^i\right)\bar{\Sigma}_t$

  - Above computation is done based on land mark $j = c_t^i$ (assuming known correspondence)

- At the end, set $\mu_t = \bar{\mu}_t, \Sigma_t = \bar{\Sigma}_t$

EKF SLAM

Preliminary steps

- Initialize μ_0, Σ_0
- Define dynamics g for augmented state y
- Define R_t for augmented state y
  - Land mark position estimates can be initialized to anything, since variance is infinite
- Calculate Jacobians G_t, H_t

Input: $\mu_{t-1}, \Sigma_{t-1}, u_{t-1}, z_t$

Output: μ_t, Σ_t

Perform prediction:

$$\bar{\mu}_t = g(\mu_{t-1}, u_{t-1})$$
$$\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^\top + R_t$$

Perform measurement update:

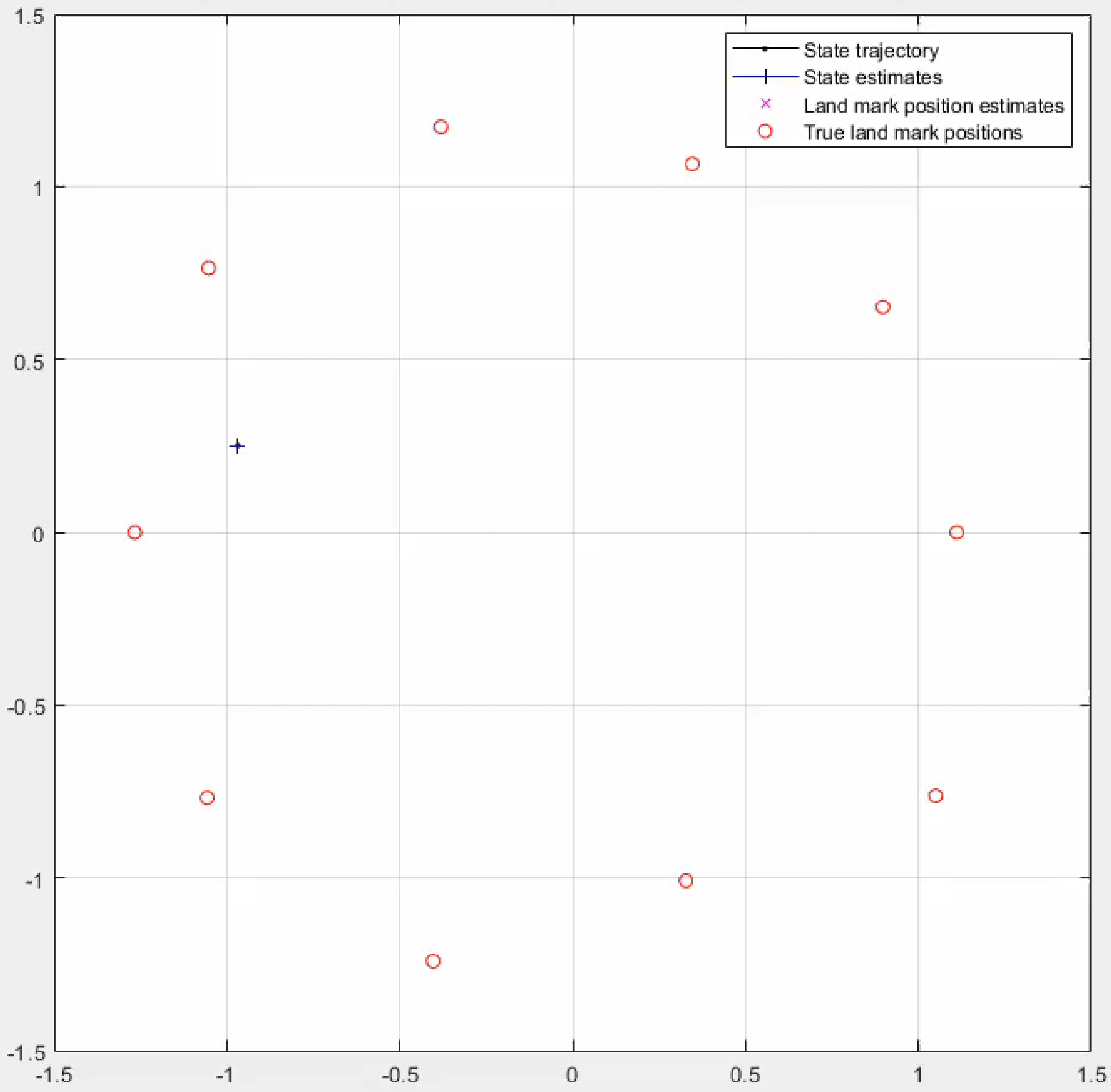
- For each $z_t^i = \left(r_t^i, \phi_t^i\right), i = 1, \ldots, N,$

1. $j = c_t^i$
2. If land mark j has not been seen, then

$$\begin{bmatrix} \bar{\mu}_{2+2j,t} \\ \bar{\mu}_{3+2j,t} \end{bmatrix} = \begin{bmatrix} \bar{\mu}_{1,t} + r_t^i \cos\left(\phi_t^i + \bar{\mu}_{3,t}\right) \\ \bar{\mu}_{2,t} + r_t^i \sin\left(\phi_t^i + \bar{\mu}_{3,t}\right) \end{bmatrix}$$

3. $K_t^i = \bar{\Sigma}_t \left(H_t^i\right)^\top \left(H_t^i \bar{\Sigma}_t \left(H_t^i\right)^\top + Q_t\right)^{-1}$
4. $\bar{\mu}_t = \bar{\mu}_t + K_t^i \left(z_t^i - h^i(\bar{\mu}_t)\right)$
5. $\bar{\Sigma}_t = \left(I - K_t^i H_t^i\right) \bar{\Sigma}_t$

Return $\mu_t = \bar{\mu}_t, \Sigma_t = \bar{\Sigma}_t$



State trajectory
State estimates
Land mark position estimates
True land mark positions
1.5
1
0.5
0
-0.5
-1
-1.5
-1.5
-1
-0.5
0
0.5
1
1.5

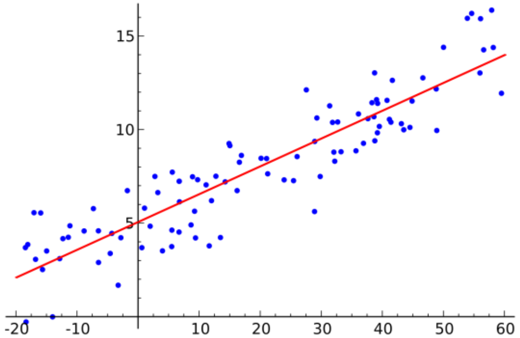
EKF SLAM: Discussion

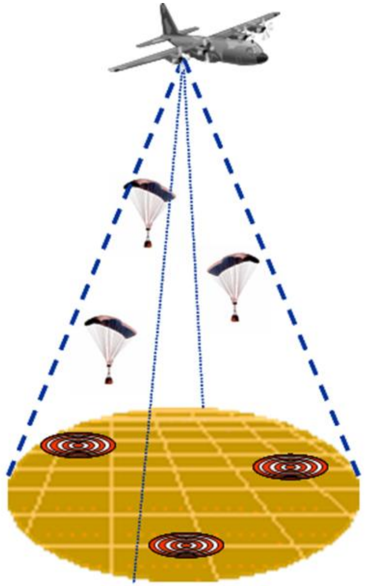
- Computational complexity: $O(N^2)$, where N is the number of land marks
- Best for feature-based maps, due to small N

- Unknown correspondences
  - Use maximum likelihood to estimate which land mark is being observed
  - Add new land mark if none of the existing land marks are likely
  - Can produce duplicates of the same land mark
  - Can incorporate more advanced techniques such as outlier rejection, or make land marks more distinct

- Accurate SLAM prefers dense maps (large N), but computation becomes expensive
- Nonparametric filters (eg. Particle filters) are popular with occupancy grids

Finished!





- Overview of algorithms used for robotic decision making
  - Theory-focused
  - Fundamentals for doing many areas of robotics research

- Dynamical systems
- Nonlinear optimization and optimal control
- Reachability analysis
- Reinforcement Learning
- Localization and mapping