

SCE5880  
Algoritmos de Estimação para  
Robótica Móvel

---

SLAM  
Localização e Mapeamento  
Simultâneos

1

Localização e Mapemanto Simultâneos

---

**O SLAM é um dos mais importantes problemas na área da robótica móvel.**

Dados:

- Deslocamento do robô
- Informações obtidas pelos sensores

Estimar:

- O mapa do ambiente
- A posição/trajeto do robô

---

SCE-5868 Denis F. Wolf

2

SLAM Summer School

---

SCE-5868 Denis F. Wolf

3

Aplicações

Indoors Undersea  
Space Underground

SLAM - problemas

---

Incerteza na localização é propogada no mapa

---

SCE-5868 Denis F. Wolf

5

SLAM - problemas

---

Erros na associação de referências no mapa são catastróficos.

---

SCE-5868 Denis F. Wolf

6

### Online SLAM

$p(x_t, m | z_{1:t}, u_{1:t})$

SCE-5868 Denis F. Wolf 7

### Full SLAM

$p(x_t, m | z_{1:t}, u_{1:t})$

SCE-5868 Denis F. Wolf 8

### Scan Matching

$$\hat{x}_t = \operatorname{argmax} \{ p(z_t | x_t, \hat{m}^{[t-1]}) \cdot p(x_t | u_{t-1}, \hat{x}_{t-1}) \}$$

- Estima a posição com a maior probabilidade, dada a posição anterior e o mapa.
- A posição é estimada maximizando a sobreposição das informações dos sensores com o mapa obtido até o momento.
- Utiliza a posição estimada para continuar a construção do mapa.

SCE-5868 Denis F. Wolf 9

### Mapeamento baseado apenas na odometria

SCE-5868 Denis F. Wolf 10

### Scan Matching

SCE-5868 Denis F. Wolf 11

### Scan Matching

**Problemas com o Scan Matching:**

- Pequenos erros vão se acumulando na trajetória do robô, e nunca são reparados.
- Pode apresentar inconsistências quando o robô completa um loop.

SCE-5868 Denis F. Wolf 12

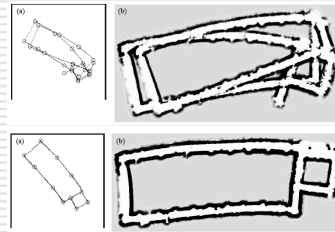
## Expectation Maximization (EM)

- Utiliza **landmarks** como referência
- Utiliza o algoritmo EM para estimar a trajetória completa do robô.
- E = estima a trajetória do robô baseado no mapa atual.
- M = estima o mapa baseado na trajetória estimada.
- O algoritmo consiste em alternar as etapas E e M.

SCE-5868 Denis F. Wolf

13

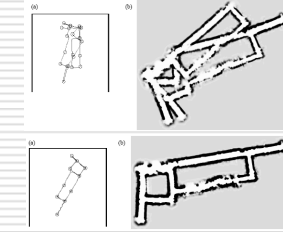
## EM SLAM



SCE-5868 Denis F. Wolf

14

## EM SLAM



SCE-5868 Denis F. Wolf

15

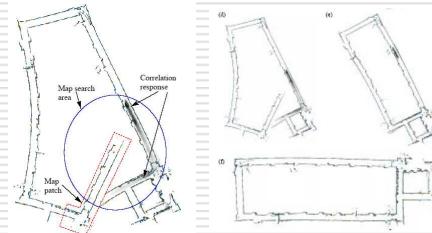
## Incremental Mapping

- O mapa é construído de maneira incremental, utilizando scan matching, até que um loop seja detectado.
- Quando um loop é detectado, tenta-se localizar o robô dentro da área já mapeada.
- Depois que o robô é localizado, é calculada o erro odométrico.
- Com base no erro, todas a trajetória do robô é corrigida proporcionalmente.
- O mapa é reconstruído.

SCE-5868 Denis F. Wolf

16

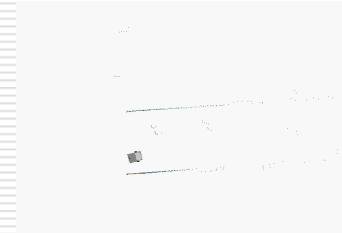
## Incremental Mapping



SCE-5868 Denis F. Wolf

17

## Incremental Mapping – Grid



SCE-5868 Denis F. Wolf

18

### Incremental Mapping – Monte Carlo

SCE-5868 Denis F. Wolf 19

### Incremental Mapping

**Problemas com o Incremental Mapping:**

- A localização fica mais difícil a medida que as trajetórias ficam mais longas.
- A correção proporcional pode não funcionar em ambientes/trajetórias complexas.

SCE-5868 Denis F. Wolf 20

### EKF SLAM

- Consiste em estimar a posição do robô e dos landmarks com o filtro de Kalman.
- As posição do robô e dos landmarks são correlacionadas através da matriz de covariância.

SCE-5868 Denis F. Wolf 21

### EKF SLAM

SCE-5868 Denis F. Wolf 22

### EKF SLAM

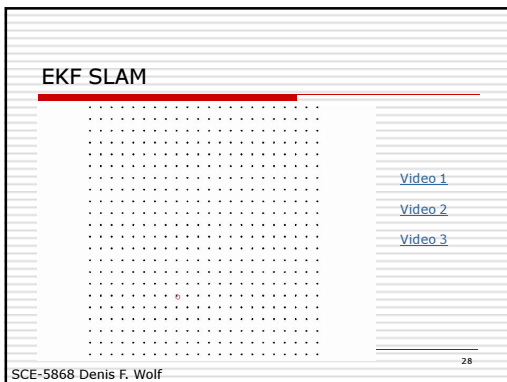
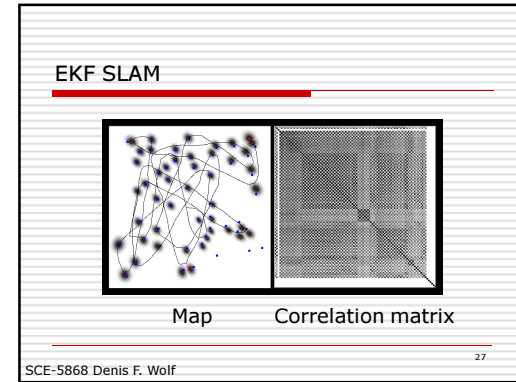
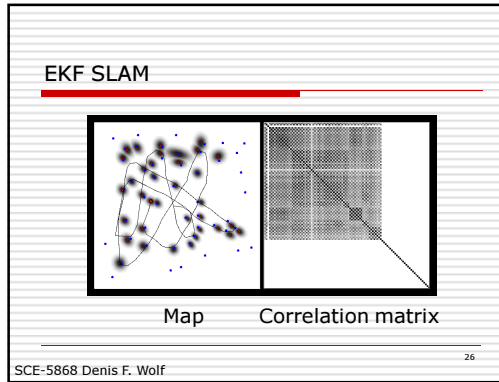
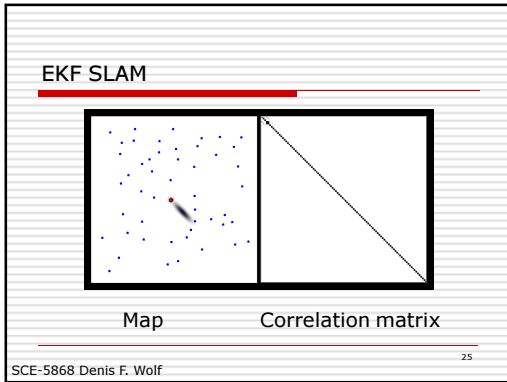
$$Bel(x_t, m) = \begin{pmatrix} x \\ y \\ \theta \\ l_1 \\ l_2 \\ \vdots \\ l_N \end{pmatrix} \begin{pmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{x\theta} & \sigma_{x1} & \sigma_{x2} & \dots & \sigma_{xN} \\ \sigma_{xy} & \sigma_y^2 & \sigma_{y\theta} & \sigma_{y1} & \sigma_{y2} & \dots & \sigma_{yN} \\ \sigma_{x\theta} & \sigma_{y\theta} & \sigma_\theta^2 & \sigma_{\theta1} & \sigma_{\theta2} & \dots & \sigma_{\theta N} \\ \sigma_{x1} & \sigma_{x2} & \sigma_{x\theta} & \sigma_{l1}^2 & \sigma_{l12} & \dots & \sigma_{l1N} \\ \sigma_{y1} & \sigma_{y2} & \sigma_{y\theta} & \sigma_{l12} & \sigma_{l2}^2 & \dots & \sigma_{l2N} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \sigma_{xN} & \sigma_{yN} & \sigma_{\theta N} & \sigma_{l1N} & \sigma_{l2N} & \dots & \sigma_N^2 \end{pmatrix}$$

- Um mapa com N landmarks estimará 2N+3 variáveis e terá uma matriz de covariância de 2N+3 x 2N+3 elementos.

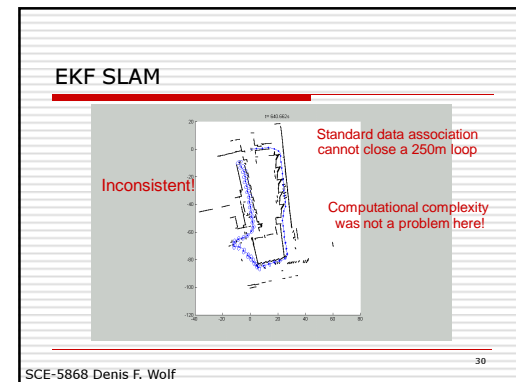
SCE-5868 Denis F. Wolf 23

### EKF SLAM

1. Algorithm **Kalman\_filter**(  $\mu_{t-1r}$ ,  $\Sigma_{t-1r}$ ,  $u_{tr}$ ,  $z_t$  ):
2. Prediction:
3.  $\bar{\mu}_t = A_t \mu_{t-1} + B_t u_{tr}$
4.  $\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$
5. Correction:
- 6.
7.  $K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$
8.  $\mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t)$
9.  $\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t$
10. Return  $\mu_t$ ,  $\Sigma_t$



- ### EKF SLAM
- Problemas com o EKF SLAM:**
- Complexidade  $O(n^2)$  em relação ao número de landmarks
  - Operações com matrizes grandes têm grande custo computacional.
  - Pode divergir em situações de grande não-linearidade
- SCE-5868 Denis F. Wolf 29



## EKF SLAM

---

### **Conclusão:**

- O EKF SLAM têm se mostrado extremamente robusto em situações reais.
- Existem diversas otimizações possíveis para se reduzir o custo computacional