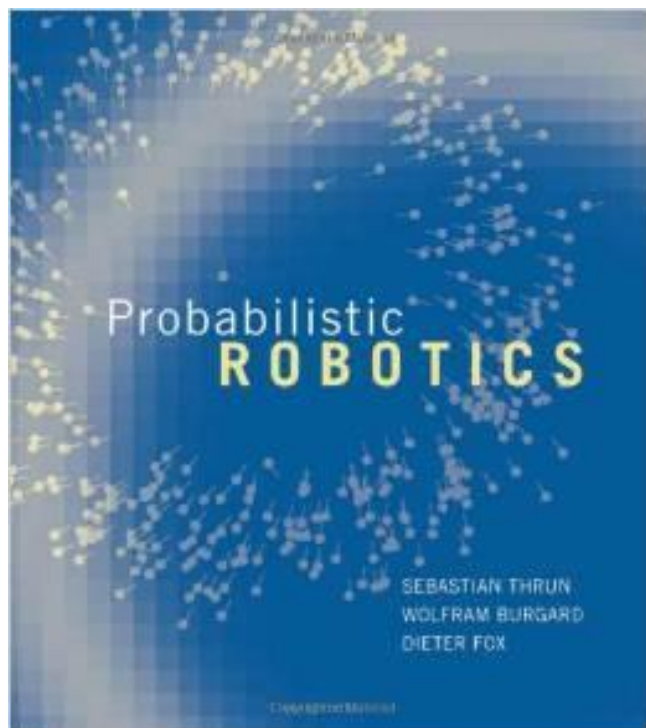


参考図書



Probabilistic Robotics

Sebastian Thrun, Wolfram Burgard and Dieter Fox

PDFファイルをインターネットで読むことができる.



確率ロボティクス

Sebastian Thrun (著),
Wolfram Burgard (著),
Dieter Fox (著),
上田 隆一 (翻訳) 2007.

12.1 制御系の状態推定にパーティクルフィルタを導入する

一期先予測

運動モデル

どのパラメータにどのような誤差が混入するのかを考慮した離散時間モデルを構築する必要がある。

粒子 $\mathbf{x}_{k-1}^{(l)}$ を $f(\mathbf{x}_k | \mathbf{x}_{k-1}^{(l)}, \mathbf{u}_{k-1})$ に従って状態推移させ

粒子集合 $\{\hat{\mathbf{x}}_k^{(l)} | l = 1, 2, \dots, N\}$ を発生させる。

る波



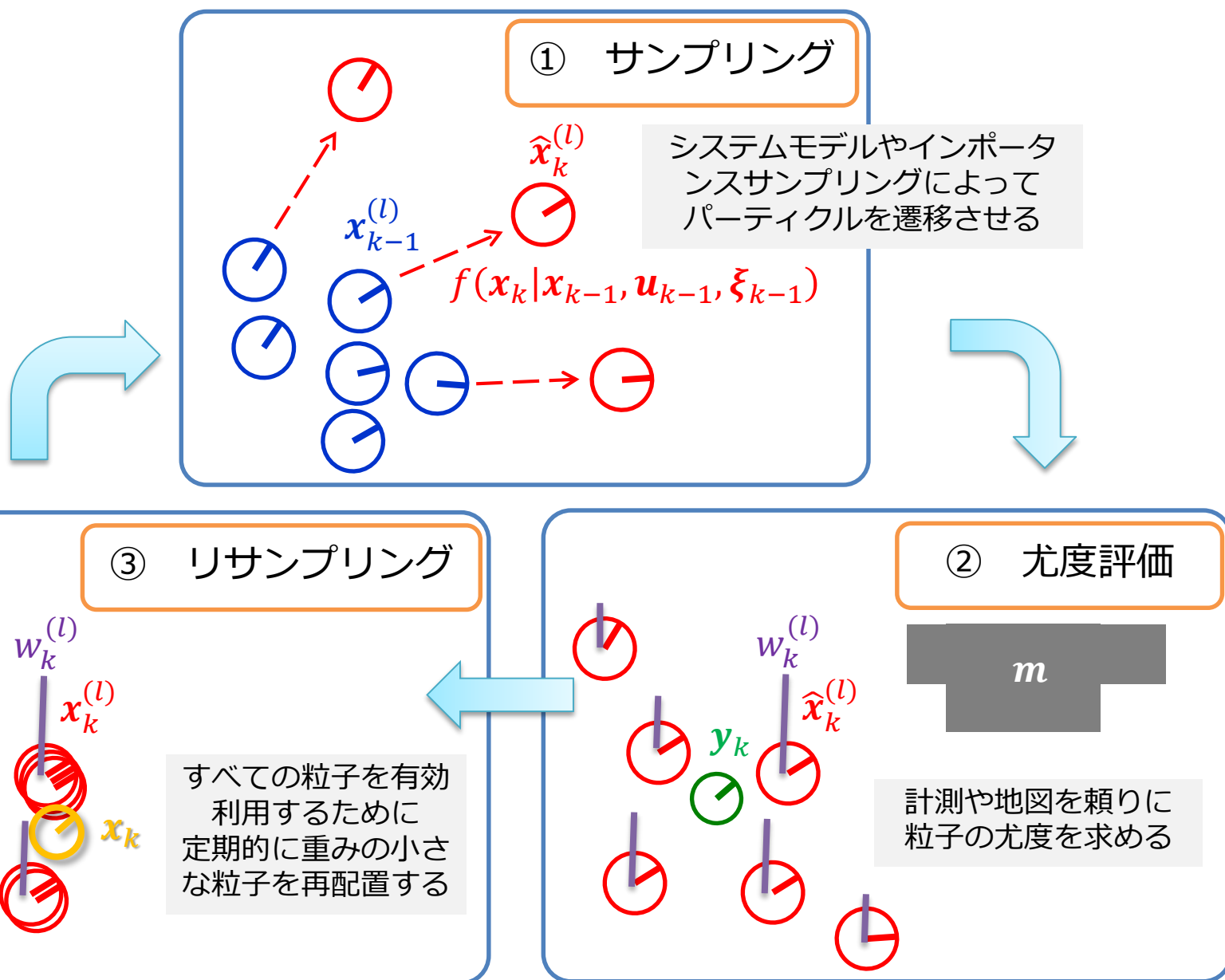
センサモデル

センサの計測誤差がどのように生ずるのかを確率的にモデル化する。

3.1 尤度計算

粒子 $\hat{\mathbf{x}}_k^{(l)}$ の尤度 $w_k^{(l)} = h(\mathbf{y}_k | \hat{\mathbf{x}}_k^{(l)}, \mathbf{m})$ を計算する。

12.1 制御系の状態推定にパーティクルフィルタを導入する



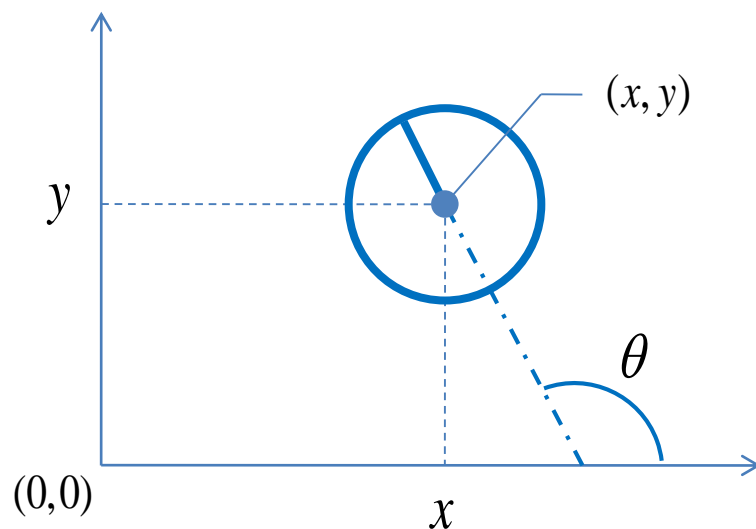
12.2 平面移動ロボットの確率状態方程式

例題として、最も単純な平面移動ロボットの機構のモデルを考える。

ロボットの状態

$$\mathbf{x}_k = (x_k \ y_k \ \theta_k)^T$$

時刻 k における慣性座標系
における位置と方向



ロボットへの入力

$$\mathbf{u}_k = (v_k \ \omega_k)^T$$

v_k : 時刻 k における並進速度

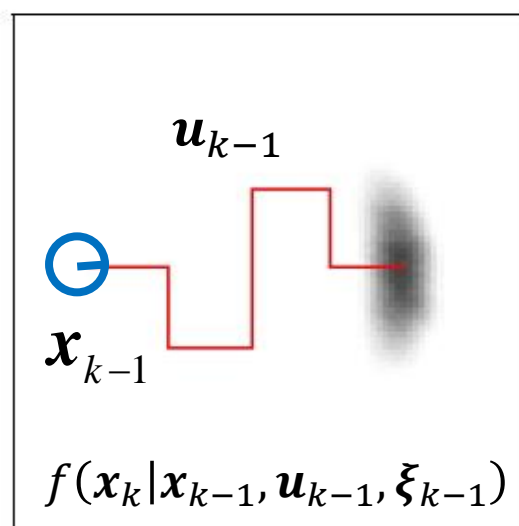
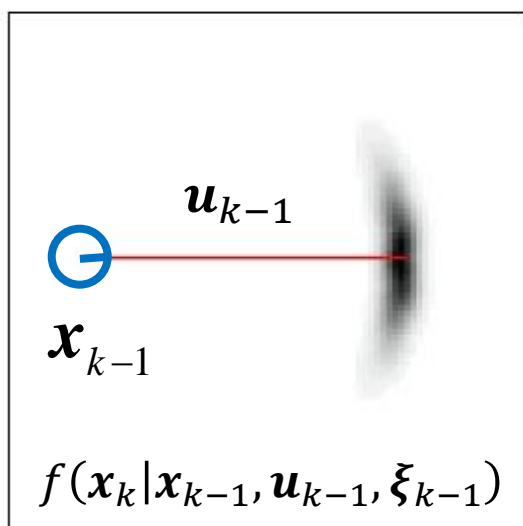
ω_k : 時刻 k における角速度



システムモデル

$$f(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \xi_{k-1})$$

時刻 $k-1$ におけるロボットの状態 \mathbf{x}_{k-1} に、制御入力 \mathbf{u}_{k-1} が実行され、システムノイズ ξ_{k-1} が印可された場合のロボットの姿勢の事後確率分布



実線で描かれた動作コマンドを実行した後のロボットの姿勢の事後分布。色が濃いほど尤度が高いことを示している。

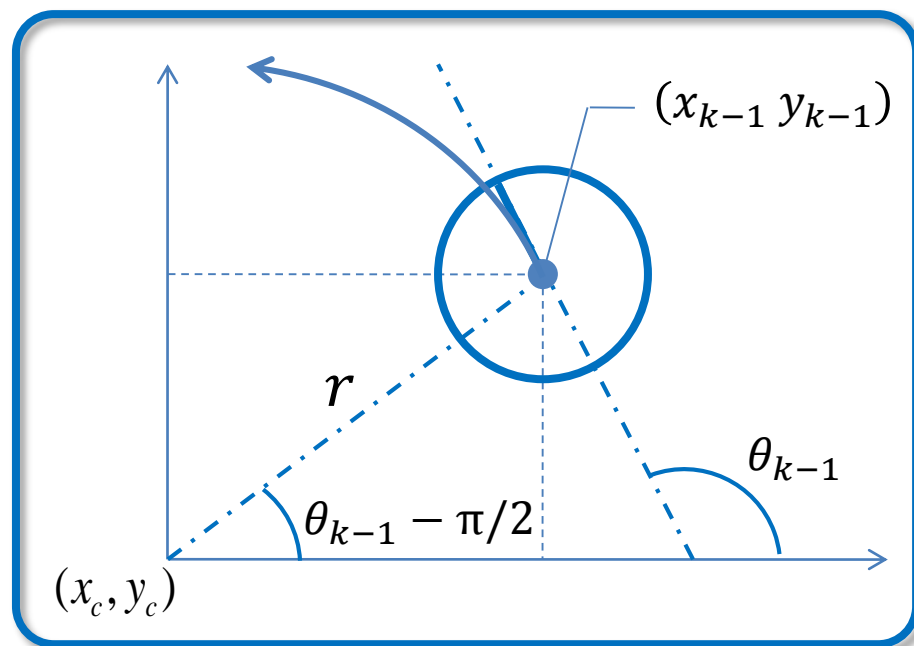
確率分布は2次元写像されているが、本来はロボットの方向も考慮した三次元の分布である。

複雑な動作コマンドが与えられるほど不確かさは広がる。

統計・試行錯誤でシステムノイズ ξ_k をモデル化する必要がある。

12.2 平面移動ロボットの確率状態方程式

ロボットの動作モデルの解析的導出



① $v_k = \omega_k \cdot r$ より

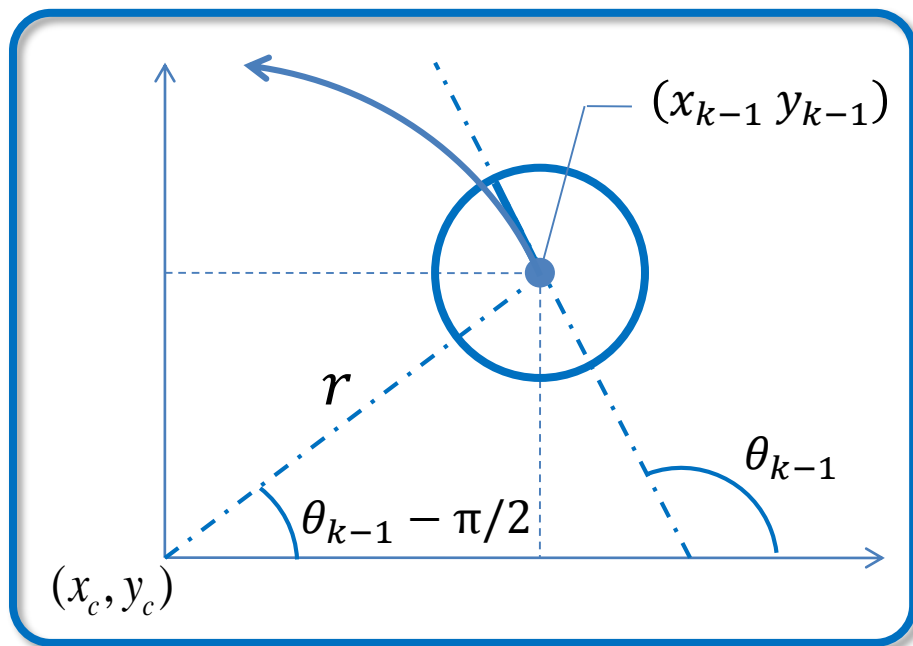
$$r = \left| \frac{v_k}{\omega_k} \right|$$

② $\mathbf{x}_{k-1} = (x_{k-1} \ y_{k-1} \ \theta_{k-1})^T$ を初期姿勢とみなし Δt の間ロボットの入力を $(v_{k-1} \ \omega_{k-1})^T$ に保つとき, 回転円の中心 $(x_c \ y_c)^T$ を以下のように定めることができる。

$$x_c = x_{k-1} - \frac{v_{k-1}}{\omega_{k-1}} \sin \theta_{k-1}, \quad y_c = y_{k-1} + \frac{v_{k-1}}{\omega_{k-1}} \cos \theta_{k-1}$$

12.2 平面移動ロボットの確率状態方程式

ロボットの動作モデルの解析的導出



③

これらより移動時間 Δt の後、ロボットの姿勢 $\mathbf{x}_k = (x_k \ y_k \ \theta_k)^T$ は

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} x_c + \frac{v_{k-1}}{\omega_{k-1}} \sin(\theta_{k-1} + \omega_{k-1} \Delta t) \\ y_c - \frac{v_{k-1}}{\omega_{k-1}} \cos(\theta_{k-1} + \omega_{k-1} \Delta t) \\ \theta_{k-1} + \omega_{k-1} \Delta t \end{bmatrix}$$

$$= \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} + \begin{bmatrix} -\frac{v_{k-1}}{\omega_{k-1}} \sin \theta_{k-1} + \frac{v_{k-1}}{\omega_{k-1}} \sin(\theta_{k-1} + \omega_{k-1} \Delta t) \\ -\frac{v_{k-1}}{\omega_{k-1}} \cos(\theta_{k-1} + \omega_{k-1} \Delta t) \\ \omega_{k-1} \Delta t \end{bmatrix}$$

12.2 平面移動ロボットの確率状態方程式

- ④ ロボットの動作に加わる雑音を、平均値が0で有限な分布に従う確率変数でモデル化する。

$$\begin{bmatrix} v_k \\ \omega_k \end{bmatrix} = \begin{bmatrix} \varepsilon_{\alpha_1} v^2 + \alpha_2 \omega^2 \\ \varepsilon_{\alpha_3} v^2 + \alpha_4 \omega^2 \end{bmatrix} \quad \varepsilon_{ab^2} = \frac{1}{\sqrt{2\pi b^2}} \exp\left(-\frac{1}{2} \frac{a^2}{b^2}\right)$$

$\alpha_1 \sim \alpha_4$ はロボットに固有な誤差パラメータ。ロボット動作の不正確さを示しており、不正確なほど値が大きくなる。


- ⑤ 回転半径に加わる雑音を、上と同様にモデル化する。

$$\theta_k = \theta_{k-1} + \hat{\omega}_{k-1} \Delta t + \hat{\gamma} \Delta t \quad \hat{\gamma} = \varepsilon_{\alpha_5} v^2 + \alpha_6 \omega^2$$

α_5 と α_6 は、回転雑音の標準偏差を決定するためのロボット固有のパラメータ。

- ⑥ すべて考慮した最終的なロボットの動作モデルは

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} + \begin{bmatrix} -\frac{\hat{v}_{k-1}}{\hat{\omega}_{k-1}} \sin \theta_{k-1} + \frac{\hat{v}_{k-1}}{\hat{\omega}_{k-1}} \sin(\theta_{k-1} + \hat{\omega}_{k-1} \Delta t) \\ \frac{\hat{v}_{k-1}}{\hat{\omega}_{k-1}} \cos \theta_{k-1} - \frac{\hat{v}_{k-1}}{\hat{\omega}_{k-1}} \cos(\theta_{k-1} + \hat{\omega}_{k-1} \Delta t) \\ \hat{\omega}_{k-1} \Delta t + \hat{\gamma} \Delta t \end{bmatrix}$$

 $f(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \boldsymbol{\xi}_{k-1})$ 8

12.2 平面移動ロボットの確率状態方程式

表 1

姿勢 $\mathbf{x}_{k-1} = (x_{k-1}, y_{k-1}, \theta_{k-1})^T$ と制御入力 $\mathbf{u}_{k-1} = (v_{k-1}, \omega_{k-1})^T$ から姿勢 $\mathbf{x}_k = (x_k, y_k, \theta_k)^T$ をサンプリングするためのアルゴリズム

1 Algorithm sample_motion_velocity

2 $\hat{v}_{k-1} = v_{k-1} + \text{sample}(\alpha_1 v_{k-1}^2 + \alpha_2 \omega_{k-1}^2)$ 注 1

3 $\hat{\omega}_{k-1} = \omega_{k-1} + \text{sample}(\alpha_3 v_{k-1}^2 + \alpha_4 \omega_{k-1}^2)$

4 $\hat{\gamma}_{k-1} = \text{sample}(\alpha_5 v_{k-1}^2 + \alpha_6 \omega_{k-1}^2)$

5 $x_k = x_{k-1} - \frac{\hat{v}_{k-1}}{\hat{\omega}_{k-1}} \sin \theta_{k-1} + \frac{\hat{v}_{k-1}}{\hat{\omega}_{k-1}} \sin(\theta_{k-1} + \hat{\omega}_{k-1} \Delta t)$

6 $y_k = y_{k-1} + \frac{\hat{v}_{k-1}}{\hat{\omega}_{k-1}} \cos \theta_{k-1} - \frac{\hat{v}_{k-1}}{\hat{\omega}_{k-1}} \cos(\theta_{k-1} + \hat{\omega}_{k-1} \Delta t)$

7 $\theta_k = \theta_{k-1} + \hat{\omega}_{k-1} \Delta t + \hat{\gamma}_{k-1} \Delta t$ 注 2

8 return $\mathbf{x}_k = (x_k, y_k, \theta_k)^T$

注 1

sample(b)は、原点を中心とする標準偏差 b の分布からランダム標本を生成するための関数。

この関数は、例えば 表 2 のようなアルゴリズムで実装される。

注 2

乱数の項 $\hat{\gamma}$ を加えて事後状態に摂動を加えていることに注意。

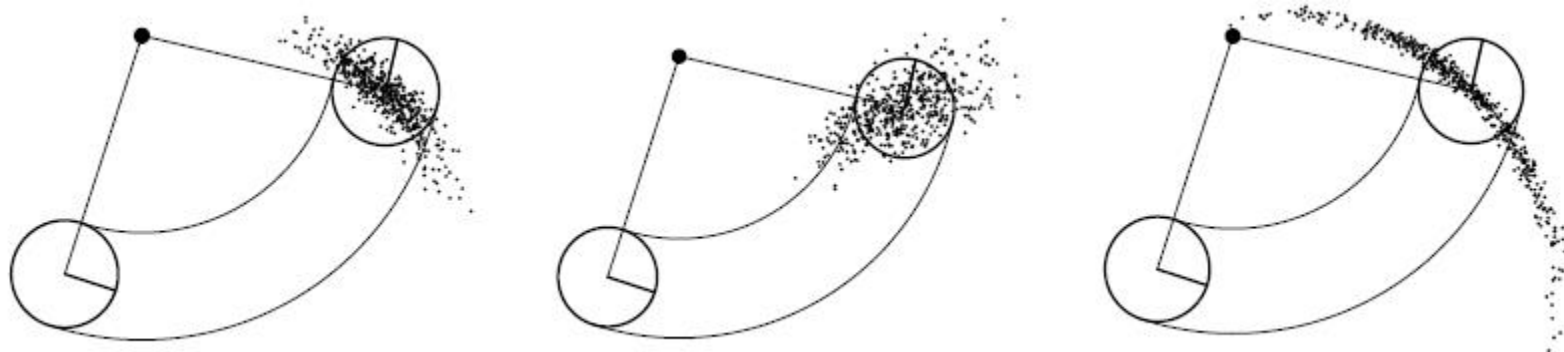
α_1 から α_6 は動作の雑音パラメータである。

表2 原点を中心とする正規分布について確率密度を計算するアルゴリズム

- 1 Algorithm sample_normal_distribution (b^2)
- 2 return $\frac{1}{2} \sum_{i=1}^{12} \text{rand}(-b, b)$

正規分布からのサンプリングの近似として一般的なアルゴリズム。
この近似は中心極限定理を利用したもの。

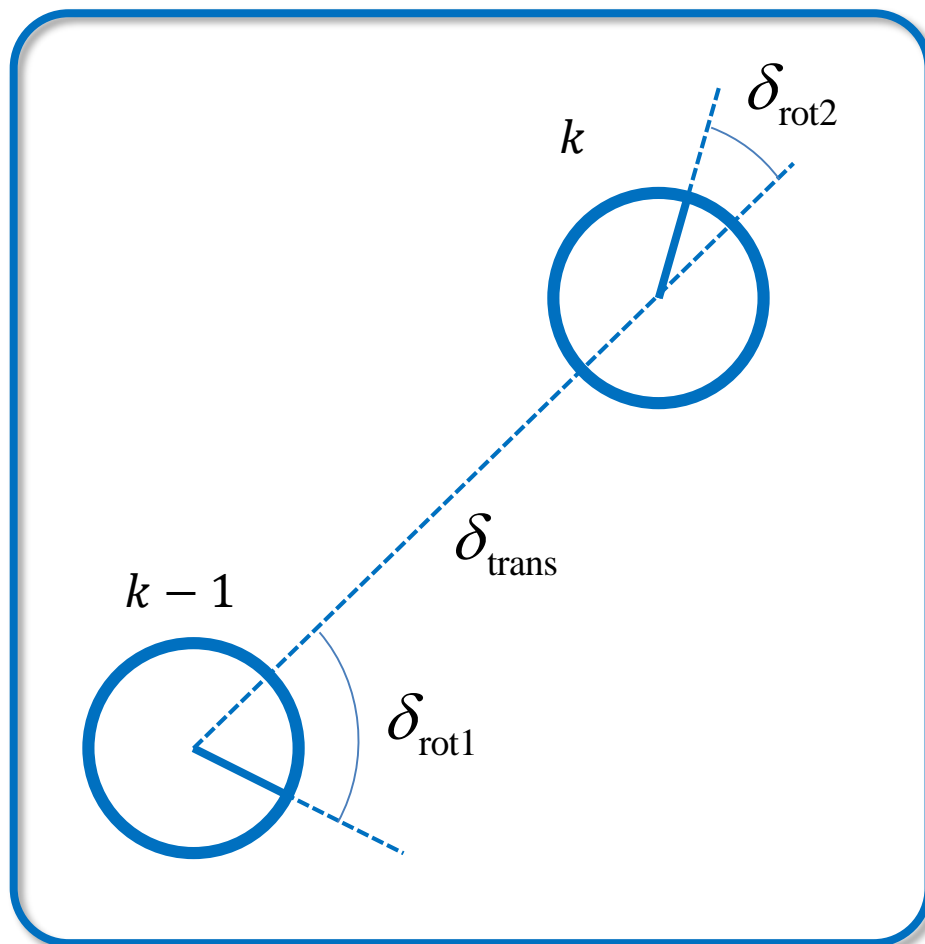
12個の一樣分布からの値を平均することで正規分布に従う値を近似的に生成するが
厳密には生成される値は $[-2b, 2b]$ 内に限定される。



速度動作モデルによるサンプリングの例。標本数はそれぞれ500。
パラメータによって動作の摂動は様々に変化する。

12.2 平面移動ロボットの確率状態方程式

オドメトリ動作モデル



オドメトリ情報

$$\bar{\mathbf{x}}_{k-1} = (\bar{x}_{k-1}, \bar{y}_{k-1}, \bar{\theta}_{k-1})^T$$

$$\bar{\mathbf{x}}_k = (\bar{x}_k, \bar{y}_k, \bar{\theta}_k)^T$$

時間 $(k-1, k]$ においてロボットは
姿勢 $\bar{\mathbf{x}}_{k-1}$ から姿勢 $\bar{\mathbf{x}}_k$ まで進む。

動作情報

$$\mathbf{u}_{k-1} = \begin{bmatrix} \bar{\mathbf{x}}_{k-1} \\ \bar{\mathbf{x}}_k \end{bmatrix}$$

オドメトリ情報に基づいて $p(\mathbf{x}_k | \mathbf{u}_{k-1}, \mathbf{x}_{k-1})$
からシステムノイズを考慮した
 $\mathbf{x}_k = (x_k, y_k, \theta_k)^T$ をサンプリング
するにはどうすればいい？



ロボットの動作を,
回転 δ_{rot1} , 直進 δ_{trans} , 回転 δ_{rot2}
の三つに変換して考える。

12.2 平面移動ロボットの確率状態方程式

①

$$\delta_{\text{rot1}} = \text{atan2}(\bar{y}_k - \bar{y}_{k-1}, \bar{x}_k - \bar{x}_{k-1})$$

$$\delta_{\text{trans}} = \sqrt{(\bar{x}_k - \bar{x}_{k-1})^2 + (\bar{y}_k - \bar{y}_{k-1})^2}$$

$$\delta_{\text{rot2}} = \bar{\theta}_k - \bar{\theta}_{k-1} - \delta_{\text{rot1}}$$

②

$$\hat{\delta}_{\text{rot1}} = \delta_{\text{rot1}} - \varepsilon_{\alpha_1} \delta_{\text{rot1}}^2 + \alpha_2 \delta_{\text{trans}}^2$$

$$\hat{\delta}_{\text{trans}} = \delta_{\text{trans}} - \varepsilon_{\alpha_3} \delta_{\text{trans}}^2 + \alpha_4 \delta_{\text{rot1}}^2 + \alpha_4 \delta_{\text{rot2}}^2$$

$$\hat{\delta}_{\text{rot2}} = \delta_{\text{rot2}} - \varepsilon_{\alpha_1} \delta_{\text{rot2}}^2 + \alpha_2 \delta_{\text{trans}}^2$$

③

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} + \begin{bmatrix} \hat{\delta}_{\text{trans}} \cos(\theta_{k-1} + \hat{\delta}_{\text{rot1}}) \\ \hat{\delta}_{\text{trans}} \sin(\theta_{k-1} + \hat{\delta}_{\text{rot1}}) \\ \hat{\delta}_{\text{rot1}} + \hat{\delta}_{\text{rot2}} \end{bmatrix}$$

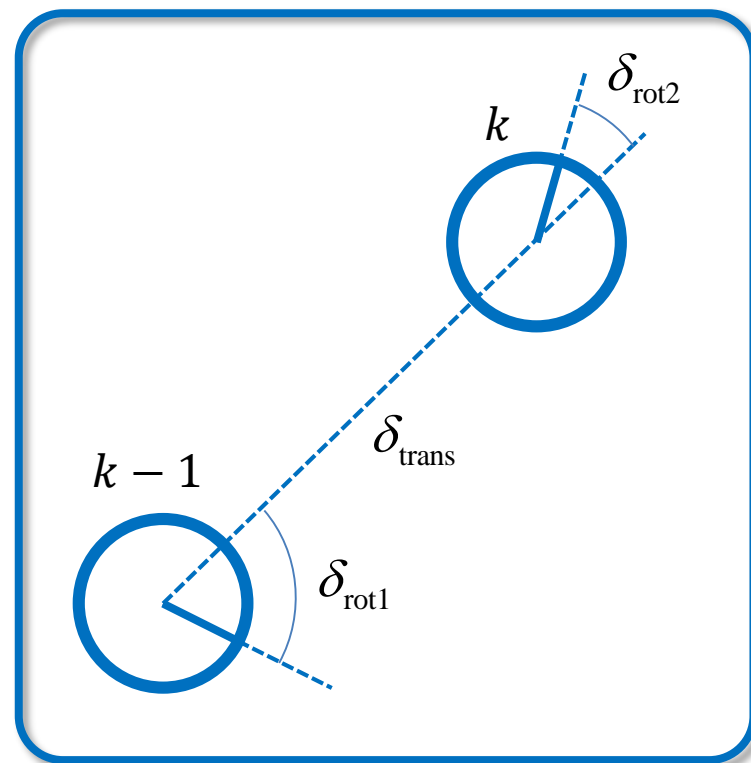


表 3

オドメトリ情報に基づいて $p(\mathbf{x}_k | \mathbf{u}_k, \mathbf{x}_{k-1})$ から

$\mathbf{x}_k = (x_k, y_k, \theta_k)^T$ をサンプリングするためのアルゴリズム

- 1 Algorithm sample_motion_model_odometry ($\mathbf{u}_k, \mathbf{x}_{k-1}$)
- 2 $\delta_{\text{rot1}} = \text{atan2}(\bar{y}_k - \bar{y}_{k-1}, \bar{x}_k - \bar{x}_{k-1})$
- 3 $\delta_{\text{trans}} = \sqrt{(\bar{x}_k - \bar{x}_{k-1})^2 + (\bar{y}_k - \bar{y}_{k-1})^2}$
- 4 $\delta_{\text{rot2}} = \bar{\theta}_k - \bar{\theta}_{k-1} - \delta_{\text{rot1}}$
- 5 $\hat{\delta}_{\text{rot1}} = \delta_{\text{rot1}} - \text{sample}(\alpha_1 \delta_{\text{rot1}}^2 + \alpha_2 \delta_{\text{trans}}^2)$
- 6 $\hat{\delta}_{\text{trans}} = \delta_{\text{trans}} - \text{sample}(\alpha_3 \delta_{\text{trans}}^2 + \alpha_4 \delta_{\text{rot1}}^2 + \alpha_4 \delta_{\text{rot2}}^2)$
- 7 $\hat{\delta}_{\text{rot2}} = \delta_{\text{rot2}} - \text{sample}(\alpha_1 \delta_{\text{rot2}}^2 + \alpha_2 \delta_{\text{trans}}^2)$
- 8 $x_k = x_{k-1} + \hat{\delta}_{\text{trans}} \cos(\theta_{k-1} + \hat{\delta}_{\text{rot1}})$
- 9 $y_k = y_{k-1} + \hat{\delta}_{\text{trans}} \sin(\theta_{k-1} + \hat{\delta}_{\text{rot1}})$
- 10 $\theta_k = \theta_{k-1} + \hat{\delta}_{\text{rot1}} + \hat{\delta}_{\text{rot2}}$
- 11 return $\mathbf{x}_k = (x_k, y_k, \theta_k)^T$

12.2 平面移動ロボットの確率状態方程式

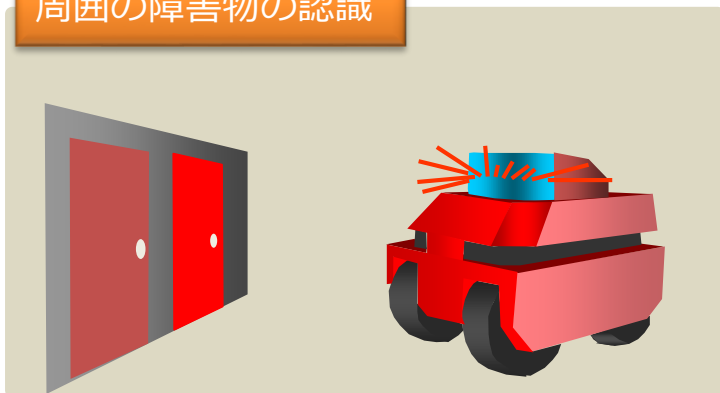
動作と地図

表4 $p(\mathbf{x}_k | \mathbf{u}_{k-1}, \mathbf{x}_{k-1}, \mathbf{m})$ を計算するアルゴリズム

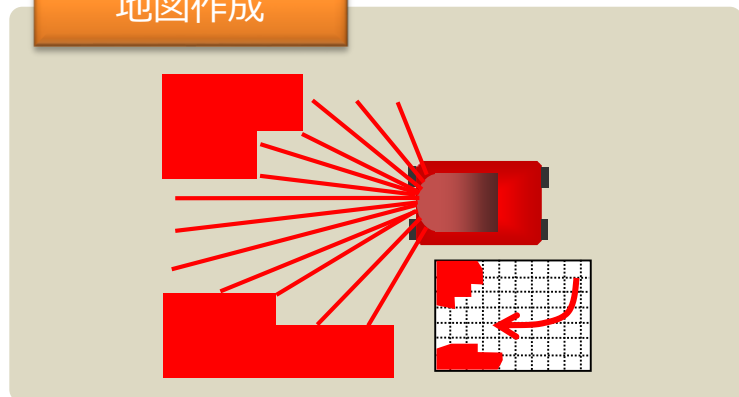
```
1  Algorithm sample_motion_model_with_map( $\mathbf{u}_{k-1}, \mathbf{x}_{k-1}, \mathbf{m}$ )  
2  do  
3     $\mathbf{x}_k = \text{sample\_motion\_model}(\mathbf{u}_{k-1}, \mathbf{x}_{k-1})$   
4     $\pi = p(\mathbf{x}_{k-1} | \mathbf{m})$   
5  until  $\pi > 0$   
6  return  $\langle \mathbf{x}_k, \pi \rangle$ 
```

この動作モデルでは、環境の地図 \mathbf{m} が利用される。このアルゴリズムは、障害物に占有された地図 \mathbf{m} 中の空間にはロボットが存在できないことを反映するために、前述の動作モデルに付け加えられる。

周囲の障害物の認識



地図作成



SICK LMS



北陽電機 URG

ロボットの知覚

環境計測モデルは次の条件付き確率分布で定義される。

$$h(y_k | \hat{x}_k^{(l)}, m)$$

時刻 k における計測

時刻 k における姿勢

地図情報

問い

得られた計測データは、与えられた地図中のどの場所・姿勢で得られたのだろうか？

このモデルは、カメラ、バーコードソナー、レーザレンジファインダなど、任意のセンサに適用できる。

12.3 ロボットの観測モデル（レーザレンジファインダの場合）

センサ信号

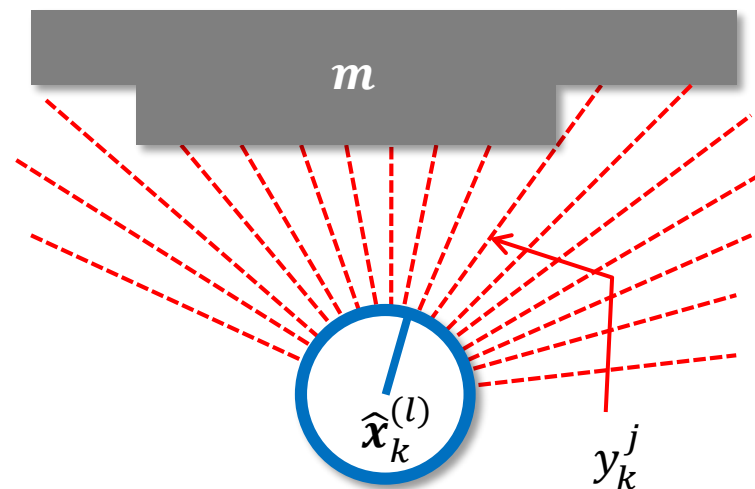
$$\mathbf{y}_k = \{y_k^1, y_k^2, \dots, y_k^j, \dots, y_k^J\}$$

センサの多くは、問い合わせがあると複数の数値 y_k^j からなる計測結果 \mathbf{y}_k を生成して返す。

知っていること：センサモデル

$$h(\mathbf{y}_k | \hat{\mathbf{x}}_k^{(l)}, \mathbf{m})$$

環境中でロボットが姿勢 $\hat{\mathbf{x}}_k^{(l)}$ と推定され、環境地図の情報 \mathbf{m} が与えられている状況でセンサの計測値 \mathbf{y}_k の尤もらしさを評価することを考える。

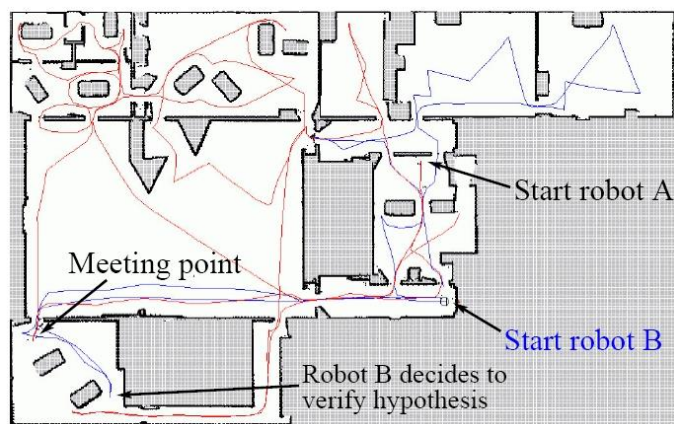


環境地図

$$\mathbf{m} = \{m_1, m_2, \dots, m_N\}$$

環境の地図は、環境中に存在する物体とそれらの置かれ方のリスト

位置ベース



n はある特徴の位置と対応付けるインデクス。
例えば世界座標系 (x, y) を直接表現するために
 m_n を $m_{x,y}$ と表現する。

ボリュームメトリック（容積測量的）な地図
であり、環境中の物体に関する情報だけでなく
物体のない場所に関する情報も記述される。

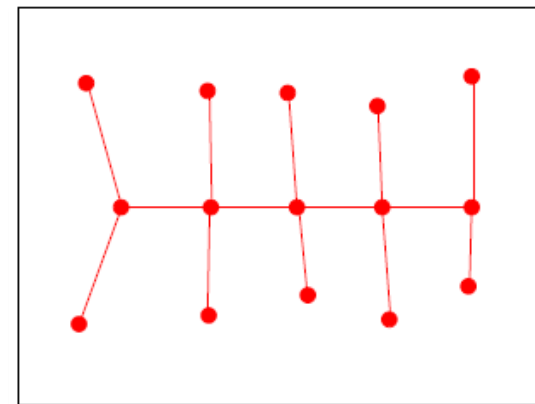
占有格子地図はこれであり、ロボットのナビゲーションなどに向く地図の形式。

特徴ベース

n は特徴に対するインデクスを表す。

m_n はデカルト座標系における特徴の位置を表す。

地図に記述されている物体の場所における環境
の形状を記録したもの。センシングによって
情報が追加された時に地図を書き換えるので、
ロボットによる地図生成によく用いられる。



例：レーザスキャナの観測モデル $h\left(\mathbf{y}_k \mid \hat{\mathbf{x}}_k^{(l)}, \mathbf{m}\right)$

求めたい観測モデルは4種類の誤差を考慮した確率分布の混合分布で表される。

- 1 正しい計測時の局所的な計測雑音
- 2 想定外の物体による計測誤差
- 3 計測失敗による計測誤差
- 4 原因不明のランダムな雑音

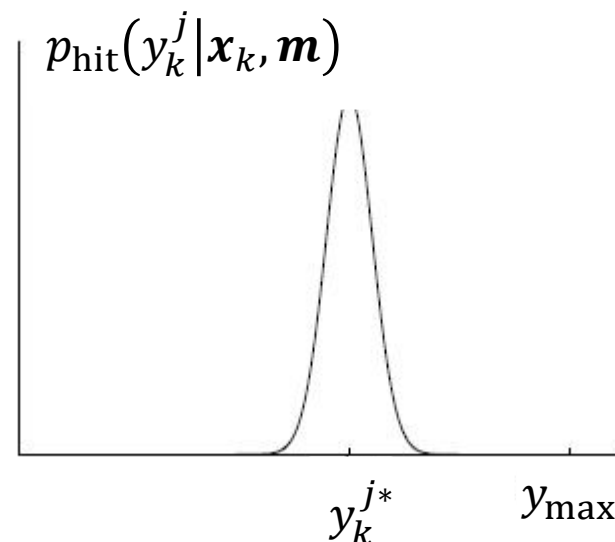
1. 正しい計測時の局所的な計測雑音

距離センサの解像度が有限であることや、計測信号に対する大気の影響などが原因で発生する測定雑音は、たいてい平均が y_k^{j*} 、標準偏差 σ_{hit} の狭いガウス分布でモデル化できる。

$$p_{\text{hit}}(y_k^j | \mathbf{x}_k, \mathbf{m}) = \begin{cases} \eta \mathcal{N}(y_k^j; y_k^{j*}, \sigma_{\text{hit}}^2) & \text{if } 0 \leq y_k^j \leq y_{\text{max}} \\ 0 & \text{otherwise} \end{cases}$$

$$\mathcal{N}(y_k^j; y_k^{j*}, \sigma_{\text{hit}}^2) = \frac{1}{\sqrt{2\pi\sigma_{\text{hit}}^2}} \exp\left\{-\frac{1}{2} \frac{(y_k^j - y_k^{j*})^2}{\sigma_{\text{hit}}^2}\right\}$$

$$\eta = \left(\int_0^{y_{\text{max}}} \mathcal{N}(y_k^j; y_k^{j*}, \sigma_{\text{hit}}^2) dy_k^j \right)^{-1}$$



σ_{hit} はセンサ固有の誤差パラメータ

2. 想定外の物体による計測誤差

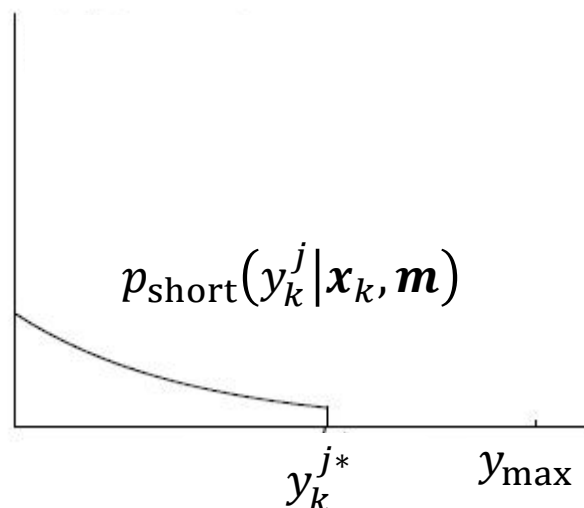
地図に含むべきではない物体をセンサが検知して極端に短い距離を出力する場合がある。そのような典型例として、ロボット周囲の人間が挙げられる。

対処法の一つとして、それらをセンサの誤差要因として扱う方法がある。その場合、モデル化されていない物体は、距離を y_k^{j*} よりも短くするという性質を持つ。

$$p_{\text{hit}}(y_k^j | \mathbf{x}_k, \mathbf{m})$$

$$= \begin{cases} \eta \lambda_{\text{short}} \exp(-\lambda_{\text{short}} y_k^j) & \text{if } 0 \leq y_k^j \leq y_k^{j*} \\ 0 & \text{otherwise} \end{cases}$$

$$\eta = \frac{1}{1 - e^{-\lambda_{\text{short}} y_k^{j*}}}$$



λ_{short} はセンサ固有の誤差パラメータ

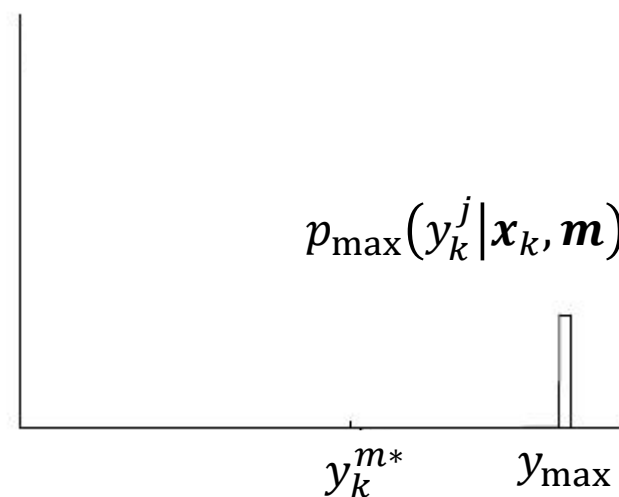
3. 計測失敗による計測誤差

時々、障害物が全く検知されないことがある。レーザレンジファインダでは、黒くて赤外線を吸収する物体の計測中や、計測対象が日光の中にあるような場合に計測失敗が生ずる。このような失敗は高頻度で起こる。

計測に失敗した場合に、典型的なセンサは最大距離 y_{\max} を出力する。

$$p_{\max}(y_k^j | \mathbf{x}_k, \mathbf{m}) = \begin{cases} 1 & \text{if } y_k^j \geq y_{\max} - \varepsilon \\ 0 & \text{otherwise} \end{cases}$$

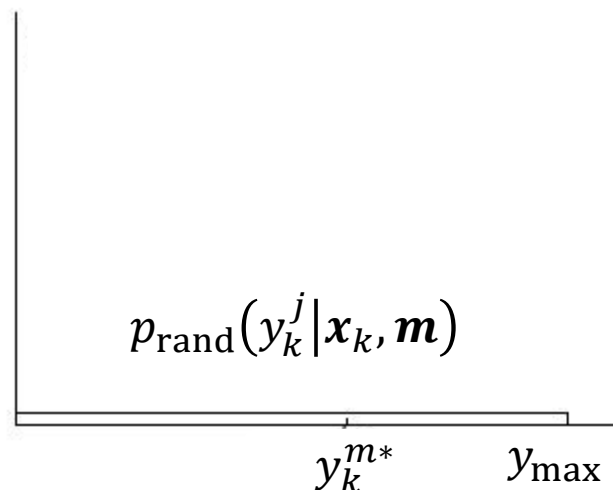
p_{\max} が確率密度関数となるように、 y_k^j が y_{\max} よりも小さなとても小さな範囲で一様分布となるようにモデル化する。



4. 原因不明のランダムな雑音

距離センサは時折、完全に説明のしようのない計測値を出力することがある（ビームの乱反射や混線など）。このような計測値は、センサ計測値の区間全域にわたる一様分布を使ってモデル化できる。

$$p_{\text{rand}}(y_k^j | \mathbf{x}_k, \mathbf{m}) = \begin{cases} \frac{1}{y_{\text{max}}} & \text{if } 0 \leq y_k^j \leq y_{\text{max}} \\ 0 & \text{otherwise} \end{cases}$$



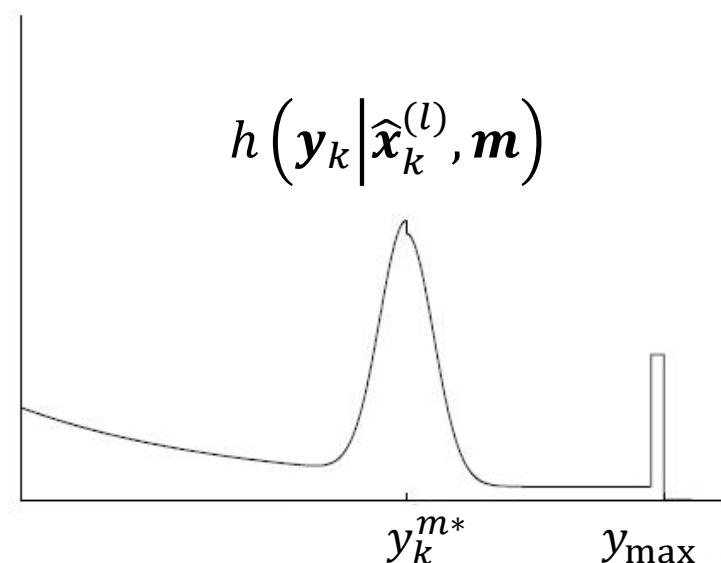
5. 4種類の計測誤差モデルの混合分布

計測誤差は4種類の分布を重みづけして混合することで得られる。

表5 レーザレンジファインダの出力 y_k の尤度を表す計測誤差確率密度関数

$$p(y_k^j | \mathbf{x}_k, \mathbf{m}) = \begin{bmatrix} y_{\text{hit}} \\ y_{\text{short}} \\ y_{\text{max}} \\ y_{\text{rand}} \end{bmatrix}^T \begin{bmatrix} p_{\text{hit}}(y_k^j | \mathbf{x}_k, \mathbf{m}) \\ p_{\text{short}}(y_k^j | \mathbf{x}_k, \mathbf{m}) \\ p_{\text{max}}(y_k^j | \mathbf{x}_k, \mathbf{m}) \\ p_{\text{rand}}(y_k^j | \mathbf{x}_k, \mathbf{m}) \end{bmatrix}$$

$$y_{\text{hit}} + y_{\text{short}} + y_{\text{max}} + y_{\text{rand}} = 1$$



センサ固有のパラメータ $\Theta = [\sigma_{\text{hit}}, \lambda_{\text{short}}, y_{\text{hit}}, y_{\text{short}}, y_{\text{max}}, y_{\text{rand}}]^T$ は、実際の計測データを利用した最尤推定によって求めることができる。

12.3 ロボットの観測モデル（レーザレンジファインダの場合）

表 6 レーザレンジファインダの出力 y_k の尤度を計算するアルゴリズム

```

1 Algorithm beam_range_finder_model ( $y_k, x_k, m$ )
2    $q = 1$ 
3   For  $j = 1$  to  $J$  do
4     compute  $y_k^{j*}$  for the measurement  $y_k^j$  using ray casting
5      $p = y_{\text{hit}} \cdot p_{\text{hit}}(y_k^j | x_k, m) + y_{\text{short}} \cdot p_{\text{short}}(y_k^j | x_k, m)$ 
        $+ y_{\text{max}} \cdot p_{\text{max}}(y_k^j | x_k, m) + y_{\text{rand}} \cdot p_{\text{rand}}(y_k^j | x_k, m)$ 
6      $q = q \cdot p$ 
7   Return  $q$ 

```

注 1

スキャン中の個別の計測値は互いに独立であると仮定する。

注 1

注 2

雑音がない場合に得られるはずの距離を計算するために、マップ情報を利用したレイキャスティングを適用。

注 2

注 3

注 3

表 5 の混合分布

表7 パーティクルフィルタに基づく位置推定アルゴリズム、モンテカルロ位置推定MCL

```
1 Algorithm MCL ( $\mathcal{X}_{k-1}, \mathbf{u}_{k-1}, \mathbf{y}_k, \mathbf{m}$ ) 注1 注2
2  $\bar{\mathcal{X}}_{k-1} = \mathcal{X}_{k-1} = \emptyset$ 
3 for  $m = 1$  to  $M$  do
4  $\mathbf{x}_k^{(m)} = \text{sample\_motion\_model}(\mathbf{u}_{k-1}, \mathbf{x}_{k-1}^{(m)})$  注3
5  $w_k^{(m)} = \text{beam\_range\_finder\_model}(\mathbf{y}_k, \mathbf{x}_k^{(m)}, \mathbf{m})$ 
6  $\bar{\mathcal{X}}_k = \bar{\mathcal{X}}_k + \langle \mathbf{x}_k^{(m)}, w_k^{(m)} \rangle$  注4
7 end for
8 for  $m = 1$  to  $M$  do
9 draw  $i$  with probability  $\propto w_k^{(i)}$  注5
10 add  $\mathbf{x}_k^{(i)}$  to  $\mathcal{X}_k$ 
11 end for
12 return  $\mathcal{X}_k$ 
```

注1

M 個のパーティクルのセット
 $\mathcal{X}_k = \{\mathbf{x}_k^{(1)}, \mathbf{x}_k^{(2)}, \dots, \mathbf{x}_k^{(M)}\}$

注2

初期のパーティクルのセット \mathcal{X}_0
は事前分布 $p(\mathbf{x}_0)$ に従って生成し
各重みを M^{-1} に設定することで
得る。

注3

例えば 表1 を利用する。

注4

例えば 表6 を利用する。

注5

表8

を利用する。

表 8

パーティクルフィルタのための等間隔リサンプリングアルゴリズム

```
1  Algorithm Low_variance_sampler ( $\mathcal{X}_k, \mathcal{W}_k$ )
2     $\bar{\mathcal{X}}_k = \emptyset$ 
3     $r = \text{rand}(0; M^{-1})$ 
4     $c = w_k^{(1)}$ 
5     $i = 1$ 
6    for  $m = 1$  to  $M$  do
7       $U = r + (m - 1) \cdot M^{-1}$ 
8      while
9         $i = i + 1$ 
10        $c = c + w_k^{(i)}$ 
11     end while
12     add  $x_k^{(i)}$  to  $\bar{\mathcal{X}}_k$ 
13   end for
14   return  $\bar{\mathcal{X}}_k$ 
```

注

このアルゴリズムでは、パーティクルの集合 \mathcal{X}_k から重み \mathcal{W}_k に基づいてサンプリングを行うために、たった一つの乱数しか用いない。

それにもかかわらず、パーティクルがリサンプリングされる確率を、その重みに比例させることができる。さらに、このサンプラは計算効率が良く、 M 個のパーティクルをサンプリングするために必要な計算量は $O(M)$ である。

表9 ランダムパーティクルを加える適応的なMCL

```

1  Algorithm Augmented_MCL ( $\mathcal{X}_{k-1}, \mathbf{u}_{k-1}, \mathbf{y}_k, \mathbf{m}$ )
2  static  $w_{\text{slow}}, w_{\text{fast}}$ 
3   $\bar{\mathcal{X}}_k = \mathcal{X}_k = \emptyset$ 
4  for  $m = 1$  to  $M$  do
5       $\mathbf{x}_k^{(m)} = \text{sample\_motion\_model}(\mathbf{u}_{k-1}, \mathbf{x}_{k-1}^{(m)})$ 
6       $w_k^{(m)} = \text{beam\_range\_finder\_model}(\mathbf{y}_k, \mathbf{x}_k^{(m)}, \mathbf{m})$ 
7       $\bar{\mathcal{X}}_k = \bar{\mathcal{X}}_k + \langle \mathbf{x}_k^{(m)}, w_k^{(m)} \rangle$ 
8       $w_{\text{ave}} = w_{\text{ave}} + 1/M \cdot w_k^{(m)}$ 
9  end for
10  $w_{\text{slow}} = w_{\text{slow}} + \alpha_{\text{slow}}(w_{\text{avg}} + w_{\text{slow}})$ 
11  $w_{\text{fast}} = w_{\text{fast}} + \alpha_{\text{fast}}(w_{\text{avg}} - w_{\text{fast}})$ 
12 for  $m = 1$  to  $M$  do
13     with probability  $\max\{0.0, 0.1 - w_{\text{fast}}/w_{\text{slow}}\}$  do
14         add random pose to  $\mathcal{X}_k$ 
15     else
16         draw  $i$  with probability  $\propto w_k^{(i)}$ 
17         add  $\mathbf{x}_k^{(i)}$  to  $\mathcal{X}_k$ 
18     end with
19 end for
20 return  $\mathcal{X}_k$ 

```

注1

注

ランダムパーティクルの数が短時間と長時間のセンサ計測値の尤度を比較することで決定されている。

もし短期の尤度が長期の尤度以上であればランダムパーティクルは加えられない。逆に短期の尤度の方が低い場合には、その値に比例した数のランダムパーティクルが加えられる。

増えすぎたパーティクルの削減方法は別に考える必要がある。

注1

$\alpha_{\text{slow}}, \alpha_{\text{long}}$ は長期、短期の平均を推定するための減衰率。

$$0 < \alpha_{\text{slow}}, \alpha_{\text{long}}$$

表 *

速度情報に基づいて $p(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_{k-1})$ を計算するアルゴリズム1 Algorithm motion_model_velocity ($\mathbf{x}_k, \mathbf{u}_{k-1}, \mathbf{x}_{k-1}$)

$$2 \quad \mu = \frac{1}{2} \frac{(x_{k-1} - x_k) \cos \theta - (y_{k-1} - y_k) \sin \theta}{(y_{k-1} - y_k) \cos \theta - (x_{k-1} - x_k) \sin \theta}$$

$$3 \quad x^* = \frac{x_{k-1} + x_k}{2} + \mu(y_{k-1} - y_k)$$

$$4 \quad y^* = \frac{y_{k-1} + y_k}{2} + \mu(x_{k-1} - x_k)$$

$$5 \quad r^* = \sqrt{(x_k - x^*)^2 + (y_k - y^*)^2}$$

$$6 \quad \Delta\theta = \text{atan2}(y_k - y^*, x_k - x^*) - \text{atan2}(y_{k-1} - y^*, x_{k-1} - x^*)$$

$$7 \quad \hat{v} = \frac{\Delta\theta}{\Delta t} r^*$$

$$8 \quad \hat{\omega} = \frac{\Delta\theta}{\Delta t}$$

$$9 \quad \hat{\gamma} = \frac{\theta_k - \theta_{k-1}}{\Delta t} - \hat{\omega}$$

$$10 \quad \text{return prob}(v_k - \hat{v}, \alpha_1 v^2 + \alpha_2 \omega^2) \cdot \text{prob}(\omega_k - \hat{\omega}, \alpha_3 v^2 + \alpha_4 \omega^2) \\ \cdot \text{prob}(\hat{\gamma}, \alpha_5 v^2 + \alpha_6 \omega^2)$$

注 1

$\mathbf{x}_{t-1} = (x \ y \ \theta)^T$, $\mathbf{u}_t = (v \ \omega)^T$ から
 $\mathbf{x}_t = (x' \ y' \ \theta')^T$ をサンプリング
 するには 表 2 を利用する。

注 1

注 2

atan2(y, x)

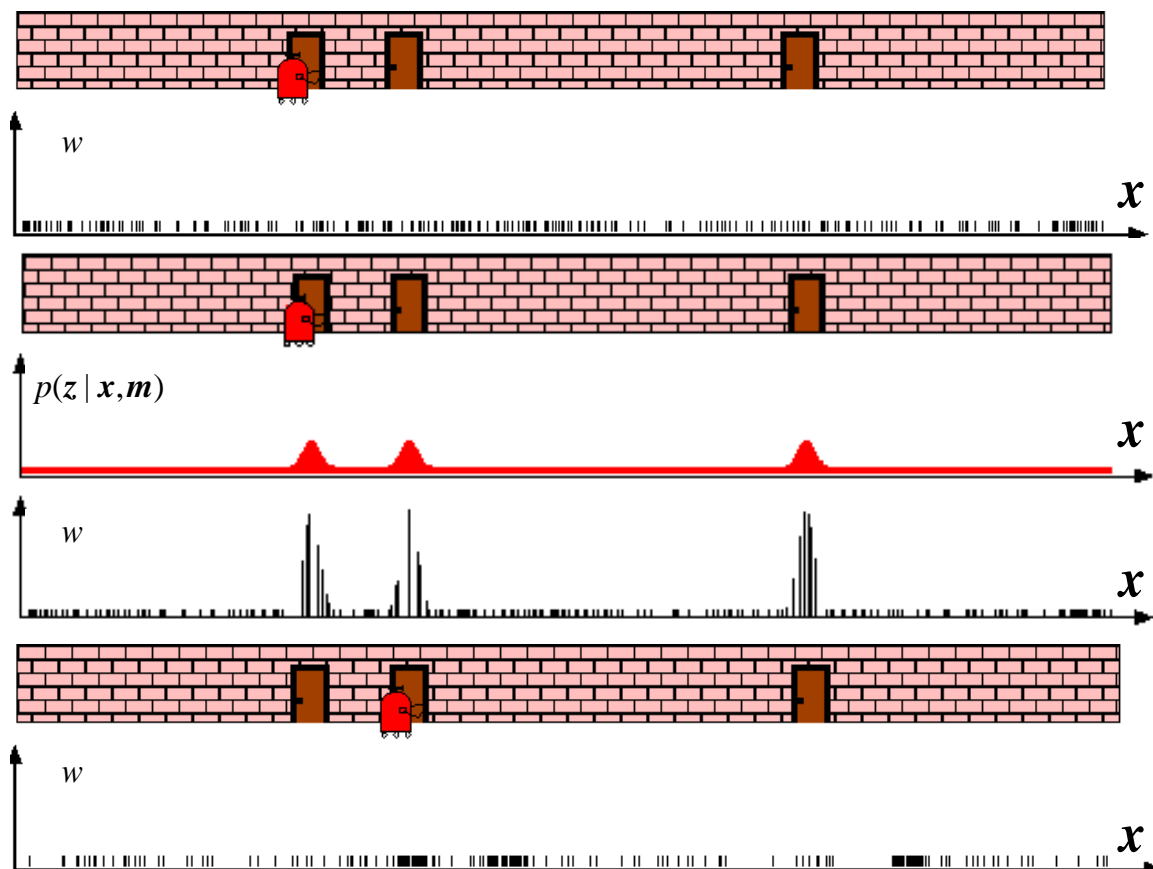
$$= \begin{cases} \text{atan}(y/x) & \text{if } x > 0 \\ \text{sign}(y)(\pi - \text{atan}(|y/x|)) & \text{if } x < 0 \\ 0 & \text{if } x = y = 0 \\ \text{sign}(y)\pi/2 & \text{if } x = 0, y \neq 0 \end{cases}$$

注 3

prob(a, b²) は、原点を中心
 とする標準偏差 b の分布に
 おいて、引数 a の確率密度
 関数を計算する関数。 表 3

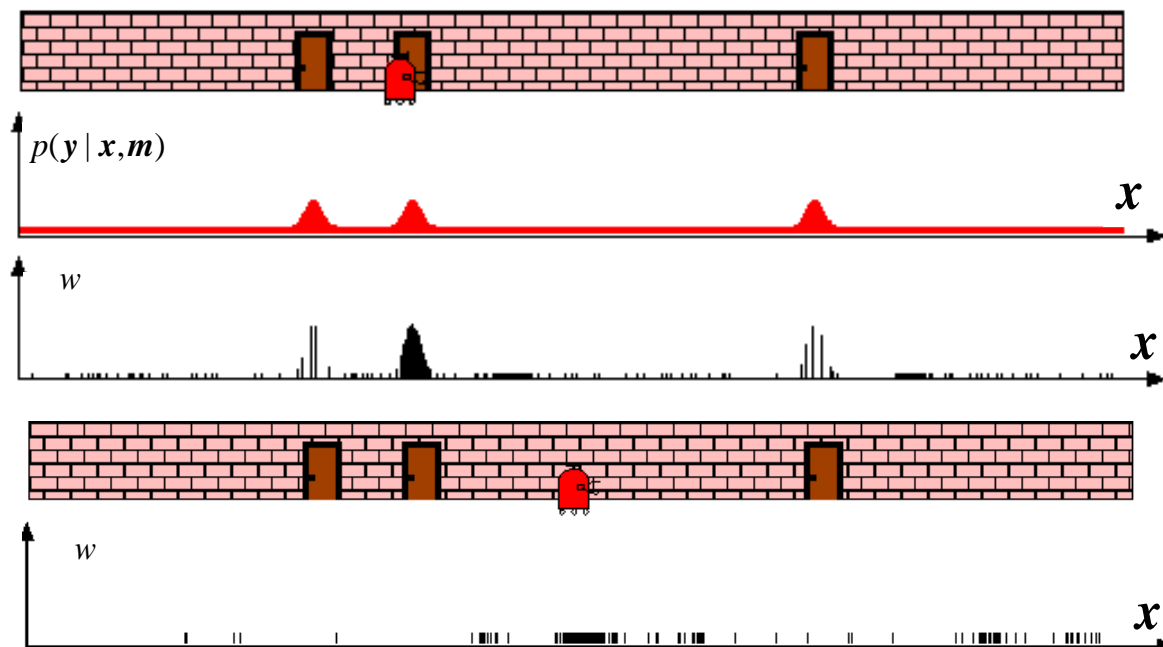
注 3

12.4 ロボットの状態推定



- 1 初期の不確かさは姿勢空間全体で一様にランダムに選ばれたパーティクルで表現される。
- 2 ロボットがドアを検知したとき、各パーティクルの重みを変更する。ここではパーティクルの位置は変化していないことに注意。
- 3 次の時刻の様子。ロボットが移動している。リサンプリング後のパーティクルは、ロボットがいそうな三か所で増加している。

12.4 ロボットの状態推定



4 新たな計測値がロボットに与えられると、各パーティクルの重みが不均一になる。この時点で、ほとんどの確率が二番目のドアに集中していることが分かる。この場所が、ロボットが最もいそうな場所である。

5 さらにロボットが移動すると、もう一度リサンプリングが行われ、新たなパーティクルのセットが事後信念を近似する。

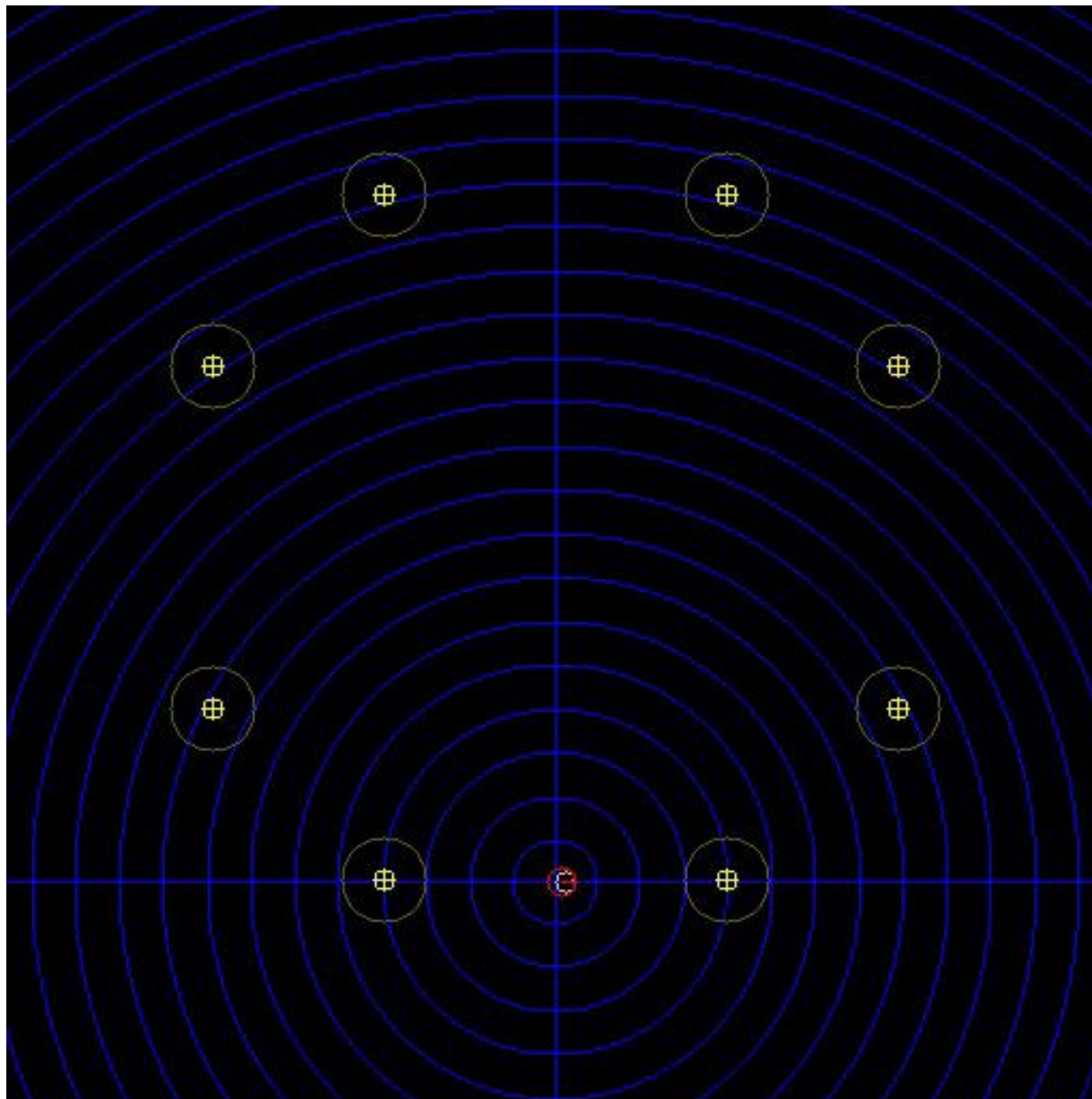
12.4 ロボットの状態推定

材料

- 状態 \mathbf{x}_k → 位置, 角度...
- 制御入力 \mathbf{u}_k → 並進速度, 回転角速度...
- 地図情報 \mathbf{m} → トポロジカルマップ, グリッドマップ...
- 観測値 \mathbf{y}_k → センサ信号...
- 初期分布 $p(\mathbf{x}_0)$ → ガウス分布, 一様分布...
- 運動モデル $f(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \boldsymbol{\xi}_{k-1})$ → ロボットの動作から解析的に導出
- システム雑音 $\boldsymbol{\xi}_{k-1}$ → ガウス分布, コーシー分布, t分布...
- 観測モデル $h(\mathbf{y}_k | \mathbf{x}_k^{(l)})$ → センサの特性に応じて実験的に決定

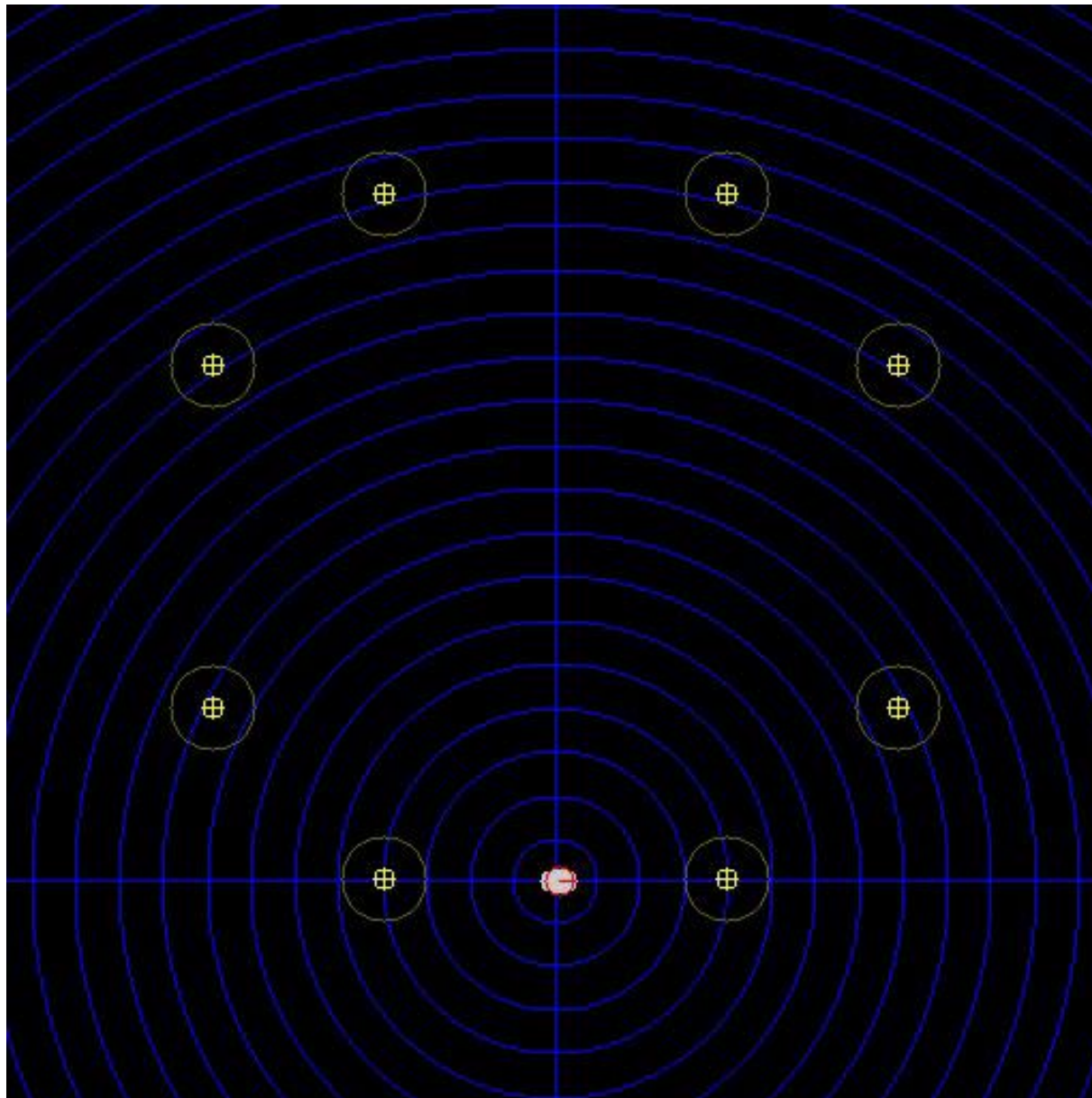
ロボットの動作モデルの例

1

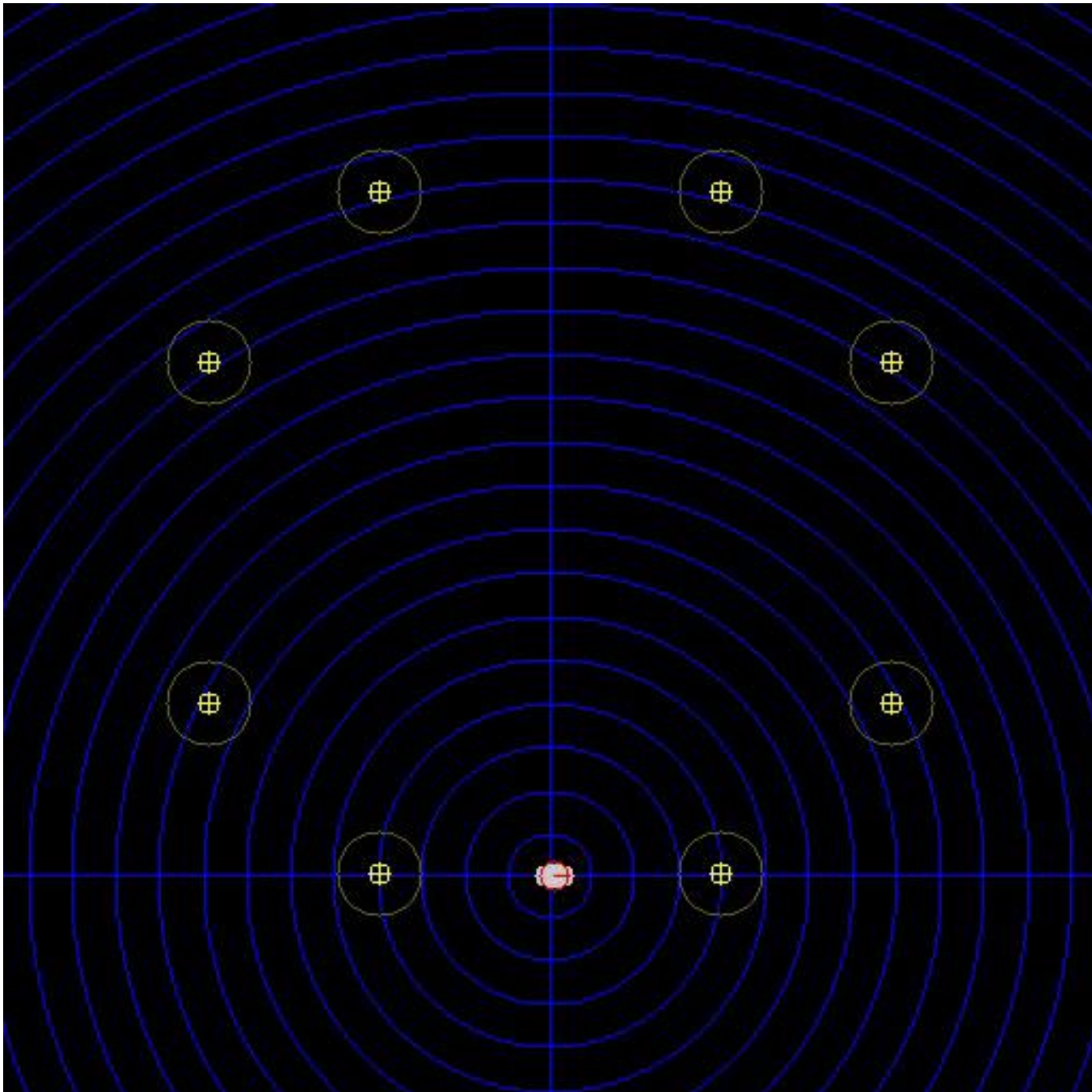


ロボットの動作モデルの例

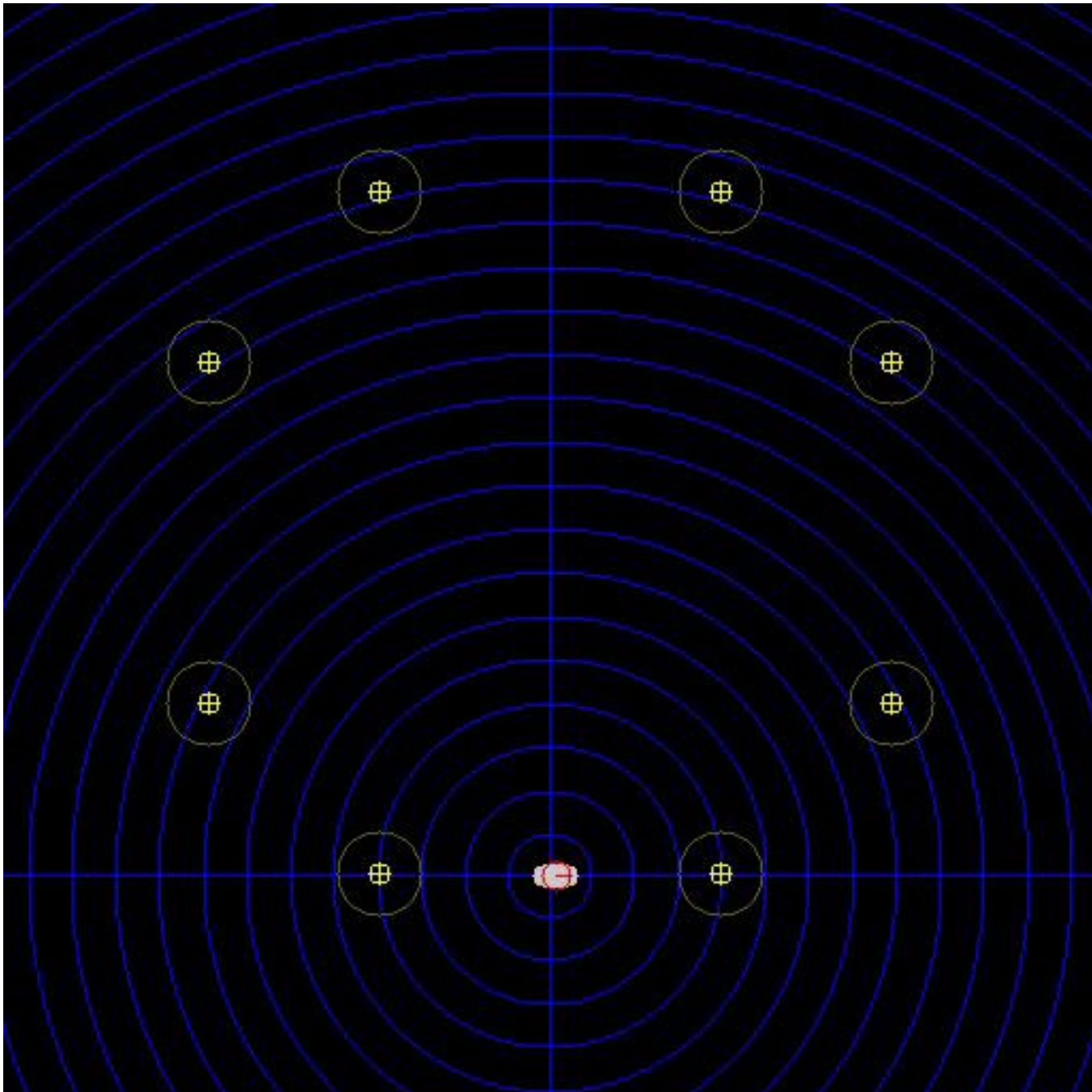
2



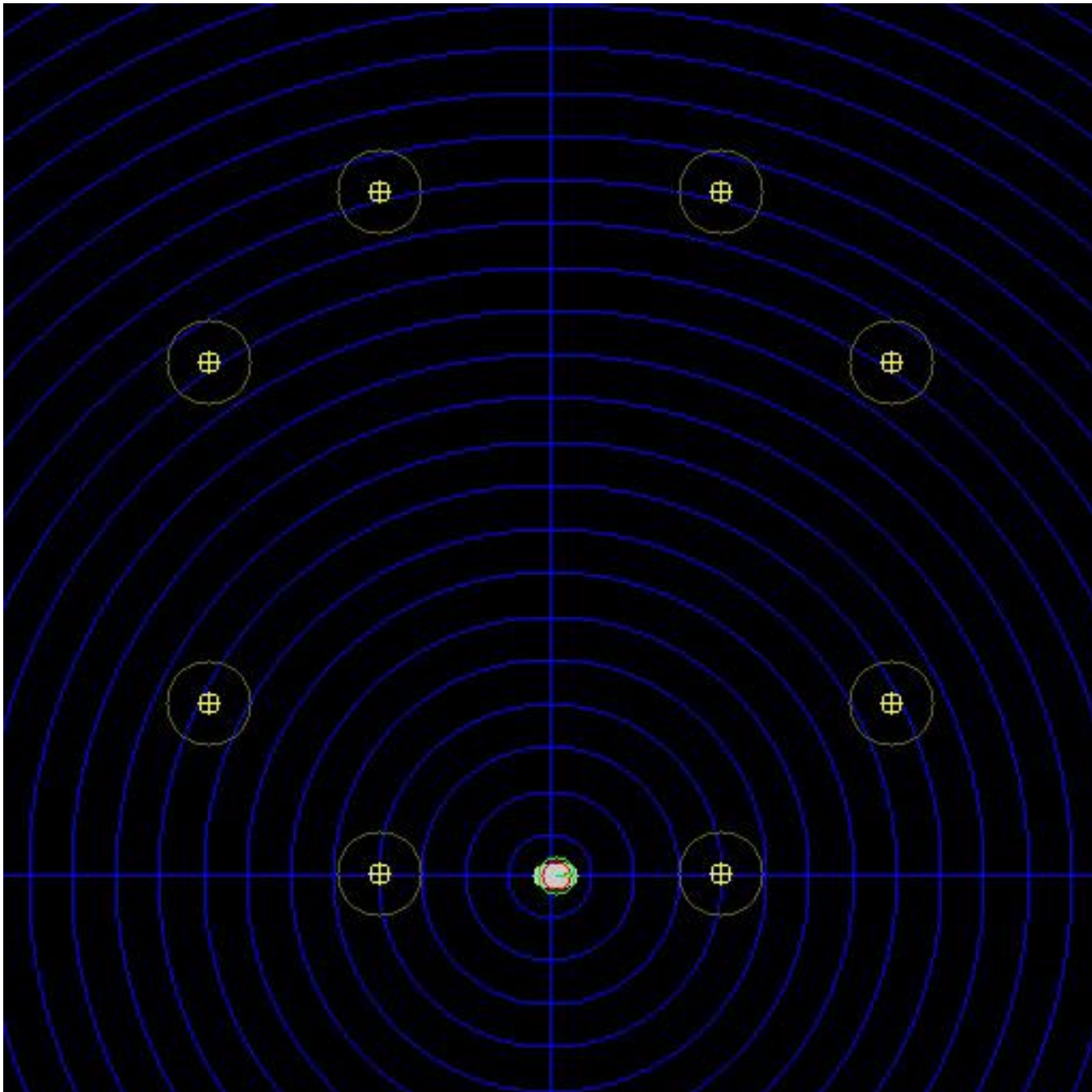
3



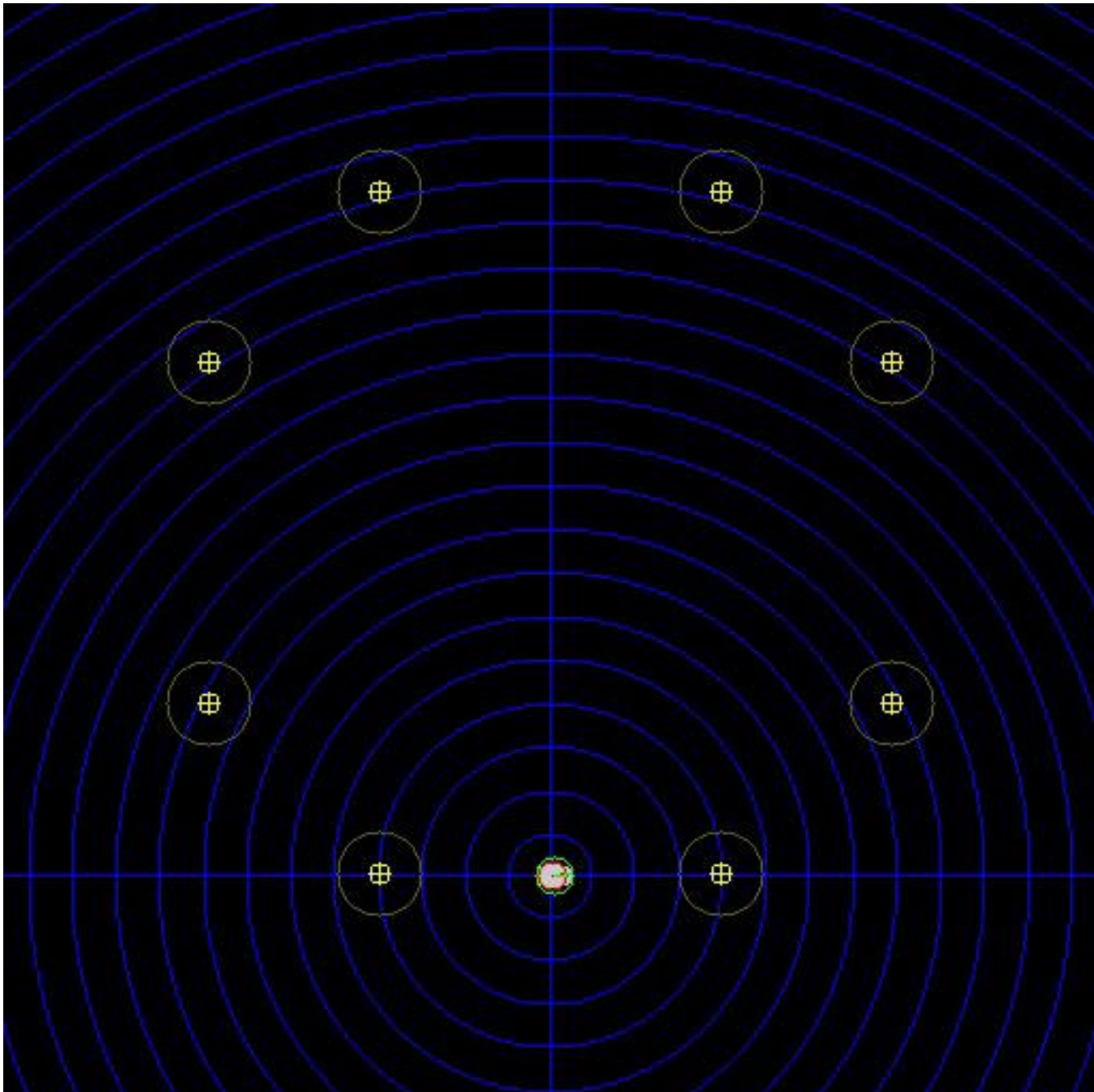
4



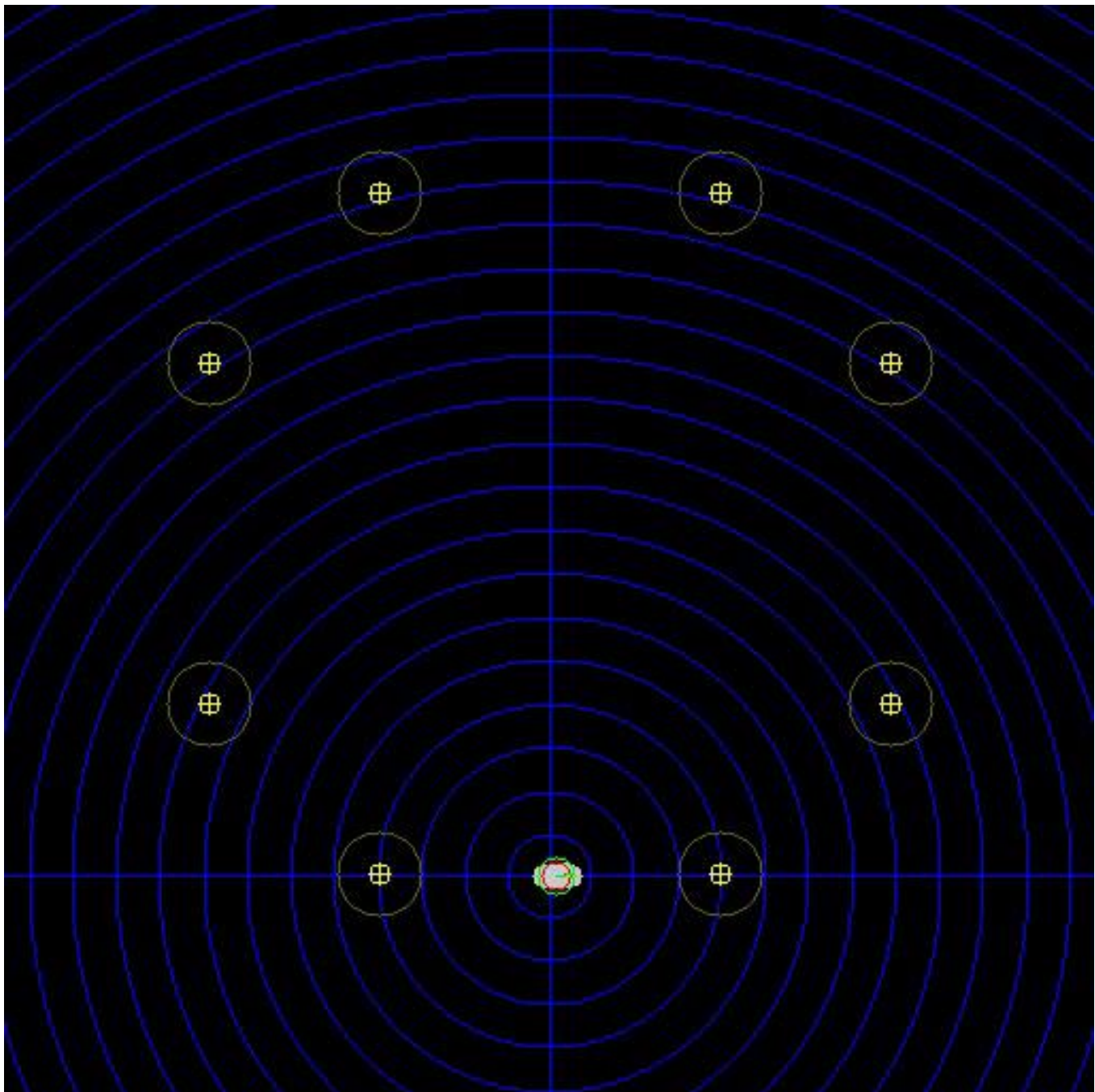
5



6



7-1



7-2

