
第 7 章 非线性 Kalman 滤波器

第 2 章讨论的线性系统滤波方法，过程方程（2-1）和测量方程（2-2）都是线性方程。在实际中，真正严格的线性系统几乎不存在，几乎所有的系统都是非线性的。现在认为的很多线性系统，也只是在一定的条件下才成立，如果超出这个范围，就会变为非线性系统，例如所熟悉的欧姆定律关系也只是在一定的电压或电流范围内才满足线性关系。

另外，还有些系统本身就具有非线性关系，比如平时见到的一些测量系统，像雷达测量系统、射频标签系统（Radio Frequency Identification，缩写为 RFID），都不是线性系统。针对这些非线性系统，第 2 章所讲的 Kalman 滤波器就不再适用了，本章将介绍针对非线性系统的滤波器。本章内容是在第 2 章 Kalman 滤波器基础上的衍生，以期能够处理非线性系统。

本章介绍两种非线性 Kalman 滤波器，一种是扩展 Kalman 滤波器（Extended Kalman Filter，简称 EKF），其基本思想是利用泰勒展开，将非线性方程直接线性化。线性化后的系统模型和系统实际的非线性模型会有差别，非线性越强，差别就会越大。不过，EKF 的优势也不容忽视，由于没有附加计算，EKF 的计算量和基本的 Kalman 滤波器差不多。另一种非线性 Kalman 滤波器在原理上和 EKF 完全不同，称之为不敏 Kalman 滤波器（Unscented Kalman Filter，简称 UKF），和 EKF 相比，该滤波器由于使用了所谓的不敏变换，线性化带来的状态均值和协方差的改变要小些。本章将详细讲解这两种非线性估计方法。

7.1 扩展 Kalman 滤波器（EKF）

设系统模型具有非线性关系：

$$x(k) = f(x(k-1), w(k-1)) \quad (7-1)$$

$$z(k) = h(x(k), v(k)) \quad (7-2)$$

其中 $x(k)$ 为系统待估计状态， $w(k-1)$ 为系统过程噪声， $z(k)$ 为系统测量， $v(k)$ 为测量噪声， $f(\cdot)$ 和 $h(\cdot)$ 是非线性的过程方程与测量方程。

(2-1)、(2-2) 式中的系统矩阵 $A(k)$ 、 $C(k)$ 在 (7-1)、(7-2) 中没有出现，(7-1)、(7-2) 中的系统关系是非线性函数 $f(\cdot)$ 及 $h(\cdot)$ ，回忆 2.4 节的离散 Kalman 滤波器 (2-24) — (2-28)，发现系统矩阵 $A(k)$ 、 $C(k)$ 还是很重要的，没有了这两个矩阵，这 5 个公式就

没有了意义。

但仔细观察，新息 $Z(k) - C(k)\hat{x}(k|k-1)$ 其实也可以使用 (7-2) 中的非线性关系实现， $C(k)\hat{x}(k|k-1)$ 表示的是向前一步预测状态 $\hat{x}(k|k-1)$ 反映的测量预测值，如果知道了向前一步预测状态 $\hat{x}(k|k-1)$ ，根据 (7-2) 可以得到相应的测量预测值 $h(\hat{x}(k|k-1), 0)$ ，则新息也就变为 $z(k) - h(\hat{x}(k|k-1), 0)$ ，即 (224) 可修改为：

$$\hat{x}(k|k) = \hat{x}(k|k-1) + K(k)(z(k) - h(\hat{x}(k|k-1), 0)) \quad (7-3)$$

同样 (2-25) 说明的是如何从估计 $\hat{x}(k-1|k-1)$ 得到向前一步预测状态 $\hat{x}(k|k-1)$ ，式 (7-1) 反映了状态的变化，也可以从 (7-1) 得到具有非线性关系的 $\hat{x}(k|k-1)$ ，来代替 (2-25)：

$$\hat{x}(k|k-1) = f(\hat{x}(k-1|k-1), 0) \quad (7-4)$$

但是滤波增益 (2-26) 及估计方差 (2-27)、(2-28) 就有点麻烦， $A(k)$ 及 $C(k)$ 的地位就无法替代了。其实 $A(k)$ 与 $f(\cdot)$ ， $C(k)$ 与 $h(\cdot)$ 还是有关系的， $f(\cdot)$ 是 $x(k-1)$ 与 $x(k)$ 的非线性关系，而 $A(k)$ 是线性关系，同样 $h(\cdot)$ 是 $x(k)$ 与 $z(k)$ 的线性关系，而 $C(k)$ 是线性关系。

而目前的情况是系统实际上是非线性的，即 $f(\cdot)$ 与 $h(\cdot)$ 是非线性函数。因此无法使用 2.4 节的 Kalman 滤波器直接估计状态，也就是说，无法直接使用线性 Kalman 滤波器获得系统的状态估计。

为了能估计非线性系统的状态，人们首先想到的是近似估计结果，这是设计算法的一个基本思想，如果不能得到最优解，但也不想交“白卷”，给出一个差不多的结果也是个答案。但必须要说明这个“差不多”的结果到底“差”多少，这是研究人员特有的严谨。一般情况下，要定量地说明到底“差多少”有两种方法：从理论上给出证明，或者设计大量合理的仿真实验来说明。

接下来说明该如何估计非线性系统 (7-1)、(7-2) 的状态。将非线性函数 $f(x(k-1), w(k-1))$ 在 $x(k-1) = \hat{x}(k-1|k-1)$ 和 $w(k-1) = 0$ 处进行一阶泰勒级数展开：

$$\begin{aligned} x(k) &= f(\hat{x}(k-1|k-1), 0) + \frac{\partial f}{\partial x} \Big|_{\hat{x}(k-1|k-1)} (x(k-1) - \hat{x}(k-1|k-1)) \\ &\quad + \frac{\partial f}{\partial w} \Big|_{\hat{x}(k-1|k-1)} \cdot w(k-1) \end{aligned}$$

$$\begin{aligned}
&= \frac{\partial f}{\partial x} \Big|_{\hat{x}(k-1|k-1)} x(k-1) + [f(\hat{x}(k-1|k-1), 0) - \frac{\partial f}{\partial x} \Big|_{\hat{x}(k-1|k-1)} \cdot \hat{x}(k-1|k-1)] \\
&\quad + \frac{\partial f}{\partial w} \Big|_{\hat{x}(k-1|k-1)} \cdot w(k-1)
\end{aligned}$$

设

$$\begin{aligned}
\frac{\partial f}{\partial x} \Big|_{\hat{x}(k-1|k-1)} &= F(k) \\
\frac{\partial f}{\partial w} \Big|_{\hat{x}(k-1|k-1)} &= L(k)
\end{aligned} \tag{7-5}$$

则上式可变为：

$$x(k) = F(k)x(k-1) + [f(\hat{x}(k-1|k-1), 0) - F(x)\hat{x}(k-1|k-1)] + L(k)w(k-1) \tag{7-6}$$

现在来看看 (7-1) 与 (7-6) 的关系，很明显，(7-1) 是与实际非线性状态转移关系一致的描述，而 (7-6) 则是 (7-1) 一阶泰勒展开的近似，两者是不相同的。从数学知识可知，泰勒级数展开的阶次越多，两者就越接近，展开为无穷阶，两者才会相同。但在一般情况下，可以取合适的阶次，并承认两者之间存在差距。不过 (7-6) 给出了一个状态转移的线性关系 $F(x)$ ，可以根据这个关系利用 Kalman 滤波器实现状态估计。

需要考虑的是过程噪声项也附加了系数，所以经过泰勒级数展开后，过程噪声的方差也有了变化，不再是原来的 $Q(k)$ ，而是 $L(k)Q(k)L^T(k)$ 。同理，也可以利用泰勒级数展开将测量方程线性化，将 (7-2) 在 $x(k) = \hat{x}(k|k-1)$ 及 $v(k) = 0$ 处展开：

$$\begin{aligned}
y(k) &= h(\hat{x}(k|k-1), 0) + \frac{\partial h}{\partial x} \Big|_{x=\hat{x}(k|k-1)} \cdot (x(k) - \hat{x}(k|k-1)) \\
&\quad + \frac{\partial h}{\partial v} \Big|_{x=\hat{x}(k|k-1)} \cdot v(k) \\
&= \frac{\partial h}{\partial x} \Big|_{x=\hat{x}(k|k-1)} \cdot x(k) + [h(\hat{x}(k|k-1), 0) - \frac{\partial h}{\partial x} \Big|_{x=\hat{x}(k|k-1)} \cdot \hat{x}(k|k-1)] \\
&\quad + \frac{\partial h}{\partial v} \Big|_{x=\hat{x}(k|k-1)} \cdot v(k)
\end{aligned}$$

设

$$\begin{aligned}
\frac{\partial h}{\partial x} \Big|_{x=\hat{x}(k|k-1)} &= H(k) \\
\frac{\partial h}{\partial v} \Big|_{x=\hat{x}(k|k-1)} &= M(k)
\end{aligned} \tag{7-7}$$

则

$$y(k) = H(k)x(k) + [h(\hat{x}(k|k-1), 0) - \frac{\partial h}{\partial x} \Big|_{x=\hat{x}(k|k-1)} \cdot \dot{\hat{x}}(k|k-1)] + M(k)v(k) \quad (7-8)$$

可见，测量噪声的方差变为 $M(k)R(k)M(k)^T$ 。所以，利用 (7-5)、(7-6) 式中的 $F(k)$ 、 $H(k)$ 及相应的噪声方差，改写 (2-26) - (2-28) 的滤波增益及估计方差如下：

$$K(k) = P(k|k-1)H^T(k)(H^T(k)P(k|k-1)H(k) + M(k)R(k)M^T(k))^{-1} \quad (7-9)$$

$$P(k|k-1) = F(k-1)P(k-1|k-1)F^T(k-1) + L(k-1)Q(k-1)L^T(k-1) \quad (7-10)$$

$$P(k|k) = (I - K(k)H(k))P(k|k-1) \quad (7-11)$$

如表 7.1 所示，给出了标准 Kalman 滤波器和 EKF 的差别。

表 7.1 标准 Kalman 滤波器和 EKF 的差别

		标准 Kalman 滤波器	EKF
系统模型描述		$x(k+1) = A(k)x(k) + w(k)$ $z(k) = C(k)x(k) + v(k)$ $E[w(k)w^T(j)] = Q(k)\delta(k-j)$ $E[v(k)v^T(j)] = R(k)\delta(k-j)$ $E[w(k)v^T(j)] = 0$	$x(k) = f(x(k-1), w(k-1))$ $z(k) = h(x(k), v(k))$ $E[w(k)w^T(j)] = Q(k)\delta(k-j)$ $E[v(k)v^T(j)] = R(k)\delta(k-j)$ $E[w(k)v^T(j)] = 0$
Kalman 滤波器	预测	$\hat{x}(k k-1) =$ $A(k-1)\hat{x}(k-1 k-1)$	$\hat{x}(k k-1) = f(\hat{x}(k-1 k-1), 0)$
	更新	$\hat{x}(k k) = \hat{x}(k k-1) +$ $K(k)(z(k) - C(k)\hat{x}(k k-1))$	$\hat{x}(k k) = \hat{x}(k k-1)$ $+ K(k)(z(k) - h(\hat{x}(k k-1), 0))$
	滤波器增益	$K(k) = P(k k-1)C^T(k)$ $\times (C(k)P(k k-1)C^T(k) + R(k))^{-1}$	$K(k) = P(k k-1)H^T(k)$ $\times (H(k)P(k k-1)H^T(k) + M(k)R(k)M^T(k))^{-1}$
	向一步预测方差	$P(k k-1) =$ $A(k-1)P(k-1 k-1)A^T(k-1)$ $+ Q(k-1)$	$P(k k-1) =$ $F(k-1)P(k-1 k-1)F^T(k-1)$ $+ L(k-1)Q(k-1)L^T(k-1)$
	状态估计方差	$P(k k) = (I - K(k)C(k))$ $\times P(k k-1)$	$P(k k) = (I - K(k)H(k))$ $\times P(k k-1)$

下面举例说明如何使用 EKF。

例 7.1 假设在二维空间内只能测量运动目标到原点的距离, 试利用 EKF 方法给出非线性的计算方法。

解析: 先来看一下系统方程, 设二维空间分别用横、纵坐标用 x 、 y 来表示, 假设 x 、 y 轴运动的过程噪声不相关, 则过程方程为:

$$\begin{bmatrix} x(k) \\ y(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x(k-1) \\ y(k-1) \end{bmatrix} + \begin{bmatrix} w_x(k-1) \\ w_y(k-1) \end{bmatrix} \quad (7-12)$$

过程噪声为:

$$Q(k-1) = \begin{bmatrix} E(w_x(k-1)w_x^T(k-1)) & 0 \\ 0 & E(w_y(k-1)w_y^T(k-1)) \end{bmatrix} \quad (7-13)$$

测量方程为:

$$z(k) = \sqrt{x^2(k) + y^2(k)} + v(k) \quad (7-14)$$

该题中的过程方程是线性的, 测量方程是非线性的, 但测量噪声与所测距离之间也是线性的, 这是 (7-1)、(7-2) 所描述非线性方程的特殊情况。那么, 应该如何使用 EKF (式 (7-3) – (7-5)、(7-7)、(7-9) – (7-11)) 的方法呢?

因为过程方程是线性的, 因此不需要进行线性化, 即不需要计算 (7-4) 和 (7-10) 中的线性化矩阵 $K(k-1)$ 和 $L(k-1)$, 而是使用 (2-25)、(2-27) 即可。

待估计状态是一个向量, 包括横轴位置 $x(k)$ 及纵轴位置 $y(k)$, 为了和前面的变量区别

开来, 本例题中的待估计状态使用 $\vec{x}(k)$ 表示, 即 $\vec{x}(k) = \begin{bmatrix} x(k) \\ y(k) \end{bmatrix}$, 则:

$$\begin{aligned} H(k) &= \frac{\partial h}{\partial \vec{x}} \Big|_{\vec{x}=[\hat{x}(k|k-1), \hat{y}(k|k-1)]^T} \\ &= \frac{[\hat{x}(k|k-1), \hat{y}(k|k-1)]^T}{\sqrt{\hat{x}^2(k|k-1) + \hat{y}^2(k|k-1)}} \end{aligned} \quad (7-15)$$

当前一步预测估计 $\hat{x}(k|k-1)$ 、 $\hat{y}(k|k-1)$ 已知时, $H(k)$ 是一个 1×2 矩阵, 利用 EKF 的估计方程为:

$$\hat{\vec{x}}(k|k) = \hat{\vec{x}}(k|k-1) + K(k)[z(k) - \sqrt{\hat{x}^2(k|k-1) + \hat{y}^2(k|k-1)}] \quad (7-16)$$

因为系统的过程方程是线性的, 所以向前一步预测估计和第 4 章线性 Kalman 滤波器一样,

即 $A(k) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, 所以:

$$\hat{\vec{x}}(k|k-1) = A(k)\hat{\vec{x}}(k-1|k-1) \quad (7-17)$$

其估计方差为:

$$P(k|k-1) = A(k)P(k-1|k-1)A^T(k) + Q(k-1) \quad (7-18)$$

又 (7-14) 中状态与测量噪声之间为线性加和关系, 则 $\frac{\partial h}{\partial v} = I$, 即 $M(k) = I$, 所以滤波增益为:

$$K(k) = P(k|k-1)H^T(k)[H^T(k)P(k|k-1)H(k) + R(k)]^{-1} \quad (7-19)$$

估计方差计算方法为:

$$P(k|k) = [I - K(k)]H(k)P(k|k-1) \quad (7-20)$$

其中 $H(k)$ 为 (7-15) 式测量方程求偏导后的矩阵。

下面总结一下 EKF。从前面的分析可以看到, EKF 的基本原理是利用一阶泰勒级数展开, 使用展开后的一阶分量来代替原来的非线性方程。从泰勒级数的原理可知, 当泰勒级数本身是无穷次级数时, 等式才成立。因此, 如果只有一阶展开, 由 EKF 获得的估计状态其实和实际的非线性方程的结果就有差别。

前面已经提到过, 如果 Kalman 滤波器参数完全已知, Kalman 滤波器就可以获得最优估计。但是经过一阶泰勒级数展开之后, EKF 使用的参数和系统的实际参数不同, 从理论上来讲, 这种在非线性 Kalman 滤波器中利用一阶泰勒级数展开的 EKF 已经不能得到最优估计了。

为了减小估计方差, 研究者也给出了二阶 EKF, 此时 Kalman 滤波器的方程也变复杂了。当然, 从理论上讲, 三阶、四阶的 EKF 可能会使系统的参数与原来的非线性关系更为接近。但可想而知, 系统在进行估计的时候, 滤波器的方程会更加复杂, 而且从理论上来讲, 这种使用线性关系来代替非线性关系的方法, 只要其阶数有限, 就一定会带有误差。所以, 不管是几阶的 EKF, 都会有系统误差。甚至有时候如果非线性非常强, 经过线性化的结果并不能很好体现原来的非线性关系, 可能会导致估计系统发散。应该了解到, EKF 的这些缺陷是由工作原理造成的, 所以, 即便扩大阶数, 进行二阶或三阶泰勒展开, 也不能从根本上消除由于线性化造成的估计结果存在偏差。

因此, 人们研究了其它的非线性 Kalman 滤波的方法, 例如, 接下来将讲到的不敏 Kalman 滤波器 (UKF), 其基本原理和上面讲到的 EKF 有着本质上的差别, 在很多时候 UKF 都会比 EKF 获得更好的结果。

7.2 不敏 Kalman 滤波器 (UKF)

先前已经讨论过, 当系统的非线性比较严重时, EKF 的估计并不准确。这是因为 EKF 是依赖线性化传播状态均值和方差的, 即 EKF 要通过线性化非线性系统函数得到线性关系, 进

而用这种线性关系来重新表达原来系统的状态和噪声均值的非线性关系。本章要讨论的 UKF 是 Kalman 滤波器的延伸，UKF 降低了 EKF 线性化带来的误差。

本小节首先讨论在已知状态均值和协方差时，通过非线性关系，非线性函数状态和协方差是如何改变的，接下来再讨论所谓的“不敏变换”。通过不敏变换，随机变量的均值及协方差的计算结果和直接用非线性函数计算更接近一些。同时，还会利用这种不敏变换来改进 Kalman 滤波器，这会比 EKF 具有更小的线性化误差，最后，还会介绍一些改进的 UKF。

7.2.1 非线性变换的均值和方差

本节将研究随机变量经过非线性函数时，状态变量的均值和协方差是如何变化的，进而说明 EKF 的线性近似是如何产生均值和协方差误差的。本节知识本质上和状态估计、Kalman 滤波器、EKF、UKF 都没有直接联系，但为后面推导 UKF 做了铺垫，也为理解依赖线性化处理的 EKF 为何会导致较大估计误差提供依据。考虑非线性函数：

$$\begin{aligned} y_1 &= r \cos \theta \\ y_2 &= r \sin \theta \end{aligned} \quad (7-21)$$

假设有传感器可以测量距离 r 和角度 θ ，想将测量数据变换到直角坐标系 y_1 和 y_2 ，这种坐标变换通常为：

$$y = h(x) \quad (7-22)$$

其中 y 是 $h(x)$ 的状态输出函数，状态的定义为 $x = \begin{bmatrix} r \\ \theta \end{bmatrix}$ 。

假设 x_1 （距离 r ）的测量数据均值为 1，并带有方差为 δ_r 均匀分布的测量噪声， x_2 （角度 θ ）的均值为 $\frac{\pi}{2}$ ，且噪声方差为 δ_θ ，并且 r 和 θ 是相互独立的。

现在考虑 y_1 、 y_2 的均值，由式 (7-21) 直观的感觉是 y_1 的均值是 0， y_2 的均值是 1，那到底是不是这样呢？对 (7-22) 进行一阶线性化，两边取期望值，得到：

$$\begin{aligned} \bar{y} &= E[h(x)] \approx E[h(\bar{x}) + \frac{\partial h}{\partial x}|_{\bar{x}}(x - \bar{x})] \\ &= h(\bar{x}) = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{aligned} \quad (7-23)$$

上式中使用了关系 $\frac{\partial h}{\partial x}|_{\bar{x}}(x - \bar{x}) = 0$ 。
(7-23) 式也说明了 y_1 的均值为 0， y_2 的均值为 1。直觉和一阶线性化得出了相同的结论。但这里我不得不提前透漏一下，这还真不是事实，下

面来看看吧。

把 r 和 θ 写成如下均值加噪声的形式：

$$\begin{aligned} r &= \bar{r} + \tilde{r} \\ \theta &= \bar{\theta} + \tilde{\theta} \end{aligned} \quad (7-24)$$

其中 \bar{r} 和 $\bar{\theta}$ 分别为 r 和 θ 的均值， \tilde{r} 和 $\tilde{\theta}$ 分别为测量噪声。现在分别来看看 y_1 和 y_2 的值，以及与 $\bar{r}, \bar{\theta}, \tilde{r}$ 及 $\tilde{\theta}$ 的关系：

$$\begin{aligned} y_1 &= r \cos \theta \\ &= (\bar{r} + \tilde{r}) \cos(\bar{\theta} + \tilde{\theta}) \\ &= (\bar{r} + \tilde{r})(\cos \bar{\theta} \cos \tilde{\theta} - \sin \bar{\theta} \sin \tilde{\theta}) \end{aligned} \quad (7-25)$$

对上式求期望值，得：

$$E(y_1) = E[(\bar{r} + \tilde{r})(\cos \bar{\theta} \cos \tilde{\theta} - \sin \bar{\theta} \sin \tilde{\theta})] \quad (7-26)$$

因为 \bar{r} 与 $\tilde{\theta}$ 是相互独立的，相乘后再求期望值，得：

$$E(y_1) = \bar{y}_1 = \bar{r} \cos \bar{\theta} = 0 \quad (7-27)$$

这里和 (7-23) 矩阵中的第一个 0 是一致的。接下来看看 y_2 ，重复上述过程，得：

$$\begin{aligned} y_2 &= E(r \sin \theta) = E[(\bar{r} + \tilde{r}) \sin(\bar{\theta} + \tilde{\theta})] \\ &= E[(\bar{r} + \tilde{r})(\sin \bar{\theta} \cos \tilde{\theta} + \cos \bar{\theta} \sin \tilde{\theta})] \end{aligned} \quad (7-28)$$

$$\text{即: } \bar{y}_2 = \bar{r} \sin \bar{\theta} E(\cos \tilde{\theta}) = E(\cos \tilde{\theta}) \quad (7-29)$$

除非假设 $\tilde{\theta}$ 的分布，否则就无法求出 \bar{y}_2 的具体值。因此，这里假设 $\tilde{\theta}$ 是在 $\pm \theta_m$ 之间的均匀分布，这样可以计算出：

$$y_2 = E(\cos \tilde{\theta}) = \frac{\sin \theta_m}{\theta_m} \quad (7-30)$$

期望 \bar{y}_2 等于 1 来证实 (523) 式的合理性，但发现对于所有的 $\theta_m > 0$ ，有 $y_2 < 0$ ，则：

$$\frac{\sin \theta_m}{\theta_m} < 1 \quad (7-31)$$

并且，只有 $\theta_m \rightarrow 0$ 时满足

$$\lim_{\theta_m \rightarrow 0} \frac{\sin \theta_m}{\theta_m} = 1 \quad (7-32)$$

这个分析解释了一阶线性化存在问题，就是与其实际的非线性变换并不一致。

下面用仿真程序来说明上式的分析过程。利用 MATLAB 产生 300 个随机生成的 r 值和 θ 值，计算 y_1 和 y_2 后得到 300 个点，分别作为横纵坐标画在图上，如图 7.1 所示。其中 \tilde{r} 在 ± 0.01 间的均匀分布， $\tilde{\theta}$ 在 0.35 弧度之间分布， \tilde{r} 的小方差和 $\tilde{\theta}$ 的大方差导致点的分布形式呈弧形，这种弧形的分布结果导致 $\bar{y}_2 < 1$ 。MATLAB 程序中利用均匀分布函数 `rand` 产生 `rnoise` 表示 r 的噪声 \tilde{r} ，与 r 均值 `rmean` 相加得到 r 。 θ 的取值方式与 r 类似，程序中 θ 用变量 `sita` 表示。然后利用非线性关系 (7-21) 求 y_1 和 y_2 ，在程序中用 `y1` 和 `y2` 表示。再利用均值函数 `mean`，分别求出 y_1 和 y_2 的值，得到 `meany1` 和 `meany2`，即随机变量 y_1 和 y_2 的真实均值。程序中使用 Monte Carlo 仿真的方法，对 300 个点的横、纵坐标分别求均值得 $(-0.0009, 0.9801)$ ，这与 (7-27) 和 (7-30) 计算值 $(0, 0.9797)$ 有一定差别，但是也非常相似，如果增加点数，二者会更接近。程序剩余部分用于绘制结果图。

```

clc
clear
a=-0.01;b=0.01;
rnoise= a + (b-a).*rand(300,1);
rmean=1;
r=rnoise+rmean;

c=-0.35;d=0.35;
sitanoise= c + (d-c).*rand(300,1);
sitamean=pi/2;
sita=sitanoise+sitamean;

y1=r.*cos(sita);
y2=r.*sin(sita);

meany1=mean(y1);
meany2=mean(y2);
%画出结果图.
plot(y1,y2,'.')
xlabel('y1');ylabel('y2')
hold on

plot(0,1,'ko')
plot(meany1,meany2,'k*')

```

最后结果如图7.1所示。

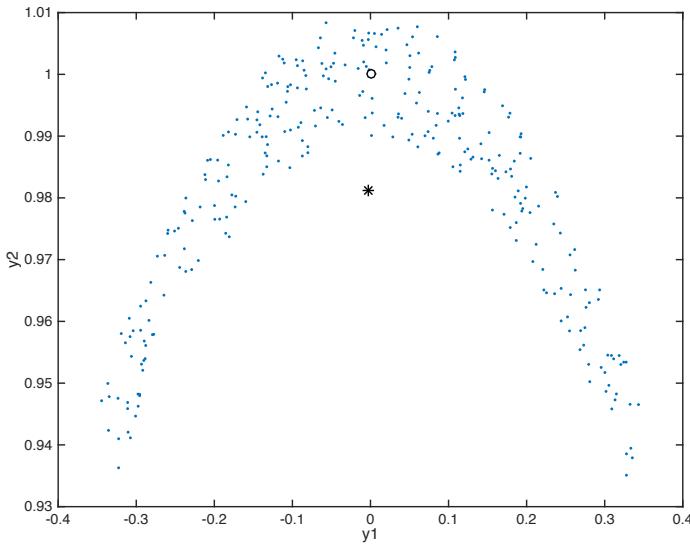


图7.1 非线性变换均值

图7.1中给出了300个点的一阶线性近似及其非线性变换的真实均值，其中‘o’表示线性变换之后的均值(0, 1)，“*”表示非线性变换的真实均值(-0.0023, 0.9811)。当EKF使用一阶线性化来更新状态均值时，得到的均值也是(0, 1)，而非真实均值(-0.0023, 0.9811)。

下面再来看看非线性变换的协方差。 y 的协方差表示为：

$$P_y = E[(y - \bar{y})(y - \bar{y})^T] \quad (7-33)$$

先来看一下基于线性变换的协方差变化。利用EKF的方法，对于(7-22)的测量方程，因为没有噪声项，所以有：

$$P_y = H P_x H^T \quad (7-34)$$

其中 H 为一阶泰勒级数展开式，即 $H = \frac{\partial h}{\partial x} \Big|_{x=\bar{x}}$ ，具体表示为：

$$H = \frac{\partial h}{\partial x} \Big|_{x=\bar{x}} = \begin{bmatrix} \cos \theta & -r \sin \theta \\ \sin \theta & r \cos \theta \end{bmatrix} \Big|_{x=\bar{x}} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \quad (7-35)$$

上式中使用 x 的均值 $\bar{x} = \begin{bmatrix} 1 \\ \frac{\pi}{2} \end{bmatrix}$ ，又 P_x 为状态 $\begin{bmatrix} r \\ \theta \end{bmatrix}$ 的方差，即：

$$P_x = E\left(\begin{bmatrix} r - \bar{r} \\ \theta - \bar{\theta} \end{bmatrix} \begin{bmatrix} r - \bar{r} & \theta - \bar{\theta} \end{bmatrix}^T\right) = \begin{bmatrix} \delta_r^2 & 0 \\ 0 & \delta_\theta^2 \end{bmatrix} \quad (7-36)$$

则由(7-34)得一阶线性化后的协方差为：

$$P_y = H P_x H^T = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \delta_r^2 & 0 \\ 0 & \delta_\theta^2 \end{bmatrix} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} \delta_\theta^2 & 0 \\ 0 & \delta_r^2 \end{bmatrix} \quad (7-37)$$

再来看看非线性方程 (7-22) 的实际协方差，利用数值计算的方法比较复杂，这里采用仿真方法求图 7.1 中 300 个数据点横、纵轴的方差。利用 Monte Carlo 仿真方法，N 个数据

点的方差为： $\text{cov}(x) = \frac{\sum_{i=1}^N (x_i - \bar{x})^2}{N}$ ，在本例中，横轴、纵轴的方差分别为：

$$\text{cov}(x) = \frac{\sum_{i=1}^{300} (x_i - \bar{x})^2}{300} \quad (7-38)$$
$$\text{cov}(y) = \frac{\sum_{i=1}^{300} (y_i - \bar{y})^2}{300}$$

其中 x_i 、 y_i 分别为横轴、纵轴的数据， \bar{x} 、 \bar{y} 分别为横轴、纵轴的均值，并且

$$\bar{x} = 0, \bar{y} = \frac{\sin(0.35)}{0.35}.$$

下面要将方差画在图上。为了更好地理解程序，先介绍一下将要使用的 rectangle 函数。该函数使用参数'Curvature',[1,1]画出圆形或椭圆形，画出图形的位置使用参数'position'来设置。

如参数设置为'position'，[0-0.35,1-0.01,0.35*2,0.01*2]，要画的圆心在 (0,1)，横轴、纵轴分别为 0.7, 0.02 的椭圆。还要说明的是，对所求方差 (7-38) 开方，得到误差的上下界，作为椭圆横、纵轴的半径，画的圆才是非线性变换的协方差范围。对应的程序语句如下：
rectangle('Position',[0-0.35,1-0.01,0.35*2,0.01*2],'Curvature',[1,1])

```
e=sqrt(sum((y1-0).^2)/300)
f=sqrt(sum((y2-sin(0.35)/0.35).^2)/300)
rectangle('Position',[0-e,sin(0.35)/0.37-f,e*2,f*2],'Curvature',[1,1])
```

其中第一个 rectangle 语句产生的是线性化的方差结果，即均值为 (0,1)， r 、 θ 的方差由 (7-37) 计算得到，开方后分别为 0.35, 0.01。要画成椭圆，则横、纵轴的直径分别为 0.7, 0.02。程序中 e 和 f 是使用 (7-38) 求的 300 个点的实际横纵轴的方差，再进行开方，利用 rectangle 函数画出的椭圆表示 y_1 和 y_2 的方差范围，结果如图 7.2 所示。

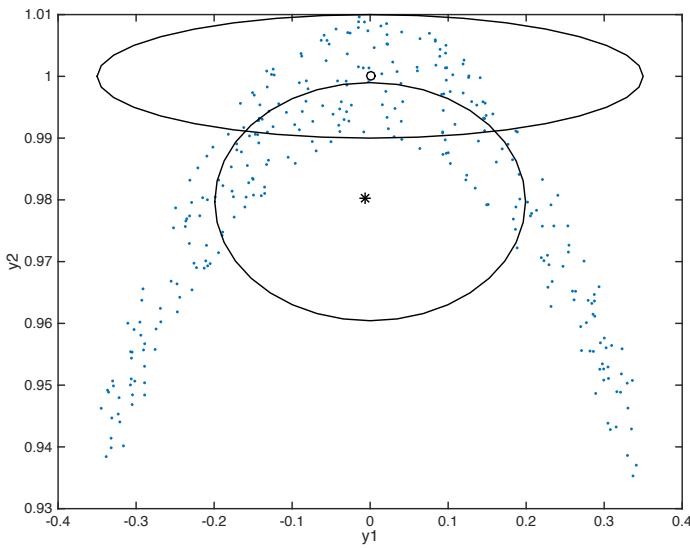


图 7.2 线性近似均值、协方差和非线性的真实均值、协方差比较

图 7.2 中 300 个随即生成点线性近似协方差和非线性的真实协方差，其中 r 的偏差在 ± 0.01 之间均匀分布， θ 在 ± 0.35 弧度间均匀分布。由于每次产生的随机数据并不完全相同，根据 (7-38) 求得的方差每次会略有偏差，但相差不多。可以确定的是经过非线性函数的方差和一阶线性近似有很大差别，一阶线性近似的结果很不准确。

7.2.2 不敏变换

前面已经知道可以使用很多点来求均值和方差，这样得到的估计值和真实值比较接近。如上面的例子，使用 300 个点通过程序计算之后，得到的结果和真实值比较接近。因此，可以不经过一次泰勒级数进行线性展开，而是使用很多点，让每个点经过非线性关系之后，得到这些点的均值和方差，进而得到相对比较准确的估计值。这种已知噪声分布情况，通过产生很多点并利用每一个点进行估计的方法叫 Monte Carlo 滤波器。

可是 300 个点比较多，由此就有人想到，可不可以在这些点当中选择几个具有足够代表性的合适的点，再利用这些点求取均值和方差，使估计值尽量接近真实均值和方差。如果这种方法可行，就不需要用 300 个点，而是使用几个点就可以达到目的。

对于这个问题需要考虑两个方面的内容：一是怎么选择这几个点，选取多少合适。二是这几个点对真实的均值和方差的重现度是多少，利用这几个点的非线性关系计算均值和方差与通过非线性函数得到的实际值相比，是否有足够的相似度。

具体来说，如何选择这几个点的方法就叫做不敏变换。这几个点选择之后对均值和方差的影响不大，被选择的点就称之为 sigma 点。至于如何选择这几个点，许多研究者也已给出

了答案，在这一小结中先来介绍几类不敏变换。

再重述一下问题：已知均值为 \bar{x} ，协方差为 P 的 n 维向量 x 和非线性函数 $z = h(x)$ ，

想估计 z 的均值 \bar{y} 和 P_z ，按照如下方式选取 $2n$ 个 sigma 点 $x^{(i)}$ ：

$$\begin{aligned} x^{(i)} &= \bar{x} + \tilde{x}^{(i)}, \quad i = 1, 2, \dots, 2n \\ \tilde{x}^{(i)} &= (\sqrt{n}P)_i^T, \quad i = 1, 2, \dots, n \\ \tilde{x}^{(n+i)} &= -(\sqrt{n}P)_i^T, \quad i = 1, 2, \dots, n \end{aligned} \quad (7-39)$$

其中 $\sqrt{n}P$ 是 nP 的矩阵平方根，即 $(\sqrt{n}P)^T(\sqrt{n}P) = nP$ ， $(\sqrt{n}P)_i$ 是 $(\sqrt{n}P)$ 的第 i 行。

接下来计算 sigma 点经过非线性变换的结果：

$$z^{(i)} = h(x^{(i)}), \quad i = 1, 2, \dots, 2n \quad (7-40)$$

则 z 的均值 \bar{z} 和协方差的近似如下：

$$\begin{aligned} \bar{z} &= \frac{1}{2n} \sum_{i=1}^{2n} z^{(i)} \\ P_{(z)} &= \frac{1}{2n} \sum_{i=1}^{2n} (z^{(i)} - \bar{z})(z^{(i)} - \bar{z})^T \end{aligned} \quad (7-41)$$

与前面按照方差选取 300 个点不同，现在先来分析一下这 $2n$ 个点的选取方式。根据方差的大小，“精心”挑选 $2n$ 个点，而且这些点均值为 \bar{x} ，协方差为 P ，那么这 $2n$ 个点经过非线性函数后， \bar{y} 和 P_z 和真实值是否接近呢？通过下面具体的例子，再给出求具体问题 sigma 点的 MATLAB 程序。

例 7.1 考虑式 (7-21) 的非线性变换。因为有两个独立变量 (r 和 θ)，所以 $n = 2$ 。 x 的

协方差是 $P = \begin{bmatrix} \delta_r^2 & 0 \\ 0 & \delta_\theta^2 \end{bmatrix}$ ，则按照 (7-39) 计算 sigma 点如下：

$$\begin{aligned} x^{(1)} &= \bar{x} + (\sqrt{2}P)_1^T = \begin{bmatrix} 1 \\ \frac{\pi}{2} \end{bmatrix} + \begin{bmatrix} \sqrt{2}\delta_r \\ 0 \end{bmatrix} = \begin{bmatrix} 1 + \sqrt{2}\delta_r \\ \frac{\pi}{2} \end{bmatrix} \\ x^{(2)} &= \bar{x} + (\sqrt{2}P)_2^T = \begin{bmatrix} \frac{1}{2} \\ \pi \end{bmatrix} + \begin{bmatrix} 0 \\ \sqrt{2}\delta_\theta \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \\ \frac{\pi}{2} + \sqrt{2}\delta_\theta \end{bmatrix} \\ x^{(3)} &= \bar{x} - (\sqrt{2}P)_1^T = \begin{bmatrix} 1 - \sqrt{2}\delta_r \\ \frac{\pi}{2} \end{bmatrix} \\ x^{(4)} &= \bar{x} - (\sqrt{2}P)_2^T = \begin{bmatrix} 1 \\ \frac{\pi}{2} - \sqrt{2}\delta_\theta \end{bmatrix} \end{aligned} \quad (7-42)$$

计算 sigma 点经过非线性变换 $z^{(i)} = h(x^{(i)})$ 得：

$$\begin{aligned}
z^{(1)} &= \begin{bmatrix} x_1^{(1)} \cos x_2^{(1)} \\ x_1^{(1)} \sin x_2^{(1)} \end{bmatrix} = \begin{bmatrix} (1 + \sqrt{2}\delta_r) \cos \frac{\pi}{2} \\ (1 + \sqrt{2}\delta_r) \sin \frac{\pi}{2} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 + \sqrt{2}\delta_r \end{bmatrix} \\
z^{(2)} &= \begin{bmatrix} x_1^{(2)} \cos x_2^{(2)} \\ x_1^{(2)} \sin x_2^{(2)} \end{bmatrix} = \begin{bmatrix} 1 \times \cos(\frac{\pi}{2} + \sqrt{2}\delta_\theta) \\ 1 \times \sin(\frac{\pi}{2} + \sqrt{2}\delta_\theta) \end{bmatrix} = \begin{bmatrix} \cos(\frac{\pi}{2} + \sqrt{2}\delta_\theta) \\ \sin(\frac{\pi}{2} + \sqrt{2}\delta_\theta) \end{bmatrix} \\
z^{(3)} &= \begin{bmatrix} x_1^{(3)} \cos x_2^{(3)} \\ x_1^{(3)} \sin x_2^{(3)} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 - \sqrt{2}\delta_r \end{bmatrix} \\
z^{(4)} &= \begin{bmatrix} x_1^{(4)} \cos x_2^{(4)} \\ x_1^{(4)} \sin x_2^{(4)} \end{bmatrix} = \begin{bmatrix} \cos(\frac{\pi}{2} - \sqrt{2}\delta_\theta) \\ \sin(\frac{\pi}{2} - \sqrt{2}\delta_\theta) \end{bmatrix}
\end{aligned} \tag{7-43}$$

则均值为：

$$\bar{z} = \frac{1}{2n} \sum_{i=1}^4 z^{(i)} = \frac{1}{4} \begin{bmatrix} \cos(\frac{\pi}{2} + \sqrt{2}\delta_\theta) + \cos(\frac{\pi}{2} - \sqrt{2}\delta_\theta) \\ 2 + \sin(\frac{\pi}{2} + \sqrt{2}\delta_\theta) + \sin(\frac{\pi}{2} - \sqrt{2}\delta_\theta) \end{bmatrix} \tag{7-44}$$

方差为：

$$P_z = \frac{1}{4} \sum_{i=1}^4 (z^{(i)} - \bar{z})(z^{(i)} - \bar{z})^T \tag{7-45}$$

下面给出 MATLAB 程序实现 (7-42) - (7-45) 的不敏变换过程。

%下面四句完成 (542) 式，求 sigma 点计算

```
clc
clear
b=0.01;d=0.35;
x(:,1)=[1+sqrt(2)*b;pi/2];
x(:,2)=[1;pi/2+sqrt(2)*d];
x(:,3)=[1-sqrt(2)*b;pi/2];
x(:,4)=[1;pi/2-sqrt(2)*d];
```

%下面利用循环完成 (7-43)，求非线性 $z^{(i)}$ 。

```
for i=1:4
    z(:,i)=[x(1,i)*cos(x(2,i));x(1,i)*sin(x(2,i))];
end
%画出所求非线性变量。
hold on
plot(z(1,1),z(2,1),'ko','MarkerSize',8,'MarkerFaceColor',[.5 1 .3]);
plot(z(1,2),z(2,2),'ko','MarkerSize',8,'MarkerFaceColor',[.5 1 .3]);
plot(z(1,3),z(2,3),'ko','MarkerSize',8,'MarkerFaceColor',[.5 1 .3]);
```

```

plot(z(1,4),z(2,4),'ko','MarkerSize',8,'MarkerFaceColor',[.5 1 .3]);

%根据 (7-44) 式求平均值, 再把均值画出来。
sigmamean=sum(z,2)/4;
plot(sigmamean(1),sigmamean(2),'^','MarkerSize',8,'MarkerFaceColor',[.5 1 .3])
%根据 (7-45) 求方差。
Pz=zeros(2);
for i=1:4
    Pz=Pz+(z(:,i)-sigmamean)*(z(:,i)-sigmamean)';
end
%将求得的方差值求平方后, 再画出一个范围, 是一个椭圆。
Pz=sqrt(Pz/4);
rectangle('Position',[sigmamean(1)-Pz(1,1),sigmamean(2)-Pz(2,2),Pz(1,1)*2,Pz(2,2)*2],'Curvature',[1,1],'LineWidth',3)
无迹变换的结果如图7.3所示。

```

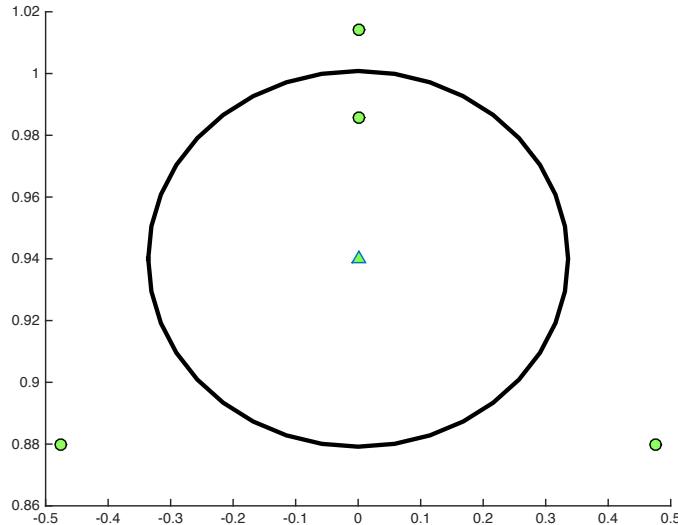


图 7.3 无迹变换的结果

程序中先用四条语句给出了 $x^{(i)}$, 为使程序运行方便, 这四个 sigma 点被排成一个矩阵, 每一列分别代表一个 sigma 点。然后用 4 次 for 循环, 产生每一个 sigma 点的非线性变换 $z^{(i)}$ 。而后使用 mean 函数得到 sigma 点的均值 sigmamean, 再用 plot 语句将 sigma 点经过非线性变换得到四个点, 用绿色的圆点画在图中。从图中可看出四个点位置比较特殊, 左下角和右下角的绿点由于方差 $\sqrt{2} \times 0.35$ 的作用, 两个点几乎离开了原来 300 个点群。 $y_1 = 0$ 点由于方差 $\sqrt{2} \times 0.01$ 的作用, 这两个绿点在所有点的上下边界附近。所化的大圆就是求得的方差, 小三角是均值。求方差的程序是经过 sigma 点的非线性变换获得 y 值, 并与其均值相减, 然后求平方和再乘以权值。此时的权值是 0.25, 将所得结果开方作为椭圆的半径。图中黑色大

圆圈就是用四个 sigma 点经过非线性变换获得方差。观察图 7.2 中的非线性变换的真实均值和协方差，发现图 7.3 中四个 sigma 点的均值和图 7.2 中的真实值（即星号表示的 300 个点的均值）相差依然较大。那么图 7.3 中“小三角”具体代表什么，下面将会讲到。

对于非线性变换，将 $y = h(x)$ 围绕 \bar{x} 用二阶泰勒级数展开：

$$y \approx h(x) = h(\bar{x}) + D\tilde{x}h + \frac{1}{2!}D_{\tilde{x}}^2h \quad (7-46)$$

代入非线性关系 (7-21) 后，得：

$$\begin{aligned} \bar{y} &\approx h(\bar{x}) + \frac{1}{2!}E(D_{\tilde{x}}^2h) \\ &= \begin{bmatrix} 0 \\ 1 \end{bmatrix} + \frac{1}{2}\delta_\theta^2 \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0.9387 \end{bmatrix} \end{aligned}$$

可以看到，图 7.3 中无迹变换的结果所获得的“小三角”其实是使用二阶泰勒展开之后所获得的均值，这两者是一致的。即经过 sigma 变换可以把 y 的均值匹配到真实均值的三阶，而线性化均值只使用 (7-46) 的第一项，仅能匹配到真实均值的一阶。所以说，通过 sigma 变换可以将均值向真实值进一步逼近。sigma 变换的方差也是一样的，向非线性变换的方差进一步逼近，感兴趣的读者可以参见相关参考文献^[15]。

必须指出，这样无迹变换得到的值并不与真值相等，而是向真值逼近。尽管这样，依然认为无迹变换是很有潜力的，和原来 EKF 的线性化相比，计算显然简单很多。这里选择四个点，就可以逼近真实值的三阶泰勒展开，与需要经过求偏导的 EKF 相比，把均值和协方差都向前推进了两阶，原来的 EKF 只是逼近到一阶。不敏变换还有其它类型，主要包括以下几种：

a. 一般型不敏变换

这种变换选取 $2n+1$ 个 sigma 点，并且取值的选择和前面也有所不同。

$$\begin{aligned} x^{(0)} &= \bar{x} \\ x^{(i)} &= \bar{x} + \tilde{x}^{(i)} \quad i = 1, 2 \dots 2n \\ \tilde{x}^{(i)} &= (\sqrt{(n+k)P})_i^T \quad i = 1, 2 \dots n \\ \tilde{x}^{(i)} &= -(\sqrt{(n+k)P})_i^T \quad i = 1, 2 \dots n \end{aligned} \quad (7-47)$$

$2n+1$ 个加权系数为：

$$\begin{aligned} w^{(0)} &= \frac{k}{n+k} \\ w^{(i)} &= \frac{1}{2(n+k)} \quad i = 1, 2 \dots 2n \end{aligned} \quad (7-48)$$

经过非线性变换：

$$z^{(i)} = h(x^{(i)}) \quad (7-49)$$

均值和协方差分别为：

$$\bar{z} = \sum_{i=0}^{2n} w^{(i)} z^{(i)} \quad (7-50)$$
$$P_z = \sum_{i=0}^{2n} w^{(i)} (z^{(i)} - \bar{z})(z^{(i)} - \bar{z})^T$$

该变换的 MATLAB 程序为：

```
clc
clear
b=0.01;d=0.35;

k=1;n=2;
x(:,1)=[1+sqrt(k+n)*b;pi/2];
x(:,2)=[1;pi/2+sqrt(k+n)*d];
x(:,3)=[1-sqrt(k+n)*b;pi/2];
x(:,4)=[1;pi/2-sqrt(k+n)*d];
x0=[1,pi/2];

nn=4;
w(1:nn)=1/(2*(n+k));
w0=k/(k+n);

y0=[x0(1)*cos(x0(2));x0(1)*sin(x0(2))];
for i=1:nn
    y(:,i)=[x(1,i)*cos(x(2,i));x(1,i)*sin(x(2,i))];
end

sigmamean=w0*y0;
for i=1:nn
    sigmamean=sigmamean+w(i)*y(:,i);
end

hold on

plot(y0(1),y0(2),'ko','MarkerSize',8,...
'MarkerFaceColor',[.5 0.10 .9]);
plot(y(1,1),y(2,1),'ko','MarkerSize',8,...
'MarkerFaceColor',[.5 0.10 .9]);
plot(y(1,2),y(2,2),'ko','MarkerSize',8,...
'MarkerFaceColor',[.5 0.10 .9]);
plot(y(1,3),y(2,3),'ko','MarkerSize',8,...
```

```

'MarkerFaceColor',[.5 0.10 .9]);
plot(y(1,4),y(2,4),'ko','MarkerSize',8,...
'MarkerFaceColor',[.5 0.10 .9]);

plot(sigmamean(1),sigmamean(2),'^','MarkerSize',...
8,'MarkerFaceColor',[.5 1 .3])

Py=w0*(y0-sigmamean)*(y0-sigmamean)';
for i=1:4
Py=Py+w(i)*(y(:,i)-sigmamean)*(y(:,i)-sigmamean)';
end
Py=sqrt(Py);
rectangle('Position',[sigmamean(1)-Py(1,1),...
sigmamean(2)-Py(2,2),Py(1,1)*2,Py(2,2)*2],...
'Curvature',[1,1],'LineWidth',3)

```

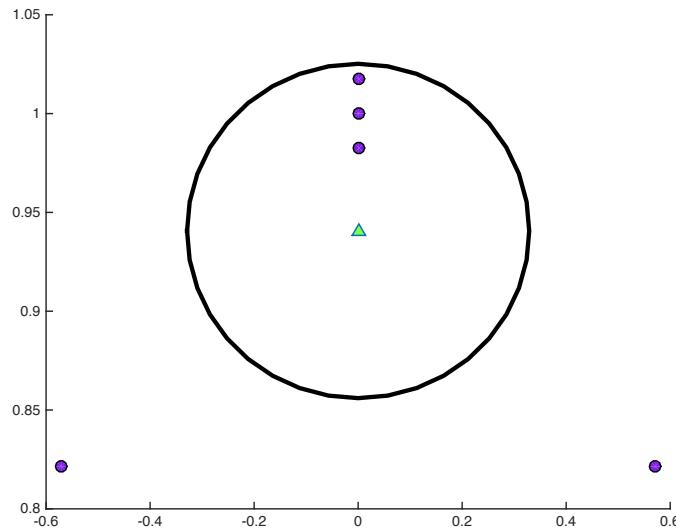


图 7.4 一般型不敏变换的均值与协方差

这一段程序和前面的非常相似，同样给出了几个 sigma 点。这里有五个 sigma 点，还包括一个 x_0 ，而且根号下的 n 变为 $k+n$ 。 k 现在取值为 1，依次计算五个 sigma 点经过非线性变换，最后通过权值求取均值，同时求出方差。在图上可以清楚地看到这五个点的分布（增加了 $(0,1)$ 点），求得的结果和图 7.3 不敏变换的均值是完全一样的，图上的小三角依然在原来的位置。但是由于式中的 $\sqrt{2}$ 变为 $\sqrt{3}$ ，所以左下角和右下角两个点向更远方伸展，这样求得的方差就比原来大，可以看到大圆变的更大了，但其均值并没有改变。

b 球型不敏变换这种变换方法，略微复杂一些，有以下几个步骤：

1) 选择权系数 $w^{(0)} \in [0,1]$

2) 选择其余的权系数

$$w^{(i)} = \frac{1-w^{(0)}}{n+1} \quad i=1,2,\dots,n+1 \quad (7-51)$$

3) 初始化线面一元向量

$$\begin{aligned} \delta_0^{(1)} &= 0 \\ \delta_1^{(1)} &= \frac{-1}{\sqrt{2w^{(1)}}} \\ \delta_2^{(1)} &= \frac{1}{\sqrt{2w^{(1)}}} \end{aligned} \quad (7-52)$$

对于 $j=2,\dots,n$, 按照下式递推获得向量 δ

$$\delta_i^{(j)} = \begin{cases} \begin{bmatrix} \delta_0^{(j-1)} \\ 0 \end{bmatrix} & i=0 \\ \begin{bmatrix} \delta_i^{(j-1)} \\ \frac{-1}{\sqrt{j(j+1)w^{(1)}}} \end{bmatrix} & i=1,\dots,j \\ \begin{bmatrix} O_j \\ \frac{j}{\sqrt{j(j+1)w^{(1)}}} \end{bmatrix} & i=j+1 \end{cases} \quad (7-53)$$

其中 O_j 是包含 j 个零的列向量。

4) 上面的递推完成后, 得到 n 维向量 $\delta_i^{(n)}, i=0,1,\dots,n+1$ 。球形不敏变换的 sigma 点为

$$x^{(i)} = \bar{x} + \sqrt{P_x} \delta_i^{(n)} \quad i=0,1,\dots,n+1 \quad (7-54)$$

其中 \bar{x} 是状态 x 的均值, P_x 为其方差。

5) 非线性变换及均值、协方差的求法同 (7.49) 与 (7.50)

球形不敏变换有点复杂, 下面通过 7.2.1 节的例子来看一下球形不敏变换具体的实现过程:

1) 选择权系数 $w^{(0)} = 0$

2) 由于 $n=2$, 即状态向量分别为 r 和 θ , 所以其余的权系数为:

$$w^{(1)} = w^{(2)} = w^{(3)} = \frac{1-w^{(0)}}{n+1} \quad (7.55)$$

3) 初始化下面的一元向量:

$$\begin{aligned}\delta_0^{(1)} &= 0 \\ \delta_1^{(1)} &= \frac{1}{\sqrt{2w^{(1)}}} \\ \delta_2^{(1)} &= \frac{1}{\sqrt{2w^{(1)}}}\end{aligned}\quad (7.56)$$

对于 $j = 2$, 得向量 δ :

$$\begin{aligned}i = 0 \text{时}, \quad \delta_0^{(2)} &= \begin{bmatrix} \delta_0^{(1)} \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ i = 1 \text{时}, \quad \delta_1^{(2)} &= \begin{bmatrix} \delta_1^{(1)} \\ -1 \\ \hline \sqrt{6w^{(1)}} \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \\ \hline \sqrt{6w^{(1)}} \end{bmatrix} \\ i = 2 \text{时}, \quad \delta_2^{(2)} &= \begin{bmatrix} \delta_2^{(1)} \\ -1 \\ \hline \sqrt{6w^{(1)}} \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \\ \hline \sqrt{6w^{(1)}} \end{bmatrix} \\ i = 3 \text{时}, \quad \delta_3^{(2)} &= \begin{bmatrix} 0 \\ 2 \\ \hline \sqrt{6w^{(1)}} \end{bmatrix}\end{aligned}\quad (7.57)$$

5) 则 sigma 点为:

$$\begin{aligned}x^{(i)} &= \bar{x} + \sqrt{P} \delta_i^{(2)} \\ &= \begin{bmatrix} 1 \\ \frac{\pi}{2} \\ \hline 2 \end{bmatrix} + \begin{bmatrix} 0.01 & 0 \\ 0 & 0.35 \end{bmatrix} \delta_i^{(2)}\end{aligned}\quad (7.58)$$

即:

$$\begin{aligned}x^{(0)} &= \begin{bmatrix} 1 \\ \frac{\pi}{2} \\ \hline 2 \end{bmatrix} \\ x^{(1)} &= \begin{bmatrix} 1 \\ \frac{\pi}{2} \\ \hline 2 \end{bmatrix} + \begin{bmatrix} 0.01 & 0 \\ 0 & 0.35 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ \hline \sqrt{2w^{(1)}} \end{bmatrix} = \begin{bmatrix} 1 + \frac{0.01}{\sqrt{2w^{(1)}}} \\ \frac{\pi}{2} + \frac{-0.35}{\sqrt{6w^{(1)}}} \\ \hline \sqrt{6w^{(1)}} \end{bmatrix} \\ x^{(2)} &= \begin{bmatrix} 1 \\ \frac{\pi}{2} \\ \hline 2 \end{bmatrix} + \begin{bmatrix} 0.01 & 0 \\ 0 & 0.35 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ \hline \sqrt{2w^{(1)}} \end{bmatrix} = \begin{bmatrix} 1 + \frac{0.01}{\sqrt{2w^{(1)}}} \\ \frac{\pi}{2} + \frac{-0.35}{\sqrt{6w^{(1)}}} \\ \hline \sqrt{6w^{(1)}} \end{bmatrix} \\ x^{(3)} &= \begin{bmatrix} 1 \\ \frac{\pi}{2} \\ \hline 2 \end{bmatrix} + \begin{bmatrix} 0.01 & 0 \\ 0 & 0.35 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \\ \hline \sqrt{6w^{(1)}} \end{bmatrix} = \begin{bmatrix} 1 \\ \frac{\pi}{2} + \frac{7}{\sqrt{6w^{(1)}}} \\ \hline \sqrt{6w^{(1)}} \end{bmatrix}\end{aligned}\quad (7.59)$$

6) 再利用非线性关系求出 $z^{(i)}$, 进而求出 z 的均值及方差。

为方便读者理解, 给出如下 MATLAB 程序来实现 (7.55) - (7.59) 的球型不敏变换。

```

clc
clear
n=2;nn=4;
W0=0;
W(1)=(1-W0)/(n+1);
W(2)=W(1);
W(3)=W(1);

gama(:, 1)=[0;0];
gama(:, 2)=[-1/sqrt(2*W(1));-1/sqrt(6*W(1))];
gama(:, 3)=[1/sqrt(2*W(1));-1/sqrt(6*W(1))];
gama(:, 4)=[0;2/sqrt(6*W(1))];

meanx=[1;pi/2];
Px=diag([0.01^2, 0.35^2]);
for i=1:4
    x(:, i)=meanx+sqrt(Px)*gama(:, i);
end

for i=1:nn
    y(:, i)=[x(1, i)*cos(x(2, i));x(1, i)*sin(x(2, i))];
end

sigmamean=W0*y(:, 1);
for i=1:nn-1
    sigmamean=sigmamean+W(i)*y(:, i+1);
end

hold on
plot(y(1, 1), y(2, 1), 'ko', 'MarkerSize', 8, 'MarkerFaceColor', [.5 0.90 .9]);
plot(y(1, 2), y(2, 2), 'ko', 'MarkerSize', 8, 'MarkerFaceColor', [.5 0.90 .9]);
plot(y(1, 3), y(2, 3), 'ko', 'MarkerSize', 8, 'MarkerFaceColor', [.5 0.90 .9]);
plot(y(1, 4), y(2, 4), 'ko', 'MarkerSize', 8, 'MarkerFaceColor', [.5 0.90 .9]);

plot(sigmamean(1), sigmamean(2), '^', 'MarkerSize', 8, 'MarkerFaceColor', [.5 1 .3])

Py=W0*(y(:, 1)-sigmamean)*(y(:, 1)-sigmamean)';
for i=1:3
    Py=Py+W(i)*(y(:, i+1)-sigmamean)*(y(:, i+1)-sigmamean)';
end
Py=sqrt(Py);
rectangle('Position', [sigmamean(1)-Py(1, 1), sigmamean(2)-Py(2, 2), Py(1, 1)*2, Py(2, 2)*2], 'Curvature', [1, 1], 'LineWidth', 1)

```

先来看看程序，这段程序和一般型不敏变换的程序相似，只是略微复杂一些。首先需要

产生四个 sigma 点，即 (7.57) 中的四个式子，在程序中用 gama 表示这四个式子中的 $\delta_i^{(2)}$ ， $i = 0, 1, 2, 3$ ；然后设置均值和方差，用 for 循环实现 (7.59) 中求 sigma 点的过程，接下来通过非线性变换求均值和方差。

球型不敏变换求取均值与协方差的结果如图 7.5 所示。从图中可以看出，均值还是和原来一样，但有两点不同：一是这四个点不再是对称的；二是如图中所示，圆表示的方差变小，说明现在的方差比原来的小，得到的值会更精确。在程序中把 w_0 ($0 \leq w_0 \leq 1$) 的值设置为 0，通过仿真可以看出如果 w_0 的取值变大，其方差表示的圆就变大，但均值还是不变。因此， w_0 一般取为 0，只是用了 3 个 sigma 点，其中 $x^{(0)}$ 这个点没被使用。

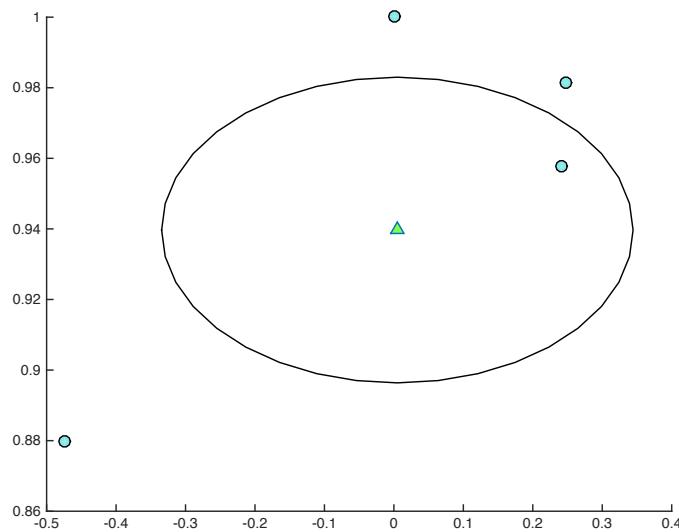


图 7.5 球型不敏变换的均值与协方差

7.2.3 UKF

通过前面学习我们知道，Kalman 滤波器的算法是利用更新时间和测量的方法来传播系统均值和协方差的。如果系统是线性的，均值和协方差可以进行准确的更新。如果是非线性的，采用扩展 Kalman 滤波器对均值和协方差更新，结果是非常不准确的，这一点从 7.2.2 的例子中即可看出。而利用不敏变换可以把均值和协方差向真值逼近到三阶，和二阶的泰勒级数一致。下面，来看一下如何利用无迹变换来修正非线性的 Kalman 滤波器。由于标准的 Kalman 滤波器是通过状态方程和测量方程两次向前预测并更新状态，如果过程方程和状态方程都是非线性的，那么这两次更新状态（一是预测，二是利用测量进行更新）都需要使用

不敏变换进行更准确的预测和更新，下面给出具体的 UKF 算法。

1) 已知 n 维离散时间非线性系统

$$\begin{aligned} x(k+1) &= f(x(k)) + w(k) \\ z(k) &= h(x(k)) + v(k) \\ w(k) &\sim (0, Q(k)) \\ v(k) &\sim (0, R(k)) \end{aligned} \quad (7.60)$$

2) 滤波器初始化如下：

$$\begin{aligned} \hat{x}(0|0) &= E(x_0) = x_0 \\ P(0|0) &= P_0 \end{aligned} \quad (7.61)$$

和前面情况一样，一般情况下， x_0 可以设为任意向量，只要与状态 x 的维数 n 相同即可， P_0

可以设为 $n \times n$ 的正定对角阵。

3) 状态向一步预测，步骤如下：

a. 利用 $k-1$ 时刻的估计状态， $\hat{x}(k-1|k-1)$ 即估计方差 $P(k-1|k-1)$ ，产生 sigma 点

$\hat{x}^{(i)}(k-1|k-1)$ 。sigma 点产生的方法可以使用最简单的不敏变换方法，也可以使用一般型不敏变换 (7-47)，或球形不敏变换 (7-52) - (7-54)，这里使用 (7-47) 作为例子说明如何使用 $\hat{x}(k-1|k-1)$ 及 $P(k-1|k-1)$ 产生向一步预测的 sigma 点：

$$\begin{aligned} \hat{x}^{(i)}(k-1|k-1) &= \hat{x}(k-1|k-1) + \tilde{x}^{(i)} \quad i = 1, \dots, 2n \\ \tilde{x}^{(i)} &= (\sqrt{n}P(k-1|k-1))_i^T, \quad i = 1, \dots, n \\ \tilde{x}^{(n+i)} &= -(\sqrt{n}P(k-1|k-1))_i^T, \quad i = 1, \dots, n \end{aligned} \quad (7.62)$$

其中 $(\sqrt{n}P(k-1|k-1))_i^T$ 表示 n 乘以估计方差 $P(k-1|k-1)$ 后，进行矩阵开方运算，选择其中的第 i 行并转置后得到 n 维向量。

b. 将 sigma 点 $\hat{x}^{(i)}(k-1|k-1)$ 代入 (7.60) 中的系统状态方程，得：

$$\hat{x}^{(i)}(k|k-1) = f(\hat{x}^{(i)}(k-1|k-1)) \quad (7.63)$$

c. 合并向量 $\hat{x}^{(i)}(k|k-1)$ 来获得 k 时刻的向一步状态估计：

$$\hat{x}(k|k-1) = \frac{1}{2n} \sum_{i=1}^{2n} \hat{x}^{(i)}(k|k-1) \quad (7.64)$$

d. 考虑过程噪声，用下面的公式求向一步预测的估计协方差：

$$P(k|k-1) = \frac{1}{2n} \sum_{i=1}^{2n} (\hat{x}^{(i)}(k|k-1) - \hat{x}(k|k-1)) (\hat{x}^{(i)}(k|k-1) - \hat{x}(k|k-1))^T + Q(k-1)$$

(7-65)

4) 利用测量更新向前一步预测状态, 步骤如下:

a. 利用向前一步预测状态, $\hat{x}(k|k-1)$ 及其方差 $P(k|k-1)$, 先产生 sigma 点 $\hat{x}^{(i)}(k|k-1)$,

方法如下:

$$\begin{aligned}\hat{x}^{(i)}(k|k-1) &= \hat{x}(k|k-1) + \tilde{x}^{(i)}, \quad i=1, \dots, 2n \\ \tilde{x}^{(i)} &= (\sqrt{n} P(k|k-1))_i^T, \quad i=1, \dots, n \\ \tilde{x}^{(n+i)} &= -(\sqrt{n} P(k|k-1))_i^T, \quad i=1, \dots, n\end{aligned}\quad (7.66)$$

注意, 也可以重复利用刚才产生的那些点 (7-61), 而不产生新的 sigma 点, 这样虽然会损失一定精度, 但可以减小计算负担。

b. 用已知的非线性测量方差将 sigma 点转换为量测预测 $\hat{z}^{(i)}(k|k-1)$:

$$\hat{z}^{(i)}(k|k-1) = h(\hat{x}^{(i)}(k|k-1)) \quad (7.67)$$

c. 合并向量 $\hat{z}^{(i)}(k|k-1)$ 获得 k 时刻的测量预测:

$$\hat{z}(k|k-1) = \frac{1}{2n} \sum_{i=1}^{2n} \hat{z}^{(i)}(k|k-1) \quad (7.68)$$

d. 考虑测量噪声量测预测的协方差为:

$$P_z = \frac{1}{2n} \sum_{i=1}^{2n} (\hat{z}^{(i)}(k|k-1) - \hat{z}(k|k-1)) (\hat{z}^{(i)}(k|k-1) - \hat{z}(k|k-1))^T + R(k) \quad (7.69)$$

e. 再来估计 $\hat{x}(k|k-1)$ 与 $\hat{z}(k|k-1)$ 之间的协方差:

$$P_{xz} = \frac{1}{2n} \sum_{i=1}^{2n} (\hat{x}^{(i)}(k|k-1) - \hat{x}(k|k-1)) (\hat{z}^{(i)}(k|k-1) - \hat{z}(k|k-1))^T \quad (7.70)$$

5) 利用 Kalman 滤波方程进行估计, 先计算滤波增益 $K(k) = P_{xz} P_z^{-1}$, 则更新的状态估计及方差为:

$$\hat{x}(k|k) = \hat{x}(k|k-1) + K(k)(z(k) - \hat{z}(k|k-1)) \quad (7.71)$$

$$P(k+1|k+1) = P(k|k+1) - K(k)P_z K^T(k) \quad (7.72)$$

上述算法中假设过程方程和量测方程相对于噪声是线性的, 而过程和测量方程相对于噪声是非线性的公式如下

$$\begin{aligned} x(k+1) &= f(x(k), w(k)) \\ z(k) &= h(x(k), v(k)) \end{aligned} \quad (7-73)$$

需要将噪声扩充到状态向量中，状态向量变为：

$$x_a(k) = \begin{bmatrix} x(k) \\ w(k) \\ v(k) \end{bmatrix} \quad (7-74)$$

然后用 UKF 估计扩充的状态 $x_a(k)$ ，UKF 初始化为：

$$\hat{x}_a = \begin{bmatrix} E(x_0) \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \hat{x}(0|0) \\ 0 \\ 0 \end{bmatrix} \quad (7-75)$$

$$P_a(0|0) = \begin{bmatrix} E[(x - \hat{x}(0|0))(x - \hat{x}(0|0))^T] & 0 & 0 \\ 0 & Q_0 & 0 \\ 0 & 0 & R_0 \end{bmatrix} \quad (7-76)$$

利用 UKF 算法估计的是扩充的均值和协方差，因此，需要从式 (7-65) 和 (7-69) 中移除 $Q(k-1)$ 和 $R(k)$ 。

如表 7.2 所示，给出了标准 Kalman 滤波器和 UKF 的差别。

表 7.2 标准 Kalman 滤波器和 UKF 的差别

		标准 Kalman 滤波器	UKF
系统模型描述	$x(k+1) = A(k)x(k) + w(k)$ $z(k) = C(k)x(k) + v(k)$ $E[w(k)w^T(j)] = Q(k)\delta(k-j)$ $E[v(k)v^T(j)] = R(k)\delta(k-j)$ $E[w(k)v^T(j)] = 0$	设置状态初值	$x(k) = f(x(k-1), w(k-1))$ $z(k) = h(x(k), v(k))$ $E[w(k)w^T(j)] = Q(k)\delta(k-j)$ $E[v(k)v^T(j)] = R(k)\delta(k-j)$ $E[w(k)v^T(j)] = 0$ $x_a(k) = \begin{bmatrix} x(k) \\ w(k) \\ v(k) \end{bmatrix}$ $\hat{x}_a = \begin{bmatrix} E(x_0) \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \hat{x}(0 0) \\ 0 \\ 0 \end{bmatrix}$ $P_a(0 0) = \begin{bmatrix} E[(x - \hat{x}(0 0))(x - \hat{x}(0 0))^T] & 0 & 0 \\ 0 & Q_0 & 0 \\ 0 & 0 & R_0 \end{bmatrix}$
Kalman 滤波器	$\hat{x}(k k-1) =$ $A(k-1)\hat{x}(k-1 k-1)$		$\hat{x}^{(i)}(k-1 k-1) = \hat{x}(k-1 k-1) + \tilde{x}^{(i)} \quad i = 1, \dots, 2n$ $\tilde{x}^{(i)} = (\sqrt{n}P(k-1 k-1))_i^T, \quad i = 1, \dots, n$ $\tilde{x}^{(n+i)} = -(\sqrt{n}P(k-1 k-1))_i^T, \quad i = 1, \dots, n$ $\hat{x}(k k-1) = \frac{1}{2n} \sum_{i=1}^{2n} \hat{x}^{(i)}(k k-1)$

更新	$\hat{x}(k k) = \hat{x}(k k-1) + K(k)(z(k) - C(k)\hat{x}(k k-1))$	$\begin{aligned}\hat{x}^{(i)}(k k-1) &= \hat{x}(k k-1) + \tilde{x}^{(i)}, \quad i=1, \dots, 2n \\ \tilde{x}^{(i)} &= (\sqrt{n}P(k k-1))_i^T, \quad i=1, \dots, n \\ \tilde{x}^{(n+i)} &= -(\sqrt{n}P(k k-1))_i^T, \quad i=1, \dots, n\end{aligned}$ $\begin{aligned}\hat{z}^{(i)}(k k-1) &= h(\hat{x}^{(i)}(k k-1)) \\ \hat{z}(k k-1) &= \frac{1}{2n} \sum_{i=1}^{2n} \hat{z}^{(i)}(k k-1) \\ \hat{x}(k k) &= \hat{x}(k k-1) + K(k)(z(k) - \hat{z}(k k-1))\end{aligned}$
滤波器增益	$K(k) = P(k k-1)C^T(k) \times (C(k)P(k k-1)C^T(k) + R(k))^{-1}$	$P_z = \frac{1}{2n} \sum_{i=1}^{2n} (\hat{z}^{(i)}(k k-1) - \hat{z}(k k-1)) (\hat{z}^{(i)}(k k-1) - \hat{z}(k k-1))^T$ $P_{xz} = \frac{1}{2n} \sum_{i=1}^{2n} (\hat{x}^{(i)}(k k-1) - \hat{x}(k k-1)) (\hat{z}^{(i)}(k k-1) - \hat{z}(k k-1))^T$ $K(k) = P_{xz} P_z^{-1}$
向前一步预测方差	$P(k k-1) = A(k-1)P(k-1 k-1)A^T(k-1) + Q(k-1)$	$P(k k-1) = \frac{1}{2n} \sum_{i=1}^{2n} (\hat{x}^{(i)}(k k-1) - \hat{x}(k k-1)) (\hat{x}^{(i)}(k k-1) - \hat{x}(k k-1))^T$
状态估计方	$P(k k) = (I - K(k)C(k)) \times P(k k-1)$	$P(k+1 k+1) = P(k k+1) - K(k)P_z K^T(k)$

	差	
--	---	--

7.3 小结

对于含有非线性过程函数及测量函数的系统，原来的只能处理线性系统的标准 Kalman 滤波器不再使用，因此，本章给出了几种处理方法。首先，研究了 EKF (EKF)，通过一阶泰勒级数将非线性函数线性化。接下来研究了不敏变换以及由此得出的 UKF，并且，UKF 比 EKF 好一些，可以将非线性近似的逼近到三阶。

在仿真时选择 300 个点，求得的均值和真值很接近。由此想到，如果选择更多的点，而不仅仅是 $2n$ 个 sigma 点，可以逼近更高阶。因此，在已知系统的非线性函数和噪声分布的情况下，使用更多的点可以逼近实际的均值和方差，但同时意味着要进行大量的计算。另一方面，要产生这些点，也要求必须已知非线性关系和噪声的分布情况。

更特殊的情况，在系统噪声分布关系未知的情况下。比如不清楚状态和测量之间的关系，也不清楚噪声的分布，只给一个条件概率密度函数，如 $P(x(k) | z(1), z(2), \dots, z(k))$ 描述的状态 $x(k)$ 和测量 $z(1), z(2), \dots, z(k)$ 关系。

同时也发现测量数据和状态的条件概率密度函数，可能不只一个峰值的。对于多峰值的概率密度函数，状态的均值并不真正代表实际状态的情况。由于不知道噪声的分布，就不能产生 sigma 点。对于这种情况，目前介绍的这些方法就都不能适用。那应该怎么办？研究者给出了粒子滤波器来处理这种特别复杂的情况。

在实际应用中到底选择哪一种滤波器呢？这就需要根据条件进行分析，如果情况比较清楚，就选择 EKF 或 UKF 滤波器。对于多峰值、非常复杂、情况不明的系统关系，只能使用粒子滤波器。关于粒子滤波器的知识，请读者参考相关书籍，本书不在此处详述了，严格讲，粒子滤波器已经超过了 Kalman 滤波器的范围了。