计算复杂度降低的基于 CDKF 的 SLAM 算法*

陈 晨，程荫杭
（北京交通大学 电子信息工程学院，北京 100044）

摘 要：为了降低移动机器人基于中心差分卡尔曼滤波(CDKF)的同时定位与地图构建(SLAM)算法的计算复杂度,使其适于较大规模环境中的应用,提出了一种改进的 CDKF SLAM 算法。该算法以 CDKF 的线性回归卡尔曼滤波(LRKF)形式为基础,利用 SLAM 自身特点,重构其预测和观测更新过程中的状态变量及相应的方差矩阵,改进 CDKF 的采样方法,从而将 CDKF SLAM 算法的计算复杂度降为 $O(n^2)$。不同规模环境中的仿真实验及停车场数据集的实验验证了在不改变 CDKF SLAM 算法估计准确度的条件下,本文算法的运行时间明显缩短,更适于大规模环境中的应用。

关键词：同时定位与地图构建；中心差分卡尔曼滤波；线性回归卡尔曼滤波；计算复杂度

中图分类号：TP24;TP301.6　　**文献标志码**：A　　**文章编号**：1001-3695(2012)09-3280-05

doi:10.3969/j.issn.1001-3695.2012.09.021

Computational complexity reduced CDKF based SLAM algorithm

CHEN Chen, CHENG Yin-hang
(*School of Electronics & Information Engineering, Beijing Jiaotong University, Beijing* 100044, *China*)

Abstract: In order to reduce the computational complexity of the CDKF SLAM algorithm for large-scale environment, this paper proposed an improved CDKF SLAM algorithm which was presented in the context of the linear-regression Kalman filter. Based on the properties of SLAM, it improved the sampling strategy by reconstructing the estimated state and its covariance during prediction and measurement update. The complexity was thus reduced to $O(n^2)$. Simulation experiments in different scale environments and experiments of the car park database proves that the proposed algorithm keep the same accuracy as the general CDKF SLAM, while its running time became much shorter. This makes it more suitable for applications in large-scale environment.

Key words: simultaneous localization and mapping (SLAM); central difference Kalman filter (CDKF); linear-regression Kalman filter (LRKF); computational complexity

0 引言

移动机器人同时定位与地图构建(simultaneous localization and mapping,SLAM)问题可以描述为:移动机器人是利用自身携带的传感器,在构建未知环境地图的同时,确定自身在该地图中位姿的过程[1]。SLAM 问题一直是移动机器人研究领域的一个热点问题,也被认为是移动机器人能否实现完全自主控制的关键。

基于卡尔曼滤波的算法是解决 SLAM 问题时最主要的算法之一。其中,基于中心差分卡尔曼滤波(central difference Kalman filter,CDKF)的 SLAM 算法以 Sterling 多项式插值[2]为基础处理非线性系统模型,能够有效地降低扩展卡尔曼滤波(extended Kalman filter,EKF)SLAM 算法中由线性化过程引入的误差,从而提高 SLAM 估计的准确度[3];且与需要确定三个经验参数的无迹卡尔曼滤波(unscented Kalman filter,UKF)SLAM 算法相比,它只需要确定一个参数[4]。然而 CDKF SLAM 算法也存在计算复杂度较高的缺点。计算复杂度是衡量算法性能的重要指标,尤其对于 SLAM 问题,随着新观测特征的加入,所估计的状态变量的维数 n 也会不断增加,计算复杂度高的算法将逐渐无法满足实时性的需要。CDKF SLAM 算法的计算复杂度为 $O(n^3)$,这使它与计算复杂度为 $O(n^2)$ 的 EKF SLAM 算法相比,更难应用于大规模的环境中。文献[5]中提出了一种基于均方根 CDKF 的 SLAM 算法,该算法虽然能够在一定程度上提高了算法的运算速度,但是并未从根本上改变算法的计算复杂度。

为此,本文提出了一种计算复杂度为 $O(n^2)$ 的 CDKF SLAM 算法。该算法中,将 CDKF 看做线性回归卡尔曼滤波(linear-regression Kalman filter,LRKF)的一种特殊形式[6],并在此形式的基础上,利用 SLAM 问题本身的特点,重新构造预测和观测更新过程中的状态变量及其相应的方差矩阵,改进 CDKF 的采样方法,从而降低了算法的计算复杂度。理论分析和实验均表明,改进后的 SLAM 算法与一般的 CDKF SLAM 算法相比,估计的准确度相当,而计算复杂度则降为 $O(n^2)$。

1 CDKF SLAM 算法的 LRKF 形式

文献[6]中提出,CDKF 可以看做是 LRKF 的一种特殊形式,即将 CDKF 视为对非线性过程模型和更新模型的一种隐含的统计线性化。此时将其用于 SLAM 问题,可得到该线性化机

收稿日期：2012-02-23；**修回日期**：2012-03-27　**基金项目**：国家教育部留学回国人员科研启动基金项目(教外司留[2010]1516 号)

作者简介：陈晨(1982-),女,辽宁大连人,博士研究生,主要研究方向为移动机器人同时定位与地图构建(chenchen_5050@163.com);程荫杭(1945-),男,教授,博导,主要研究方向为运输自动化与控制、移动机器人.

制下的 CDKF SLAM 算法,称之为 CDKF SLAM 算法的 LRKF 形式。同时,它也对计算复杂度降低的 CDKF SLAM 算法的提出具有重要的基础作用。

1.1 线性回归

对一个非线性函数 $y=g(x)$,LRKF 用如下线性模型对其进行估计:

$$y=\boldsymbol{A}x+\boldsymbol{b}+\boldsymbol{e} \tag{1}$$

其中:$\boldsymbol{A}$ 和 $\boldsymbol{b}$ 分别表示回归矩阵和回归矢量;$\boldsymbol{e}$ 表示由于线性化产生的加性误差。估计得到的 $\boldsymbol{A}$、$\boldsymbol{b}$ 应满足式(2)中定义的目标函数,即进行最小均方误差估计:

$$\min_{\boldsymbol{A},\boldsymbol{b}}\int_{-\infty}^{+\infty}[y-(\boldsymbol{A}x+\boldsymbol{b})]^{\mathrm{T}}[y-(\boldsymbol{A}x+\boldsymbol{b})]p(x)\mathrm{d}x \tag{2}$$

其中:$p(x)$ 表示状态矢量 $\boldsymbol{x}$ 的概率密度函数。当确定该线性模型后,LRKF 即可利用常规的卡尔曼滤波公式进行递推估计。确定线性模型的具体方法为:LRKF 首先选择 $r+1$ 个回归点 $\{\chi_i,\omega_i\}_{i=0}^{r}$,其中 ω_i 表示每个回归点相应的权值,且它们的均值和方差应满足以下条件:

$$\hat{x}=\sum_{i=0}^{r}\omega_i\chi_i=E\{x\} \tag{3}$$

$$P_{xx}=\sum_{i=0}^{r}\omega_i(\chi_i-\hat{x})(\chi_i-\hat{x})^{\mathrm{T}}=E\{(\boldsymbol{x}-\hat{x})(\boldsymbol{x}-\hat{x})^{\mathrm{T}}\} \tag{4}$$

将这些回归点分别代入非线性函数 $g(x)$,可得到 $\{y_i=g(\chi_i)\}_{i=0}^{r}$。则求 $\boldsymbol{A}$ 和 $\boldsymbol{b}$ 的公式[7]分别可表示为

$$\boldsymbol{A}=P_{xy}^{\mathrm{T}}P_{xx}^{-1} \tag{5}$$

$$\boldsymbol{b}=\hat{y}-\boldsymbol{A}\hat{x} \tag{6}$$

其中:

$$\hat{y}=\sum_{i=0}^{r}\omega_iy_i,\ P_{yx}=\sum_{i=0}^{r}\omega_i(y_i-\hat{y})(\chi_i-\hat{x})^{\mathrm{T}} \tag{7}$$

当用于 SLAM 问题时,LRKF 即在递推时采用上面的公式分别估计非线性的机器人运动模型和传感器观测模型,从而完成预测和观测更新过程。

1.2 预测过程

设机器人的运动模型可表示:

$$\boldsymbol{x}_k=f(x_{k-1},u_k) \tag{8}$$

其中:$f(\cdot)$ 是机器人的运动模型;x_k 表示 k 时刻由机器人位姿 $x_{r,k}$ 和地图中的特征位置 $x_{f_i,k}(i=1,\cdots,n_f,n_f$ 表示特征的数量)所构成的联合矢量;u_k 表示 $k-1$ 时刻到 k 时刻的控制输入(如里程计),且 $u_k=u_{m,k}-w_k$,其中 $u_{m,k}$ 表示由里程计实际测得的输入变量,w_k 为相应的噪声,并假设其是均值为 0、方差为 Q_k 的高斯白噪声。

设 LRKF 用如下的线性函数对运动模型进行估计:

$$\begin{aligned}\boldsymbol{x}_k&=\breve{\Phi}_{k-1}x_{k-1}+\breve{G}_{k-1}u_k+b_{k-1}+e_{k-1}=\\&[\breve{\Phi}_{k-1}\quad \breve{G}_{k-1}]\begin{bmatrix}x_{k-1}\\u_k\end{bmatrix}+b_{k-1}+e_{k-1}\end{aligned} \tag{9}$$

在预测过程之前,首先对状态变量略微进行重构,将其增广为与控制输入 u_k 的联合矢量。设已知 $k-1$ 时刻对状态变量估计的均值和方差分别为 $\hat{x}_{k-1}$ 和 $P_{k-1|k-1}$,则增广后得到 x_{k-1}^a 的均值和方差应分别为

$$\hat{x}_{k-1}^a=\begin{bmatrix}\hat{x}_{k-1}\\u_{m,k}\end{bmatrix},\ P_{k-1|k-1}^a=\begin{bmatrix}P_{k-1|k-1}&0\\0&Q_{k-1}\end{bmatrix} \tag{10}$$

LRKF 的预测过程首先产生采样点 $\{\chi_{i,k-1|k-1}^a\}_{i=0}^{r}$,并满足这些采样点的均值为 $\hat{x}_k^a$、方差为 $P_{k-1|k-1}^a$。这里采样规则遵照 CDKF 中的规则[3]进行,共应产生 $(2n+1)$ 个点(即 $r=2n$,n 表示 $\hat{x}_k^a$ 的维数)。

$$\chi_{i,k-1|k-1}^a=\begin{cases}\hat{x}_{k-1}^a & i=0\\ \hat{x}_{k-1}^a+\left(h\sqrt{P_{k-1|k-1}^a}\right)_i & i=1,2,\cdots,n\\ \hat{x}_{k-1}^a-\left(h\sqrt{P_{k-1|k-1}^a}\right)_{i-n} & i=n+1,\cdots,2n\end{cases} \tag{11}$$

其中:h 为尺度参数,对高斯分布有确定的取值 $h=\sqrt{3}$。随后,LRKF 将 $\chi_{i,k-1|k-1}^a$ 代入式(8)所示的非线性运动模型,进而得到新的回归点集 $\{y_{i,k}=\chi_{i,k|k-1}\}_{i=0}^{r}$。

$$y_{i,k}=\chi_{i,k|k-1}=f(\chi_{i,k-1|k-1}^a,u_k) \tag{12}$$

并且新的回归点的均值和方差即作为代入运动模型后,预测得到的状态变量的均值 $\hat{x}_{k|k-1}$ 和方差 $P_{k|k-1}$,具体表达式为

$$\hat{x}_{k|k-1}=\sum_{i=0}^{2n}\omega_i^m\chi_{i,k|k-1} \tag{13}$$

$$\begin{aligned}P_{k|k-1}&=\sum_{i=1}^{n}\omega_i^{c1}[\chi_{i,k|k-1}-\chi_{i+n,k|k-1}][\chi_{i,k|k-1}-\chi_{i+n,k|k-1}]^{\mathrm{T}}+\\&\sum_{i=1}^{n}\omega_i^{c2}[\chi_{i,k|k-1}+\chi_{i+n,k|k-1}-2\chi_{0,k|k-1}][\chi_{i,k|k-1}+\chi_{i+n,k|k-1}-2\chi_{0,k|k-1}]^{\mathrm{T}}\end{aligned} \tag{14}$$

其中:ω_i^m 表示与各采样点相应的均值序列的权值;ω_i^{c1}、ω_i^{c2} 则表示相应方差序列的权值[3]。

进一步可得到 $\breve{\Phi}_{k-1}$ 和 $\breve{G}_{k-1}$ 的表达式类似式(5)的形式[8]:

$$[\breve{\Phi}_{k-1}\quad \breve{G}_{k-1}]=P_{xy}^{\mathrm{T}}(P_{k-1|k-1}^a)^{-1} \tag{15}$$

其中:$\boldsymbol{P}_{xy}$ 表示代入运动模型前及代入运动模型后的状态矢量之间的相交协方差矩阵。

$$\boldsymbol{P}_{xy}=\sqrt{\omega_1^{c1}P_{k-1|k-1}^a}[\chi_{1:n,k|k-1}-\chi_{(n+1):2n,k|k-1}]^{\mathrm{T}} \tag{16}$$

1.3 观测更新过程

设机器人的观测模型为

$$z_k=h(x_k)+\delta_k \tag{17}$$

其中:$h(\cdot)$ 是传感器的观测模型;z_k 表示 k 时刻的观测矢量;δ_k 表示观测噪声,并假设其是均值为 0、方差为 R_k 的高斯白噪声。

设 LRKF 用如下的线性函数对观测模型进行估计:

$$z_k=\breve{H}_kx_k+b'_k+\delta_k+e'_k \tag{18}$$

观测更新过程中仍先产生采样点,并满足这些采样点的均值为 $\hat{x}_{k|k-1}$,方差为 $P_{k|k-1}$,且仍遵循 CDKF 的采样规则,共产生 $(2n_o+1)$ 个采样点(即 $r=2n_o$,n_o 为 $\hat{x}_{k|k-1}$ 的维数),与式(11)中的方法类似。随后将这些采样点代入式(17)所示的非线性观测模型中,得到新的回归点集:$\{Z_{i,k}=h(\chi_{i,k|k-1})\}_{i=0}^{r}$,并进一步得到经过观测预测后的均值 $\hat{z}_{k|k-1}$ 和方差 S_k:

$$\hat{z}_{k|k-1}=\sum_{i=0}^{2n_o}\omega_i^mZ_{i,k} \tag{19}$$

$$\begin{aligned}S_k&=\sum_{i=1}^{n_o}\omega_i^{c1}[Z_{i,k}-Z_{i+n_o,k}][Z_{i,k}-Z_{i+n_o,k}]^{\mathrm{T}}+\\&\sum_{i=1}^{n_o}\omega_i^{c2}[Z_{i,k}+Z_{i+n_o,k}-2Z_{0,k}][Z_{i,k}+Z_{i+n_o,k}-2Z_{0,k}]^{\mathrm{T}}+R_k\end{aligned} \tag{20}$$

用 $\boldsymbol{P}_{xz}$ 表示代入观测模型前及代入观测模型后的状态矢量之间的相交协方差矩阵,即

$$\boldsymbol{P}_{xz}=\sqrt{\omega_1^{c1}P_{k|k-1}}[Z_{1:n_o,k}-Z_{(n_o+1):2n_o,k}]^{\mathrm{T}} \tag{21}$$

则进一步得到回归矩阵 $\breve{H}_k$ 的表达式为

$$\breve{H}_k=P_{xz}^{\mathrm{T}}(P_{k|k-1})^{-1} \tag{22}$$

最后,根据卡尔曼滤波的观测更新过程,可得到 LRKF 形式下,CDKF SLAM 算法计算观测更新后的状态均值 $\hat{x}_k$ 和方差 $P_{k|k}$ 的表达式:

$$\hat{x}_k = \hat{x}_{k|k-1} + W_k(z_k - \hat{z}_{k|k-1}) \tag{23}$$

$$P_{k|k} = P_{k|k-1} - W_k S_k W_k^{\mathrm{T}} \tag{24}$$

其中：W_k 为卡尔曼增益，可用式(25)进行计算。

$$W_k = P_{k|k-1}\breve{H}_k^{\mathrm{T}} S_k^{-1} \tag{25}$$

CDKF SLAM 算法的 LRKF 形式与一般 CDKF SLAM 算法相比，在预测和观测更新时分别采用了线性回归的原理，进而完成递推过程。其在观测更新前需要进行数据关联来判断观测所对应的特征与地图中已有特征之间的对应关系。最大似然法[9]是一种常用的数据关联方法，该方法首先计算地图中每个特征的预测观测值，然后计算出实际观测值与预测观测值之间的新息和新息方差，进而得到相应的规范距离，并选择最小的规范距离作为与观测对应的特征。在观测更新后还需要进行地图增广，将初次观测到的特征加入到地图中。具体的方法则与一般的 CDKF SLAM 算法相同，在本文中不作详细介绍。

2　计算复杂度降低的 CDKF SLAM 算法

CDKF SLAM 算法的计算复杂度为 $O(n^3)$。由于在 SLAM 问题中机器人在探索周围环境时会不断观测到新的特征，状态维数 n 也会随之增加，使得 CDKF SLAM 算法在应用到大规模环境中时很难满足实时性的要求。分析可知，该计算复杂度是在确定采样点时由计算状态方差矩阵的均方根矩阵的复杂度所决定的。为此，本文以 CDKF SLAM 算法的 LRKF 形式为基础，提出了一种改进的采样点确定方法，使该步骤的计算复杂度降为常数，从而使整个算法的计算复杂度降为 $O(n^2)$。

2.1　改进算法的基本思想

如前文所述，在 SLAM 问题中，其估计的状态变量 x_k 是由 k 时刻机器人的位姿 $x_{r,k}$（包括机器人的位置和方向角）和地图中的特征位置 x_{f_i}（$i=1,\cdots,n_f$，n_f 表示特征的数量）所构成，即

$$x_k^{\mathrm{T}} = [x_{r,k}^{\mathrm{T}} \quad x_{f_1}^{\mathrm{T}} \quad \cdots \quad x_{f_{n_f}}^{\mathrm{T}}] \tag{26}$$

由于在预测和观测更新过程中，确定采样点时，均需要计算状态方差矩阵的均方根阵，而该计算的复杂度 $O(n^3)$ 恰好决定了 CDKF SLAM 算法的计算复杂度。同时，通过观察也发现，无论在预测还是在观测更新过程，实际上只有状态变量的一小部分子集分别被代入运动模型和观测模型中进行了非线性变换。具体地，在预测过程中，只有表示机器人位姿的状态矢量被代入了运动模型；在观测更新过程中，则只有表示机器人位姿及表示其中一个特征位置的状态矢量被代入了观测模型。若只依据状态变量中参与非线性变化的这一小部分子集来确定采样点，则计算状态方差矩阵的均方根矩阵时的计算复杂度即会变为与状态变量维数 n 无关的常数，这就是改进算法的基本思想。

为了利用 SLAM 问题上述的特点，进一步得到改进的 SLAM 算法，需要用到如下所示的定理[8]：

定理 1　设状态矢量 $\boldsymbol{x}$ 可表示为 $\boldsymbol{x}^{\mathrm{T}} = [x_1^{\mathrm{T}} \quad x_2^{\mathrm{T}}]$，且满足 $g(x) = g(x_1)$（即状态矢量 $\boldsymbol{x}$ 中只有 x_1 作为 $g(\cdot)$ 的输入矢量）；进一步地，设线性回归矩阵 $\boldsymbol{A}$ 可表示为 $\boldsymbol{A} = [A_1 \quad A_2]$，且满足

$$y = \boldsymbol{A}x + \boldsymbol{b} + \boldsymbol{e} = A_1 x_1 + A_2 x_2 + \boldsymbol{b} + \boldsymbol{e}$$

则式(2)中所示的目标函数的最优解为

$$A_1 = P_{x_1 y}^{\mathrm{T}} P_{x_1 x_1}^{-1}, A_2 = 0, \boldsymbol{b} = \hat{y} - A_1 \hat{x}_1$$

将上述的基本思想和定理 1 分别用于 CDKF SLAM 算法的预测和观测更新过程中，即可降低这两个过程在确定采样点时的计算复杂度。

2.2　预测过程计算复杂度的降低

由于对任一 k 时刻的预测过程，代入运动模型的输入变量只有表示机器人位姿的 $x_{r,k-1}$ 和控制输入矢量 $\boldsymbol{u}_k$，且预测过程后整个状态矢量 x_{k-1} 中只有对 $x_{r,k-1}$ 的估计发生了变化，因此在改进的预测过程中，用 $x_{r,k-1}$ 代替原预测过程中的整个状态矢量 x_{k-1}，与 u_k 组成新的状态矢量，并据此产生采样回归点。此时，经重构后得到的新的状态矢量的均值 $\bar{x}_{k-1}^a$ 和方差 $\bar{P}_{k-1|k-1}^a$ 分别为

$$\bar{x}_{k-1}^a = \begin{bmatrix} \hat{x}_{r,k-1} \\ u_{m,k} \end{bmatrix}, \bar{P}_{k-1|k-1}^a = \begin{bmatrix} P_{r,k-1|k-1} & 0 \\ 0 & Q_{k-1} \end{bmatrix} \tag{27}$$

其中：$P_{r,k-1|k-1}$ 是与机器人位姿相对应的方差矩阵，可通过将 $P_{k-1|k-1}$ 分块为如下形式得到

$$P_{k-1|k-1} = \begin{bmatrix} P_{r,k-1|k-1} & P_{rf,k-1|k-1} \\ P_{fr,k-1|k-1} & P_{f,k-1|k-1} \end{bmatrix} \tag{28}$$

再根据 CDKF 的采样规则式(11)，以 $\bar{x}_{k-1}^a$ 为均值、$\bar{P}_{k-1|k-1}^a$ 为方差产生采样点集。注意到，$\bar{x}_{k-1}^a$ 为 5 维的（假设里程计的测量值 $u_{m,k}$ 为 2 维），而 $\bar{P}_{k-1|k-1}^a$ 始终为 5×5 维的方阵，因此计算 $\sqrt{\bar{P}_{k-1|k-1}^a}$ 的复杂度始终为常数，与状态变量的维数无关。产生的采样点集为 $\{\bar{\chi}_{i,k-1|k-1}^a\}_{i=0}^r$ $(r=10)$。

将采样点代入运动模型中得到新的采样点集 $\bar{y}_{i,k} = \bar{\chi}_{i,k|k-1}$ $(i=0,\cdots,10)$，再根据式(13)和(14)分别计算新的采样点的均值和方差，作为预测的机器人位姿的均值 $\hat{x}_{r,k|k-1}$ 和方差 $P_{r,k|k-1}$。进一步地，根据定理 1 得到该改进的预测过程中的回归矩阵为

$$[\breve{\Phi}_{r,k-1} \quad \breve{G}_{r,k-1}] = \bar{P}_{xy}^{\mathrm{T}}(\bar{P}_{k-1|k-1}^a)^{-1} \tag{29}$$

又由相关协方差的定义可知，预测的机器人位姿与特征位置矢量之间的相关协方差为

$$P_{rf,k|k-1} = \breve{\Phi}_{r,k-1} P_{rf,k-1|k-1} \tag{30}$$

且在以上过程中，特征的位置估计的均值和方差并未发生变化，即 $\hat{x}_{f,k|k-1} = \hat{x}_{f,k-1}$、$P_{f,k|k-1} = P_{f,k-1|k-1}$，故最后得到预测的整个状态变量的均值和方差为

$$\hat{x}_{k|k-1} = \begin{bmatrix} \hat{x}_{r,k|k-1} \\ \hat{x}_{f,k-1} \end{bmatrix}, \begin{bmatrix} P_{r,k|k-1} & \breve{\Phi}_{r,k-1} P_{rf,k-1|k-1} \\ P_{fr,k-1}\breve{\Phi}_{r,k-1}^{\mathrm{T}} & P_{f,k-1|k-1} \end{bmatrix} \tag{31}$$

以上的预测过程，其计算复杂度与整个状态变量的维数无关，仅为常数。

2.3　观测更新过程计算复杂度的降低

因为对任意时刻 k 的观测更新过程，代入观测模型的输入变量只有机器人的位姿和被观测到的特征位置矢量，因此在改进的观测更新过程中，只依据状态矢量中这两者构成的子集产生采样点，从而降低计算复杂度。具体地，设在 k 时刻机器人观测到了第 j 个特征，则产生的采样点 $\{\bar{\chi}_{i,k|k-1}^*\}_{i=0}^r$ $(r=10)$，其均值 $\bar{x}_{k|k-1}$ 和方差 $\bar{P}_{k|k-1}$ 应分别满足

$$\bar{x}_{k|k-1} = \begin{bmatrix} \hat{x}_{r,k|k-1} \\ \hat{x}_{f_j,k|k-1} \end{bmatrix}, \bar{P}_{k|k-1} = \begin{bmatrix} P_{r,k|k-1} & P_{rf_j,k|k-1} \\ P_{f_j r,k|k-1} & P_{f_j,k|k-1} \end{bmatrix} \tag{32}$$

显然，$\bar{x}_{k|k-1}$ 为 5 维，而 $\bar{P}_{k|k-1}$ 为 5×5 维的方阵，计算 $\sqrt{P_{k|k-1}}$ 的复杂度仍为常数，与整个状态矢量的维数无关。

按照与 1.3 节中的观测更新过程类似的方法，计算观测预

测后的均值 $\bar{z}_{k|k-1}$ 和方差 $\bar{S}_k$,并进一步得到改进的观测更新过程中的回归矩阵:

$$[\breve{H}_{r,k} \quad \breve{H}_{f_j,k}] = \bar{P}_{xz}^{\mathrm{T}}(\bar{P}_{k|k-1})^{-1} \tag{33}$$

其中:回归矩阵 $\breve{H}_{r,k}$ 与机器人的位姿矢量相对应,回归矩阵 $\breve{H}_{f_j,k}$ 与第 j 个特征位置矢量相对应。根据定理 1,与未被观测到的特征的位置矢量所对应的回归矩阵应为 0,由此可得到观测更新过程完整的回归矩阵 $\breve{H}_k$ 为

$$\breve{H}_k = [\breve{H}_{r,k} \quad 0 \quad \cdots \quad 0 \quad \breve{H}_{f_j,k} \quad 0 \quad \cdots \quad 0] \tag{34}$$

将结果代入式(23)~(25),可得到观测更新后状态矢量估计的均值 $\hat{x}_k$ 和方差 $P_{k|k}$。

在以上的观测更新过程中,由于确定采样点时的计算复杂度降为常数,故该过程的计算复杂度取决于整个状态方差矩阵更新时的计算复杂度,为 $O(n^2)$。

2.4　改进算法的流程及计算复杂度

为了便于应用,本节中给出计算复杂度降低的 CDKF SLAM 算法的流程。假设运动模型和观测模型分别如式(8)和(17)所示,且已知相应的输入矢量,根据 $k-1$ 时刻状态矢量的均值 $\hat{x}_{k-1}$ 和方差 $P_{k-1|k-1}$,算法经预测、数据关联、观测更新和地图增广四个过程,得到 k 时刻状态矢量的均值 $\hat{x}_k$ 和方差 $P_{k|k}$,从而完成一次递推过程。具体如下:

a)预测过程。

(a)在对状态矢量的估计中,取与机器人位姿对应的估计子集 $\hat{x}_{r,k-1}$ 和 $P_{r,k-1|k-1}$。

(b)与控制矢量 u_k 重构得到新的状态矢量,其均值 $\bar{x}_{k-1}^a$ 和方差 $\bar{P}_{k-1|k-1}^a$ 如式(27)所示。

(c)以 $\bar{x}_{k-1}^a$ 为均值、$\bar{P}_{k-1|k-1}^a$ 为方差,根据式(11)产生采样点集 $\{\bar{\chi}_{i,k-1|k-1}^a\}_{i=0}^r(r=10)$。

(d)将采样点集代入运动模型,得到新的采样点集 $\{\bar{y}_{i,k} = \bar{\chi}_{i,k|k-1}\}_{i=0}^r(r=10)$。

(e)根据式(13)和(14)计算新的采样点集的均值和方差,作为预测后机器人位姿的均值 $\hat{x}_{r,k|k-1}$ 和方差 $P_{r,k|k-1}$。

(f)根据式(16)计算 $\bar{P}_{xy}$,并代入式(29)得到回归矩阵 $\breve{\Phi}_{r,k-1}$。

(g)根据式(30)计算预测的机器人位姿与特征位置矢量之间的相关协方差 $P_{rf,k|k-1}$。

(h)根据式(31)得到预测的状态矢量的均值 $\hat{x}_{k|k-1}$ 和方差 $P_{k|k-1}$。

b)数据关联过程。可用最大似然数据关联方法[9]。

c)观测更新过程。

(a)用机器人的位姿和被观测到的特征位置矢量(设观测到了第 j 个特征)重构一新的状态矢量,其均值 $\bar{x}_{k|k-1}$ 和方差 $\bar{P}_{k|k-1}$ 如式(32)所示。

(b)以 $\bar{x}_{k|k-1}$ 为均值、$\bar{P}_{k|k-1}$ 为方差,根据式(11)产生采样点集 $\{\bar{\chi}_{i,k|k-1}^*\}_{i=0}^r(r=10)$。

(c)将采样点集代入观测模型,得到新的采样点集 $\{\bar{Z}_{i,k} = h(x_{i,k|k-1}^*)_{i=0}^r\}(r=10)$。

(d)根据式(19)和(20)计算观测预测后的均值 $\bar{z}_{k|k-1}$ 和方差 $\bar{S}_k$。

(e)根据式(21)计算 $\bar{P}_{xz}$,并代入式(33)得到回归矩阵 $[\breve{H}_{r,k} \quad \breve{H}_{f_j,k}]$。

(f)根据式(34)得到观测更新过程完整的回归矩阵 $\breve{H}_k$。

(g)根据式(25)计算卡尔曼增益 W_k。

(h)根据式(23)和(24)计算观测更新后状态矢量的均值 $\hat{x}_k$ 和方差 $P_{k|k}$。

d)地图增广过程。

对初次观测到的新特征,用对其位置的估计扩展 $\hat{x}_k$,并计算相应的方差矩阵 $P_{k|k}$。

对于 CDKF SLAM 算法,其计算复杂度是由预测和观测更新这两个过程的计算复杂度决定的,在上面给出的改进算法中,这两个过程的计算复杂度分别为常数和 $O(n^2)$,因此算法的计算复杂度为 $O(n^2)$。

3　仿真及实验结果分析

为了进一步验证改进算法的性能,本文分别选取了 EKF SLAM、CDKF SLAM 算法以及计算复杂度降低的 UKF SLAM 算法[8],并基于不同规模环境中的仿真实验和停车场数据的实验,与本文提出的计算复杂度降低的 CDKF SLAM 算法进行了比较。实验中的计算机采用 Intel 酷睿双核 CPU,主频 2.8 GHz,内存 2 GB。程序运行环境为 MATLAB 6.5。

3.1　仿真实验结果和分析

在文献[10]中,Bailey 公布了用于测试 SLAM 算法的仿真软件,可借助其比较不同 SLAM 算法的性能。其中 EKF SLAM 算法采用软件包自带的程序,在此基础上笔者编写了其余算法的程序。由于 SLAM 算法的计算复杂度与环境中的特征数量相关,本文分别设置了特征数量不同的两种仿真环境,如图 1 所示。其中,图 1(a)为较小规模的仿真环境,环境中设置了机器人运行路径的 17 个关键点和 35 个静止的点特征;(b)为较大规模的仿真环境,环境中机器人运行路径保持不变,但将静止的点特征的数量增加为 70 个。

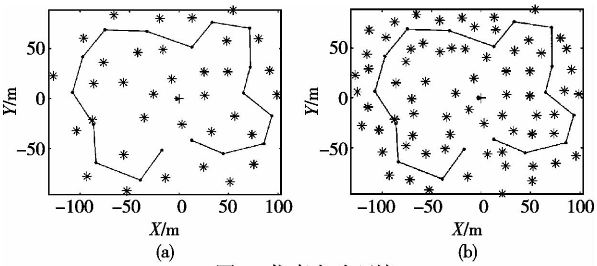

图1　仿真实验环境

实验中采用移动机器人速度运动模型和测距传感器的距离—转向角模型[5]。具体的参数设置包括机器人的轴距设为 4 m,运动速度为 3 m/s,测距传感器的最大测距范围为 30 m,且具有 180°的前向视角,里程计和测距传感器的采样间隔分别为 25 ms 和 200 ms;运动模型噪声设为 0.3 m/s 和 3°,观测模型噪声设为 0.1 m 和 1°。实验中设数据关联的结果已知,在两个环境中,机器人均由原点开始沿关键点逆时针运动两圈,并分别采用上面提到的四种不同的 SLAM 算法进行机器人位姿估计和环境中特征位置的估计,且每种算法均进行 50 次 Monte Carlo 仿真实验。

a)分析小规模环境中的实验结果。图 2 给出了这四种算法所估计的机器人位置的均方根(root mean square,RMS)误差(图 2(a))和方向角 RMS 误差(图 2(b))随时间变化的情况。为了表示方便,将复杂度降低的 UKF SLAM 算法和复杂度降低的 CDKF SLAM 算法分别记做 CR(complexity reduced)-UKF

SLAM 和 CR-CDKF SLAM。由图 2 可知,除 EKF SLAM 算法外,其他三种算法的估计误差相当,且均比 EKF SLAM 算法的估计误差有明显减少。表 1 中进一步列出了经 50 次 Monte Carlo 仿真实验得到的这四种算法估计的误差值。由表 1 可知,在四种算法中,本文提出的 CR-CDKF SLAM 算法估计的机器人位置和方向角的 RMS 误差最小,且其估计的特征位置的 RMS 误差也最小。

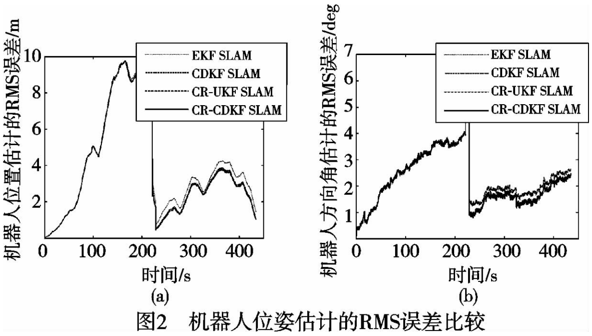

图2　机器人位姿估计的RMS误差比较

表 1　较小规模环境中 EKF、CDKF、CR-UKF 和 CR-CDKF SLAM 算法的性能比较

SLAM 算法	机器人位置估计误差 /m	机器人方向角估计误差 /deg	特征位置估计误差 /m	总运行时间平均值 /s	单步运行时间平均值 /ms	计算复杂度
EKF	4.007 0	2.312 5	3.034 0	10.87	0.63	$O(n^2)$
CDKF	3.819 3	2.187 4	2.685 3	84.24	4.85	$O(n^3)$
CR-UKF	3.810 2	2.183 7	2.672 3	15.75	0.91	$O(n^2)$
CR-CDKF	3.778 4	2.160 7	2.634 0	15.68	0.90	$O(n^2)$

表 1 中还列出了四种算法的计算复杂度,并比较了它们的运行时间。仿真实验中机器人共运行了 17 383 步,因此单步运行时间是通过总运行时间除以该运行步数得到的。由于传感器的最小采样时间间隔为 25 ms,而四种算法单步运行时间的平均值均远小于该值,说明四种算法在图 1(a)中所示的环境中均可以满足实时性的要求。而 EKF、CR-UKF 和 CR-CDKF 三种 SLAM 算法的计算复杂度为 $O(n^2)$,CDKF SLAM 的计算复杂度则为 $O(n^3)$,因此它们的运行时间也远小于 CDKF SLAM 算法。

b)对较大规模环境(图 1(b))中的实验结果进行分析。实验中运动模型噪声增大为 0.6 m/s 和 6°,其他参数设置仍保持不变。由于 CDKF SLAM 算法在该环境中的运行时间过长,表 2 中只列出了其他三种算法经 50 次 Monte Carlo 仿真实验得到的相关性能的统计比较。

表 2　较大规模环境中 EKF、CR-UKF 和 CR-CDKF SLAM 算法性能比较

SLAM 算法	机器人位置估计误差 /m	机器人方向角估计误差 /deg	特征位置估计误差/m	总运行时间平均值 /s	单步运行时间平均值 /ms
EKF	2.691 8	1.549 5	2.0683	11.93	0.69
CR-UKF	2.609 3	1.501 4	1.8543	17.24	0.99
CR-CDKF	2.595 4	1.487 8	1.8258	16.84	0.97

与表 1 中对应的估计误差相比,虽然在参数设置时增大了系统误差,但特征数量的增加反而使估计误差有所减小。表 2 中 CR-UKF 和 CR-CDKF SLAM 算法的估计误差均比 EKF SLAM 算法的误差有较明显减小,且 CR-CDKF SLAM 算法估计的准确度仍是最高的。比较运行时间可知,虽然特征数量变为原来的 2 倍,但三种算法由于计算复杂度仅为 $O(n^2)$,故运行时间与在较小规模环境中的运行时间相比并未明显增加,且每种算法的单次递推时间的平均值仍远小于 25 ms,能满足实时性的需要。相比之下,CDKF SLAM 算法的单次递推大致需要 30 ms,已无法满足实时性的需要。

通过以上仿真实验可知,本文提出的 CR-CDKF SLAM 算法,与 EKF SLAM 算法相比,能够明显降低 SLAM 的估计误差,提高估计的准确度;与 CDKF SLAM 算法相比,计算复杂度由 $O(n^3)$ 变为 $O(n^2)$,从而使算法的运行时间有明显缩短,更易满足 SLAM 实时性的需要。虽然与 CR-UKF SLAM 算法相比,其估计的误差略有减少,且运行时间也大致相同,但考虑到 CR-UKF SLAM 算法在确定采样点时需要根据被估计变量的性质和系统状态方程的非线性程度同时确定三个参数[4];而 CR-CDKF SLAM算法则只需要确定一个参数 h,且对于高斯随机变量,该参数有确定的最优值 3。由此可以认为,CR-CDKF SLAM 算法在所比较的四种算法中性能最优。

3.2　停车场数据集的实验结果及分析

下面采用检验 SLAM 算法时广泛应用的标准数据集之一——停车场数据集对 EKF、CDKF 和 CR-CDKF SLAM 算法进行进一步比较。该数据集是在一个面积约为 45 m×30 m 的停车场中,由研究人员预先在其中放置一些人工路标作为地图特征,然后驾驶智能车辆(机器人)低速行驶约 2 min 而采集得到的。采集的数据包括车辆左后轮上安装的里程计测得的线速度和车辆转向角、由激光测距仪测得的人工路标的距离和方向,以及由 GPS 测得的车辆运动过程中的经纬度信息。它们采集的时间间隔分别为 25 ms、214 ms 和 200 ms。其中,激光测距仪的有效距离为 81 m,且具有 180°的前向视角。GPS 的信息并不用于 SLAM 的整个过程,而用来验证 SLAM 估计结果的准确度。在算法的程序中,采用了最大似然的数据关联方法。

图 3 是运动模型噪声和观测模型噪声分别设为 0.7 m/s、5.5°和 0.1 m、1°时,分别采用三种算法得到的实验结果比较。

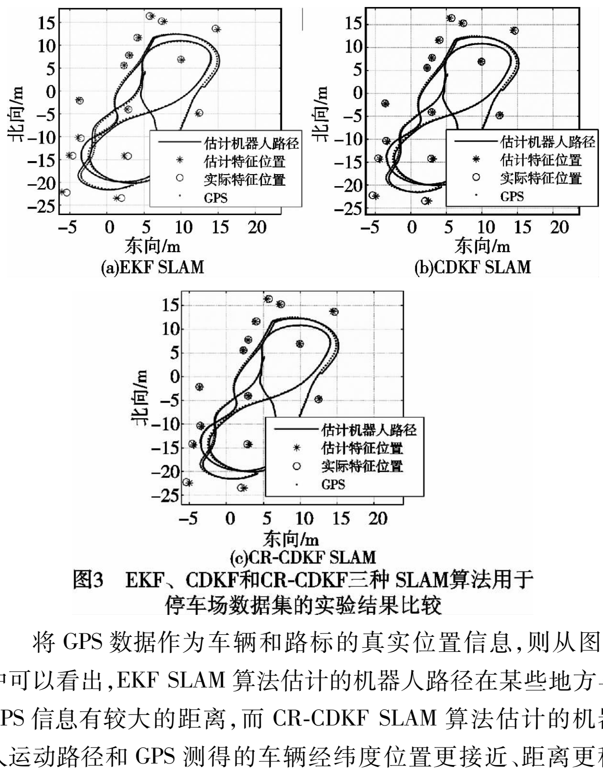

图3　EKF、CDKF和CR-CDKF三种 SLAM算法用于停车场数据集的实验结果比较

将 GPS 数据作为车辆和路标的真实位置信息,则从图 3 中可以看出,EKF SLAM 算法估计的机器人路径在某些地方与 GPS 信息有较大的距离,而 CR-CDKF SLAM 算法估计的机器人运动路径和 GPS 测得的车辆经纬度位置更接近、距离更稳定,尤其是其估计的路标位置与实际路标位置(下转第 3298 页)

(上接第 3284 页)之间的误差更小,表明其准确度高于 EKF SLAM 算法。从运行时间来看,EKF、CDKF 和 CR-CDKF 三种 SLAM 算法总运行时间的平均值分别为 20.67 s、28.09 s 和 20.75 s,相对于数据的总采集时间 112 s,三种算法均能够满足实时性的需要。而与 CDKF SLAM 算法相比,CR-CDKF SLAM 算法的运行时间明显缩短。

4 结束语

SLAM 算法的计算复杂度是衡量算法性能的重要指标之一。为此,本文在 CDKF SLAM 算法的基础上提出了一种降低其计算复杂度的算法。实验表明,本文所提出的计算复杂度降低的 CDKF SLAM 算法在保持较高估计准确度的同时,明显缩短了 CDKF SLAM 算法的运行时间,可以满足在较大规模环境中实时运行的需求。下一步将以此改进的 CDKF SLAM 算法为基础,研究改进其准确度、一致性等其他性能指标的方法。

参考文献:

[1] PAZ L M, ENSFELT P J, TARDOS J D, *et al*. EKF SLAM updates in $O(n)$ with divide and conquer SLAM[C]//Proc of IEEE International Conference on Robotics and Automation. 2007:1657-1663.

[2] NORGAARD M, POULSEN N, RAVN O. New developments in state estimation for nonlinear systems[J]. Automatica, 2000, 36(11):1627-1638.

[3] ZHU Ji-hua, ZHENG Nan-ning, YUAN Ze-jian. A SLAM algorithm based on the central difference Kalman filter[C]//Proc of IEEE Intelligent Vehicle Symposium. 2009:123-128.

[4] 祝继华,郑南宁,袁泽剑,等. 基于中心差分粒子滤波的 SLAM 算法[J]. 自动化学报, 2010,36(2):249-257.

[5] CHEN Chen, CHENG Yin-hang. Sqaure root sigma point Kalman filter based SLAM[C]//Proc of the 13th IEEE Joint International Computer Science and Information Technology Conference. 2011:245-249.

[6] Van Der MERWE R. Sigma-point Kalman filters for probabilistic inference in dynamic state-space models[D]. Hillsboro:Oregon Graduate Institute of Science and Technology. 2004.

[7] LEFEBVRE T, BRUYNINCKX H, De SCHUTTER J. Comment on "a new method for the nonlinear transformation of means and covariances in filters and estimators"[J]. IEEE Trans on Automatic Control, 2002, 47(8):1406-1408.

[8] HUANG Shou-dong, MOUIKIS A I, ROUNMELIOTIS S I. On the complexity and consistency of UKF-based SLAM[C]//Proc of IEEE International Conference on Robotics and Automation. 2009:4401-4408.

[9] 曾文静,张铁栋,姜大鹏. SLAM 数据关联方法的比较分析[J]. 系统工程与电子技术,2010,32(4):860-864.

[10] BAILEY T. SLAM simulations [DB/OL]. (2008-06-10) [2010-10-24]. http://www-personal.acfr.usyd.edu.au/tbailey/.