

Automatic Vehicle Location and State Estimation Based on Multi-Sensor Data Fusion

PENG Wenzheng, AO Yinhui*, HUANG Xiaotao, WANG Pengfei

(School of Electromechanical Engineering, Guangdong University of Technology, Guangzhou Guangdong 510006, China)

Abstract: In order to deal with the high-precision positioning and speed problem for Formula SAE(FSAE) vehicle, a new multi-sensor data fusion estimation method based on filter is proposed and applied on real vehicle platform. The method is designed based on Iterative Extended Kalman Filter(iEKF), and combine different sensors, including the IMU, wheel speed and steering encoders, the GPS, Camera and Lidar. First, the IMU is used to predict vehicle prior state; Then, the other sensors are respectively subjected to different information extraction in parallel fusion architecture, and updated the prior state. Because of the parallel fusion architecture, different sensors may update the vehicle state vector independently. Experimental results show that the proposed method has high precision dealing with the positioning in map and speed estimation problems, and has sufficient redundancy and real-time performance.

Key words: automatic vehicle; self-localization; state estimation; iEKF; online estimation; multi-sensor data fusion
EEACC: 7230 **doi:** 10.3969/j.issn.1004-1699.2020.08.010

多传感器信息融合的自动驾驶车辆定位与速度估计

彭文正, 敖银辉*, 黄晓涛, 王鹏飞

(广东工业大学机电工程学院, 广东 广州 510006)

摘要: 针对大学生无人驾驶方程式(FSAC)于地图内高精定位以及速度观测的问题,设计基于多传感器信息融合的状态估计算法,并应用于自研实车平台。算法基于迭代扩展卡尔曼滤波(iEKF)进行设计,融合多类传感器,包括惯性测量单元(IMU)、转角及轮速编码器、全球卫星定位(GPS)、相机与激光雷达(Lidar)。首先,利用IMU预测车辆先验状态;然后,建立并联融合架构,对各类传感器数据进行不同的信息处理,用于更新先验状态;由于并联融合的架构,不同传感器可独立地维护车辆的状态观测。实验结果表明,所提出的算法对地图内定位、速度观测有较好的精度,且具有足够的冗余性和实时性。

关键词: 自动驾驶; 自主定位; 状态估计; 迭代扩展卡尔曼滤波; 在线估计; 多传感器信息融合

中图分类号: TP393

文献标识码: A

文章编号: 1004-1699(2020)08-1140-09

随着机器人、无人机、自动驾驶车辆等领域的技术与实际应用的发展,各类算法正面临着复杂的性能与场景需求。对于无人驾驶车辆,一般需要实时精准地测量其位姿与速度,以用于车辆的精准观测、行为决策与控制输入。因此,状态估计(State Estimation)问题^[1-2]成为了其中的关键问题。而由于一般车辆上存在多类传感器,基于多传感器融合思路的状态估计算法在自动驾驶车辆应用上具有较好的适用性与应用前景。

一般多传感器融合的算法可以分为两类:一类从频域的角度来分辨噪声,通过截取频率范围来处理信号,再进行融合,如互补滤波器;另一类是从时

域的角度,对状态进行预测与校正,如扩展卡尔曼滤波(EKF, Extended Kalman Filter)。

为了进一步提高基于多传感器融合的状态估计算法的精度,研究者们提出了多种融合方案与滤波算法的变形。张勇刚等^[3]结合互补滤波器,利用分级滤波的思路,对实时的惯性测量单元(IMU)、全球卫星定位(GPS)、地磁传感器进行融合滤波,求解姿态、速度与位置,能达到与 EKF 相近的精度。Jun N I 等^[4]结合互补滤波器,耦合激光雷达与组合导航定位系统(GPS-INS)设计其里程计,并应用于大学生方程式平台。郭安等^[5]结合卡尔曼滤波(KF, Kalman Filter)算法,融合 GPS、IMU、气压计、空速计,利用三级串联思路分别求

解姿态、位置与速度,并对无人机状态估计进行仿真,该算法由于是串联的思路,会造成不同频率传感器中高频传感器数据的丢失,且难以应对传感器失效。Ko N 等^[6]结合不变扩展卡尔曼滤波(Invariant EKF)设计多传感器融合算法,融合 GPS(测量定位与速度)、姿态航向测量系统 MEMS-AHRS、气压计(测量海拔)实现无人机的海拔、位姿、速度的估计,并通过实验验证相比 EKF 精度有所提高。Goh S T 等^[7]提出了加权测量融合卡尔曼滤波算法,对本地定位系统中多个接发器的数据进行加权融合用于状态预测,并应用于无人机仿真中,证明较 KF 算法效果更优。Gao Z 等^[8]利用 Rauch-Tung-Striebel 对数据进行平滑,基于 EKF 融合北斗 RTK 差分定位、里程计、IMU、航向角等传感器信号,完成对铁路车的状态估计。Cui B 等^[9]提出了改进的迭代数值积分卡尔曼滤波应用于组合导航定位设备(GPS+IMU),较 EKF 算法精度有所提升。黄伟杰等^[10]结合不变卡尔曼滤波,将多层次最近点配准(ICP)求解的相机位姿变换与 IMU 进行融合,求解位姿,当里程计定位失效时,可以利用 IMU 建立模型辅助里程计重新获取定位。另外,在非线性未知系统的应用中,刘济等^[11]提出的无迹卡尔曼滤波(UKF, Unscented Kalman Filter)与神经网络结合的算法,设计包含一个隐含层的动态 BP 神经网络用于对非线性未知模型进行模拟辨识,将网络权值与状态作为输入共同进行更新。Hertzberg C 等^[12]提出了一种基于流行的数学封装,可用于最小乘方优化或 KF 中三维状态的运算,并结合无迹卡尔曼滤波,对 GPS 与 IMU 信号进行融合实现组合导航定位。从现有的研究可以发现几个主要的问题:为了达到较高的定位精度,一些算法需要配置组合导航系统提高的高精定位,这带来极高的成本;对于多传感器融合,很多算法采用的思路逻辑上是串联的,使其难以解决不同传感器数据的时间对准和失效的问题。

此外,Bell B M 等^[13]证明迭代卡尔曼滤波(iKF, Iterative Extended Kalman Filter)是高斯-牛顿法用于近似最大似然估计的一种应用,且相比 EKF 更为精确。Havlík J 等^[14]详细对比了 EKF、iEKF 与其变形,证明此类算法相较 EKF 具有更优的状态估计精度与更低的不可信指数。这为这一类的算法设计提供了更丰富的理论基础。

本文围绕低成本的多传感器在车辆定位与速度观测的应用,基于滤波提出了一种多传感器融合状态估计算法;算法定义观测目标包含车辆的水平位姿与速度,并结合各传感器的数据采集精度与可能出现的失效加以考虑。算法主要特点如下:①利用车辆较为普及的传感器进行融合,在不增加过多成

本的前提下,提出一种新的基于多传感器信息融合的框架,结合多种传感器的信息进行估计;②基于 iEKF 理论设计算法,充分考虑不同传感器数据的不确定性(方差),提高算法的精度,且对各雅各比矩阵的求法做了说明,有效简化了计算过程;③各传感器数据并联处理;各传感器虽采集频率不同,结合本文并联的思路与时间截补偿,即可独立维护车辆的状态观测,实现了算法的冗余性设计,且应对传感器失效的风险。最后通过实验对算法进行了多项性能的验证分析,证明算法切实有效。

1 状态估计中的滤波方法

1.1 基本定义

定义先验状态为 $\bar{\mathbf{x}}_k$,后验状态为 $\hat{\mathbf{x}}_k$;本文中讨论的状态向量与数据皆为概率模型,并假设服从高斯分布,即 $\mathbf{x}_k \sim N(\mathbf{0}, \mathbf{Q}')$,括号内分别为均值向量与协方差矩阵。

运动噪声与观测噪声分别为:

$$\mathbf{w} \sim N(\mathbf{0}, \mathbf{Q}'), \mathbf{v} \sim N(\mathbf{0}, \mathbf{R}')$$

\mathbf{Q}' 为运动协方差矩阵, \mathbf{R}' 为运动协方差矩阵,系统运动模型递推的非线性映射与观测模型的非线性映射分别为:

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_k) + \mathbf{w}_k \quad (1)$$

$$\mathbf{Z}_k = g(\mathbf{x}_k) + \mathbf{v}_k \quad (2)$$

1.2 扩展卡尔曼滤波

在处理符合高斯分布的状态量或者噪声的状态估计问题中,EKF 是数据融合的主流算法,广泛应用于非线性系统中。

EKF 算法主要分为预测与更新两个环节。其中预测过程结合运动模型与当前实际控制量,对上一帧状态 $\hat{\mathbf{x}}_{k-1} \sim N(\hat{\boldsymbol{\mu}}_{k-1}, \hat{\Sigma}_{k-1})$ 进行递推,计算当前的状态先验 $\check{\mathbf{x}} \sim N(\check{\boldsymbol{\mu}}_k, \check{\Sigma}_k)$ 。而更新环节则是通过计算观测模型求得的期望观测与传感器实际观测作误差,进而递归更新状态先验,得到状态后验 $\hat{\mathbf{x}}_k \sim N(\hat{\boldsymbol{\mu}}_k, \hat{\Sigma}_k)$ 。预测方程如下:

$$\check{\Sigma}_k = F_{k-1} \hat{\Sigma}_{k-1} F_{k-1}^T + Q'_k \quad (3a)$$

$$\check{\boldsymbol{\mu}}_k = f(\hat{\boldsymbol{\mu}}_{k-1}, \mathbf{u}_k) \quad (3b)$$

$$F_{k-1} = \frac{\partial \check{\boldsymbol{\mu}}_k}{\partial \hat{\boldsymbol{\mu}}_{k-1}} = \frac{\partial f(\hat{\boldsymbol{\mu}}_{k-1}, \mathbf{u}_k)}{\partial \hat{\boldsymbol{\mu}}_{k-1}} \quad (3c)$$

更新环节方程如下

$$K_k = \check{\Sigma}_k G_k^T (G_k \check{\Sigma}_k G_k^T + R'_k)^{-1} \quad (4a)$$

$$\hat{\Sigma}_k = (I - K_k G_k) \check{\Sigma}_k \quad (4b)$$

$$\hat{\boldsymbol{\mu}}_k = \check{\boldsymbol{\mu}}_k + K_k E_k = \check{\boldsymbol{\mu}}_k + K_k (\bar{\mathbf{Z}}_k - g(\check{\boldsymbol{\mu}}_k)) \quad (4c)$$

$$\mathbf{G}_k = \frac{\partial \mathbf{g}}{\partial \dot{\boldsymbol{\mu}}_k} \quad (4d)$$

式中, $\bar{\mathbf{Z}}_k$ 为实际观测, $\check{\mathbf{Z}}_k = \mathbf{g}(\dot{\boldsymbol{\mu}}_k)$ 为期望观测, \mathbf{K}_k 为卡尔曼增益, \mathbf{R}'_k 为修正用的数据的协方差矩阵。

1.3 迭代扩展卡尔曼滤波(iEKF)

iEKF 算法是 EKF 的改进算法, 其理论上通过迭代逐步逼近全后验概率的极大值。其与 EKF 的不同体现在更新环节, 对状态后验均值进行迭代修正。迭代开始时, 令迭代状态向量 $\mathbf{x}_{op,k} \leftarrow \check{\mathbf{x}}_k$, 然后在每次迭代中, 将工作点设置为上一次迭代的后验均值:

$$\mathbf{K}_{op,k} = \check{\mathbf{Z}}_k \mathbf{G}_{op,k}^T (\mathbf{G}_{op,k} \check{\mathbf{Z}}_k \mathbf{G}_{op,k}^T + \mathbf{R}'_k)^{-1} \quad (5a)$$

$$\hat{\Sigma}_k = (\mathbf{I} - \mathbf{K}_{op,k} \mathbf{G}_{op,k}) \check{\Sigma}_k \quad (5b)$$

$$\hat{\boldsymbol{\mu}}_k = \dot{\boldsymbol{\mu}}_k + \mathbf{K}_k (\bar{\mathbf{Z}}_k - \mathbf{g}(\boldsymbol{\mu}_{op,k}) - \mathbf{G}_k (\dot{\boldsymbol{\mu}}_k - \boldsymbol{\mu}_{op,k})) \quad (5c)$$

式中, 下标 op 代表迭代次数。此处判断收敛可以状态量更新足够小, 或者达到最大迭代步数作为依据。本文设计收敛标准如下:

$$\begin{aligned} & (\hat{\boldsymbol{\mu}}_k - \boldsymbol{\mu}_{op,k})^T \mathbf{M} (\hat{\boldsymbol{\mu}}_k - \boldsymbol{\mu}_{op,k}) < e_{re} \\ & e_{re} = \alpha (\dot{\boldsymbol{\mu}}_k - \boldsymbol{\mu}_{op,1})^T \mathbf{M} (\dot{\boldsymbol{\mu}}_k - \boldsymbol{\mu}_{op,1}) \end{aligned} \quad (5d)$$

式中, α 为收敛系数, 本文 $\alpha=0.01$, \mathbf{M} 为距离权重矩阵。若复合收敛标准, 则退出迭代完成更新; 否则令 $\mathbf{x}_{op,k} = \hat{\mathbf{x}}_k$, 继续迭代。

2 多传感器融合算法

多传感器融合状态估计算法框架如图 1, 传感器包含 IMU、GPS、运动传感器(轮速与转角)、相机与激光雷达。其中:①IMU 数据去除重力加速度后, 作为算法的预测环节, 对观测状态向量进行预测;②GPS 数据作地心坐标系到地图坐标系的变换后, 用于更新状态量;③运动传感器数据解算出质心速度后, 用于更新;④图像与点云信息通过感知算法提取前方路标, 结合先验路标坐标, 求路标的地图定位误差, 用于更新。其中, 状态向量的时间戳与频率由 IMU 数据决定, 其他 3 个更新环节互不影响, 接收到对应传感器数据才产生更新作用, 各自对状态向量进行维护。

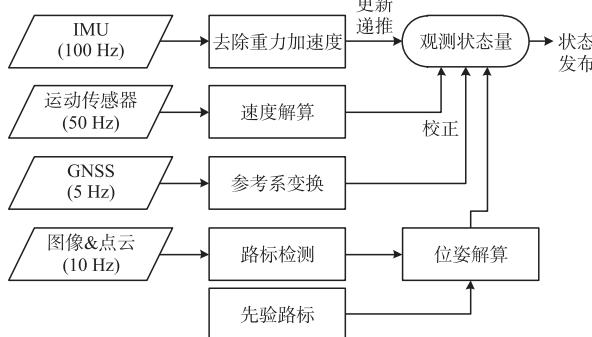


图 1 多传感器融合架构

另外, 考虑到不同传感器数据采集时间和频率不同步, 直接用于更新会造成新的误差, 所以在数据进行处理前会利用时间戳差进行补偿处理。

2.1 状态量定义

本文车辆行驶在封闭的场景, 需要完成地图内的精准定位, 以及对车辆的行驶速度进行观测。而对于一般车辆来说, 平面二维的定位与速度定义基本满足其需求, 所以定义状态量 \mathbf{x}_k :

$$\mathbf{x}_k = [X \ Y \ \varphi \ v_x \ v_y \ w]$$

各分量分别代表了车辆的水平定位(坐标 X, Y 以及航向 φ)与运动状态(纵向速度 v_x , 侧箱速度 v_y , 横摆角速度 w)。

定义 I 系为车辆质心坐标系(IMU); W 系为地心坐标系, GPS 采集数据是基于此系的; M 系为先验地图坐标系; L 系为 Lidar 点云坐标系; 符号定义 \mathbf{T}_l^M 为上标参考系到下标目标系的位姿变换矩阵, 其乘法符合链式法则, 如:

$$\mathbf{T}_L^M = \mathbf{T}_I^M \mathbf{T}_L^I$$

$\mathbf{Z}_{lm,i}^M$ 为上标参考系中对目标的观测。如果符号没有标注上标参考系(如 \mathbf{x}_k), 位姿默认参考 M 系; 而所有速度默认参考 I 系。

2.2 IMU 预测环节

IMU 的离散数据可以结合车辆运动学模型直接对状态进行预测, 由于本文只讨论水平方向的运动, 所以对原始的 IMU 数据进行去偏置与重力加速度后(节 3.1), 取纵侧向加速度与横摆角速度 $[\bar{a}_x \ \bar{a}_y \ \bar{w}]^T$ 表示原始数据, 以及前后两帧 IMU 数据的时间差作为 Δt_k 。车辆二自由度运动学模型状态方程如下:

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\varphi} \\ \dot{v}_x \\ \dot{v}_y \end{bmatrix} = \begin{bmatrix} v_x \cos\varphi - v_y \sin\varphi \\ v_x \sin\varphi + v_y \cos\varphi \\ w \\ \bar{a}_x + v_y w \\ \bar{a}_y - v_x w \end{bmatrix} \quad (6)$$

状态量预测的非线性映射如下:

$$\dot{\boldsymbol{\mu}}_k = \hat{\boldsymbol{\mu}}_{k-1} + \begin{bmatrix} (\hat{v}_{x,k-1} \cos\hat{\varphi}_{k-1} - \hat{v}_{y,k-1} \sin\hat{\varphi}_{k-1}) \Delta t_k \\ (\hat{v}_{x,k-1} \sin\hat{\varphi}_{k-1} + \hat{v}_{y,k-1} \cos\hat{\varphi}_{k-1}) \Delta t_k \\ \hat{w}_{k-1} \Delta t_k \\ (\hat{a}_x + \hat{v}_{y,k-1} \hat{w}_{k-1}) \Delta t_k \\ (\hat{a}_y - \hat{v}_{x,k-1} \hat{w}_{k-1}) \Delta t_k \\ \bar{w} - \hat{w}_{k-1} \end{bmatrix} \quad (7a)$$

$$\mathbf{Q}_k = \mathbf{V}_k \mathbf{M}_k \mathbf{V}_k^T \quad (7b)$$

$$\mathbf{F}_{k-1} = \begin{bmatrix} 1 & 0 & (-\hat{v}_{x,k-1}\sin\hat{\varphi}_{k-1} - \hat{v}_{y,k-1}\cos\hat{\varphi}_{k-1})\Delta t_k & \Delta t_k \cos\hat{\varphi}_{k-1} & -\Delta t_k \sin\hat{\varphi}_{k-1} & 0 \\ 0 & 1 & (\hat{v}_{x,k-1}\cos\hat{\varphi}_{k-1} - \hat{v}_{y,k-1}\sin\hat{\varphi}_{k-1})\Delta t_k & \Delta t_k \sin\hat{\varphi}_{k-1} & \Delta t_k \cos\hat{\varphi}_{k-1} & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t_k \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0_{3 \times 3} & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (7c)$$

$$\mathbf{V}_k = \frac{\partial \check{\boldsymbol{\mu}}_k}{\partial \boldsymbol{u}_k} = \begin{bmatrix} \mathbf{0}_{3 \times 3} \\ \Delta t_k \\ \Delta t_k \\ 1 \end{bmatrix} \quad (7d)$$

式中, \mathbf{M}_k 为 IMU 数据协方差矩阵。结合式(3), 即可完成对状态先验 $\check{\boldsymbol{x}}_k$ 的预测。

2.3 先验路标更新

定义在已知地图内的稀疏路标(landmark)坐标为 $\mathbf{Z}_{lm,n}^M$, 即在地图中路标的坐标。另外, 在 $k+\Delta t_{lm}$ 时刻, 相机及雷达采集到了车前路标相对雷达坐标系为 $\bar{\mathbf{Z}}_{lm,i}^L$ (黑色箭头), i 指代当前感知的路标序号, Δt_{lm} 代表感知数据与当前状态的时间戳差。如图 2 所示, 将车辆感知到的前方路标匹配到稀疏地图(右)内的对应路标(橙色箭头过程), 以进行车辆状态更新(绿色箭头过程)。

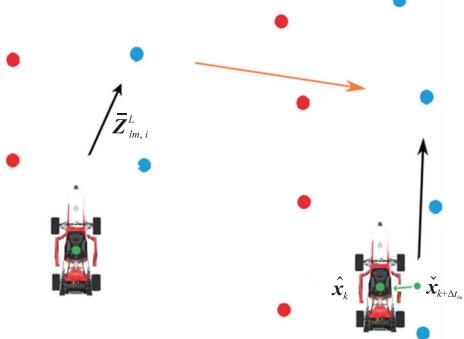


图 2 利用路标进行状态更新

算法步骤如下:

Step 1 对该数据作时间戳补偿如下:

$$\check{\boldsymbol{\mu}}_{k+\Delta t_{lm}} = \check{\boldsymbol{\mu}}_k + \begin{bmatrix} (\check{v}_{x,k} \cos\check{\varphi}_k - \check{v}_{y,k} \sin\check{\varphi}_k) \Delta t_{lm} \\ (\check{v}_{x,k} \sin\check{\varphi}_k + \check{v}_{y,k} \cos\check{\varphi}_k) \Delta t_{lm} \\ \check{w}_k \Delta t_{lm} \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (8)$$

在忽略车辆实际加速度影响的前提下, 上式能消除数据之间时间不匹配引起的误差。

Step 2 结合状态先验, 将实时感知的路标点投影到 M 坐标系, 公式如下:

$$\mathbf{T}_l^M \leftarrow \check{\boldsymbol{\mu}}_{k+\Delta t_{lm}} \quad (9a)$$

$$\bar{\mathbf{Z}}_{lm,i}^M = \mathbf{T}_l^M \mathbf{T}_L^I \bar{\mathbf{Z}}_{lm,i}^L \quad (9b)$$

式中 \mathbf{T}_L^I 是车辆坐标系到 Lidar 坐标系的位姿变换矩阵。

Step 3 如图 2 所示, 左侧为车辆实时感知的路标信息, 右侧为地图内定位, 搜索先验路标 $\mathbf{Z}_{lm,n}^M$ 中与 $\bar{\mathbf{Z}}_{lm,i}^M$ 距离最近的路标点进行 ID 匹配。若 ID 匹配成功, 进行如下运算:

$$\mathbf{G}_{k,i} = \frac{\partial \check{\mathbf{Z}}_{lm,i}^M}{\partial \check{\boldsymbol{\mu}}_{k+\Delta t_{lm}}} \frac{\partial \check{\boldsymbol{\mu}}_{k+\Delta t_{lm}}}{\partial \check{\boldsymbol{\mu}}_k} \quad (10a)$$

$$\mathbf{E}_{k,i} = \mathbf{Z}_{lm,i}^M - \check{\mathbf{Z}}_{lm,i}^M \quad (10b)$$

式(10a)的求导过程比较复杂, 本文采用微小扰动法, 即对 $\check{\boldsymbol{\mu}}_k$ 的不同维度依次增加微小增量 Δd , 代入式(9), 计算坐标增量得到 $\Delta \bar{\mathbf{Z}}_{lm,i}^M$ 除以 Δd , 从而求出 $\mathbf{G}_{k,i}$ 对应维度的值。

Step 4 将所有 ID 匹配成功的矩阵进行合并, 设有 m 个匹配成功的路标。方法如下:

$$\mathbf{G}_k = \begin{bmatrix} \mathbf{G}_{k,0} \\ \vdots \\ \mathbf{G}_{k,m} \end{bmatrix}, \mathbf{E}_k = \begin{bmatrix} \mathbf{E}_{k,0} \\ \vdots \\ \mathbf{E}_{k,m} \end{bmatrix} \quad (10c)$$

将结果代入式(5)完成该环节对状态量的校正。如果路标匹配成功较多, 容易造成上述矩阵维度过大, 所以本文选取离车辆最近 4 个路标用于更新。

2.4 GPS 更新

本文采用双天线 GPS 设备, 可以测量车辆航向, 以及主天线的坐标, 定义 GPS 采集到的数据为 $\bar{\mathbf{Z}}_G^W = [\bar{X} \ \bar{Y} \ \bar{\varphi}]^T$, $\bar{\varphi}$ 即地心坐标系下的 GPS 主天线与副天线连线航向。这里需要在车辆启动时, 对地图 M 坐标系与地心 W 坐标系之间的转换进行初始化, 得到 \mathbf{T}_M^W 。与式(8)类似, 利用 GPS 数据与状态数据的时间差对位姿进行补偿, 得到 GPS 数据时间下状态先验 $\check{\boldsymbol{\mu}}_{k+\Delta t_G}$, 然后作以下计算;

$$\check{\mathbf{Z}}_I^M = \mathbf{g}(\check{\boldsymbol{\mu}}_{k+\Delta t_G}) = \begin{bmatrix} \check{X}_{k+\Delta t_G} \\ \check{Y}_{k+\Delta t_G} \\ \check{\varphi}_{k+\Delta t_G} \end{bmatrix} \quad (11a)$$

$$\bar{\mathbf{Z}}_I^M = \text{Log}((\mathbf{T}_M^W)^{-1} \bar{\mathbf{T}}_G^W \mathbf{T}_I^G) \quad (11b)$$

$$\bar{\mathbf{T}}_G^W \leftarrow \bar{\mathbf{Z}}_G^W \quad (11c)$$

$$\begin{aligned} \mathbf{G}_k &= \frac{\partial \check{\mathbf{Z}}_I^M}{\partial \check{\boldsymbol{\mu}}_{k+\Delta t_G}} = \\ & \left[\begin{array}{ccccc} 1 & 0 & (-\check{v}_{x,k} \sin \check{\varphi}_k - \check{v}_{y,k} \cos \check{\varphi}_k) \Delta t_G & \Delta t_G \cos \check{\varphi}_k & -\Delta t_G \sin \check{\varphi}_k & 0 \\ 0 & 1 & (\check{v}_{x,k} \cos \check{\varphi}_k - \check{v}_{y,k} \sin \check{\varphi}_k) \Delta t_G & \Delta t_G \sin \check{\varphi}_k & \Delta t_G \cos \check{\varphi}_k & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t_G \\ & & \mathbf{0}_{3 \times 3} & & & \mathbf{1}_{3 \times 3} \end{array} \right] \quad (11d) \end{aligned}$$

式中 $\text{Log}(\cdot)$ 为指数映射,由位姿变换矩阵转换为平移与旋转向量。将上式结果代入式(5)完成 GPS 对状态估计的更新。

2.5 运动传感器更新

轮速传感器与前轮转角传感器的作用较为关键,对前后轮分别取左右轮速平均值作为前后轮的轮速,定义采集到的数据为 $\bar{\mathbf{Z}}_v = [\bar{v}_{xf} \bar{v}_{xr} \bar{\delta}]^T$ 。前面的更新环节主要是对坐标起修正作用。在此,假设车轮转动线速度等于车轮纵向速度,即忽略车轮滑移率。由于状态量不包含加速度,所以此环节不作时间差补偿。通过车辆运动模型状态先验到期望传感器数据的映射关系:

$$\check{\mathbf{Z}}_v = \begin{bmatrix} \check{v}_{xf} \\ \check{v}_{xr} \\ \delta \end{bmatrix} = \mathbf{g}(\check{\boldsymbol{\mu}}_k) = \begin{bmatrix} \check{v}_x \cos \alpha_f + (\check{v}_y + a\check{w}) \sin \alpha_f \\ \check{v}_x \cos \alpha_r + (\check{v}_y - b\check{w}) \sin \alpha_r \\ \arctan(L\check{w}/\check{v}_x) \end{bmatrix} \quad (12a)$$

$$\begin{cases} \alpha_f = \arctan((\check{v}_y + a\check{w})/\check{v}_x) \\ \alpha_r = \arctan((\check{v}_y - b\check{w})/\check{v}_x) \end{cases} \quad (12b)$$

$$\mathbf{G}_k = \frac{\partial \check{\mathbf{Z}}_v}{\partial \check{\boldsymbol{\mu}}_k} \quad (12c)$$

由于此处求偏微分较为复杂,此处同样使用微小扰动法求解 \mathbf{G}_k 。

由于选择了低成本的传感器,所以一般不会增加额外的成本,且即使 GPS 或者运动传感器出现失效等情况,算法仍然可以继续保持准确定位,冗余度较高,后续也将用实验数据证明。

3 传感器预处理

在状态估计算法运行时,需要进行一些预处理,才能提取第 2 章中的所需要的信息,主要包括传感器标定、感知路标检测、传感器外参变换矩阵。

3.1 IMU 标定

IMU 传感器在车辆状态估计问题中起关键的作用,一般 IMU 的标定包括偏置(Bias)标定、方差(White Noise)标定、偏置随机游走(Bias Instability),其中偏置标定与方差标定对本算法是

必要的。偏置标定^[15]过程需要将 IMU 摆放 30 个以上的不同的工位进行短暂静置,最终计算 IMU 数据校正偏置所使用的旋转与向量。方差与偏置游走标定^[16]利用去除偏置后的 IMU,静置 2 h 采集数据,以分析获取 Allan 方差,标定数据如表 1。

IMU 数据通过去偏置以及重力加速度分量后,提取获得 $[\bar{a}_x \bar{a}_y \bar{w}]^T$; IMU 与 Lidar 的外参变换 \mathbf{T}_L^I 、IMU 与 GPS 的外参变换 \mathbf{T}_G^I ,可以直接通过车辆上传传感器定位安装获取。

3.2 Lidar-Camera 联合标定与路标检测

感知信息提取是基于 Lidar-Camera 的时间空间信息融合完成的,首要需要标定其外参变换,以给图像的信息重投影到点云空间中,标定可以利用开源的联合标定工具^[17]完成,也可以通过传感器安装定位获取。本文应用场景的路标提取步骤如下:

Step 1 对于路标检测,我们对到的图像与点云数据进行时间戳对准,以点云时间戳为准,搜索与其时间戳差绝对值最小的图像帧进行匹配;

Step 2 点云经过去地面算法得到非地面点云,图像通过目标检测得到的路标外框;

Step 3 检测到了桩桶框通过外参变换投影到点云空间中对非地面点云进行过滤,用以分割出目标点云。分割结果如图 3,通过对目标点云进行坐标的解算,可以得到每个路标的三维坐标 $\bar{\mathbf{Z}}_{lm,i}^L$ 。



图 3 识别车辆前方路标的感知分割算法

4 实验结果与分析

4.1 实车测试平台与工况

图 4 为团队自研发的车辆(FSAC 赛车如^[18])。该车辆安装了多种传感器,包括 IMU(AH-100B AHRS)、单目相机(HIKVISION MV-CA013-20GC)、



图 4 无人驾驶方程式测试平台示意

16线激光雷达、车前后两个GPS天线(不需差分)、以及后轮轮速编码器与转角偏码器;计算器组成包括ECU(RapidECU-E3 Huahai)、工控机(ADVANTECH MIC-7700,不带GPU,基于Linux与ROS开发程序框架)、以及GPU(Nvidia TX2)。本文后续的数据分析与算法验证皆基于该平台实时运行,表1记录了涉及本文算法的传感器数据方差。

表1 不同传感器的数据方差

| 采集数据 | | 方差 |
|-------|--------------------|------------------|
| IMU | \bar{a}_x | 0.000 08 |
| | \bar{a}_y | 0.000 22 |
| | w | 0.057 63 |
| GPS | \bar{X} | 0.500 67 |
| | \bar{Y} | 0.544 95 |
| | φ | 0.003 53 |
| 运动传感器 | \bar{v}_{xr} | 0.002 30 |
| | $\bar{\delta}$ | 0.001 21 |
| 感知路标 | $\bar{Z}_{lm,i}^L$ | Diag(0.05, 0.05) |

测试场景主要有八字绕环(图5,约50 m×30 m)与直线加速(图6,约5 m×90 m),道路主要由桩桶组成,且有先验地图即已有桩桶的摆放坐标。八字绕环要求车辆先绕右两周、再绕左两周,圆周直径约为20 m;直线加速要求车辆加速行驶在直线道路中,且在75 m后开始制动。测试过程中,车辆以自动驾驶模式完成行驶,从而验证算法与程序的可行性。对于一般无人驾驶车辆,普遍认为感知定位方面算法较注重算法的精度以及冗余性(鲁棒性、系统稳定性)。后续验证主要包含:①定位与速度观测的有效性与冗余性;②定位与速度观测的精度;③iEKF迭代的必要性;④算法的实时性。



图5 八字绕环测试



图6 直线加速测试

4.2 定位分析及冗余度验证

图7展示了本文算法与文献[9]算法及GPS原始数据的定位轨迹对比,本文多传感器融合下的估计算法得到的定位轨迹与GPS轨迹基本一致。分析本文误差的几个主要来源,包括GPS的定位数据误差(一般介于米级到分米级)、人为桩桶摆放与先验坐标的差异、非线性模型的线性化。在此基础上,该轨迹精度基本准确。

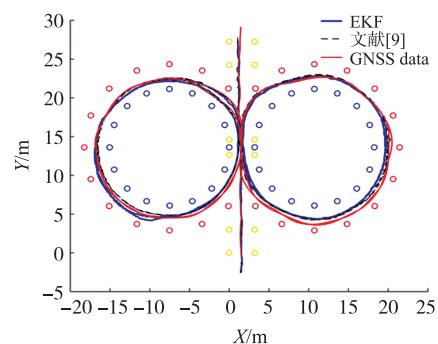


图7 观测轨迹与GPS原始数据对比

另外,通过图8(图7的局部放大)可以看出,由于本文算法是一种模型的递推状态估计,而文献[9]的定位是基于当前有关姿态的不同传感器数据结合进行的修正,所以平顺性上较差,出现比较明显的抖动甚至后退(而车辆实际是一直前进的)。所以本文算法对车辆的定位估计更贴合实际。

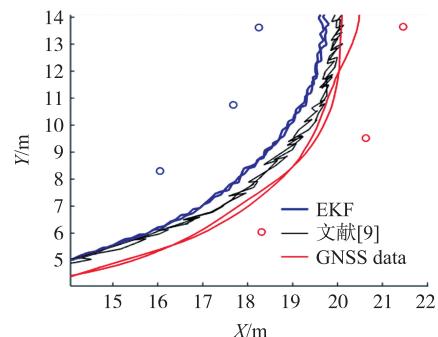


图8 图7的局部放大

图9与图10展示了模拟车辆在不同环节的传感器失效下,轨迹定位的结果,分别有感知传感器失

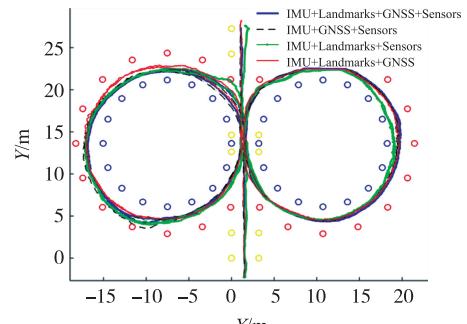


图9 八字工况失效模拟测试的轨迹对比

效(即先验环节无效)、GPS失效、运动传感器失效。由于算法设计是并行的,所以传感器失效下,算法仍能保持定位的基本精准,系统稳定可靠,而文献[9]的三级串联思路则无法进行冗余度验证。八字工况由于几个误差因素,轨迹重合度较好,而直线工况下几种失效的轨迹基本重合。

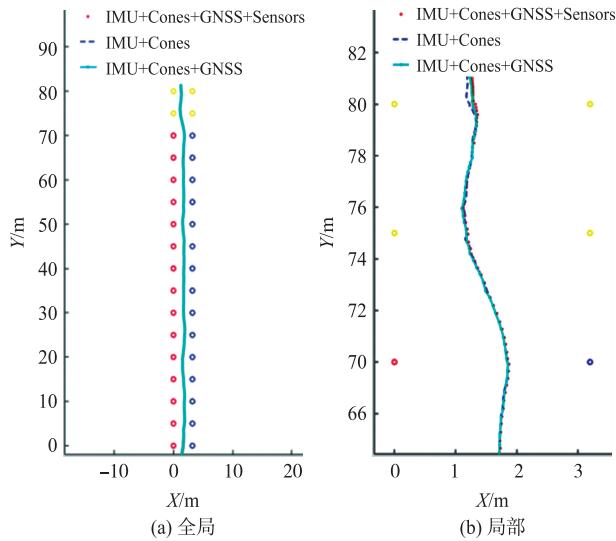


图10 直线工况失效模拟测试(左全局,右局部)

4.3 速度观测分析及冗余度验证

图11展示了速度观测的结果,结合状态后三维速度代入式(12a)可以反算后轴速度与转向角,然后与实际传感器数据进行比较。由图可以看出,车辆速度观测基本精准,而较原始数据平顺度更差,是由于本文并行思路认为GPS、先验路标也与车辆速度存在间接联系,从而对速度产生更新作用,且数据采集频率相对较低,所以是速度观测的结果较为不平,不过基本精准。且同样的只要运动传感器不失效,速度的观测就能保持有效,由于其对速度观测存在直接联系;反之,如果运动传

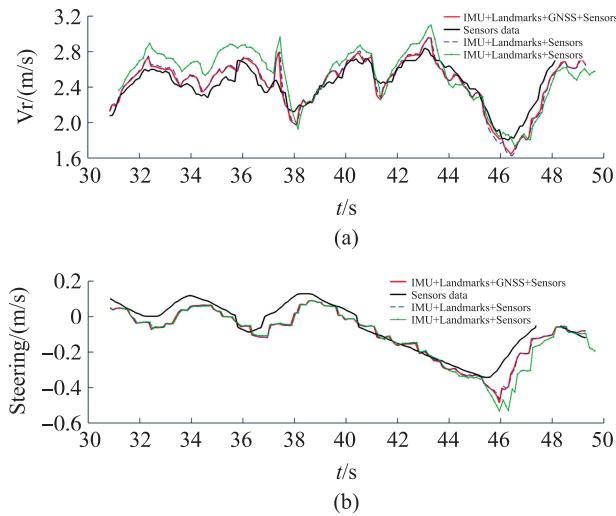


图11 不同传感器失效模拟下的速度观测

感器失效,则速度观测的精度会明显下降,这可以继续基于并行的思路,通过增加其他车辆基本的运动传感器加入环节。

4.4 iEKF验证

图12展示了EKF与iEKF的定位轨迹,图13展示了3个校正环节在采样时间段中迭代次数的直方图统计。可以看出,iEKF与EKF的轨迹基本重合,算法同样有效而方差直方图说明了,由于GPS、先验路标与定位有直接联系,而与速度估计通过时间截补偿与状态的协方差矩阵产生间接联系,所以收敛快;而运动传感器对定位递推与速度估计都有直接联系,所以一般迭代多步后收敛,所以iEKF对于进一步提高精度具有必要性。

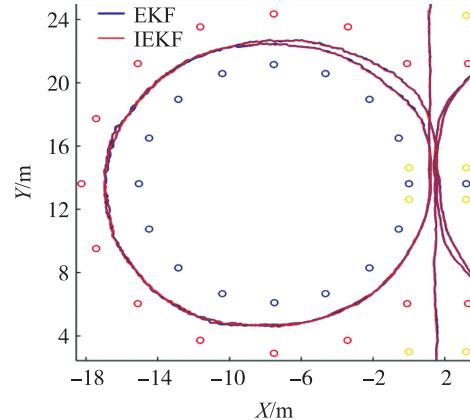
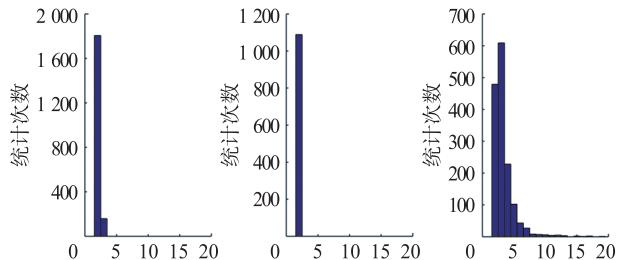


图12 EKF与iEKF的定位轨迹对比



依次为先验更新、GPS更新、运动传感器更新

图13 iEKF更新环节迭代次数直方图

4.5 状态误差分析

表2展示了EKF、iEKF的状态误差对比,分别相对于轮速、转角、GPS的原始数据。iEKF的结果平均绝对误差(MAE)与均方根误差(RMSE)较之EKF基本有所改善。而文献[9]的方法用于本文工况是直接以GPS数据作为状态量被校正,而本文是维护一个独立的递推状态,所以前者与GPS原始数据的误差理应更小。

表3展示了冗余度测试下定位的误差对比。其中,不可信系数NCI(Noncredibility index^[14])可以评估算法的估计是否可信,是面向此类算法的较为普遍的评估指标;若NCI的值越小,则算法的估计值

越可信。不同失效下,精度有所下降但并不发散,进一步证明了本文算法在不同的失效下仍能保持算法基本精准,且具有切实的冗余性与有效性。

表2 EKF、iEKF 观测结果误差分析

| | | MAE | RMSE |
|-------|-----------------------|---------|---------|
| EKF | $v_{xr}/(\text{m/s})$ | 0.100 1 | 0.124 6 |
| | δ/rad | 0.050 4 | 0.064 9 |
| | 定位/m | 0.322 3 | 0.379 3 |
| iEKF | $v_{xr}/(\text{m/s})$ | 0.097 0 | 0.119 2 |
| | δ/rad | 0.050 9 | 0.067 8 |
| | 定位/m | 0.316 2 | 0.373 1 |
| 文献[9] | 定位/m | 0.273 5 | 0.363 6 |

表3 不同传感器失效模拟下的定位误差分析

| | MAE | RMSE | NCI |
|--------------------|---------|---------|-------|
| IMU+GPS+先验+Sensors | 0.322 3 | 0.379 3 | 23.22 |
| IMU+GPS+Sensors | 0.421 2 | 0.503 5 | 35.72 |
| IMU+先验+Sensors | 0.462 6 | 0.583 3 | 53.92 |
| IMU+GPS+先验 | 0.334 8 | 0.412 1 | 36.59 |

4.6 实时性验证

表4展示了预测与校正的4个环节的耗时统计,其中校正的时间指每步所需,即迭代的情况下需要乘上迭代次数才是该环节的实际耗时。由表可知,在本文的测试平台上,以算法100~200 Hz的运行频率来说,算法基本满足实时运算的要求。

表4 不同环节的每步实时性分析

| | mean \pm stddev($\times 10^{-5}$ s) |
|----------------|--|
| IMU 预测(100 Hz) | 6.141 7 \pm 2.521 4 |
| 先验更新(10 Hz) | 7.584 2 \pm 2.375 3 |
| GPS 更新(5 Hz) | 1.850 7 \pm 4.599 9 |
| 运动传感器更新(50 Hz) | 3.913 0 \pm 1.976 9 |

5 结论

本文结合iEKF设计了针对场地无人驾驶车辆常见传感器的多传感器融合状态估计算法,以解决几个方面的问题:①融合车辆较为普及的传感器,包括IMU、GPS、轮速、转角、感知与先验地图数据,来对车辆速度与定位进行估计;②结合iEKF理论进行设计,充分考虑各传感器数据的不确定性,提高估计的精度;③提出不同的传感器数据各自处理,实现并联的思路,维护一个独立的车辆状态量,保证了算法的冗余度。

本文将其应用在大学生无人驾驶方程式平台上进行验证;通过实验可知,算法能够精准地估计车辆状态,且具有良好的实时性和冗余度。

参考文献:

- [1] Thrun S. Probabilistic Robotics[J]. Communications of the ACM, 2002, 45(3):52–57.
- [2] Barfoot T D. State Estimation for Robotics[M]. Cambridge: Cambridge University Press, 2017.
- [3] 张勇刚,张云浩,李宁.基于互补滤波器的MEMS/GPS/地磁组合导航系统[J].系统工程与电子技术,2014,36(11):2272–2279.
- [4] Jun N I, Jibin H U. Autonomous Driving System Design for Formula Student Driverless Racecar[C]//2018 IEEE Intelligent Vehicles Symposium(IV). IEEE, 2018;1–6.
- [5] 郭安,周洲,祝小平,等.基于EKF算法的太阳能无人机低成本飞控状态估计[J/OL].控制与决策. <https://doi.org/10.13195/j.kzyjc.2019.0091>. 2020-08-10.
- [6] Ko N, Youn W, Choi I, et al. Features of Invariant Extended Kalman Filter Applied to Unmanned Aerial Vehicle Navigation[J]. Sensors 2018, 18(9):2855.
- [7] Goh S T, Abdelkhalik O, Zekavat S A R. A Weighted Measurement Fusion Kalman Filter Implementation for UAV Navigation[J]. Aerospace Science and Technology, 2013, 28(1):315–323.
- [8] Gao Z, Ge M, Li Y, et al. Modeling of Multi-Sensor Tightly Aided BDS Triple-Frequency Precise Point Positioning and Initial Assessments[J]. Information Fusion, 2020, 55:184–198.
- [9] Cui B, Chen X, Xu Y, et al. Performance Analysis of Improved Iterated Cubature Kalman Filter and Its Application to GNSS/INS [J]. ISA Transactions, 2017, 66:460–468.
- [10] 黄伟杰,张国山.基于快速不变卡尔曼滤波的视觉惯性里程计[J].控制与决策,2019,34(12):2585–2593.
- [11] 刘济,高丽君.基于UKF和神经网络的一类非线性系统状态估计[J].控制与决策,2014,29(11):2076–2080.
- [12] Hertzberg C, Wagner R, Frese U, et al. Integrating Generic Sensor Fusion Algorithms with Sound State Representations Through Encapsulation of Manifolds[J]. Information Fusion, 2013, 14(1):57–77.
- [13] Bell B M, Cathey F W. The Iterated Kalman Filter Update as a Gauss-Newton Method[J]. IEEE Transactions on Automatic Control, 1993, 38(2):294–297.
- [14] Havlík J, Straka O. Performance Evaluation of Iterated Extended Kalman Filter with Variable Step-Length[C]//Journal of Physics: Conference Series. IOP Publishing, 2015, 659(1):012022.
- [15] Tedaldi D, Pretto A, Menegatti E. A Robust and Easy to Implement Method for IMU Calibration without External Equipments[C]//2014 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2014:3042–3049.
- [16] El-Sheimy N, Hou H, Niu X. Analysis and Modeling of Inertial Sensors Using Allan Variance[J]. IEEE Transactions on Instrumentation & Measurement, 2007, 57(1):140–149.
- [17] Kato S, Tokunaga S, Maruyama Y, et al. Autoware on Board: Enabling Autonomous Vehicles with Embedded Systems[C]//2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPs). IEEE, 2018:287–296.
- [18] Kabzan J, Valls M I, Reijgwart V, et al. AMZ Driverless: The Full Autonomous Racing System[J/OL]. <https://www.researchgate.net/publication/333077486>. 2020-08-10.



彭文正(1996—),男,广东工业大学机电工程学院研究生,现研究方向为SLAM与无人驾驶技术,358799851@qq.com;



敖银辉(1973—),男,广东工业大学机电系教授,主要研究方向有机电一体化、新能源与无人驾驶车辆,发表学术论文50余篇,aoyinhui@gdut.edu.cn;



黄晓涛(1999—),男,广东工业大学机电工程学院本科生,现研究方向为无人驾驶技术。