Multi-Sensor Fusion Algorithm in Localization for vehicles based on extended Kalman filter

1st Zijian Luo
Dept. of Electrical and Computer Engineering(ECE)
National Univeristy of Singapore(NUS)
Singapore, Singapore
luozijian@u.nus.edu

2nd Rui Jiang
Dept. of Electrical and Computer Engineering(ECE)
National Univeristy of Singapore(NUS)
Singapore, Singapore
elejiangrui@nus.edu.sg

***Abstract*—This report is about multi-sensor fusion in assisted positioning based on extended Kalman filter. For the measurement accuracy of different sensors in the measurement, a unified prediction model needs to be proposed to simulate the uncertainty of these measurement data. In addition, the estimation accuracy will decrease under the condition of high non-linearity because some filters are approximated by a first-order Taylor series in the error covariance matrix. To solve this problem, we propose a new multi-sensor fusion algorithm for localization, which can improve the final result of state estimation. Despite the high degree of nonlinearity, modeling uncertainty and external interference, our proposed method can still provide better navigation and positioning results.**

***Index Terms*—Multi-Sensor Fusion, extended Kalman filter, state estimation, Localization**

I. Introduction

Autonomous technology is a fast-developing field, and both academia and industry are interested in it, so this research is often at the forefront of innovation and technology. However, even in some well-known and successful autonomous vehicle projects, it is often common practice to use relatively simple and sometimes naive control strategies and/or system models for vehicle control. For example, Stanley [1].

It explains in detail the concept of a very common motion model in professional applications, that is, a constant turning rate model composed of a hexapod based on a linear guide. Using these commonly used motion models, autonomous robotic vehicles can simulate 99% actual problems.

The contributions of this work are three-fold:

- For linear motion, common motion models can simulate the motion situation well, but for non-linear motion, we use the constant steering model in VO [2] to simulate the rotational drift of the vehicle. Through the corresponding state transition equation, the change of the next state is obtained.
- In order to simplify the simulation, in our framework, the positioning of the vehicle is abstracted as a point in the graph model. On this basis, the problem is expressed as a point's trajectory change. Therefore, there is no need to consider the width and height of the vehicle, so the bicycle model is not applicable in our simulation.
- We have proposed a multi-sensor framework to contain different types of sensors. Our simulation algorithm can collect multiple measurement positions from different sensors and provide a final estimate based on the corrected results. Compared with the ground truth, the experimental results show that the multi-sensor framework can perform better than the single-sensor framework, which is very useful and high-quality for the research of autonomous driving.

The remaining part of the paper is organized as follows: After presenting related work in Section II, we introduce principles of sensors at Section IV. Section V provides three levels of abstraction motion models involving constant acceleration, constant velocity, constant turning rate and so on. In Section VI, a general multi-sensor framework of pose estimation algorithm is proposed. Section VII detail the implementation of the proposed approach and demonstrate the experimental results. Finally, VIII concludes the paper and discusses on possible future work.

II. Related work

In recent years, the problem of robot positioning has been a hot research topic, especially in the field of autonomous ground vehicles. The nonlinear control law that is used for auto-tracking the trajectory of the car and provided in real-time attitude estimation is very valuable. The following are some examples of successful self-driving car projects in the figures 1 and 2

Figure 1 is Stanley, this self-driving car uses intuitive steering control laws based on a simple kinematics vehicle model to win the DARPA Challenge. Figure 2 shows the self-driving car boss that won the DARPA Urban Challenge because Boss uses a more complex model predictive control strategy to perform vehicle control.

The author [3] told that vehicle controller series are called path trackers. The goal of the path tracking controller is to minimize the lateral distance between the vehicle and the

Fig. 1. Example of Stanley.

Fig. 2. Example of Boss.

defined path, minimize the difference between the vehicle heading and the heading of the defined path, and limit the steering input to smooth motion while maintaining stability.

III. Sensors in Data Tracking

As the sensors [4] are the key components of the self-driving vehicles, the fusion of the information from the sensors and their proper interpretation followed by control of the vehicle has the central point in the autonomous driving. Figure 3 shows the procedure example of autonomous vehicle system.

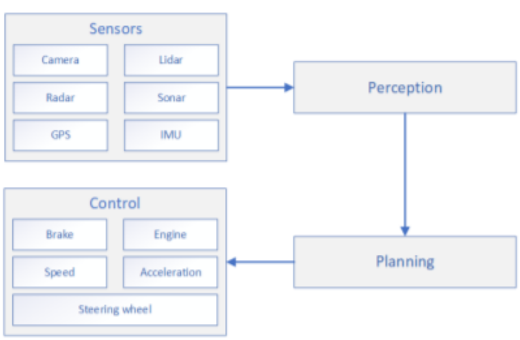

Fig. 3. Block diagram of the autonomous vehicle system.

In actual applications [5] [6], cameras, lidars, radars, sonar, global positioning systems (GPS), inertial measurement units (IMU) and wheel odometers can be part of the sensors for autonomous vehicles. The sensors in the car are used to collect data. These data are analyzed by the computer in the self-driving car and used to control the fusion of the sensors in the self-driving car with other sensors to realize the steering, braking and speed of the vehicle.

A. Camera

Although the perception of autonomous vehicles is achieved through many sensors and sensor systems, cameras [7] are one of the earliest sensors used in unmanned vehicles and are still the main choice of automakers. In practical applications, today's new vehicles generally have dozens of different cameras installed.

B. Radar

Radar [8] is a sensor integrated into the vehicle for different purposes such as adaptive cruise control, blind spot warning, collision warning, and collision avoidance. Unlike cameras, radar does not need to process any video input with a large amount of data, but compared with lidar and cameras, the processing speed required to process data output is lower.

IV. Motion Models

For motion models [9], systemization can be achieved by defining different levels of complexity. The linear motion model is at the low end of this definition. These models include, assuming constant velocity (CV) or constant acceleration (CA). Their main advantage is the linearity of the state transition equation and allows the best propagation of the state probability distribution. On the other hand, these models assume linear motion, so rotation (especially yaw rate) cannot be considered, so nonlinear models are introduced. One example is constant turning rate(CT),which can simulate the basic motion change in non-linear model.

A. Constant Velocity model

As for the CV model, we annote its space state with

$$\vec{x}(t) = \begin{pmatrix} x & v_x & y & v_y \end{pmatrix}^T \tag{1}$$

is a linear motion model, the linear state transition

$$\vec{x}(t+T) = F * \vec{x}(t) \tag{2}$$

is substituted by the state transition function vector

$$F = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

B. Constant Accelaretion model

As for the CA model, we annote its space state with

$$\vec{x}(t) = \begin{pmatrix} x & v_x & a_x & y & v_y & a_y \end{pmatrix}^T \tag{4}$$

is a linear motion model, the linear state transition

$$\vec{x}(t+T) = F * \vec{x}(t) \tag{5}$$

is substituted by the state transition function vector

$$F=\begin{bmatrix}1 & T & \frac{T^2}{2} & 0 & 0 & 0\\ 0 & 1 & T & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & T & \frac{T^2}{2}\\ 0 & 0 & 0 & 0 & 1 & T\\ 0 & 0 & 0 & 0 & 0 & 1\end{bmatrix} \tag{6}$$

C. Constant turning rate model

As for the CT model, we annote its space state with

$$\vec{x}(t)=\begin{pmatrix} x & v_x & y & v_y \end{pmatrix}^T \tag{7}$$

is a linear motion model, the linear state transition

$$\vec{x}(t+T)=F(\omega_t)*\vec{x}(t)+\omega_t \tag{8}$$

is substituted by the state transition function vector [10]

$$F(\omega_t)=\begin{bmatrix}1 & \frac{\sin(\omega_t T)}{\omega_t} & 0 & -\frac{1-\cos(\omega_t T)}{\omega_t}\\ 0 & \cos(\omega_t T) & 0 & -\sin(\omega_t T)\\ 0 & \frac{1-\cos(\omega_t T)}{\omega_t} & 1 & \frac{\sin(\omega_t T)}{\omega_t}\\ 0 & \sin(\omega_t T) & 0 & \cos(\omega_t T)\end{bmatrix} \tag{9}$$

V. Multi-Sensor Framework

In order to compare the performance between single sensor pose estimation and multi-sensor pose estimation. According to the measurement's reference frame, we category sensors into local and global types [4].

- Local sensors: Camera, radar, IMU (accelerometer and gyroscope), etc. Such sensors are not globally referenced, so a reference system is usually required. Generally, the first state of the vehicle is set as the origin to activate the sensor.
- Global sensors: including GPS, magnetometer, barometer, etc. This sensor is globally referenced. It always works under a fixed global frame, such as the earth frame, which can reflect the relative position of the frame of reference. Therefore, the origin of the corresponding reference system is fixed, and the position is known in advance. So their measurements are globally referenced, but there is noise.

A. Radar Sensor Detection Mode

The fusionRadarSensor System object can model three detection modes: monostatic, bistatic, and electronic support measures (ESM) as shown in the following figures 4. In our simulation, we only consider the monostatic detection condition.

For the monostatic detection mode, the transmitter and the receiver are collocated. In this mode, the range measurement R can be expressed as $R = R_T = R_R$, where R_T and R_R are the ranges from the transmitter to the target and from the target to the receiver, respectively. In the radar sensor, the range measurement is $R = ct/2$, where c is the speed of light and t is the total time of the signal transmission. Other than the range measurement, a monostatic sensor can also optionally report range-rate, azimuth, and elevation measurements of the target. Therefore, it is very useful as the reference localization data in our simulation.

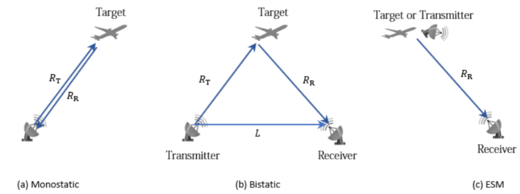

Fig. 4. three detection modes in radar system

B. GPS Sensor Detection Mode

For GPS, it measures absolute longitude, latitude and altitude with respect to the earth. The longitude, latitude, and altitude can be converted to x, y and z coordinate. For magnetometer, it measures magnetic field direction and strength, which can determine the orientation.

In our simulation, we explore the relationship between process noise and estimation accuracy in different sensors. More details about simulation can be extended in the next section.

C. multi-sensor fusion algorithm

If we have both local sensor data and global sensor data, we can get the most accurate location prediction. Therefore, it is necessary to establish a comprehensive multi-sensor prediction algorithm. In my simulation, I used the following algorithm.

Algorithm 1 Multi-Sensor Fusion Algorithm

```
Data: motion model, state space, ground truth and sensors setting
Result: pose estimation
initialization;
while iterations requirement is satisfied do
    get measurement position from radar and GPS;
    do data fusion from multi-sensors;
    if satisfy the requirement of accuracy then
        current state space updates this one;
    else
        get the average value from measurement positions;
    end
    return pose estimation;
end
```

The idea of this algorithm is mainly based on the position data of traditional sensors, which is GPS prediction. If the accuracy requirements are met, there is no need to combine radar data. However, if the accuracy requirements are not met, we judge that the traditional forecast is invalid. We need to combine the radar data to correct the final forecast. In my case, I will simply deal with it, and only take the average result.

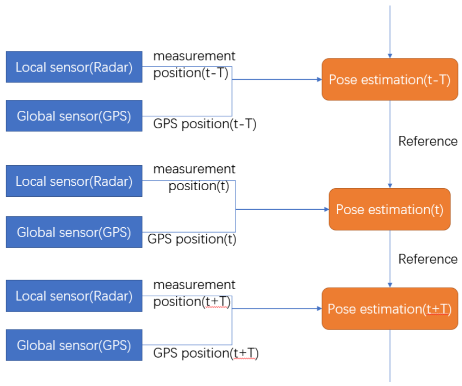

Fig. 5. Block of multi-sensors algorithm

VI. Implementation of the Proposed Approach

A. Generating the ground truth of a vehicle

In the part of generating data, I used the three common functions (constvel, costacc, constturn) in matlab toolbox. For the convenience of unified use, I use the true-state matrix to record the corresponding state changes. Here is the definition of state in Table I.

TABLE I
STATE SPACE IN DEFINITION

Focus contents	Row number	Definition
x	1	position in the coordinate x
v_x	2	velocity in the coordinate x
a_x	3	acceleration in the coordinate x
y	4	position in the coordinate y
v_y	5	velocity in the coordinate y
a_y	6	acceleration in the coordinate y
z	7	position in the coordinate z
v_z	8	velocity in the coordinate z
a_z	9	acceleration in the coordinate z
ω	10	turning rate in 2-D view

Then a trajectory is simulated, the first segment is constant velocity, the second segment is constant turning rate, and the last segment is constant accelaration. During the conversion process, according to the needs of the built-in functions, we continuously take out the state of the previous stage from the true state matrix, obtain the state of the next stage according to the corresponding state transition equation, and then record it in the matrix.

B. Collecting position data from global sensors with different process noise

Based on the ground truth of the previous stage, we simulated two sensors, and their corresponding process noise is different. Here we distinguish, one is high process noise, and the other is low process noise. Then combined with the EKF method, we get the corresponding position prediction, and then in order to facilitate the visual comparison, we use the regularization method to calculate Corresponding to the distance from the real position, and then get the following Figure 6.

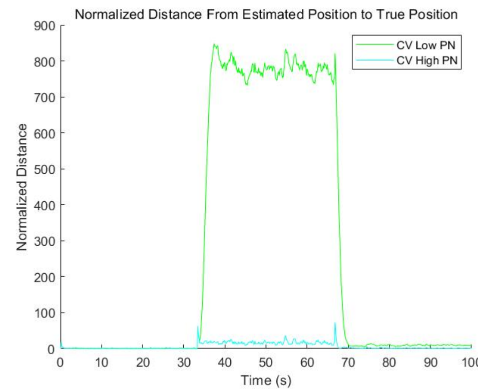

Fig. 6. Distance accuracy in Hign process noise and Low process noise

Increasing process noise will significantly improve the filter's ability to track targets during turns. However, this comes at a price: this is because the filter has a poor ability to eliminate measurement noise during uniform motion. Although the normalized distance when turning is significantly reduced, the normalized distance increases in the first 33 seconds during the constant speed of the movement.

C. Radar estimation sensor

Then we have to introduce radar prediction, which is a local sensor. In addition, in matlab, there is a built-in tool function of fusionRadarSensor, which can create parameter information of radar, and then as long as the corresponding reference state matrix is input, the corresponding predicted position can be obtained.Corresponding to the distance from the real position, and then we get the following Figure 7.

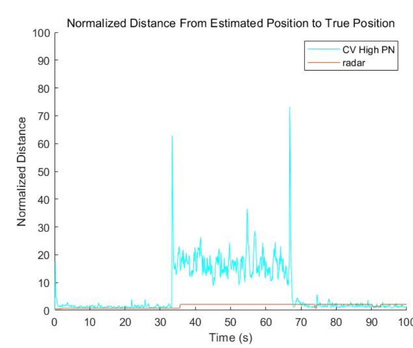

Fig. 7. Distance accuracy in Hign process noise and radar sensor

We can conclude that compared to traditional sensors with high process noise, radar sensors can provide better attitude estimation in the cornering field, but perform poorly in the field of constant acceleration.

So for our simulation, I choose the radar data as the measurement position of turning filed, and then use the comprehensive sensor data in other fileds if needed.

D. Interacting Motion-Model Filter

Combining the pros and cons of different filters, one solution is to use a filter that can consider all motion models at the same time, which is called an interactive multiple model (IMM) filter in matlab. The IMM filter can maintain as many motion models as needed. For this example, three models are sufficient: a constant speed model, a constant turn model, and a constant acceleration model.

With similar simulation, we can get the estimation of IMM in Figure 8

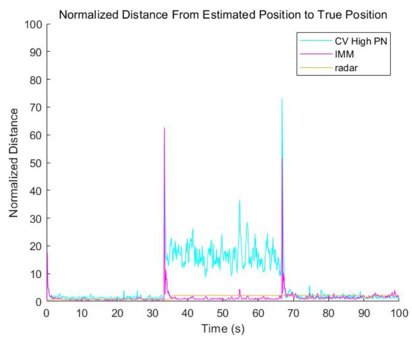

Fig. 8. Distance accuracy in Hign process noise, IMM filter and radar sensor

Under ideal conditions, this result fits the radar data very well. It can be seen that the performance of the radar sensor is very close to that of the IMM filter.

E. Final estimation based on multi-sensor algorithm

After running our multi-sensor algorithm with same procedure process, we show the difference between final estimation trajectory and true position. Here is the Figure 9

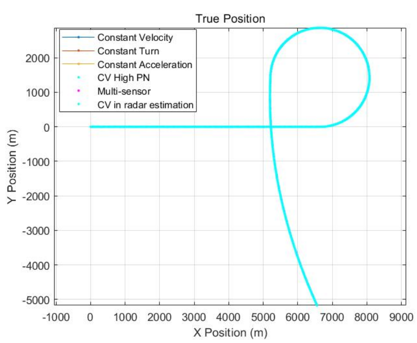

Fig. 9. Difference between final estimation trajectory and true position

VII. Conclusions

We propose a geometric map-assisted positioning algorithm, which uses the estimated trajectories from multiple sensors to obtain a position estimate. In addition, we propose a framework that combines position measurement information from VO [2] and angle measurement information. Experiments show that, in the case of unconstrained, compared with traditional GPS sensors, the positioning error of this algorithm has been significantly reduced. Currently, we are conducting more experiments on our data set, and future work involves the application of localization methods in image sensors [11].

References

[1] Hoffmann, Gabriel M and Tomlin, Claire J and Montemerlo, Michael and Thrun, Sebastian,"Autonomous automobile trajectory tracking for off-road driving: Controller design, experimental validation and racing"

[2] Wang, Han and Jiang, Rui and Zhang, Handuo and Ge, Shuzhi,"Heading Reference-Assisted Pose Estimation for Ground Vehicles"

[3] Snider, Jarrod M and others,"Automatic steering methods for autonomous automobile path tracking"

[4] Qin, Tong and Cao, Shaozu and Pan, Jie and Shen, Shaojie,"A general optimization-based framework for global pose estimation with multiple sensors"

[5] Jiang, Rui and Yang, Shuai and Ge, Shuzhi and Wang, Han and Lee, Tong,"Geometric Map-Assisted Localization for Mobile Robots Based on Uniform-Gaussian Distribution"

[6] Kocic, Jelena and Jovičić, Nenad and Drndarevic, Vujo,"Sensors and Sensor Fusion in Autonomous Vehicles"

[7] Lex Fridman,"Human-Centered Autonomous Vehicle Systems: Principles of Effective Shared Autonomy"

[8] Steinbaeck, Josef and Steger, Christian and Holweg, Gerald and Druml, Norbert,"Next Generation Radar Sensors in Automotive Sensor Fusion Systems"

[9] Schubert, Robin and Richter, Eric and Wanielik, Gerd,"Comparison and evaluation of advanced motion models for vehicle tracking"

[10] Yuan, Xianghui and Han, Feng and Han, Chongzhao,"Models and Algorithms for Tracking Target with Coordinated Turn Motion"

[11] Shaukat, Nabil and Ali, Ahmed and Javed Iqbal, Muhammad and Moinuddin, Muhammad and Otero, Pablo,"Multi-Sensor Fusion for Underwater Vehicle Localization by Augmentation of RBF Neural Network and Error-State Kalman Filter"