**Multi-sensor fusion based on Kalman filter**

Chenxin Yu, Yao Fang

**Abstract**

The inertial measurement unit (IMU) is a device that measures the three-axis attitude angle (or angular rate) and acceleration of an object. Integrating the measurements from the IMU can be used to estimate the pose of the robot. The integration of the IMU in a short time is more accurate, but the integration of a long time will have a large cumulative error. Therefore, we are going to use the UAV kinematics equation and the IMU measurement equation to perform extended Kalman filter fusion to estimate the robot pose to reduce the cumulative error of IMU measurement. Experiments have proved that the EKF algorithm does reduce the cumulative error of the pose, but since the equation of motion is also in a recursive form, there will be small errors in the long-term operation, which need to be eliminated by the measurement of other sensors.

**Index Terms**

Estimation, EKF

I. INTRODUCTION

With the development of unmanned system, a UAV can fly throught a complex unknown environment only using the sensor and computer equipped on it. Pose estimation is one of the most important capabilities for unmanned system. IMU is widely used for pose estimation because of its capability of measuring acceleration and body rates of the robot. However, a large cumulative error exsits when we have to estimate robot’s pose for a long time. So, some works fuse IMU and camera or lidar data to eliminate cumulative error. But these methods can not be used for robot with poor computation resource. In this report, we use control input to predict the pose of the robot and data from IMU is regraded as measurement, and EKF is used to fuse these two kind of data.

II. IMU MEASUREMENT MODEL AND POSE ESTIMATION

This part introduces the measurement model of IMU and pose estimation using IMU alone.

A. IMU Measurement Model

IMU measures the angular velocity and the acceleration in the body coordinate system, as shown in the following formula

Where, the subscript m represents the measured value. The subscript w represents the world coordinate system, the subscript b represents the body coordinate system, and represents the vector converted from the world coordinate system to the body coordinate system. b represents the IMU measurement bias, and n represents the IMU measurement noise.

B. IMU integral estimation pose

Using the median integral in the discrete case, the state of the robot can be obtained: position p, velocity v, rotation q, as shown in the following formula:

where,

where, represents the translation at time k in world coordinate, k represents the velocity at time k, represents the attitude at time k, and t represents the time interval. Among them, we use quaternion representation for rotation. The subscript mid represents the calculated value of the median integral. We take the mean value of the measured values at the two moments as the measured value at the current moment and perform integration to roughly obtain the position, speed, and rotation of the robot.

III. PROCESS

We use quaternions as the rotation state of the drone:

The rotational equation is given:

where is input body rates. Use the first-order Ryuga Kuta method and we can obtain

The translation stage is set as

where are the position and velocity of the drone, respectly. The translation equation is given:

where is the third column of the rotation matrix of the drone, is input acceleration along and g is the gravitational acceleration.

IV. EKF PART

We use an extended Kalman filter to fuse the kinematic equation predictions with the IMU measurements

A. Predition Part

We can get predition equation from equation(3).

B. Measurement Part

We get the acceleration measurement equation:

Since the state is set to a quaternion, we convert the quaternion to a rotation matrix representation as shown in the following formula:

where, the rotation matrix is an orthogonal matrix, which means the inverse of the rotation matrix is its transpose. So we have:

After obtaining the rotation matrix, we assume that the current acceleration is very small, and we can regard the acceleration as the noise of a small disturbance. The IMU measures the acceleration of gravity in the body coordinate system, as shown in the following formula:

In this formula, the partial derivative is calculated for the quaternion state variable, and the following formula can be obtained:

This is the Jacobianx of the linearization of the measurement equation, which needs to be used in the subsequent extended Kalman filter algorithm.

C. Update Part

First, calculate the Kalman gain:

V. THROTTLE NORMALIZATION FACTOR ESTIMATIONThe throttle value entered for the drone in airsim is a number between 0 and 1, but we can only get the desire acceleration along from the trajectory. So we have to estimate the throttle normalization factor to map desire acceleration to between 0 and 1.

A. Predition and Measurement Part

Assume a linear relationship between the throttle value and the actual acceleration along :

where is the throttle normalization factor, is the acceleration along get from imu and tcmd is the throttle value entered for the drone.

B. Update Part

We set to be constant and then we have

VI. APRILTTAG MEASUREMENT MODEL AND POSE ESTIMATION

The ApriltTag fiducial marking system can be used for a variety of tasks, including augmented reality, robotics and camera calibration. ApriltTag detection calculates the precise 3D position, orientation and ID of the calibration plate relative to the camera. The system is implemented in C language and does not rely on external library data. It can be transplanted to embedded devices. It is also relatively simple to call ros and can be used as an independent node to send and receive external topic data.

Several features of the AprilTag open source library are as follows:

1. Based on C;

2. Does not rely on other third-party libraries;

3. BDS is open source.

AprilTag is similar in concept to QR code, both are a type of two-dimensional barcode. However, AprilTag is designed to encode a smaller data payload, typically between 4 and 12 bits, which makes them more reliable during detection and able to be accurately detected from longer ranges. In addition, AprilTag is specially designed for high positioning accuracy, and users can calculate the precise three-dimensional position of AprilTag relative to the camera. This gives AprilTag a great advantage in many positioning and tracking applications. Its family (AprilTag code type) is shown in fig1:

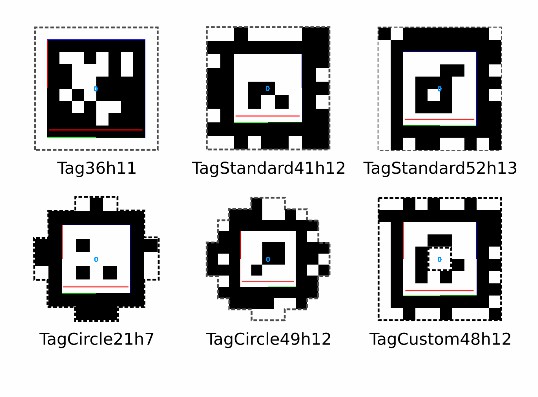


fig1: AprilTag family

The recognition process of Apriltag mainly includes two steps: detection and decoding. First, in the detection stage, Apriltag binarizes the input image to separate the black and white pixels in the image to form a binary image. Apriltag then analyzes each pixel in the image, looking for the intersection between black and white pixels to determine the position and orientation information of the QR code. In the decoding stage, Apriltag decodes the detected QR code. Through the decoding process, Apriltag converts the encoded information in the QR code into corresponding numbers or characters to obtain the actual meaning or data contained in the QR code.

The use of Apriltag can achieve efficient detection and recognition of target QR codes in images, providing basic data for subsequent applications. Through this technology, it can be widely used in fields such as robot navigation, augmented reality, object tracking, etc., providing reliable visual positioning and recognition capabilities for related research and applications.

Assume that the coordinates are, and the quaternion , then represents the coordinates of the AprilTag code center in the camera reference system, represents the rotation of the camera reference system to the AprilTag code reference system.

In fact, in this experiment, the AprilTag code remained stationary as a foundation, so this article needs to reversely deduce the coordinates of the camera in the AprilTag code reference system.

For this purpose, assume that the rotation matrix from the camera reference system to the AprilTag code reference system is , the coordinates of the camera under the AprilTag code reference are, and the direction vector is .

Thus:

Thus

In this formula, the partial derivative is calculated for the quaternion state variable, and the following formula can be obtained:

This is the Jacobianx of the linearization of the measurement equation, which needs to be used in the subsequent extended Kalman filter algorithm.

And:

VI. EXPERIMENT

The experimental platform is based on ros noetic. First, we executed a trajectory planning program and se3 controller in the px4 simulation environment to collect the UAV’s IMU data, odometer data, desired attitude angular velocity and throttle and throttle normalization coefficient. Use the rosbag function to record packets for subsequent use. Such as fig 1.

We run the program and compare the pose estimated by the EKF with the pose estimated by a separate IMU. It can be seen that the pose estimated by the EKF is not much different from the real pose when running for 20s, while the pose estimated by the IMU The pose has deviated far from the true pose due to accumulated errors, such as fig 2.

The figure below shows the comparison between the rotation attitude estimated by the EKF and the rotation attitude estimated by the IMU within 120s. It can be seen from the figure 3 and 4 that the cumulative error of the rotation attitude estimated by the IMU is gradually increasing, while the cumulative error of the rotation attitude estimated by the EKF The error drops to about 7° in about 40s. However, since the equation of motion is also in a recursive form, and in the case of ignoring the bias measured by the IMU, the rotation attitude estimated by the EKF method also has cumulative errors. To completely eliminate the cumulative error, other sensors should be added to assist in the measurement, or loopback detection should be performed to eliminate the cumulative error. It is more appropriate to use this method to estimate the state of the robot in a short period of time.

VII. WORK ALLOCATION

A. Chenxin Yu

• Data process for IMU

• Formula derivation and code for IMU data

• Frame of the code

• Code for EKF

• PPT

• Report

B. Yao Fang

• Formula derivation and code for dynamics of UAV

• Estimation of throttle normalization coefficient

• Data clooection and visualization

• Code for EKF

• PPT

• Report