

К ВОПРОСУ НАХОЖДЕНИЯ ОРИЕНТАЦИИ И ТЕКУЩЕЙ ПОЗИЦИИ ОБЪЕКТА В ПРОСТРАНСТВЕ НА ПРИМЕРЕ СОМАТОСЕНСОРНОЙ ПЕРЧАТКИ

аспирант Старокожев С.В., доц., к.т.н. Свид И.В.

Харьковский национальный университет радиоэлектроники,
кафедра микропроцессорных технологий и систем,
e-mail: sviatoslav.starokozhev@nure.ua

Abstract. The paper considers the problem of finding the orientation and current position of an object in space using the example of a somatosensory glove. A variant of finding the spatial position of an object is also presented and a variant of visualization in 3D using hardware and software is proposed.

Введение. Задача нахождения ориентации и текущей позиции объекта в пространстве на примере соматосенсорной перчатки может быть актуальна для управления игровыми объектами, малыми БПЛА, образовательные проекты и многое другое.

Используя соматосенсорную перчатку на базе Arduino UNO R3, к которой подключены инерциальные датчики акселерометр и гироскоп, можно найти текущее положение объекта и его пространственную ориентацию. Были использованы методы калмановской фильтрации с учетом дрейфа нуля для калибровки датчиков. Данный проект был реализован в среде математического моделирования MATLAB и имеет хорошие перспективы для дальнейшего развития.

Основная часть.

Для оценки пути пройденного телом используется датчик акселерометра MPU6050 в соматосенсорной перчатке при следующих допущениях [1, 2]:

- для того, что бы найти путь необходимо дважды проинтегрировать ускорение, что измерено акселерометром [2].

- точно так же производная функции положения по времени является функцией скорости;

- исходя из предыдущего допущения, можно использовать одни и те же математические преобразования для нахождения скорости и перемещения.

После математических преобразований получаем следующее выражение для нахождения скорости и перемещения:

$$x(t) = x_0 + v_0 t + \frac{1}{2} a t^2, \quad (1)$$

где $v_0 = (0)$ – начальная скорость; a – константа.

Для определения углов наклона устройства в пространстве будут использоваться такие программные среды как MATLAB и Arduino Studio.

В среде моделирования MATLAB, пакет Sensor Fusion and Tracking Toolbox™ позволяет объединить данные, считанные с инерциального измерительного блока (IMU), для оценки ориентации и угловой скорости:

Можно использовать разные связки инерционных датчиков: `escompass` – объединение показаний акселерометра и магнитометра; `imufilter` – объединение показания акселерометра и гироскопа; `ahrsfilter` – объединение показаний акселерометра, гироскопа и магнитометра.

Больше датчиков на IMU приводит к более надежной оценке ориентации. Данные датчиков могут быть проверены перекрестно, а информация, которую передают датчики, является ортогональной [1].

Используя 6-осевые алгоритмы слияния для вычисления ориентации можно описать ориентацию объекта описывая его вращение относительно некоторой системы координат, иногда называемой родительской системой координат, в трех измерениях.

Для оценки ориентации обычно используются три типа датчиков: акселерометры, гироскопы и магнитометры. Акселерометры измеряют правильное ускорение. Гироскопы измеряют угловую скорость. Магнитометры измеряют местное магнитное поле. Различные алгоритмы используются для объединения различных комбинаций датчиков для оценки ориентации.

Данные датчика акселерометра, гироскопа считываются в реальном времени и выводятся на график, пока устройство вращается вокруг трех разных осей.

На рис. 1 показана работа программы с нахождением ориентации датчика в пространстве и показания инерциальных датчиков при использовании фильтра Каллмана [3].

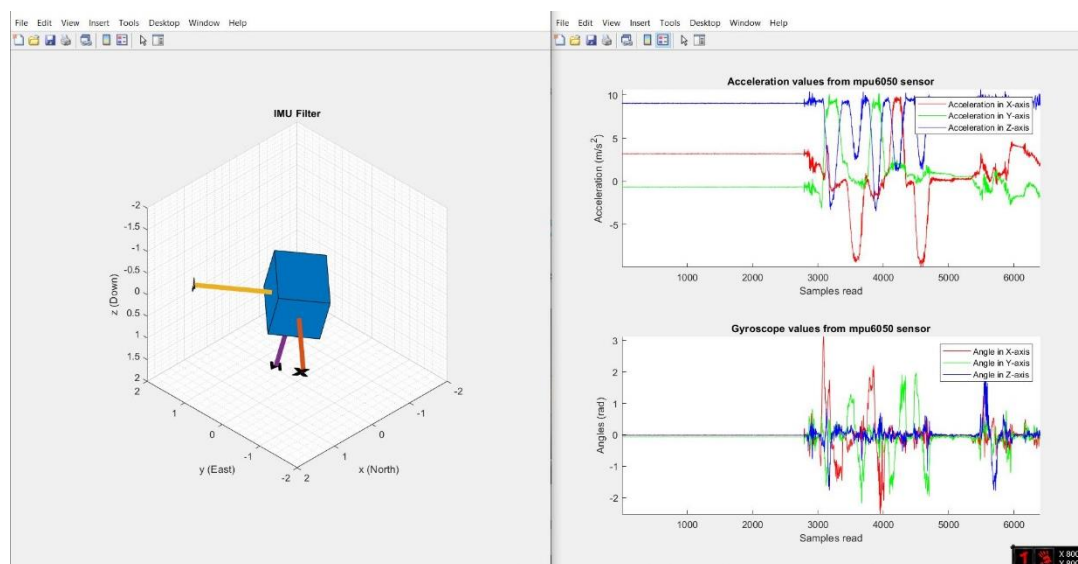


Рис. 1 – Нахождение ориентации датчика в пространстве и показания инерциальных датчиков, акселерометра и гироскопа

Есть несколько проблем, которые нужно решить для правильной оценки пройденного пути, используя показания ускорений с акселерометра [2]: показания датчика акселерометра «сырые» и их нужно преобразовать в

нормальное представление в m/s^2 ; когда датчик наклонен просто по одной из осей, то он показывает ускорение свободного падения на этой оси; у датчика есть «дрейф нуля» и необходимо его вычитать, что бы показания были правдоподобными; во время прекращения движения тело как бы не движется, так как при равномерной скорости ускорение нулевое [4]. Поэтому в данном опыте введено такое допущение, что при нулевом ускорении тело не движется, а значит, пройденный текущий путь будет нулевым [5].

При испытании использовались неоткалиброванные и откалиброванные датчики, а также фильтрация полученных результатов проводилась фильтром Каллмана [6]. Также был измерить дрейф нуля и дисперсия ошибки датчиков для корректной настройки фильтра Каллмана. Датчик калибровался простым методом. Для этого нужно соматосенсорную перчатку с датчиком положить на ровную поверхность включить и подождать 30 секунд, пока запишутся средние значения ошибки по всем трем осям ориентации, потом устройством можно пользоваться.

Выводы. В ходе данной работы показано, что можно определить ориентацию объекта и его положение с помощью инерциальных датчиков, таких как акселерометр и гироскоп. Показания их будут намного точнее, если их откалибровать перед использованием. Фильтрация фильтром Кальмана дает более плавную отрисовку и позиционирование. Нарботки этого проекта можно использовать как для дополненной реальности, так и для работы с БПЛА.

Перечень использованных ссылок.

1. І.І. Обод, І.В. Свид, І.В. Рубан, Г.Е. Заволодько. Математичне моделювання інформаційних систем: навчальний посібник. – Харків : Друкарня Мадрид, 2019. – 270 с.

2. Determine Orientation Using Inertial Sensors. [Access mode]: <https://ch.mathworks.com/help/fusion/gs/determine-orientation-through-sensor-fusion.html>.

3. Finding Velocity and Displacement from Acceleration. [Access mode]: <https://openstax.org/books/university-physics-volume-1/pages/3-6-finding-velocity-and-displacement-from-acceleration>.

4. Pose Estimation From Asynchronous Sensors. [Access mode]: <https://ch.mathworks.com/help/fusion/ug/pose-estimation-from-asynchronous-sensors.html>.

5. EstimateDistanceIntegration.m demo example. [Access mode]: <https://www.mathworks.com/matlabcentral/mlc-downloads/downloads/submissions/53872/versions/4/previews/help/html/Example/estimateDistanceIntegration.html#5>

6. Definite and indefinite integrals. [Access mode]: https://ch.mathworks.com/help/symbolic/sym.int.html?s_tid=srchtitle.

7. Estimating Orientation Using Inertial Sensor Fusion and MPU-9250. [Access mode]: https://ch.mathworks.com/help/fusion/ug/Estimating-Orientation-Using-Inertial-Sensor-Fusion-and-MPU-9250.html?s_eid=PSM_15028#EstimateOrientationUsingInertialSensorFusionAndMPU9250Example-9.