

Программное обеспечение оптико-электронной системы межсамолетной навигации беспилотного летательного аппарата

В. Г. Бондарев, email: dimkaao@yandex.ru

Д. В. Лопаткин, email: dimkaao@yandex.ru

О. Н. Роговенко, email: dimkaao@yandex.ru

М. А. Чернышов, email: mixail.chernyshov.00@mail.ru

Военный учебно-научный центр Военно-воздушных сил «Военно-воздушная академия имени профессора Н.Е. Жуковского и Ю.А. Гагарина»

***Аннотация.** В работе рассмотрены методы, подходы и средства, используемые для разработки оптико-электронной системы межсамолетной навигации, в том числе, алгоритм работы системы технического зрения, обработки изображения, передачи данных в вычислитель и отображение в графическом интерфейсе оператора.*

***Ключевые слова:** Система технического зрения, обработка изображения, вычисление координат, передача данных через websocket соединение, использование библиотеки OpenCV, графический интерфейс.*

Введение

Неоспоримым фактом является то, что на сегодняшний день тактика группового применения беспилотных летательных аппаратов (БпЛА) в различных видах и формах строя вызывает к себе все больший интерес и способствует бурному развитию методов применения, а также прогрессу в сфере технических средств, обеспечивающих данный режим полета [1]. Ключевым средством, позволяющим реализовать данную идею является система межсамолетной навигации, от выбора которой во многом будет зависеть эффективность действий БпЛА при выполнении поставленных задач. Поэтому в ходе анализа существующих систем межсамолетной навигации и оценки их результативности в различных условиях и обстановке, рассмотрев все достоинства и недостатки, был предложен способ использования оптико-электронной системы межсамолетной навигации.

Основным её достоинством, особенно в сравнении с радиотехническими аналогами, является полная нечувствительность к средствам радиоэлектронной борьбы, что особенно актуально в

современных военных конфликтах, учитывая высокие темпы развития этих средств [2]. Также стоит отметить еще немаловажное достоинство, а именно, высокую точность определения относительных координат, которая позволяет предупреждать столкновения между БПЛА, обеспечивать полет в различных конфигурация строя, в том числе и в сомкнутом, а также изменять их положение в зависимости от сложившейся ситуации [3]. При всех перечисленных достоинствах опико-электронной системы существует и одна особенность. Работая в трудных метеорологических условиях могут возникнуть затруднения, обусловленные возникновением помех из-за густого тумана, дождя, обильного потока снега. Решение данной проблемы, связано с выбором диапазона излучения, в котором работает система, имеющим окно прозрачности и работой в импульсном режиме.

1. Алгоритм работы опико-электронной системы

Алгоритм функционирования опико-электронной системы межсамолетной навигации представлен следующей последовательностью действий [4]:

1. Обработка изображений ведомого БПЛА и определение координат изображений навигационных ИК-меток на фоточувствительной матрице цифровой фотокамеры ведомого БПЛА.

2. Вычисление координат ведомого относительно ведущего.

3. Вычисление отклонения текущих относительных координат от заданных.

Вычисление относительных координат происходит так: сначала находится нулевое приближение и рассчитывается первое приближение. Затем происходит вычисление Якобиана и обратной ему матрицы, а на основе дельты этой обратной матрицы можно получить разность между нулевым и первым приближениями.

Программная реализация данного алгоритма выглядит представлена на рисунках 1 и 2.

```
// вычисление нулевых приближения
const double x0 = ((l21 - l22) * Fo) / sqrt( pow( (S1.y - S2.y, 2) + pow( (S1.x - S2.x, 2) );
const double y0 = (x0 * (x0 - (lx3-lx2)) * (S2.y - S3.y)) / ((lx3-lx2) * Fo);
const double z0 = ((-x0 * (-x0 + lx3) * (S2.x * ((l23 / l22)) - S3.x)) / (Fo * lx3)) + l23;
const double gamma0 = atan( (S1.y - S2.y) / (S1.x - S2.x));
const double psi0 = 0;
const double epsilon0 = 0;
```

Рис. 1. Реализация вычисления нулевых приближений на языке программирования C++

```

double Farray[6][6] = {
{
    (f1(x0 + dx, y0, z0, gamma0, psi0, epsilon0) - f10) / dx,
    (f2(x0 + dx, y0, z0, gamma0, psi0, epsilon0) - f20) / dx,
    (f3(x0 + dx, y0, z0, gamma0, psi0, epsilon0) - f30) / dx,
    (f4(x0 + dx, y0, z0, gamma0, psi0, epsilon0) - f40) / dx,
    (f5(x0 + dx, y0, z0, gamma0, psi0, epsilon0) - f50) / dx,
    (f6(x0 + dx, y0, z0, gamma0, psi0, epsilon0) - f60) / dx
},
{
    (f1(x0, y0 + dx, z0, gamma0, psi0, epsilon0) - f10) / dx,
    (f2(x0, y0 + dx, z0, gamma0, psi0, epsilon0) - f20) / dx,
    (f3(x0, y0 + dx, z0, gamma0, psi0, epsilon0) - f30) / dx,
    (f4(x0, y0 + dx, z0, gamma0, psi0, epsilon0) - f40) / dx,
    (f5(x0, y0 + dx, z0, gamma0, psi0, epsilon0) - f50) / dx,
    (f6(x0, y0 + dx, z0, gamma0, psi0, epsilon0) - f60) / dx
},
},
{
    (f1(x0, y0, z0 + dx, gamma0, psi0, epsilon0) - f10) / dx,
    (f2(x0, y0, z0 + dx, gamma0, psi0, epsilon0) - f20) / dx,
    (f3(x0, y0, z0 + dx, gamma0, psi0, epsilon0) - f30) / dx,
    (f4(x0, y0, z0 + dx, gamma0, psi0, epsilon0) - f40) / dx,
    (f5(x0, y0, z0 + dx, gamma0, psi0, epsilon0) - f50) / dx,
    (f6(x0, y0, z0 + dx, gamma0, psi0, epsilon0) - f60) / dx
},
},
},
{
    (f1(x0, y0, z0, gamma0 + dx, psi0, epsilon0) - f10) / dx,
    (f2(x0, y0, z0, gamma0 + dx, psi0, epsilon0) - f20) / dx,
    (f3(x0, y0, z0, gamma0 + dx, psi0, epsilon0) - f30) / dx,
    (f4(x0, y0, z0, gamma0 + dx, psi0, epsilon0) - f40) / dx,
    (f5(x0, y0, z0, gamma0 + dx, psi0, epsilon0) - f50) / dx,
    (f6(x0, y0, z0, gamma0 + dx, psi0, epsilon0) - f60) / dx
},
{
    (f1(x0, y0, z0, gamma0, psi0 + dx, epsilon0) - f10) / dx,
    (f2(x0, y0, z0, gamma0, psi0 + dx, epsilon0) - f20) / dx,
    (f3(x0, y0, z0, gamma0, psi0 + dx, epsilon0) - f30) / dx,
    (f4(x0, y0, z0, gamma0, psi0 + dx, epsilon0) - f40) / dx,
    (f5(x0, y0, z0, gamma0, psi0 + dx, epsilon0) - f50) / dx,
    (f6(x0, y0, z0, gamma0, psi0 + dx, epsilon0) - f60) / dx
},
},
{
    (f1(x0, y0, z0, gamma0, psi0, epsilon0 + dx) - f10) / dx,
    (f2(x0, y0, z0, gamma0, psi0, epsilon0 + dx) - f20) / dx,
    (f3(x0, y0, z0, gamma0, psi0, epsilon0 + dx) - f30) / dx,
    (f4(x0, y0, z0, gamma0, psi0, epsilon0 + dx) - f40) / dx,
    (f5(x0, y0, z0, gamma0, psi0, epsilon0 + dx) - f50) / dx,
    (f6(x0, y0, z0, gamma0, psi0, epsilon0 + dx) - f60) / dx
},
},
},
};

```

Рис. 2. Вычисление матрицы Якоби (Ф) на языке программирования C++

2. Разработка программного обеспечения

В ходе разработки программного обеспечения были выдвинуты следующие требования [5]:

- обеспечение контроля использования памяти в программе;
- обеспечение контроля использования вычислительных ресурсов процессора;
- обеспечение непосредственного контроля устройства видекамеры;
- обеспечение гибкости в использовании повторяющихся архитектурных конструкций программирования.

Данный алгоритм программно реализован с помощью следующих средств разработки:

1. Язык программирования C++.
2. Среда разработки CLion.
3. Система сборки Cmake.
4. Система контроля версий git.
5. Язык программирования TypeScript.

Общая структура программы, включает два основных модуля:

1. Модуль вычисления относительных координат ведомого БПЛА.
2. Модуль графического интерфейса оператора.

Все операции, связанные с захватом и обработкой изображения, были выполнены с помощью функций и методов, взятых из открытой библиотеки компьютерного зрения OpenCV. Набор возможностей данной библиотеки достаточно обширный и позволяет решать огромный перечень задач, связанных с обработкой изображения.

Веб интерфейс представляет собой веб-приложение, написанное на языке TypeScript расширяющем возможности языка javascript, отличительной особенностью которого является поддержка статической типизации, которая позволяет проверять правильность типа во время компиляции, а также его объектно-ориентированная направленность. Большая часть графической части была реализована при помощи библиотеки „react“, предназначенной для разработки веб приложений и предоставляющей обширный инструментарий для их разработки, в частности базовые функции многих стандартных элементов пользовательского интерфейса, что дает гибкость в работе. Для создания отдельных элементов интерфейса использовался фреймворк „Material UI“.

Получение данных из модуля вычисления координат в пользовательский интерфейс происходит путем передачи видеопотока через „websocket“ соединение, а значения вычисленных координат через „http api“. В результате система получает данные в реальном времени и практически не имеет временных задержек.

Отдельное внимание стоит обратить на использование в программе цветовой модели HSV суть которой заключается в нелинейном преобразовании RGB модели. Как можно заметить в пользовательском интерфейсе есть слайдеры изменяющие ее параметры. Параметрами же этой модели являются:

- H („hue“ — цветовой фон), который варьируется в пределах 0...360;
- S („saturation“ — насыщенность) варьируется в пределах 0...100 или 0...1. Чем больше этот параметр, тем «чище этот цвет» поэтому к нему также применимо название «чистота света»;
- V („value“ — значение цвета или „brightness“ — яркость.), который так же задается в пределах 0...100 и 0...1.

Изменение значений этих параметров влияет на распознавание системой технического зрения ИК-меток. При этом если использовать интерференционный фильтр, представленный на рисунке 3, изменение параметров цветового фона и насыщенность не требуется.

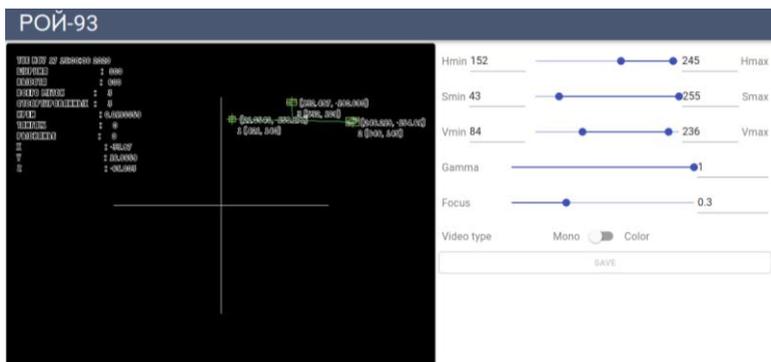


Рис. 3. Панель графического интерфейса оператора

Заключение

Рассмотрев все особенности использования оптико-электронной системы межсамолетной навигации, можно сделать вывод, что значительное повышение точности навигации, возможности, предоставляемые для группового полета строя БпЛА, полная нечувствительность к средствам радиоэлектронной борьбы и низкие экономические затраты сократят трудозатраты на ее производство и позиционируют данную систему как оптимальный выбор по соотношению цены и качества производителю.

Список литературы

1. Новичков Н.Н. Российское оружие в Сирии: анализ, итоги. М.: ООО «Статус», 2018. 336 с.
2. Приказчиков А.В., Оганесян А.А., Пастухов А.В. Помехоустойчивая аппаратура ГЛОНАСС/GPS промышленного назначения. Аэрокосмический курьер. №1 2013, С. 2-3.
3. Верба В.С. Комплексы с беспилотными летательными аппаратами. Кн. 1. Принципы построения и особенности применения комплексов с БЛА. Монография. – М.: Радиотехника, – 2016. – 512 с.
4. Пат. 2626017 Российская Федерация, МПК51 G 01S 13/46. Способ навигации подвижного объекта [Текст] / Кудавев А.Н., Косенко А.А., Бондарев В.Г., Ипполитов С.В., Озеров Е.В., Лопаткин Д.В. (РФ); заявители и патентообладатели ВУНЦ ВВС «ВВА». – № 2016130484; заявл. 25.07.16. – 7 с.
5. Молчалов А.С. Иконические системы воздушной разведки. Основы построения, оценка качества и их применение в комплексах с БЛА / Молчалов А.С., Абрамов Д.В. – Волгоград: Панорама, – 2017. С. 45-62.

6.