

ВНУТРЕННИЙ SLAM ДЛЯ УПРАВЛЕНИЯ МИКРО ЛЕТАТЕЛЬНЫМИ АППАРАТАМИ С ИСПОЛЬЗОВАНИЕМ МОНОКУЛЯРНОЙ КАМЕРЫ И SENSOR FUSION

Горшков С.Н., специалист, старший оператор 4 научной роты ФГАУ «Военный инновационный технополис «ЭРА», г. Анапа, Российская Федерация, e-mail: cngorshkov@mail.ru, тел.: 89097633050,

Прокофьев М.А., магистр, старший оператор 4 научной роты ФГАУ «Военный инновационный технополис «ЭРА», г. Анапа, Российская Федерация, e-mail: maprokofev@mail.ru, тел.: 89663135366,

Аничин А.М., бакалавр, оператор 5 научной роты ФГАУ «Военный инновационный технополис «ЭРА», г. Анапа, Российская Федерация, e-mail: amanichin@yandex.ru, тел.: 89531946794,

Колузов А.В., младший научный сотрудник, командир 1 взвода 4 научной роты ФГАУ «Военный инновационный технополис «ЭРА», г. Анапа, Российская Федерация,

Патрушев С.М., начальник 3 лаборатории испытательной (робототехники) ФГАУ «Военный инновационный технополис «ЭРА», г. Анапа, Российская Федерация.

Аннотация

Данная статья представляет собой исследование, проводимое в Simultaneous Localization and Mapping (SLAM) для Микро Летательных Аппаратов (MAV) в контексте навигационных задач по спасению и/или распознаванию в помещении. В такого рода приложениях MAV должны полагаться на собственные встроенные датчики для автономной навигации в неизвестных, враждебных условиях и в условиях отсутствия GPS, таких как разрушенные или полуразрушенные здания. Цель данной статьи – исследовать методику SLAM, которая объединяет визуальную информацию и измерения от инерциального измерительного блока (IMU), чтобы надежно получить оценку позиционирования 6DOF MAV в локальной карте окружающей среды [1,2,3]. Монокулярный визуальный алгоритм SLAM вместе с IMU вычисляют позицию с помощью расширенного фильтра Калмана (EKF). Система состоит из недорогого коммерческого беспилотника и пульта дистанционного управления, который предоставит вычислительные возможности для алгоритмов SLAM с использованием системы распределенных узлов на основе ROS (Robot Operating System). Некоторые экспериментальные результаты показывают, как Sensor Fusion улучшает оценку положения и полученную карту в различных условиях испытаний.

Ключевые слова: военный инновационный технополис «ЭРА»; микро летательные аппараты; внутренняя навигация; sensor fusion; одновременная локализация и картографирование; ROS.

Микро летательные аппараты (MAV) широко использовались в различных областях, от военных до гражданских, включая операции по наблюдению, наблюдению за погодой, оказанию помощи при бедствиях и гражданский инженерный надзор. Благодаря инерциальным датчикам GPS и MEMS были разработаны MAV, которые демонстрируют впечатляющий набор возможностей в окружающей среде. К сожалению, большинство внутренних помещений остаются без доступа к внешним системам позиционирования, и автономные MAV очень ограничены в своей способности работать в этих областях.

Двумя основными проблемами внутренней навигации MAV являются отказ в приеме сигнала GPS и ограничения внутренних воздушных платформ. В отличие от обычной навигации на основе GPS / INS, в которой глобальное положение и скорость MAV получаются непосредственно, навигация в помещении должна получать эту информацию с помощью сложных алгоритмов, основанных на относительном зондировании. Кроме того, внутренние MAV обычно проектируются так, чтобы быть небольшими и иметь очень ограниченную полезную нагрузку, и это приводит к ограниченной вычислительной мощности на борту, что делает алгоритмы еще сложнее для реализации.

В частности, определение положения важно для многих задач навигации, включая локализацию, картографирование и управление. Используемая техника в основном зависит от имеющихся на борту датчиков, которые в воздушной навигации должны тщательно выбираться из-за ограничений полезной нагрузки. Благодаря небольшому весу и расходу, большинство коммерческих MAV включают, по крайней мере, одну монокулярную камеру, поэтому методы VSLAM (Visual SLAM) широко используются [4,5]. Тем не менее, большинство из этих работ были ограничены небольшими рабочими пространствами, которые имеют определенные особенности окружения и достаточно солнечного света. Кроме того, время вычислений слишком велико для быстрой динамики летательных аппаратов, что затрудняет их управление. С другой стороны, несмотря на их большой вес и расход, датчики дальности, такие как камеры RGB-D или датчики лазерной дальности, также использовались на MAV из-за их быстрого определения расстояния.

В качестве стратегии алгоритма слияния каждый из датчиков должен иметь возможность обеспечить своё собственное определение положения, чтобы наделить систему некоторой избыточностью, которая позволяет ей работать в различных условиях окружающей среды.

Для решения вычислительных требований, система состоит из летного и наземного блоков, поэтому код может быть распределен по разным узлам с помощью ROS (Robot Operating System).

Исследование, описанное в данной статье, используются два монокулярных алгоритма VSLAM для расчета позиционирования (вместе с измерениями из IMU) и карты окружающей среды: LSD-SLAM [6] и ORB-SLAM [7].

Одной из основных проблем алгоритмов монокулярной камеры VSLAM является то, что она не может рассчитать масштаб данных пройденного пути и картографирования. Это приводит к созданию системы, которая не работает с данными реального масштаба, что может повлиять на целостность летательного аппарата. Чтобы решить эту проблему, наша система использует данные из IMU для расчета динамического масштаба SLAM и возврата позиции MAV в реальном времени без неоднозначности масштаба.

Мы рассматриваем проблему автономной локализации MAV в помещении как программную проблему, уделяя особое внимание интеграции алгоритмов высокого уровня, а не конкретного оборудования. По этой причине мы используем недорогую коммерческую платформу с небольшими изменениями и платформу разработки с открытым исходным кодом (ROS), так что драйверы датчиков и некоторые алгоритмы могут использоваться без разработки.

Аппаратная архитектура. Квадрокоптер MAV, показанный на рис. 1, представляет квадрокоптер собственной сборки. Он имеет следующую комплектацию

- бортовой компьютер, обеспечивающий работу алгоритмов автономного полета – процессорный модуль NVIDIA Jetson TX2;
- полетный контроллер для обеспечения контроля за движением беспилотного летательного аппарата – Pixhawk 2;
- лазерный дальномер RPLidar A1M8;
- модуль телеметрии FrSky X8R;
- 4 ротора DJI 3510.



Рисунок 1 – Квадрокоптер для летных испытаний

Архитектура программного обеспечения. Как показано на рис. 2, встроенный контроллер и процессор выполняют показания датчиков и основное управление MAV. Наземная станция использует нашу систему SLAM, а также стратегии управления и планирования, последние из которых выходят за рамки данной статьи.

Система SLAM, описанная в этой статье, состоит из двух основных компонентов: (а) монокулярная система VSLAM, которая получает позиции 6DOF (и трехмерную карту окружающей среды); (b) расширенный фильтр Калмана, который объединяет последнюю оценку с навигационными данными, предоставленными бортовыми датчиками MAV, для получения надежной оценки 6DOF положения робота на сгенерированной карте. Кроме того, мы внедрили ПИД-регулятор, который позволяет MAV достигать поставленных целей с использованием предполагаемой позиции.

SLAM подход. В следующих подразделах мы опишем модули системы SLAM.

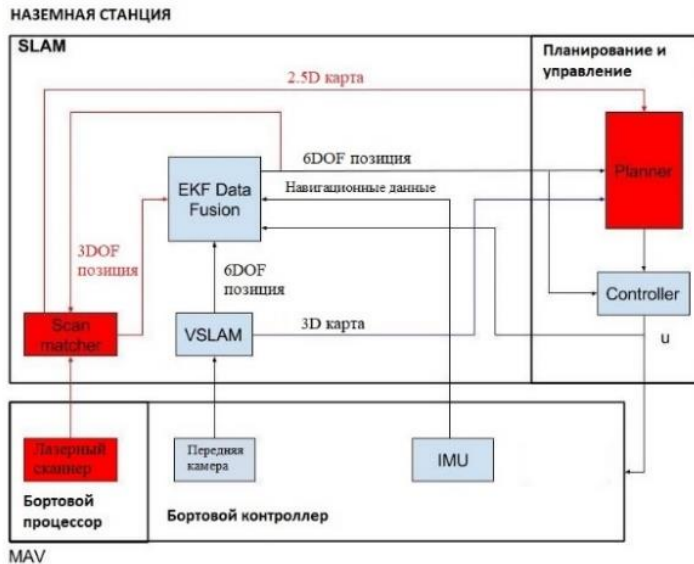


Рисунок 2 – Архитектура программного обеспечения проекта ISLAMAV (красные модули выходят за рамки данной статьи)

Монокулярный VSLAM. После изучения современных монокулярных алгоритмов VSLAM мы решили реализовать в нашей системе два таких алгоритма: LSD-SLAM (крупномасштабный прямой монокулярный SLAM) и ORB-SLAM (ориентированный FAST и повернутый BRIEF SLAM), оба доступны в виде пакетов ROS.

LSD-SLAM – это прямой (не имеющий функций) монокулярный алгоритм SLAM, который, наряду с высокоточной оценкой положения на основе прямого выравнивания изображения, восстанавливает трехмерную среду в режиме реального времени в виде графа позиций ключевых кадров с соответствующими полу-плотными картами глубины. Из-за более поздней реализации узла laser SLAM и его карты 2,5D, мы использовали только позиции 6DOF этого алгоритма в качестве входных данных для фильтра слияния данных. Мы решили использовать карту лазера вместо карты, созданной LSD-SLAM, из-за лучшей точности первой и из-за вычислительных требований, необходимых для последней.

Хотя в этом случае результаты хороши, системе необходимо большое количество визуальных характеристик, недоступных в темных зонах, где она должна быть объединена с другими датчиками. Кроме того, он очень чувствителен к чистому вращательному движению.

С другой стороны, ORB-SLAM является монокулярным SLAM на основе функций. ORB-SLAM очень точно оценивает положение дрона. Это делает его идеальным для реализации в системе на основе MAV из-за его быстрой и нестабильной динамики. Кроме того, благодаря продуманной разработке алгоритма он может быть надежно зацикленным.

Можно сделать вывод, что данные от других датчиков необходимы для правильной оценки положения MAV. Хотя трекинг в комнате и вдоль коридора правильный, он не рассчитывает угол поворота после поворота. Кроме того, изменение масштаба позволяет получить неправильную оценку расстояний (длина коридора после угла укорачивается).

На рис. 3 показаны результаты, полученные, когда алгоритм приблизительно оценивает положение камеры вокруг квадрата площадью 35 м². Алгоритм замыкания петли позволяет технике VSLAM точно отслеживать позицию камеры в реальном времени.

Как уже было сказано, одной из основных проблем при работе с монокулярным VSLAM является неоднозначность масштаба. Поскольку нам необходимо работать с системой, учитывающей масштабирование, мы разработали метод расчета масштаба на основе бортового зондирования. Благодаря этому наша система работает в реальных масштабах. Чтобы решить эту проблему, система использует измерения высоты из альтиметра и VSLAM для расчета шкалы следующим образом:

$$scale = \frac{h_{IMU}}{h_{VSLAM}} \quad (2)$$

$$x_{REAL-SCALE} = x_{VSLAM} * scale \quad (3)$$

$$y_{REAL-SCALE} = y_{VSLAM} * scale \quad (4)$$

$$z_{REAL-SCALE} = z_{VSLAM} * scale \quad (5)$$

Масштаб вычисляется на каждой итерации узла перед объединением данных, чтобы избежать проблем из-за динамических изменений.

Слияние данных с EKF. Чтобы объединить все доступные данные, мы используем расширенный фильтр Калмана (EKF). EKF также используется для компенсации различных временных задержек в системе, это подробно описано в [8], возникающих в связи с беспроводной локальной сетью и вычислительно сложным визуальным маршрутом.

EKF использует следующий вектор состояния:

$$x_t := (x_t, y_t, z_t, vx_t, vy_t, vz_t, \Phi_t, \Theta_t, \Psi_t, \dot{\Psi}_t)^T \in \mathbb{R}^{10} \quad (6)$$

где (x_t, y_t, z_t) – положение MAV в метрах (м); (vx_t, vy_t, vz_t) скорость в метрах/секунду (м/с); Φ_t, Θ_t, Ψ_t углы крена, тангажа и рыскания в радианах (рад); и $(\dot{\Psi}_t)$ скорость вращения рыскания в радианах/секунду (рад/с). Все они оцениваются в мировых координатах. Далее мы определим модели прогнозирования и наблюдения.

Модель прогнозирования. Модель прогнозирования основана на полной модели движения динамики полета квадрокоптера и реакции на команды управления.

Модель устанавливает, что горизонтальное ускорение MAV пропорционально горизонтальной силе, действующей на квадрокоптер, то есть ускоряющая сила минус сила сопротивления. Сопротивление пропорционально горизонтальной скорости квадрокоптера, а ускоряющая сила пропорциональна проекции оси z дрона на горизонтальную плоскость, что приводит к:

$$vx_t = K_1(K_2(\cos \Phi \sin \Theta \cos \Psi + \sin \Phi \sin \Psi)) \quad (7)$$

$$vy_t = K_1(K_2(\cos \Phi \sin \Theta \sin \Psi - \sin \Phi \cos \Psi)) \quad (8)$$

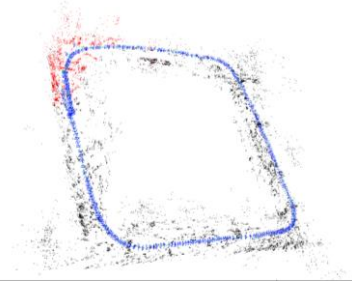


Рисунок 3 – Результаты ORB-SLAM, оценивающие положение камеры вокруг квадрата

Кроме того, влияние отправленной команды управления $u = (\hat{v}\bar{x}, \hat{v}\bar{y}, \hat{v}\bar{z}, \hat{\Psi})$ описывается следующей линейной моделью:

$$\dot{\Phi}_t = -K_3(K_4\hat{v}\bar{y}_t + \Phi_t) \quad (9)$$

$$\dot{\Theta}_t = -K_3(K_4\hat{v}\bar{y}_t + \Phi_t) \quad (9)$$

$$\dot{\Phi}_t = K_3(K_4\hat{v}\bar{x}_t + \Theta_t) \quad (10)$$

$$\dot{vz}_t = K_7(K_8\hat{v}\bar{z}_t + vz_t) \quad (11)$$

$$\dot{\Psi}_t = K_5(K_6\hat{\Psi}_t + \Psi_t) \quad (12)$$

Мы оценили пропорциональные коэффициенты от K1 до K8 по данным, собранным в серии испытательных полетов. Из уравнений (7) – (12) получаем общую функцию перехода состояний:

$$\begin{pmatrix} x \\ y \\ z \\ vx \\ vy \\ vz \\ \Phi \\ \Theta \\ \Psi \\ \dot{\Psi} \end{pmatrix}_{t+1} \leftarrow \begin{pmatrix} x \\ y \\ z \\ vx \\ vy \\ vz \\ \Phi \\ \Theta \\ \Psi \\ \dot{\Psi} \end{pmatrix}_t + \Delta_t \begin{pmatrix} vx_t \\ vy_t \\ vz_t \\ K_1(K_2(\cos \Phi \sin \Theta \cos \Psi + \sin \Phi \sin \Psi)) - vx_t \\ K_1(K_2(\cos \Phi \sin \Theta \sin \Psi - \sin \Phi \cos \Psi)) - vy_t \\ K_7(K_8\hat{v}\bar{z}_t + vz_t) \\ -K_3(K_4\hat{v}\bar{y}_t + \Phi_t) \\ K_3(K_4\hat{v}\bar{x}_t + \Theta_t) \\ \Psi_t \\ K_5(K_6\hat{\Psi}_t + \Psi_t) \end{pmatrix} \quad (13)$$

Модель инерциального навигационного наблюдения. Эта модель связывает бортовые измерения, полученные через навигационный канал квадрокоптера, описанного в разделе III.A, который мы назвали «navdata» на рис. 2, и вектор состояния. Квадрокоптер измеряет свою горизонтальную скорость $(\overline{v}dx, \overline{v}dy)$ в своей локальной системе координат, которую мы преобразуем в мировую систему координат (vx, vy) . Углы крена и тангажа, измеренные акселерометром, являются прямыми наблюдениями соответствующих переменных состояния. С другой стороны, мы различаем измерение высоты и рыскание как наблюдения соответствующих скоростей. Результирующий вектор измерения $z_{NAVDATA}$ и функция наблюдения $h_{NAVDATA}(x_t)$:

$$z_{NAVDATA} := (\overline{v}dx, \overline{v}dy, \bar{h}_t, \bar{\Phi}, \bar{\Theta}, \bar{\Psi}_t) \quad (14)$$

$$h_{NAVDATA}(x_t) := \begin{pmatrix} vx_t \cos \Psi_t + vy_t \sin \Psi_t \\ -vx_t \sin \Psi_t + vy_t \cos \Psi_t \\ z_t \\ \Phi_t \\ \Theta_t \\ \Psi_t \end{pmatrix} \quad (15)$$

Модель наблюдения VSLAM. Когда алгоритм VSLAM успешно отслеживает видеоквадр, его позиция 6DOF преобразуется из системы координат передней камеры в систему координат квадрокоптера, что приводит к непосредственному наблюдению за позицией квадрокоптера, определяемой как:

$$z_{VSLAM} := f(E_{DC}E_{C,t}) \in \mathbb{R}^6 \quad (16)$$

$$h_{VSLAM}(x_t) := (x_t, y_t, z_t, \Phi_t, \Theta_t, \Psi_t)^T \in \mathbb{R}^6 \quad (17)$$

где $E_{c,t} \in SE(3)$ – оценочная позиция камеры с учетом масштаба, $E_{dc} \in SE(3)$ – постоянное преобразование из камеры в систему координат квадрокоптера $SE(3) \rightarrow \mathbb{R}^6$ – преобразование из элемента $SE(3)$ к представлению качки-рысканья $(x, y, z, \Phi, \Theta, \Psi)$.

ПИД-регулятор. ПИД-регулятор был разработан для того, чтобы управлять движениями MAV на основе предполагаемой позиции. Ссылка $(\hat{x}, \hat{y}, \hat{z}, \hat{\Psi})$ необходима в качестве требуемой позиции дрона по отношению к окружающей среде. EKF принесет оценку позиции, как показано на рис. 4. Разница между опорным и оцененным положением – это ошибка, которая будет минимизирована ПИД-регулятором путем отправки в MAV соответствующей управляющей команды $u = (\hat{v}\hat{x}, \hat{v}\hat{y}, \hat{v}\hat{z}, \hat{\Psi})$, который рассчитывается следующим образом:

$$\hat{v}\hat{x} = \cos \Psi [Kp(\hat{x} - x) + Kd * \dot{\hat{x}}] + \sin \Psi [Kp(\hat{y} - y) + Kd * \dot{\hat{y}}] \quad (18)$$

$$\hat{v}\hat{y} = -\sin \Psi [Kp(\hat{x} - x) + Kd * \dot{\hat{x}}] + \cos \Psi [Kp(\hat{y} - y) + Kd * \dot{\hat{y}}] \quad (19)$$

$$\hat{v}\hat{z} = Kp * (\hat{z} - z) + Kd * \dot{\hat{z}} + Ki * \int (\hat{z} - z) \quad (20)$$

$$\hat{\Psi} = Kp(\hat{\Psi} - \Psi) \quad (21)$$

Это позволяет алгоритму вести MAV вдоль ряда точек на карте, чтобы он мог следовать определенному маршруту.

Результаты. Для тестирования нашей системы на земле мы использовали горизонтальную камеру с детектором движения, которая была установлена в потолке тестовой среды. Это позволяет нам измерять XY движения дрона с помощью внешнего датчика. С помощью этого метода невозможно определить высоту, поэтому мы доверяем данным высотометра, встроенного в MAV, по умолчанию. Эта процедура позволяет нам сравнивать положение, оцениваемое нашим алгоритмом, с истинным положением, обнаруженным внешней камерой.

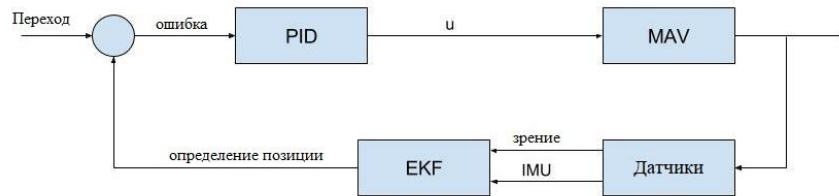


Рисунок 4 – Схема блоков ПИД-регулятора

Эксперимент. ORB-SLAM использовался во время тестов, результаты которых представлен на рис. 5-7.

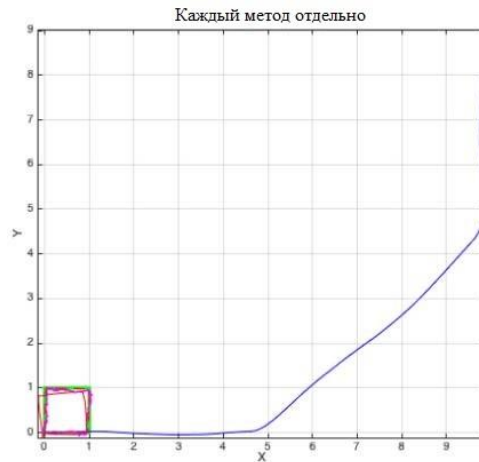


Рисунок 5 – Результаты разных методов, реализованных отдельно (красная линия – реальное позиционирование по Predictor; пунктирная красная линия – определенное позиционирование по Predictor; синяя линия – реальное позиционирование по IMU; синяя пунктирная линия – определенное позиционирование по IMU; розовая линия – реальное позиционирование по VSLAM; розовая пунктирная линия – определенное позиционирование по VSLAM)

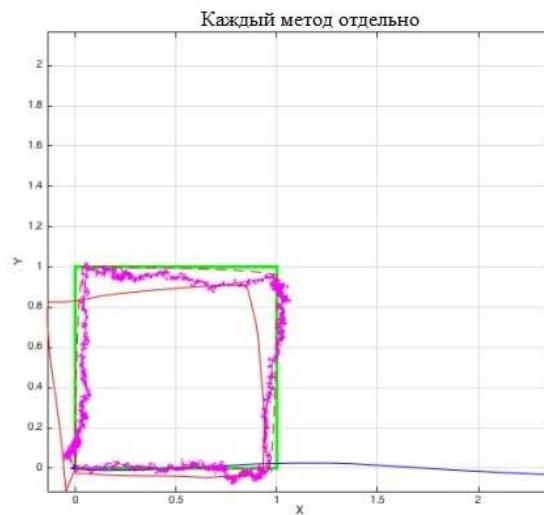


Рисунок 6 – Увеличенные результаты разных методов, реализованных отдельно

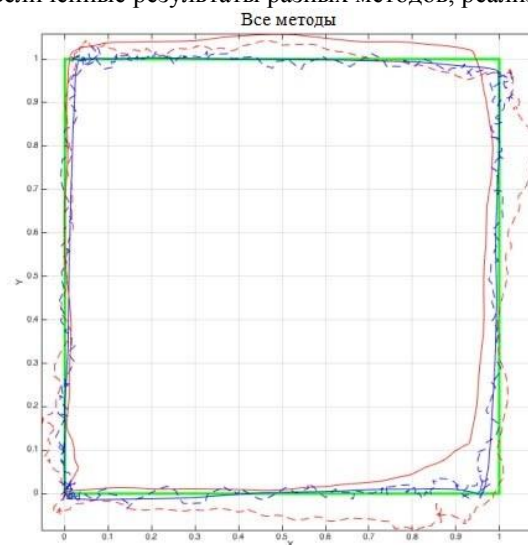


Рисунок 7 – Выполнение всех методов

Мы использовали этот алгоритм VSLAM вместо LSD-SLAM, потому что нам не требовалась трехмерная карта, которую может принести LSD-SLAM – так вычислительные требования были исключены. Для этой работы условия освещения были оптимальными, и мы предоставили достаточно функций, которые могут быть извлечены алгоритмом для обеспечения его работы. В других условиях, например, из-за отсутствия функций, где выделяется выполнение LSD-SLAM – мы могли бы выбрать другой алгоритм. Кроме того, мы поняли, что ORB-SLAM представляет собой более надежную технику VSLAM, которая сталкивается с чистым вращательным движением и быстрым переводом.

Как было сказано ранее, PID-контроллер позволяет MAV проходить путь через ряд точек. В качестве теста мы заставили беспилотника летать, пытаясь воссоздать квадрат размером 1 x 1 м, который изображен зеленым квадратом на рис. 5-7.

В качестве первого теста мы запускаем алгоритм с каждым из этапов EKF по отдельности – только со стадией прогнозирования, только со стадией коррекции IMU и только со стадией коррекции VSLAM, как показано на рис. 5. Поскольку мы не имеем возможность проверить вертикальную точность алгоритма, где производительность IMU отличается, лучшее отслеживание этапов прогнозирования и коррекции VLSAM очевидно. Чтобы показать плохую производительность алгоритма, когда он использует только измерения IMU, на рис. 5 представлены результаты метода со всеми его отклонениями. Ниже рис. 6 представлен тот же график, но увеличенный, чтобы было легче увидеть различия между реализациями.

Улучшение производительности системы с добавлением этапов, суммированных на IV-B, оценивается на рис. 7. Как показано, система наиболее точна с прогнозированием и этапами коррекции как IMU, так и VSLAM.

Литература

1. Mellinger, D. Kumar, V.: Minimum snap trajectory generation and control for quadrotors. In: Proc.IEEE Intelligent Conference on Robotics and Automation (ICRA). pp. 2520- 2525. (2011).

2. Lindsey, Q., Mellinger, D. Kumar, V.: Construction of cubic structures with quadrotor teams. In: Proceedings on Robotics: Science and Systems (RSS). (2011).
3. Kushleyev, A., Mellinger, D., Kumar, V: Towards a swarm of agile micro quadrotors. In: Proceedings of Robotics: Science and Systems (RRS). (2012).
4. Achtelik, M., Lynen, S., Weiss, S., Kneip, L., Chli, M.: Visualinertial SLAM for a small helicopter in large outdoor environments. In: Proceedings IEEE Conference on Intelligent Robots and Systems (IROS).pp. 2651-2652 (2012).
5. Fraundorfer, F., Heng, L., Honegger, D., Lee, L., Meier, L., Tanskanen, P., Pollefeys, M.: Vision-based autonomous mapping and exploration using a quadrotor MAV. In: Proceedings IEEE Inter. Conference on Intelligent Robots and Systems (IROS). (2012).
6. Engel, J., Schöps, T., Cremers, D.: LSD-SLAM: Large-scale direct monocular SLAM. In Computer Vision (ECCV 2014), pp.834-849. (2014).
7. Mur-Artal, R., Montiel, J., Tardos, J.: ORB-SLAM: A Versatile and Accurate Monocular SLAM System. In IEEE Transactions on Robotics, vol. 31, no. 5, pp. 1147-1163 (2015).
8. Engel, J., Sturm, J., Cremers, D.: Scale-Aware Navigation of a Low-Cost Quadcopter with a Monocular Camera. In Robotics and Autonomous Systems (RAS), volume 62, 2014 (2014).

