

# СИСТЕМА СЛИЯНИЯ ДАННЫХ IMU, GPS, LIDAR, СТЕРЕОКАМЕРЫ И МАГНЕТОМЕТРА ДЛЯ РЕШЕНИЯ ЗАДАЧИ ЛОКАЛИЗАЦИИ БЕСПИЛОТНОГО АВТОМОБИЛЯ

## SYSTEM OF IMU, GPS, LIDAR, STEREO CAMERA AND MAGNETOMETER DATA FUSION FOR SELF-DRIVING CAR LOCALIZATION

A. Izvin

*Summary.* This work describes the localization system of an unmanned vehicle, which consists of IMU, GPS, LIDAR, stereo camera and magnetometer sensors. Unscented Kalman filter is responsible for merging the data produced by these sensors. System performance is tested on synthetic data generated by the Carla simulator. The car's state values predicted by the system are compared with the ground truth values. As a result of comparison it is established that the system determines the position, speed and direction of the vehicle with accuracy  $\pm 4.13$  cm,  $\pm 0.08$  m/s and  $\pm 0.01$  rad, respectively. This makes it possible to use that system in practice.

*Keywords:* localization, Kalman filter, GPS, IMU, stereo camera, LIDAR, magnetometer, unmanned vehicle, simulator, data fusion, Carla.

Извин Андрей Валерьевич

МГТУ им. Н.Э. Баумана, г. Москва  
izvinav309@gmail.com

*Аннотация.* В данной работе описывается система локализации беспилотного автомобиля, которая состоит из сенсоров IMU, GPS, LIDAR, стереокамеры и магнетометра, а за слияние данных этих сенсоров отвечает сигма-точечный фильтр Калмана. Работоспособность системы тестируется на синтетических данных, сгенерированных с помощью симулятора Carla. Проводится сравнение значений состояния автомобиля, спрогнозированных системой, с их истинными значениями, в результате чего устанавливается, что система определяет позицию, скорость и направление движения автомобиля с точностями  $\pm 4.13$  см,  $\pm 0.08$  м/с и  $\pm 0.01$  рад соответственно, что говорит о возможности ее применения на практике.

*Ключевые слова:* локализация, фильтр Калмана, GPS, IMU, стереокамера, LIDAR, магнетометр, беспилотный автомобиль, симулятор, слияние данных, Carla.

## Введение

**В** настоящее время, для решения задачи локализации беспилотных транспортных средств, широко используются относительно дешевые инерциальные измерительные модули (IMU). IMU модуль обычно состоит из гироскопа и акселерометра, которые позволяют измерять угловую скорость и ускорение движущегося объекта. Интегрирование показаний IMU бортовым компьютером, позволяет на высоких частотах обновлять данные о координатах, скорости и направлении движения автомобиля. Поскольку показания гироскопа и акселерометра подвержены воздействию шумов (например, проскальзывание шин), данные о положении, направлении и скорости движущегося объекта дрейфуют во времени [1]. Увеличение точности навигации за счет установки высокоточных IMU модулей, значительно увеличивает стоимость навигационных систем. Как показали современные исследования [2], для решения данной проблемы целесообразно использовать связку дешевых IMU модулей с вспомогательными навигационными системами и сенсорами, такими как: GPS, магнетометр, спидометр и др. Такой подход позволяет справиться

с проблемой падения точности навигации за счет слияния показаний IMU модуля с высокой частотой обновления и показаний вспомогательных модулей с низкой частотой обновления, измерения которых практически не зависят от времени.

Системы GNSS и GPS, в частности, хорошо подходят для роли вспомогательного модуля для IMU. Однако такие системы навигации имеют очень большую погрешность измерения позиции (от 3 до 10 метров), кроме того, в больших городах здания могут перекрывать спутниковые сигналы, что приводит к обрывам системы и падению точности. В работах [3–6] исследуется возможность применения стерео-камер для дальнейшего увеличения точности локализации, хотя такой подход и дает положительные результаты, нельзя забывать, что видеокamеры сильно зависят от освещенности окружающей среды. Для решения данной проблемы совместно с камерами можно использовать системы ночного видения, либо отказаться от камер в пользу лидаров, показания которых не зависят от освещенности, поскольку используют собственный световой луч для построения неструктурированного облака точек. Как было установ-

Таблица 1. частоты обновления сенсоров

Наименование сенсора	Частота обновления, Hz
IMU	100
Стерео-камера	60
Магнитометр	20
GPS	5
Lidar	10



Рис. 1. схема объединения измерений с GPS, IMU и лидаров

лено в работах [7–11], лидары способны значительно увеличить точность локализации.

Поскольку GPS, стереокамеры и лидары позволяют измерять только координаты и скорость движущегося объекта, но нам также нужно уточнять направление его движения, для синтеза действительно робастной системы навигации, требуется еще хотя бы один сенсор. В качестве такого сенсора предлагается использовать магнитометр (компас), который обычно входит в состав современных IMU модулей вместе с акселерометром и гироскопом.

Обычно слияние показаний нескольких сенсоров с целью корректировки ошибок измерения, проводят с помощью расширенного фильтра Калмана [12]. В настоящее время внимание исследователей приковано с сигма-точечному фильтру Калмана [13], ввиду следующих его преимуществ: отсутствие необходимости

расчета сложного якобиана, сохранение нелинейности модели вплоть до второго порядка, относительно малая вычислительная сложность.

### Асинхронная схема слияния сенсоров

На рисунке 1 представлена схема объединения измерений с IMU, LIDAR, магнитометра и GPS при помощи сигма-точечного фильтра Калмана. Поскольку данные с IMU, поступают чаще, чем с GPS, LIDAR и магнитометра, то данная схема строится по асинхронному принципу. Если в данный момент времени нам доступны только измерения с IMU, то на их основе делается предсказание текущего состояния системы с последующей корректировкой по норме кватерниона направления, однако, если нам также доступны измерения с любого другого сенсора, то корректировка предсказания проводится с помощью этих измерений.

$$x = [p, v, q, b_a, b_\omega]^T_{16 \times 1} \tag{1}$$

$$x_t = f(x_{t-1}, u_{t-1}) + \varepsilon \tag{2}$$

$$x_t = \begin{bmatrix} p_{t-1} + v_{t-1}\Delta t + (C(q_{t-1})(f_{b,t-1}) + g) \frac{\Delta t^2}{2} \\ v_{t-1} + (C(q_{t-1})(f_{b,t-1}) + g)\Delta t \\ \cos\left(\frac{\|\omega_{b,t-1}\|\Delta t}{2}\right) I_4 + \omega_{b,t-1} \frac{\sin\left(\frac{\|\omega_{b,t-1}\|\Delta t}{2}\right)}{\|\omega_{b,t-1}\|} \\ b_{a,t-1} \\ b_{\omega,t-1} \end{bmatrix} + \begin{bmatrix} e_p \\ e_v \\ e_q \\ e_a \\ e_\omega \end{bmatrix} \tag{3}$$

Для иллюстрации последовательности, в которой будет проходить корректировка измерений, в таблице 1 представлены приблизительные частоты обновления сенсоров.

### Координатные системы

Для решения задачи локализации транспортных средств выделяют следующие четыре координатных системы:

Геоцентрическая система координат, связанная с Землей (ECEF) — система координат с началом в центре масс Земли, ось Z которой проходит через точку истинного севера, ось X проходит через главный меридиан, а ось Y определяют таким образом, чтобы тройка координатных векторов была положительна (правило правой руки);

Глобальная система координат — координатная система север-восток-вниз (NED), ось X которой проходит через точку истинного севера, ось Y через точку истинного востока, а ось Z направлена к центру масс земли (вдоль направления гравитации);

Система координат, связанная с автомобилем — система координат, ось X которой проходит вдоль положительного движения автомобиля, ось Z указывает вверх, а ось Y влево.

Системы координат сенсоров автомобиля — после калибровки, совпадают с системой координат автомобиля.

Состояние автомобиля в конкретный момент времени можно описать следующими характеристиками:

позицией, скоростью и направлением. Каждая из этих характеристики представляется в виде вектора.

Векторы  $p, v \in \mathbb{R}^3$  — соответственно характеризуют позицию и скорость автомобиля в глобальной системе координат относительно осей XYZ.

Вектор  $q \in \mathbb{R}^4$  — характеризует поворот системы координат автомобиля относительно глобальной системы координат. Поворот хранится в виде единичного кватерниона, поэтому  $q$  состоит из векторной части  $e = (q_1, q_2, q_3)^T \in \mathbb{R}^3$  и скалярной части  $q_4 \in \mathbb{R}$ . Норма вектора  $q$  равна единице:

$$\|q\| = 1$$

Поскольку, акселерометр и гироскоп имеют смещения показаний, которые можно моделировать процессом случайного блуждания, то нам требуется опре-

делить еще два вектора  $b_a, b_\omega \in \mathbb{R}^3$ , которые будут компенсировать смещения акселерометра и гироскопа, соответственно.

Все эти характеристики можно объединить в вектор состояния с размерностью  $16 \times 1$  (1).

Модель процесса, устанавливающая связь между состоянием автомобиля в текущий момент времени  $t$ , и предыдущий момент времени  $t - 1$ , имеет следующий вид (2).

Где  $x_t$  — вектор состояния в момент времени  $t$ ;

$x_{t-1}$  — вектор состояния в момент времени  $t - 1$

$\varepsilon$  — вектор ошибок;

$u_{t-1} = (f_{t-1}, w_{t-1})$  — показания акселерометра и гироскопа в момент времени  $t - 1$ .

$$C(q_{t-1}) = (q_{4,t-1}^2 - e_{t-1}^T e_{t-1})I_3 + 2e_{t-1}e_{t-1}^T - 2q_{4,t-1}[e_{t-1}]_x \quad (4)$$

$$\mathbb{C}[a] = \begin{bmatrix} -[a]_x & a \\ -a^T & 0 \end{bmatrix} \quad (5)$$

$$[a]_x = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \quad (6)$$

$$Q = \text{diag}(Q_p, Q_v, Q_q, Q_a, Q_{\mathbb{C}}) \quad (7)$$

$$\hat{y}_t = g(x_t) + e_m \quad (8)$$

$$\hat{p}_t = p_{cam} + e_{cam} \quad (9)$$

В данной работе используется следующая форма модели процесса (3).

Где  $\Delta t$  — приращение по времени;  
 $g = (0, 0, -9.8)^T$  — вектор гравитационного ускорения;

$f_{b,t-1} = f_{t-1} - b_{a,t-1}$  — показания акселерометра с поправкой на их смещение;  
 $\mathbb{C}_{b,t-1} = \mathbb{C}_{t-1} - b_{\mathbb{C},t-1}$  — показания гироскопа с поправкой на их смещение;  $C(q_{t-1})$  — матрица поворота, соответствующая кватерниону  $q_{t-1}$ , и которая связана с ним выражением (4).

$\mathbb{C}[a]$  — это операция отображения вектора  $a \in \mathbb{R}^3$  в косимметричную матрицу, которая имеет вид (5).

$[a]_x$  — в выражениях 4 и 5 — это операция отображения трехмерного вектора  $a = (a_x, a_y, a_z)^T$  в виде косимметричной матрицы (6).

Ошибка процесса представлена в виде вектора  $e = (e_p, e_v, e_q, e_a, e_{\mathbb{C}})^T$ . Каждая составляющая вектора ошибки — независима, моделируется функцией Гаусса со средним значением равным нулю, и имеет собственную матрицу ковариации. Таким образом, матрицу ковариации процесса можно выразить следующим образом (7).

### Модели измерений

В момент, когда становятся доступны новые измерения с магнитометра, стерео-камеры, LIDAR или GPS, появляется возможность уточнить значения вектора состояния. В общем случае модель измерений имеет вид (8).

Где:  $\hat{y}_t$  — измерение сенсора;  
 $g(x_t)$  — модель измерений;  
 $e_m$  — вектор ошибки измерений.

Каждому сенсору соответствует своя модель измерений в зависимости от специфики его работы.

Для того, чтобы получить положение автомобиля с помощью RGBD камеры, к двум последовательно снятым кадрам видео-потока применяется алгоритм ORB [14], в основе которого лежит комбинация детектора FAST [15] и дескриптора BRIEF [16] с некоторыми изменениями, после каждому дескриптору с одного изображения ставится в соответствие дескриптор с другого за счет сравнения расстояний между этими дескрипторами, в итоге, к полученным парам дескрипторов применяется алгоритм RANSAC для нахождения матрицы трансформации положения камеры за время снятия кадров. С помощью матрицы трансформации находится новое положение автомобиля, а формула измерений RGBD камеры принимает вид (9).

Где:  $p_{cam}$  — матрица кросс-ковариации;  
 $e_{cam}$  — матрица коэффициентов усиления;

Аналогичным образом происходит определение положения автомобиля с помощью LIDAR. Для этого к двум последовательно зарегистрированным облакам точек применяется алгоритм ICP (итерация по ближайшим точкам) [17], который требует приблизительную матрицу трансформации положения одного облака относительно другого, которую легко получить в результате сравнения вектора состояния автомобиля в момент регистрации первого облака и текущего нескорректированного

$$\hat{p}_t = p_{lidar} + e_{lidar} \tag{10}$$

$$\hat{p}_t = p_{gps} + e_{pgps} \tag{11}$$

$$\hat{v}_t = v_{gps} + e_{vgps} \tag{12}$$

$$\hat{\psi}_t = \psi_{mag} + e_{mag} = \arctan\left(\frac{2(q_2q_3 + q_1q_4)}{q_1^2 + q_2^2 - q_3^2 - q_4^2}\right) + e_{mag} \tag{13}$$

$$\mathbf{1} = \|\mathbf{q}\|^2 = q_1^2 + q_2^2 + q_3^2 + q_4^2 + e_{qat} \tag{14}$$

$$\hat{P}_{t-1} = L_{t-1}L_{t-1}^T \tag{15}$$

$$K = \mathbf{3} - N \tag{16}$$

$$x_{t-1}^{(0)} = \hat{x}_{t-1} \tag{17}$$

$$x_{t-1}^{(i)} = \hat{x}_{t-1} + \sqrt{N + K}col_i(L_{t-1}), i = 1 \dots N \tag{18}$$

$$x_{t-1}^{(i+N)} = \hat{x}_{t-1} + \sqrt{N + K}col_i(L_{t-1}), i = 1 \dots N \tag{19}$$

вектора состояния. Результатом выполнения алгоритма ISF, является матрица трансформации между облаками точек, с ее помощью находится новое положение автомобиля, а формула измерений с LIDAR принимает вид (10).

Где:  $p_{lidar}$  — положение автомобиля в глобальной системе координат;  
 $e_{lidar}$  — ошибка измерений LIDAR.

С помощью GPS приемника измеряется скорость и позиция автомобиля в данный момент времени. Позиция приходит в виде широты, высоты и долготы (LLA), затем, она переводится в координатную систему ECEF, а после этого в глобальную систему координат. Модель измерений GPS состоит из уравнений позиции и скорости (11), (12).

Где:  $p_{gps}$  — положение автомобиля в глобальной системе координат;  
 $e_{pgps}$  — ошибка измерения позиции GPS приемником.

$v_{gps}$  — положение автомобиля в глобальной системе координат;

$e_{vgps}$  — ошибка измерения скорости GPS приемником.

Магнитометр измеряет магнитное поле Земли, с помощью которого находится угол рыскания автомобиля. Модель измерений магнитометра имеет вид (13).

Где:  $\psi_{mag}$  — угол рыскания автомобиля;  
 $e_{mag}$  — ошибка измерения угла рыскания магнитометром.

$q_1, q_2, q_3, q_4$  — параметры кватерниона направления движения;

Поскольку кватернион, который хранит направление движения автомобиля в векторе состояния — единичный, то можно добавить еще одну модель псевдо-измерений (14)

Где:  $\|\mathbf{q}\|^2 = \mathbf{1}$  — норма кватерниона направления движения в квадрате;

$$\tilde{x}_t^{(i)} = f(x_{t-1}^{(i)}, u_{t-1}) + e \quad (20)$$

$$\sigma^{(i)} = \begin{cases} \frac{K}{N+K}, \text{ для } i = 0 \\ \frac{1}{2(N+K)}, \text{ для } i > 0 \end{cases} \quad (21)$$

$$\tilde{x}_t = \sum_{i=0}^{2N} \sigma^{(i)} \tilde{x}_t^{(i)} \quad (22)$$

$$Q_t = Q \Delta t^2 \quad (23)$$

$$\tilde{P}_t = \sum_{i=0}^{2N} \sigma^{(i)} (\tilde{x}_t^{(i)} - \tilde{x}_t) (\tilde{x}_t^{(i)} - \tilde{x}_t)^T + Q_t \quad (24)$$

$e_{quat}$  — псевдо-ошибка измерения нормы кватерниона.

$q_1, q_2, q_3, q_4$  — параметры кватерниона направления движения.

### Сигма-точечный фильтр Калмана (UKF)

UKF состоит из двух основных этапов: этапа предсказания и этапа корректировки. На этапе предсказания, каждый раз, когда становятся доступны измерения с IMU, происходит расчет сигма-точек для вектора состояния системы (15), (16), (17), (18), (19).

Где:  $\tilde{P}_{t-1}$  — матрица ковариации состояния системы;

$L_{t-1}$  — матрица, полученная из матрицы ковариации с помощью разложения методом Холецкого.

$\hat{x}_{t-1}$  — вектор состояния системы в предыдущий момент времени;

$x_{t-1}^{(i)}$  — сигма-точка вектора состояния;

$col_i(L_{t-1})$  —  $i$ -ый столбец матрицы;

$N$  — число столбцов матрицы  $L_{t-1}$ ;

Далее, с помощью модели процесса (формула 3) и данных IMU- каждая сигма-точка обновляется, и с их помощью происходит расчет среднего значения вектора состояния и матрицы ковариации для текущего момента времени (20), (21), (22), (23), (24).

Где:  $Q_t$  — аддитивный белый шум процесса;

$\tilde{x}_t$  — среднее значение вектора состояния;

$\tilde{P}_t$  — матрица ковариации вектора состояния;

$\sigma^{(i)}$  — коэффициенты, требуемые для расчета среднего значения и матрицы ковариации;

На этапе корректировки происходит уточнение полученной матрицы ковариации и вектора состояния, используя одну из моделей измерений, описываемых формулами 8–14. Для этого, сначала рассчитывается новый сет сигма-точек из  $\tilde{x}_t$  и  $\tilde{P}_t$ , по формулам 15–19. Затем, эти сигма-точки проходят через модель измерения и, на их основе рассчитываются среднее значение и матрица ковариации измерения (25), (26), (27), (28), (29).

Где:  $\tilde{x}_t^{(i)}$  — сигма-точка, полученная из  $\tilde{x}_t$  и  $\tilde{P}_t$ ;

$R_t$  — аддитивный белый шум измерения;

$\hat{y}_t$  — среднее значение вектора измерения;

$\tilde{P}_Y$  — матрица ковариации вектора измерения.

Далее, идет расчет матрицы кросс-ковариации и матрицы коэффициентов усиления (30), (31).

Где:  $P_{xy}$  — матрица кросс-ковариации;

$K_K$  — матрица коэффициентов усиления;

На последнем этапе слияния данных с сенсоров происходит расчет скорректированного значения вектора состояния системы и матрицы  $\tilde{x}_t$  ковариации  $\tilde{P}_t$  (32), (33).

### Исследование точности работы системы

Для тестирования представленной системы, использовался набор данных, полученный имитацией езды

$$\hat{y}_t^{(i)} = g(\tilde{x}_t^{(i)}) + e_m \quad (25)$$

$$\sigma^{(i)} = \begin{cases} \frac{K}{N+K}, \text{ для } i = 0 \\ \frac{1}{2(N+K)}, \text{ для } i > 0 \end{cases} \quad (26)$$

$$\hat{y}_t = \sum_{i=0}^{2N} \sigma^{(i)} \hat{y}_t^{(i)} \quad (27)$$

$$R_t = R_{t-1} \Delta t^2 \quad (28)$$

$$\hat{P}_Y = \sum_{i=0}^{2N} \sigma^{(i)} (\hat{y}_t^{(i)} - \hat{y}_t) (\hat{y}_t^{(i)} - \hat{y}_t)^T + R_t \quad (29)$$

$$P_{xy} = \sum_{i=0}^{2N} \sigma^{(i)} (\tilde{x}_t^{(i)} - \tilde{x}_t) (\hat{y}_t^{(i)} - \hat{y}_t)^T \quad (30)$$

$$K_k = P_{xy} P^{-1} \quad (31)$$

$$\hat{x}_t = \tilde{x}_t + K_k (y_t - \hat{y}_t) \quad (32)$$

$$\hat{P}_t = \check{P}_t - K_k \hat{P}_Y K_k^T \quad (33)$$

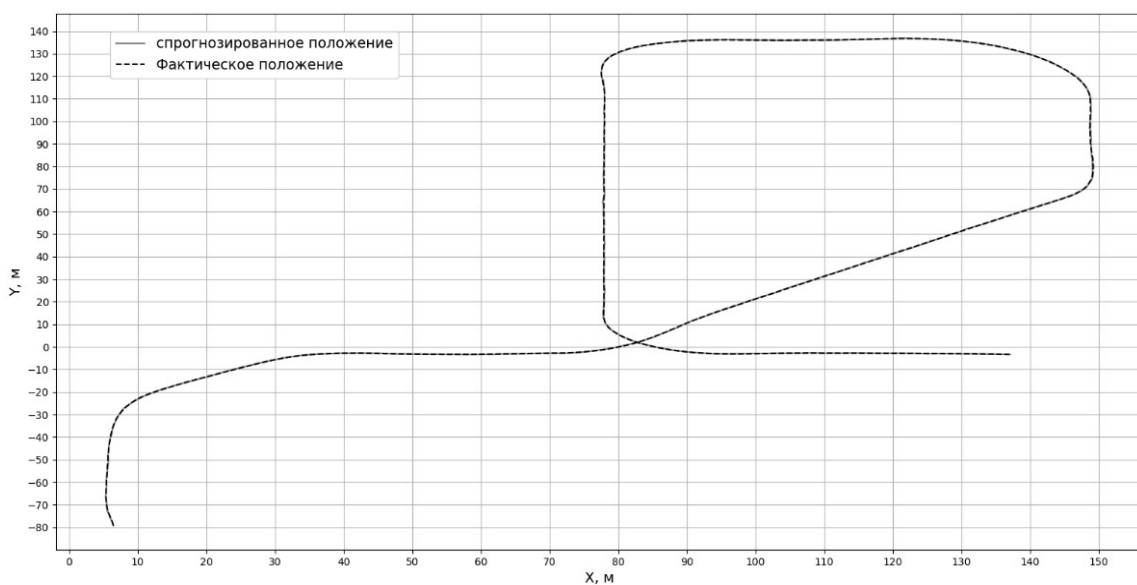


Рис. 2. Маршрут движения (вид сверху)

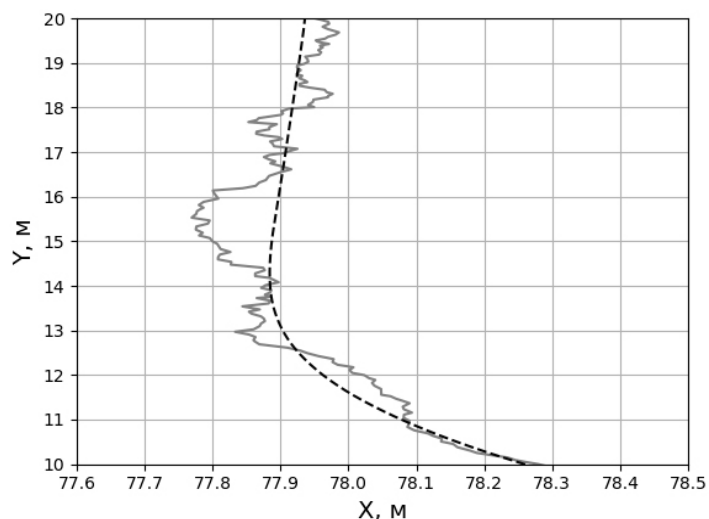


Рис. 3. Участок маршрута движения

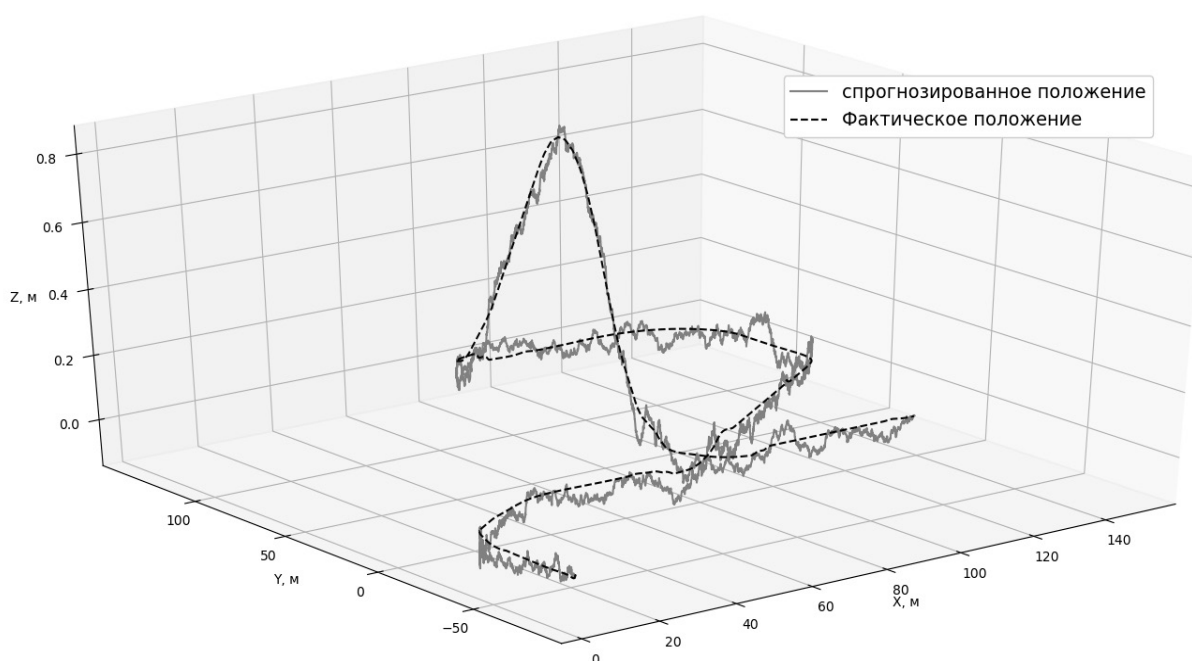


Рис. 4. Маршрут движения в 3-х мерном виде

беспилотного автомобиля, с заданным набором сенсоров, по виртуальной карте города, в среде симулятора Carla[18]. В результате, был сгенерирован маршрут, представленный на рисунке 2.

Для наглядной демонстрации точности локализации автомобиля, на рисунке 3 представлен отрезок маршрута движения, на котором представлены фактическое положение автомобиля, и положения, которой сгенерировал локализатор.

На рисунке 4 можно наблюдать тот же маршрут, что и на рисунках выше, но в трехмерном виде, что позволяет оценить точность локализации по оси Z.

Оценки позиции, направления и скорости автомобиля представленной в статье системой, сравнивались с истинными значениями данных параметров, выдаваемых симулятором Carla. Результаты организованы в виде графиков ошибок (рисунок 5), с границами  $3\sigma$ , которые характеризуют интервал,



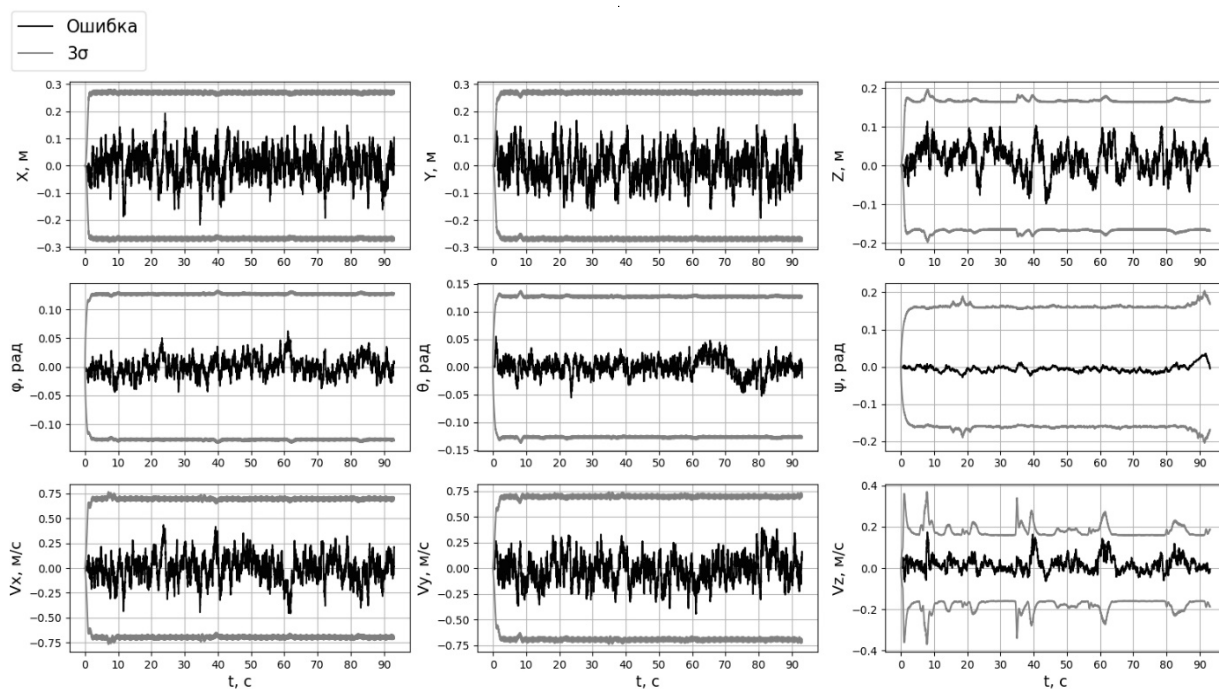


Рис. 5. Ошибки оценки состояния системы

за пределы которого ошибка практически не должна выходить.

### Заключение

Представленная в данной работе система позволяет проводить оценку позиции, скорости и направления движения беспилотного автомобиля с точностями

$\pm 4.13$  см,  $\pm 0.08$  м/с и  $\pm 0.01$  рад соответственно. Большое число разнородных сенсоров, используемых в системе, должно обеспечить высокую надежность локализации при разных природных условиях, а также в условиях, когда данные с одного или нескольких датчиков зашумлены, или не поступают вообще. Таким образом, высокая точность и надежность системы позволяют сделать вывод о возможности ее применения на практике.

### ЛИТЕРАТУРА

1. D.J. Biezad. Integrated Navigation and Guidance Systems. AIAA Education Series, Reston, VA, 1999.
2. R. van der Merwe and E. A. Wan. Sigma-point kalman filters for nonlinear estimation and sensor-fusion — application to integrated navigation. AIAA guidance, navigation, and control conference, Providence, Rhode Island, August 2004.
3. Tao, Z.; Bonnifait, P.; Fremont, V.; Ibanez-Guzman, J. Mapping and localization using gps, lane markings and proprioceptive sensors. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems, Tokyo, Japan, 3–7 November 2013; pp. 406–412.
4. Schindler, A. Vehicle self-localization with high-precision digital maps. In Proceedings of the IEEE Intelligent Vehicle Symposium, Gold Coast, Australia, 23–26 June 2013; pp. 141–146.
5. Gruyer, D.; Belaroussi, R.; Revilloud, M. Map-aided localization with lateral perception. In Proceedings of the 2014 IEEE Intelligent Vehicles Symposium (IV), Dearborn, MI, USA, 8–11 June 2014; pp. 674–680.
6. Pandey, G.; McBride, J.R.; Eustice, R. M. Ford campus vision and LiDAR data set. Int. J. Rob. Res. 2011, 30, 1543–1552.
7. Zhao, G.Q.; Yuan, J. S. Curb detection and tracking using 3D-LIDAR scanner. In Proceedings of the 19th IEEE International Conference on Image Processing (ICIP), Orlando, FL, USA, 30 September–3 October 2012.
8. Huang, A.S.; Teller, S. Lane boundary and curb estimation with lateral uncertainties. In Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO, USA, 10–15 October 2009; pp. 1729–1734.
9. Hervieu, A.; Soheilian, B. Road side detection and reconstruction using LIDAR sensor. In Proceedings of the 2013 IEEE on the Intelligent Vehicles Symposium (IV), Gold Coast, Australia, 23–26 June 2013; pp. 1247–1252.
10. Yeonsik, K.; Chiwon, R.; Seung-Beum, S.; Bongsob, S. A Lidar-based decision-making method for road boundary detection using multiple Kalman filters. IEEE Trans. Ind. Electron. 2012, 59, 4360–4368.

11. Kim, Z. Robust lane detection and tracking in challenging scenarios. *IEEE Trans. Intell. Transp. Syst.* 2008, 9, 16–26.
12. D.B. Kingston. Implementation issues of real-time trajectory generation on small uavs, 2004.
13. R. van der Merwe and E. A. Wan. Sigma-point kalman filters for probabilistic inference in dynamic state-space models. *Workshop on advances in machine learning*, Montreal, Canada, 2003.
14. Ethan Rublee, Vincent Rabaud, Kurt Konolige, Gary Bradski: "ORB: an efficient alternative to SIFT or SURF", *Computer Vision (ICCV)*, IEEE International Conference on. IEEE, pp.2564–2571, 2011.
15. Rosten, Edward, Tom Drummond: "Machine learning for high-speed corner detection", 9th European Conference on Computer Vision (ECCV), pp.430–443, 2006.
16. Michael Calonder, Vincent Lepetit, Christoph Strecha, Pascal Fua, "BRIEF: Binary Robust Independent Elementary Features", 11th European Conference on Computer Vision (ECCV), pp.778–792, 2010.
17. Besl P.J., McKay N. D. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp.239–256, 1992.
18. Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, Vladlen Koltun. CARLA: An Open Urban Driving Simulator // 1st Conference on Robot Learning. 2017.

© Извин Андрей Валерьевич (izvinav309@gmail.com).

Журнал «Современная наука: актуальные проблемы теории и практики»



Московский государственный технический университет имени Н.Э. Баумана