

УДК 004.896:621.865

## НЕОПРЕДЕЛЁННЫЕ МОДЕЛИ В ЗАДАЧЕ ЛОКАЛИЗАЦИИ МОБИЛЬНОГО РОБОТА

А.Д. Московский (*moscowskyad@gmail.com*)  
Национальный исследовательский центр  
«Курчатовский институт», Москва

**Аннотация.** В работе исследуется возможность применения метода недоопределённых вычислений для задачи определения мобильным роботом своего положения. Рассматривается классическая проблема локализации по данным одометрии и наблюдаемым при помощи камеры ориентирам. Особенностью рассматриваемой задачи является наличие зон неопределённости, в которых отсутствуют данные об ориентирах. Определены преимущества разрабатываемого метода на основе недоопределённых вычислений в применении к рассматриваемому классу задач в сравнении с классическими методами локализации. Разработанная система локализации сравнивается с методами на основе расширенного фильтра Калмана и фильтра частиц. Проведенные вычислительные эксперименты показали преимущество разрабатываемой системы на определенном классе задач.<sup>1</sup>

**Ключевые слова:** локализация, мобильный робот, недоопределённые вычисления.

### Введение и постановка задачи

Задача локализации робота в пространстве возникла с момента зарождения мобильной робототехники, т.к. любая задача, связанная с движением, неминуемо требует ответа на вопрос «где я?». Классической задачей можно назвать задачу, когда робот совершает движение в среде, основываясь на данных о своем перемещении, и корректирует свое положение, ориентируясь на расположенные в его окружении метки. Предполагается, что положения данных меток в среде известны, а также возможно определить расстояние до них или угол наблюдения. Метки могут быть объектами весьма широкого класса, например, специальные радиомаяки. Однако наиболее удобными в использовании являются

---

<sup>1</sup> Работа выполнена при поддержке НИЦ «Курчатовский институт» (приказ от 05.07.2018 №1601).

визуальные ориентиры, т.к. веб-камера является одним из самых доступных датчиков, а также в качестве ориентиров зачастую можно использовать уже расположенные в среде функционирования объекты. Первая проблема такой задачи - это погрешности измерений, которые касаются как определения своего смещения по внутренним датчикам, так и измерений, связанных с определением параметров меток. Вторая проблема связана с зонами неопределённости, в которых робот не наблюдает ни одного ориентира. Если первая проблема неизбежна, то вторая часто не рассматривается, т.к. исследователи обычно пытаются поместить в среду достаточное количество ориентиров. В данной же работе такие случаи рассматриваются отдельно.

## **1. Классические методы локализации**

Основным столпом, на котором основано большинство используемых методов навигации, являются вероятностные подходы. Они подробно описаны в монографии Себастьяна Трана [Thrun и др., 2005]. К вероятностным подходам относятся такие известные методы, как фильтр Калмана и фильтр частиц. Они используются для того, чтобы бороться с неопределённостью, которая возникает ввиду таких факторов, как: неструктурированность среды оперирования; зашумленность сенсоров робота; неточности исполняемых механизмов (актуаторов). Обычно такие подходы работают по схеме «предсказание-коррекция». На каждом шаге при помощи модели системы вычисляется новое положение робота, которое потом корректируется в соответствии с данными от сенсоров робота. Далее будут рассмотрены самые часто используемые на практике подходы: фильтр Калмана и фильтр частиц.

### **1.1 Фильтр Калмана**

Существует много реализаций фильтра Калмана, в робототехнике в основном используется расширенный фильтр Калмана (Extended Kalman Filter) [Einicke и др., 1999], который будет рассмотрен далее. EKF строится на знании о модели движения робота и её погрешности, а также о доступных сенсорных измерениях и их ошибках. Этап предсказания на каждом шаге работы, используя модель движения робота, определяет его смещение и выдвигает гипотезу о новом положении робота. Эта гипотеза также содержит компоненту погрешности. На этом же этапе вычисляется ковариационная матрица для экстраполированного вектора положения. Далее на этапе коррекции рассчитываются отклонения показаний сенсоров от ожидаемых экстраполированных значений для получения матрицы усиления, на основе которой вычисляется коррекция. Положение робота рассчитываются как предсказанное положение робота, сложенное с рассчитанной коррекцией. Матрица коэффициентов также обновляется.

Несмотря на свою распространённость, ЕKF имеет ряд недостатков: требуется знать модель робота с достаточно хорошей приближенностью; обязательна информация о погрешностях измерений; требуется серьезная перенастройка при меняющихся измеряемых данных; нелинейные задачи требуют перестроения матриц моделей на каждом шаге алгоритма.

## **1.2. Фильтр частиц**

Впервые понятие фильтр частиц было озвучено в 1996 году [Moral Del, 1996], в русскоязычной литературе еще встречается название многочастичный фильтр. Фильтр частиц относят к методам семейства Монте-Карло. Стартовая область заполняется случайным образом «частицами», которые являются моделями робота. Между частицами равномерно распределяется вес, в совокупности равный единице. Соответственно, когда робот совершает движение, совершает движение и каждая частица, меняя свое положение. Когда поступает информация, позволяющая уточнить свое положение, вес каждой частицы пересчитывается. Пересчет веса происходит обычно с использованием распределения Гаусса. Чем больше отдельно взятая частица удовлетворяет пришедшим данным, тем больше её вес. После перераспределения весов и их нормировки происходит главный «трюк» фильтра частиц, называющийся в англоязычной литературе *resampling*. В результате отбрасываются частицы с наименьшим весом, а частицы с большим весом копируются до изначального числа. Таким образом, в ходе нескольких итераций остаются только частицы в области истинного положения робота. Финальное положение робота вычисляется как сумма положений частиц, помноженных на свой вес.

В сравнении с фильтром Калмана фильтр частиц лучше справляется с нелинейными задачами. Однако среди недостатков данного метода можно отметить то, что они более «тяжелые» по вычислениям, ибо корректность фильтрации улучшается с ростом количества частиц. Поскольку инициализация фильтра и *resampling* выполняются при помощи генератора случайных чисел, это может сильно повлиять на стабильность метода. Также в случае если фильтр «пошел по ложному следу», ему очень сложно перестроиться на истинный путь.

## **2 Предлагаемый подход**

### **2.1 Общие понятия в недоопределенных вычислениях**

Недоопределённые модели (далее *n*-модели) были впервые предложены отечественным выдающимся ученым Александром Семеновичем Нариньяни в 1986 [Нариньяни, 1986]. Основная идея данного подхода заключается в том, чтобы не описывать всю исследуемую систему целиком,

а лишь её некоторую модель, которая задается набором переменных и функций. Также вводится понятие недоопределённой переменной (далее *n*-переменной), которая представляет собой некоторую область, в которой находится истинное значение переменной в её классическом понимании. Область может быть представлена как интервалом или мультиинтервалом, так и дискретным списком. *N*-переменные связаны между собой функциями присваивания (иногда встречаются как функции интерпретации). Модель представима в виде двудольного графа, где один набор вершин это *N*-переменные, а другой - функции присваивания. Когда в систему попадают новые данные извне, то они изменяют *N*-переменные, которые в свою очередь активируют функции, для которых они являются входными аргументами, функции изменяют другие *N*-переменные и т.д. В ходе вычислительного процесса область неопределённости переменных сужается и должна сходиться к своему точному значению.

Формально обобщенная вычислительная модель задаётся следующей четверкой [Нариньяни, 2007]:

$$M = (V, R, W, C), \quad (1)$$

где *V* – множество параметров из заданной области, *R* – множество ограничений, *W* – множество функций присваивания, *C* – множество функций проверки корректности. *N*-вычисления позволяют решать широкий круг задач, в т. ч. применялись для задачи навигации мобильного робота по набору маяков с известным направлениями на них [Карпов, 2009].

## 2.2 Недоопределенные вычисления для задачи локализации

Предложенный метод также относится к классу методов «предсказание-коррекция» и параллельно рассчитывает положение робота, определяемое его координатами, и ограничивающую область, заданную *n*-переменными. Ограничивающая область - это область, в которой потенциально может находиться робот, основываясь на начальных и входных данных. При поступлении данных о перемещении положение робота и ограничивающая область изменяются. Далее на основе данных о наблюдаемых ориентирах ограничивающая область изменяется, и если измененное положение робота выходит за пределы ограничивающей области, то оно корректируется на максимально близкое ему значение, принадлежащее области ограничения.

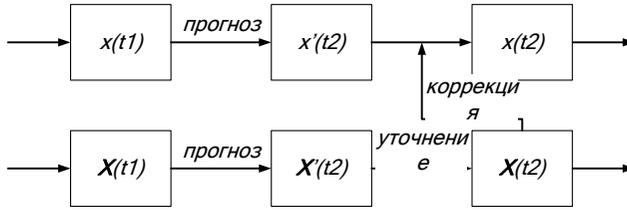


Рис. 1. Схема работы метода локализации на основе n-моделей

В двумерном случае положение робота  $x_r$  задается тремя переменными: координатами на плоскости ( $x$ ,  $y$ ) и углом поворота  $\alpha$ . Ограничивающая область  $X_r$  задается тремя N-переменными: координата  $*x$ , координата  $*y$  и угол поворота робота  $*\alpha$ . Каждая N-переменная представляет собой непрерывный интервал от минимально возможного значения до максимально возможного значения.

$$X_r = \begin{pmatrix} x_{min} & x_{max} \\ y_{min} & y_{max} \\ \alpha_{min} & \alpha_{max} \end{pmatrix} \quad (2)$$

На старте метода положение робота должно быть инициализировано либо его примерным положением, либо, если таковое не известно, то всей зоной, в которой оперирует робот. Ориентация робота в отсутствии знаний о стартовом положении задается от  $-\pi$  до  $\pi$ .

Данные о смещении робота за момент времени  $t_2-t_1$  представляют собой линейное  $dr$  и угловое  $da$  смещения. При смещении  $x_r$  изменяется следующим образом:

$$x_r' = (x + dr \cdot \cos(\alpha + da), y + dr \cdot \sin(\alpha + da), \alpha + da) \quad (3)$$

Когда совершается движение, предполагается, что робот перемещается в области, заданной смещением, измененным на относительную погрешность измерений  $\delta$ . Т.к. угол задается в некотором диапазоне значений, то область расширяется на линейное смещение, изменённое на погрешность измерений, в каждом направлении, принадлежащему диапазону углов.

$$X_r' = \bigcup_{\alpha = \alpha_{min}(1-\delta_\alpha)}^{\alpha_{max}(1+\delta_\alpha)} X_r + \begin{pmatrix} \min(dr \cdot \cos(\alpha) \cdot (1 \pm \delta_r)) & \max(dr \cdot \cos(\alpha) \cdot (1 \pm \delta_r)) \\ \min(dr \cdot \sin(\alpha) \cdot (1 \pm \delta_r)) & \max(dr \cdot \sin(\alpha) \cdot (1 \pm \delta_r)) \\ \min(da \cdot (1 \pm \delta_\alpha)) & \max(da \cdot (1 \pm \delta_\alpha)) \end{pmatrix} \quad (4)$$

Согласно обобщенной вычислительной модели, данная операция «раздвижения» области является функцией присваивания, обозначим её  $w^+$ .

На следующем этапе происходит процесс уточнения. В общем случае уточняющими данными может быть практически любая информация, получаемая от датчиков робота, которая каким-либо образом влияет на  $X_r$ . В данном случае происходит уточнение положения по данным маяков  $O$ , координаты которых известны, а расстояние до них определяется с известной погрешностью. Когда маяк попадает в «поле зрения» робота, то область максимального удаления от маяка  $X_o$  можно определить как квадрат со стороной, равной удвоенному расстоянию от маяка  $d$ , увеличенному на погрешность измерений  $\delta_d$ .

$$X_o = \begin{pmatrix} x_o - d(1 + \delta_d) & x_o + d(1 + \delta_d) \\ y_o - d(1 + \delta_d) & y_o + d(1 + \delta_d) \\ -\pi & \pi \end{pmatrix} \quad (5)$$

Соответственно, робот должен находиться как в области  $X_r$  так и в области  $X_o$ , а это значит, что он находится на пересечении данных областей, если таковое существует. Для определения области пересечения также требуется определить соответствующую функцию присваивания  $w^*$  из обобщенной вычислительной модели (1). Если пересечение областей существует, то функция присваивания  $w^*$  его определяет, в Н-вычислениях это называется сужением области неопределённости. В противном случае берется объединение областей, т.к. неопределённость не удалось сузить. Аналогично происходит и с диапазоном углов. После данной операции Н-переменные могут представлять собой мультиинтервалы.

$$w^*(X_1, X_2) = \begin{cases} X_1 \cap X_2, & X_1 \cap X_2 \neq \emptyset \\ X_1 \cup X_2, & X_1 \cap X_2 = \emptyset \end{cases} \quad (6)$$

$$X_1 \cap X_2 = \begin{pmatrix} \max(x_{min}^1, x_{min}^2) & \min(x_{max}^1, x_{max}^2) \\ \max(y_{min}^1, y_{min}^2) & \min(y_{max}^1, y_{max}^2) \\ \max(\alpha_{min}^1, \alpha_{min}^2) & \min(\alpha_{max}^1, \alpha_{max}^2) \end{pmatrix} \quad (7)$$

$$X = \emptyset \Leftrightarrow \begin{cases} x_{min} > x_{max} \\ y_{min} > y_{max} \\ \alpha_{min} > \alpha_{max} \end{cases} \quad (8)$$

$$X_1 \cup X_2 = \langle X_1, X_2 \rangle \quad (9)$$

Относительно мультиинтервалов все рассматриваемые операции с n-переменными являются дистрибутивными. Поскольку разрабатываемый метод предполагается использовать совместно с системой видеораспознавания ориентиров, то ориентир наблюдаем, только когда он

попадает в поле зрения камеры. Следовательно, для каждой области можно определить растров углов, под которыми может быть наблюдаем каждый ориентир. Вводится ещё одна функция присваивания  $w^<$ , которая пересчитывает  $n$ -переменную угла в зависимости от маяка-положения ориентира. Для крайних точек области определяются углы, под которыми виден тот или иной ориентир, образуя максимально возможный интервал. Т.к. может быть видно несколько ориентиров одновременно, то будет получено столько же углов, под которыми эти ориентиры видны, каждому углу соответствует своя область:

$$w^<(X, \{O_i\}) = \langle X_j \rangle \quad (10)$$

Также у области, для которой выполняется пересчет углов, имеется свой стартовый угол до пересчета. Чтобы привести все эти области в один мультиинтервал без пересечений, опять используется функция  $w^+$  (4).

Когда была определена максимально возможная область нахождения робота, приходится переходить к процедуре дискретизации, т.к.  $n$ -переменные в данной задаче имеют нелинейную зависимость, и тем самым не позволяют сузить область до приемлемых значений. Для того, чтобы решить эту проблему, полученная область равномерно разбивается на  $N^*M$  равных областей, операцией  $w^\#$ . Далее каждая из полученных областей проверяется на соответствие расстоянию до маяка и на соответствие диапазону углов. Области, которые не удовлетворяют данным критериям, исключаются из выборки. Обозначим эту функцию присваивания  $w$ . Области, которые «пережили» отбор, образуют мультиинтервал уточненного положения робота. На этом этапе полученный мультиинтервал и рассчитанное ранее положение робота  $x'_r$  (3) обрабатываются функцией проверки коррекции  $c$  (1). Соответственно, если рассчитанное принадлежит полученному мультиинтервалу, то оно сохраняется. В противном случае положение робота изменяется на значение, принадлежащее итоговому мультиинтервалу, максимально приближенное по расстоянию и углу к рассчитанному значению  $x'_r$ .

$$c(x'_r, X) = \begin{cases} x'_r, x'_r \in X \\ x_r, x'_r \notin X, x_r \in X, |x_r - x'_r| = \min \end{cases} \quad (11)$$

где  $|x_r - x'_r|$  - расстояние между точками. Полученный мультиинтервал объединяется одной областью операцией  $w^*$ , которая содержит в себе все элементы мультиинтервала, эта область будет являться входными данными для следующей итерации алгоритма:

$$X_r^{t_2} = w^* \cdot w^- \cdot w^\# \cdot w^< \cdot w^\times (X_o^{t_2}, w^+(X_r^{t_1})) \quad (12)$$

### 3 Эксперименты

Эксперименты проводились на модели, написанной в среде Python с использованием библиотеки Python Robotics [Sakai, 2018], робот двигался по замкнутой траектории радиусом 10 м в среде, в которой было размещено восемь ориентиров. Гауссовый шум с нулевым средним, помноженный на стандартное отклонение, добавлялся в положение робота и определение расстояния до ориентиров для эмуляции погрешностей датчиков. Стандартное отклонение для эмуляции одометрии составляла 1м для расстояния и 0.5 радиан для угла, для определения расстояния стандартное отклонение составляло 0.2 м. Расстояние до ориентира поступало на вход методов локализации, когда ориентир попадал в поле зрения робота (угол раствора 120 градусов), и когда оно не превышало заданного максимума (10 м). Соответственно данные одометрии и о расстояниях подавались на вход трех методов: расширенный фильтр Калмана, фильтр частиц и предложенный метод на основе n-вычислений.

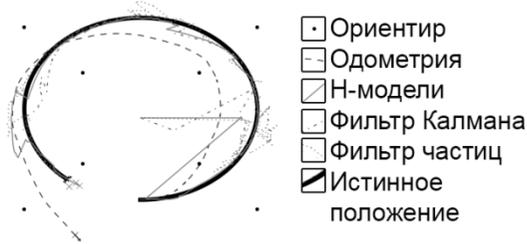


Рис. 2. Среда моделирования

Эффективность работы методов локализации сравнивалась по отклонению с истинным значением.

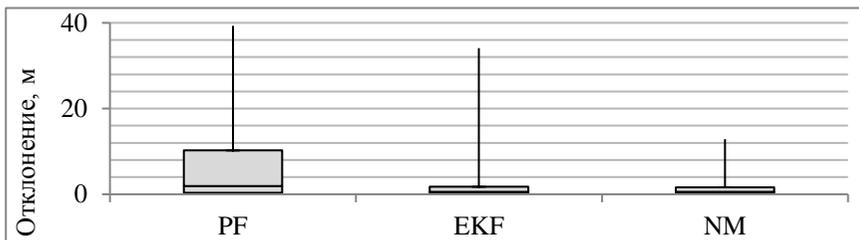


Рис. 3. Диаграмма размаха для сравниваемых методов локализации, PF – фильтр частиц, EKF – расширенный фильтр Калмана, NM – n-вычисления

Результаты отклонений приведены в виде диаграмм размаха, где «усуши» показаны максимальное и минимальное значение, средней линией

указана медиана, а высота столбиков соответствует дисперсии распределения. На диаграмме (Рис. 3) видно, что фильтр Калмана и предложенный метод на основе  $n$ -вычислений примерно схожи по медиане и дисперсии, однако у фильтра Калмана выше выбросы, которые происходят ввиду нелинейности рассматриваемой задачи. Фильтр частиц показывает хуже результаты в среднем. Но т.к. фильтр частиц опирается на генератор случайных чисел, то в одних случаях он показывает приемлемые результаты, а в других нет, на Рис. 4 представлены данные для «удачного» и «неудачного» случая на фоне средних данных для  $N$ -вычислений.

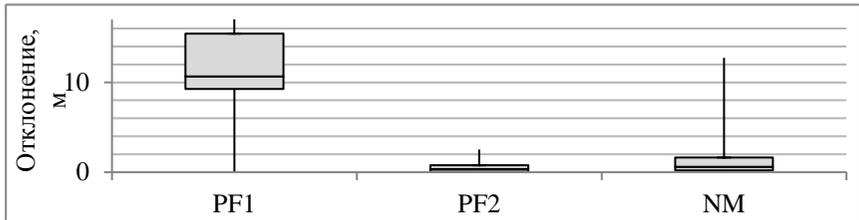


Рис. 4. "Удачный" (PF2) и "неудачный" (PF1) случай для фильтра частиц на фоне средних значений для  $n$ -вычислений (NM)

Как видно из графика, фильтр частиц для «удачного» случая превосходит другие методы локализации, однако такие результаты при количестве частиц равным 300 были достигнуты примерно в 50% случаев. При увеличении количества частиц в два раза, «неудачные» случаи были сокращены примерно до 35%, однако время работы алгоритма выросло более чем в два раза.

При рассмотрении поведения методов на отрезках с разным количеством видимых ориентиров следует, что для областей, где не видно ориентиров, фильтр Калмана показывает лучшие результаты, т.к. подстраивается под конкретную задачу, в то время как фильтр частиц и метод на  $N$ -вычислениях полагаются в этом случае полностью на одометрию.

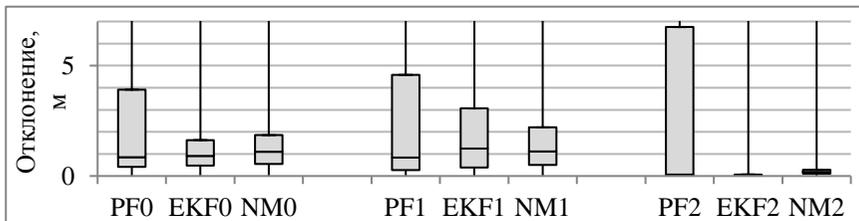


Рис. 5. Диаграмма разброса для областей с 0, 1 и 2 ориентирами

Для случая с одним ориентиром минимальный разброс у предложенного метода, для варианта с двумя ориентирами Н-модели показывают результат немного хуже, т.к. применяется механизм дискретизации с шагом 0.5 м.

Поставленные эксперименты в сравнении с фильтром Калмана и фильтром частиц выявили следующие особенности исследуемых методов:

Табл. 1. Сравнительная таблица методов локализации

Метод	ЕКФ	Фильтр частиц	Н-модели
Быстродействие	высокое	низкое	среднее
Чувствительность к стартовым усл.	низкое	низкое	низкое
Скорость перестройки при отклонении	высокая	низкая	средняя
Чувствительность к нелинейности	высокое	низкое	низкое
Стабильность	высокая	средняя	высокая
Расширяемость	низкая	высокая	средняя

## Заключение

Предложенный метод на основе парадигмы Н-вычислений на рассматриваемой задаче локализации по одометрии и видимым ориентирам показал большую стабильность (немного в угоду точности), чем фильтр частиц, и в то же время большую точность и меньшую чувствительность к нелинейности, чем расширенный фильтр Калмана.

## Список литературы

- [Einicke, White, 1999] Einicke G.A., White L.B. Robust Extended Kalman Filtering // IEEE Trans. Signal Process. 1999. Т. 47. № 9. с. 2596–2599.
- [Moral Del, 1996] Moral P. Del. Non Linear Filtering: Interacting Particle Solution // Markov Process. Relat. Fields. 1996. Т. 2. № 4. с. 555–580.
- [Sakai, 2018] Sakai A. Python Robotics [Электронный ресурс]. URL: <https://github.com/AtsushiSakai/PythonRobotics> (дата обращения: 17.02.2019).
- [Thrun, Wolfram, Fox, 2005] Thrun S., Wolfram B., Fox D. Probabilistic Robotics (Intelligent Robotics and Autonomous Agents). , 2005.
- [Карпов, 2009] Карпов В.Э. О некоторых особенностях применения недоопределенных моделей в робототехнике // Международная научно-практическая конференция «Интегрированные модели и мягкие вычисления в искусственном интеллекте» (28-30 мая 2009) Сб. научных трудов.Т.1. М.: Физматлит. , 2009. с. 520–532.
- [Нариньяни, 1986] Нариньяни А.С. Недоопределенность в системах представления и обработки знаний // Известия АН СССР. Техн. кибернетика. 1986. Т. 5.
- [Нариньяни, 2007] Нариньяни А.С. Введение в недоопределенность // Информационные технологии. 2007. № 4. с. 1–32.