

Ван Гуоянь, аспирант, guoyan@yandex.ru, А. В. Фомичев, канд. техн. наук, доц., a.v.fomichev@bmstu.ru,
Ду Ижань, магистр,
Московский государственный технический университет им. Н. Э. Баумана
(национальный исследовательский университет), г. Москва

Исследование модификаций сглаживающего фильтра Гаусса для их применения совместно с методом SLAM

При проведении научных исследований на поверхности небесных тел Солнечной системы с помощью планетоходов для решения задач их навигации и построения карты местности в неизвестном пространстве можно применить метод одновременной локализации и построения карты — метод SLAM. В случае применения лазерного дальномера в системе навигации в качестве широко используемых методов фильтрации информации используют методы, основанные на фильтре Калмана, а также его модификации, например, алгоритм EKF и алгоритм UKF. В алгоритме EKF присутствует процесс линеаризации. Алгоритм UKF по сравнению с алгоритмом EKF лучше согласован, но обладает большей вычислительной сложностью. Поэтому данные методы отличаются достаточно низкой точностью и большим объемом вычислений. Алгоритм фильтрации GP-RTSS, основанный на фильтре Гаусса, не требует процедуры прогнозирования выборки модели, линеаризации и численного интегрирования и способен адаптироваться для различных динамических систем. Таким образом, алгоритм фильтрации GP-RTSS существенно превосходит алгоритмы EKF и UKF по данным показателям. Самым большим препятствием в процессе работы алгоритма GP-RTSS является то, что функция ядра Гаусса требует множества вычислений. В статье исследуется модификация фильтра DIS RTSS, использующая схему распределенных вычислений, реализованную в гауссовском сглаживающем фильтре GP-RTSS для повышения скорости алгоритма GP-RTSS. Использование схемы распределенных вычислений позволяет провести эффективное преобразование данных, а следовательно, значительно уменьшить общие затраты на вычисления (время вычислений и требуемый объем памяти). Предложены четыре алгоритма распределенных вычислений для фильтра DIS RTSS: DIS RTP, DIS RTGP, DIS RTB, DIS RTrB. Моделирование показало, что среди данных алгоритмов алгоритм DIS RTGP является наиболее эффективным решением для рассматриваемого класса практических задач.

Алгоритм фильтрации DIS RTSS обладает высокой скоростью обработки информации и теоретически может обрабатывать бесконечное число выборок данных.

Ключевые слова: гауссовский процесс, распределенная система вычислений, алгоритм фильтрации, алгоритм DIS RTSS, алгоритм DIS RTP, алгоритм DIS RTGP, алгоритм DIS RTB, алгоритм DIS RTrB

Введение

Для обеспечения функционирования автономной системы управления планетоходом при проведении с его помощью научных исследований на поверхности небесных тел Солнечной системы необходимо разработать эффективные алгоритмы для системы навигации. Для решения данной задачи можно применить метод одновременной локализации и построения карты местности, который получил в литературе название SLAM (simultaneous localization and mapping) [1]. Метод SLAM используется в мобильных автономных средствах для построения карты в неизвестном пространстве или для обновления карты в заранее известном пространстве с одновременным контролем текущего местоположения объекта. Лазерный SLAM (используется информация от лазерных дальнометров, например, LRFs) имеет высокую точность при построении карт и локализации местоположения объекта и не имеет существенного накопления ошибок. При решении навигационных задач наряду с ме-

тодом лазерного SLAM широко используются различные методы фильтрации измерительной информации, например, фильтр Калмана (EKF и UKF) [2] или фильтр частиц, чтобы получить апостериорное распределение позиций и ориентиров для планетохода на карте. Так, в марсоходе Curiosity (Канада) использовался метод SLAM с алгоритмом расширенного фильтра Калмана (EKF-SLAM) [3]. Однако на практике у этих методов имеются существенные проблемы. В алгоритме EKF присутствует процесс линеаризации, поэтому он менее точен. Алгоритм UKF по сравнению с алгоритмом EKF лучше согласован, но обладает большей вычислительной сложностью. Метод фильтра частиц имеет проблему, связанную с деградацией частиц. Поэтому данные алгоритмы не могут использоваться в режиме реального времени.

В последние годы разработчики в области робототехники, управления и машинного обучения [4–7] большое внимание уделяют использованию фильтра Гаусса для обработки информации и вычислений.

Как показано в статье [8], в отличие от вышеупомянутых методов нелинейный алгоритм фильтра Гаусса со сглаживанием GP-RTSS (Rauch Tung Stried smoother) позволяет получить в аналитическом виде формулу для процесса фильтрации, где не требуются процедуры прогнозирования выборки модели, линеаризации и численного интегрирования. Кроме того, в данной модификации фильтра Гаусса отсутствует проблема деградации частиц. В статьях [8, 9] показано, что алгоритм GP-RTSS по сравнению с алгоритмами EKF, UKF, SKF (кубический фильтр Калмана) обладает меньшей неопределенностью и несогласованностью, более устойчив и способен адаптироваться в различных динамических средах. Поэтому алгоритм GP-RTSS является хорошим вариантом для использования в неизвестной среде совместно с системой SLAM.

Однако у алгоритма GP-RTSS имеется существенный недостаток. Поскольку в основу данного алгоритма фильтрации положен полный гауссов процесс, то это предполагает большой объем вычислений ядра Гаусса, в то время как для планетохода при решении задач навигации и управления в реальном режиме времени требуется как можно меньше затрат времени на вычисления.

Для устранения данного недостатка в статье предлагается использовать систему распределенных вычислений для алгоритма фильтрации GP-RTSS, в результате чего был разработан новый алгоритм DIS RTSS. Этот алгоритм может более эффективно использовать ресурсы данных для обеспечения процесса фильтрации, а также позволяет значительно снизить объем вычислений.

Описание алгоритма фильтра

Гауссовский процесс (далее — ГП) определяется математическим ожиданием $\mu(\cdot)$, дисперсией $\sigma(\cdot)$ и положительной полуопределенной ковариационной функцией $K(\cdot, \cdot)$, которая называется ядром.

Если имеется некоторый набор данных $D = (x, y)$, то в начале его необходимо преобразовать одним из известных способов к виду гауссовского процесса:

$$y_* = h(x_*) + \varepsilon, \varepsilon \sim N(0, \sigma_\varepsilon^2), \quad (1)$$

где x_* — вектор переменных состояния динамической системы; y_* — вектор выхода; $h(\cdot)$ —

функция, описывающая гауссовский процесс; ε — шум, описываемый нормальным законом распределения с нулевым математическим ожиданием и дисперсией σ_ε^2 .

Учитывая квадратичную зависимость в показателе экспоненты ядра в ГП, составленного из скалярных произведений пар векторов (x, x') , получим выражение для ковариационной функции в следующем виде:

$$K(x_*, x') = \alpha^2 \exp\left(-\frac{1}{2}(x_* - x')^T \Lambda (x_* - x')\right), \quad (2)$$

где Λ — диагональная матрица характеристических масштабов длины квадратного экспоненциального ядра, а α^2 — дисперсия скрытой функции h . Параметры Λ и α^2 являются гиперпараметрами, которые могут быть получены методом градиентного спуска.

Последующее предсказанное распределение значений функции $h_* = h(x_*)$ для вектора переменных состояния x_* является гауссовским процессом со следующим математическим ожиданием и дисперсией:

$$\mu_h(x_*) = M_h[h_*] = K_*^T (K' + \sigma_\varepsilon^2 I)^{-1} y = K_*^T \beta; \quad (3)$$

$$\sigma_h^2(x_*) = D_h[h_*] = K_{**} - K_*^T (K' + \sigma_\varepsilon^2 I)^{-1} K_*, \quad (4)$$

где $K_* = K(x, x_*)$, $K_{**} = K(x_*, x_*)$, $\beta = (K' + \sigma_\varepsilon^2 I)^{-1} y$, K' — матрица ядра с элементами $K'_{ij} = K(x_i, x_j)$; I — единичная матрица.

Алгоритм фильтра GP-RTSS

Пусть модель динамической системы описывается следующими уравнениями в дискретном виде:

$$x_t = f(x_{t-1}) + w; \quad (5)$$

$$y_t = g(x_t) + v, \quad (6)$$

где f, g — нелинейные функции; x_t — D -мерный вектор непрерывного многозначного состояния; y_t — D -мерный вектор выходных переменных; $w \sim N(0, \sigma_w)$ и $v \sim N(0, \sigma_v)$ — аддитивный белый гауссовский шум; t — дискретный индекс времени, $t = 1, \dots, T$.

В данной статье состояние системы представляется моделью марковского процесса, поскольку переменные скрытого состояния в момент времени t зависят только от момента времени $t - 1$.

Алгоритм GP-RTSS включает в себя три этапа: прогнозирование, обновление фильтра и сглаживание.

Этап 1. Прогнозирование. Априорное распределение вероятностей $p(x_t | y_{1:t-1})$ приближенно определяется следующим образом:

$$p(x_t | y_{1:t-1}) \approx N(\mu_{t|t-1}^x, \sigma_{t|t-1}^x), \quad (7)$$

где $y_{1:t-1}$ — состояние y в промежутке времени от $t = 1$ до $t - 1$; $\mu_{t|t-1}^x$ — математическое ожидание в момент времени t , полученное при известном математическом ожидании в момент времени $t - 1$;

$$\begin{aligned} \mu_{t|t-1}^x &= M_{x_{t-1}} [M_h [h(x_{t-1})] | \mu', \sigma'] = \\ &= \int \mu_h(x_{t-1}) N(h(x_{t-1}) | \mu', \sigma') dx_{t-1} = \beta^T L; \end{aligned} \quad (8)$$

$$(\sigma_{t|t-1}^x)^2 = \beta^T \tilde{L} \beta + \alpha^2 - \text{tr}[(K' + \sigma_\varepsilon^2 I)^{-1} \tilde{L}] - (\mu_{t|t-1}^x)^2. \quad (9)$$

В формулах (8) и (9) обозначено:

$$\begin{aligned} L &= [l_1, \dots, l_n]^T; \\ l_i &= \alpha^2 \left| \sigma' \Lambda^{-1} + I \right|^{-1/2} \times \\ &\times \exp \left(-\frac{1}{2} (x_i - \mu')^T (\sigma' + \Lambda)^{-1} (x_i - \mu') \right); \\ \tilde{L}_{ij} &= \frac{K(x_i, \mu') K(x_j, \mu')}{\left| 2\sigma' \Lambda^{-1} + I \right|^{1/2}} \times \\ &\times \exp \left((\tilde{z}_{ij} - \mu')^T \left(\sigma' + \frac{1}{2} \Lambda \right)^{-1} \sigma' \Lambda^{-1} (\tilde{z}_{ij} - \mu') \right); \\ \tilde{z}_{ij} &= \frac{1}{2} (x_i + x_j); \end{aligned}$$

$\text{tr}[(K' + \sigma_\varepsilon^2 I)^{-1} \tilde{L}]$ — след матрицы $(K' + \sigma_\varepsilon^2 I)^{-1} \tilde{L}$; $p(h(x_*)) \sim N(\mu', \sigma')$ и μ', σ' определяются по формулам (3) и (4).

Этап 2. Обновление фильтра. Апостериорное распределение вероятностей $p(x_t | y_{1:t})$ определяется следующим образом:

$$p(x_t | y_{1:t}) = N(x_t | \mu_t^e, \sigma_t^e), \quad (10)$$

где

$$\mu_t^e = \mu_{t|t-1}^x + \sigma_{t|t-1}^{xy} (\sigma_{t|t-1}^y)^{-1} (y_t - \mu_{t|t-1}^y); \quad (11)$$

$$\sigma_t^e = \sigma_{t|t-1}^x - \sigma_{t|t-1}^{xy} (\sigma_{t|t-1}^y)^{-1} (\sigma_{t|t-1}^{yx})^T. \quad (12)$$

В формуле (11), (12) обозначено:

$$\sigma_{t|t-1}^{xy} = M_{x,g} [xy^T] - \mu_{t|t-1}^x (\mu_{t|t-1}^y)^T;$$

$$M_{x,g} [xy^T] = \sum_{i=1}^n \beta_i \int x c_1 N(x | x_i, \Lambda_i) N(x | \mu_{t|t-1}^x, \sigma_{t|t-1}^x) dx,$$

c_1 — константа.

Этап 3. Сглаживание. Чтобы получить более точные результаты, необходимо выполнить процедуру сглаживания гауссовой системы.

Цель сглаживания — повышение качества идентификации за счет повторного использования данных наблюдений, причем в обратном порядке от x_t до x_1 . После объединения результатов обновления фильтра $p(x_t | y_{1:T})$ (см. выражение (10)) и окончательных результатов этапа сглаживания (для $t = T, \dots, 1$) получим:

$$p(x_{t-1} | y_{1:T}) = N(x_{t-1} | \mu_{t-1|T}^R, \sigma_{t-1|T}^R), \quad (13)$$

где

$$\mu_{t-1|T}^R = \mu_{t-1|t-1}^e + J_{t-1} (\mu_{t|T} - \mu_{t|t}^e); \quad (14)$$

$$\sigma_{t-1|T}^R = \sigma_{t-1|t-1}^e + J_{t-1} (\sigma_{t|T} - \sigma_{t|t}^e) J_{t-1}^T; \quad (15)$$

$$J_{t-1} = \sigma_{t-1,t|t-1} (\sigma_{t|t-1}^x)^{-1}; \quad (16)$$

$$\begin{aligned} \sigma_{t-1,t|t-1} &= \\ &= \sum_{i=1}^n \beta_i^x l_i^x \sigma_{t-1|t-1}^x (\sigma_{t-1|t-1}^e + \Lambda)^{-1} (x_i - \mu_{t-1|t-1}^e). \end{aligned} \quad (17)$$

Алгоритм фильтра GP-RTSS имеет меньшую неопределенность и несогласованность, а также способен к адаптации для различных динамических систем. Однако при применении фильтра GP-RTSS возникает серьезная проблема, заключающаяся в том, что функция ядра Гаусса требует большого объема вычислений. Чтобы уменьшить объем вычислений гауссовского процесса, можно, например, использовать разреженный гауссовский процесс [10]. Однако даже при разреженных аппроксимациях нецелесообразно применять ГП к данным, имеющим размерность десятков или сотен миллионов точек. Поэтому далее в статье показано, что для устранения описанной проблемы и уменьшения объема вычислений ГП можно использовать метод распределенных вычислений.

Модификация алгоритма GP-RTSS

Использование схемы распределенных вычислений, описанной в работе [11], позволяет

провести эффективное преобразование данных к виду ГП, следовательно, значительно уменьшить общие затраты на вычисления (время вычислений и требуемый объем памяти).

Согласно гипотезе максимума правдоподобия логарифм распределения вероятности $p(h|x, y)$ в гауссовой системе можно представить следующим образом:

$$\log p(y|x, \theta) \approx \prod_{j=1}^S \log p(y^{(j)}|x^{(j)}, \theta), \quad (18)$$

где j — число экспертов ($j = 1, \dots, S$); θ — гиперпараметр ядра, включающий Λ и α^2 . Эти S экспертов пользуются одним и тем же набором гиперпараметров. Здесь и далее $\log$ — любой логарифмический оператор, например $\ln$. Для произвольного j -го эксперта выражение для логарифма под знаком суммы в формуле (18) может быть вычислено аналитически:

$$\log p(y^{(j)}|x^{(j)}, \theta) = -\frac{1}{2} y^{(j)} (K_\phi^{(j)} + \sigma_\epsilon^2 I)^{-1} y^{(j)} - \frac{1}{2} \log(K_\phi^{(j)} + \sigma_\epsilon^2 I) + \text{const}, \quad (19)$$

где $K_\phi^{(j)} = K(x^{(j)}, x^{(j)})$ — ядро ГП.

Предположим, согласно статье [11], что для использования алгоритма GP-RTSS имеется множество данных $D = (x, y)$ размерностью N . Полное множество данных $D = (x, y)$ можно разделить на S множеств $D^{(j)} = \{x^{(j)}, y^{(j)}\}$, ($j = 1, \dots, S$), где $D^{(j)}$ — локальное множество данных размером n , которое будем называть локальным экспертом в схеме распределенных вычислений.

На основе приведенной выше теории для увеличения скорости вычислений алгоритма GP-RTSS (рис. 1, а, см. третью сторону обложки), функционирующего совместно с системой SLAM, разработана новая модификация этого алгоритма DIS RTSS (рис. 1, б, см. третью сторону обложки), в основу которой положена схема распределенных вычислений. Конечный результат фильтрации получается в результате применения одного из методов слияния данных экспертов.

На рис. 1 обозначено: x — переменные состояния; y — переменные наблюдения; t — время; f — нелинейные функции движения; g — нелинейные функции наблюдений. Локальные эксперты используют один и тот же набор гиперпараметров θ в одном нелинейном процессе,

например, θ_f — гиперпараметр для функции движения f ; θ_g — гиперпараметр для функции наблюдений g ; j — число экспертов.

Сложность вычислений гауссовского процесса в основном определяется функцией ядра Гаусса. Для набора данных $D = (x, y)$, если использовать для их обработки алгоритм GP-RTSS, то из формул (8), (9), (11) и (12) следует, что сложность функции ядра равна $O(N^3)$ (рис. 2, а). Аналогично можно получить, что сложность функции ядра для алгоритма DIS RTSS равна $O(Sn^3)$, где S — число экспертов, n — число данных j -го эксперта (причем $S \ll N$, $n \ll N$) (рис. 2, б). Таким образом, как следует из сравнения рис. 2, а и 2, б, сложность при вычислении ядра Гаусса для алгоритма DIS RTSS уменьшается.

В данной работе для слияния результатов экспертов используются следующие четыре метода: метод произведения гауссовских

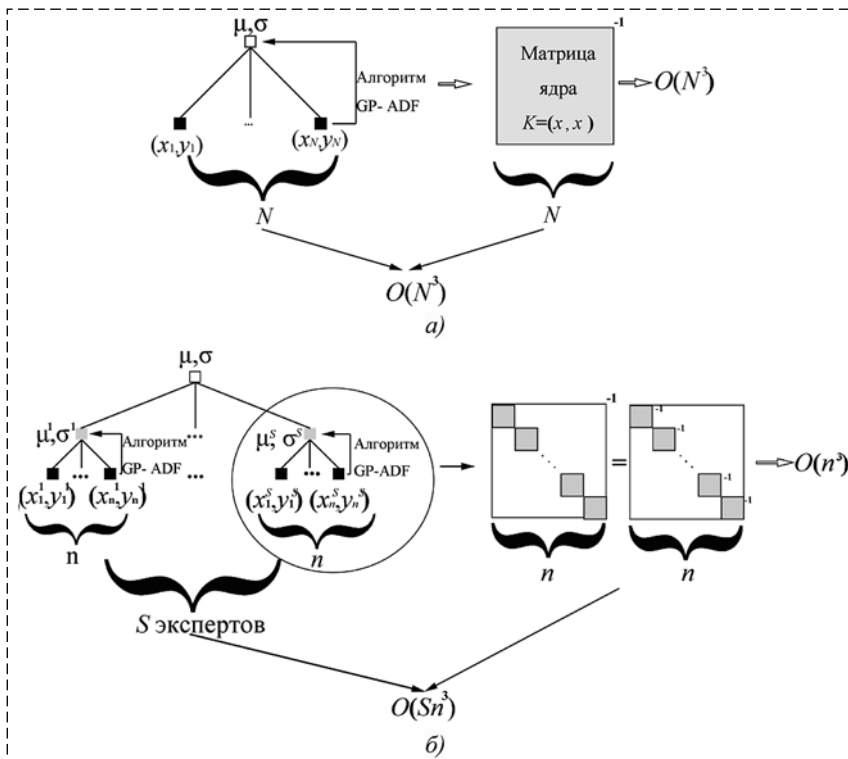


Рис. 2. Схемы вычислений матрицы ядра гауссовского процесса:

а — алгоритм GP-RTSS; б — алгоритм DIS RTSS

Fig. 2. The framework for the matrix of the kernel of the Gaussian process:

а — GP-RTSS algorithm; б — DIS RTSS algorithm

экспертов (PoE — Product of GP Experts) [12], метод обобщенного произведения гауссовских экспертов (gPoE — Generalised Product of GP Experts) [13], метод машины Байесовых комитетов (BCM — Bayesian Committee Machine) [14], метод робастной машины Байесовых комитетов (rBCM — Robust Bayesian Committee Machine) [15], которые в сочетании с алгоритмом DIS RTSS образуют четыре новые модификации исходного алгоритма: алгоритм DIS RTP, алгоритм DIS RTGP, алгоритм DIS RTB и алгоритм DIS RTTrB.

Алгоритм DIS RTP

Метод PoE позволяет вычислить целевое распределение вероятностей из формулы (18) как произведение множества конечных плотностей вероятностей в следующем виде:

$$p(y_* | x_*, (x, y)) = \prod_j p(y_*^{(j)} | x_*^{(j)}, (x^{(j)}, y^{(j)})).$$

Здесь и далее $\prod_{j=1}^S (\cdot) = \prod_{j=1}^S (\cdot)$.

В алгоритме DIS RTP для каждого j -го ($j = 1, \dots, S$) эксперта плотность вероятности вычисляется согласно следующему выражению:

$$p(y_*^{(j)} | x_*^{(j)}, (x^{(j)}, y^{(j)})) = N(y_*^{(j)} | \mu_j^R(x_*), \sigma_j^R(x_*)),$$

где $\mu_j^R(x_*)$, $\sigma_j^R(x_*)$ — математическое ожидание и дисперсия соответственно, полученные как результат экспертного прогноза, выполненного с помощью алгоритма GP-RTSS (см. выражения (14) и (15)).

Полученный при использовании модифицированного алгоритма DIS RTP результат можно представить выражением

$$p(y_* | x_*, (x, y)) = N(y_* | \mu^{RTP}, \sigma^{RTP}),$$

где математическое ожидание и дисперсия определяются следующим образом:

$$\begin{aligned} \mu^{RTP} &= (\sigma^{RTP})^2 \sum_j (\sigma_j^R)^{-2}(x_*) \mu_j^R(x_*); \\ (\sigma^{RTP})^{-2} &= \sum_j (\sigma_j^R)^{-2}(x_*). \end{aligned}$$

Здесь и далее $\sum_{j=1}^S (\cdot) = \sum_{j=1}^S (\cdot)$.

При $S = 1$ результаты, полученные с помощью алгоритма DIS RTP, будут идентичны результатам алгоритма GP-RTSS.

Алгоритм DIS RTGP

Метод gPoE согласно работе [13] позволяет рассчитать увеличение или уменьшение весового коэффициента эксперта β , а также вычислить целевое распределение вероятностей по формуле (18). Конечное значение плотности вероятностей определяется в следующем виде:

$$p(y_* | x_*, (x, y)) = \prod_j p^{\beta_j}(y_*^{(j)} | x_*^{(j)}, (x^{(j)}, y^{(j)})).$$

В алгоритме DIS RTGP для каждого j -го эксперта плотность вероятности вычисляется согласно выражению

$$p(y_*^{(j)} | x_*^{(j)}, (x^{(j)}, y^{(j)})) = N(y_*^{(j)} | \mu_j^R(x_*), \sigma_j^R(x_*)),$$

где $\mu_j^R(x_*)$, $\sigma_j^R(x_*)$ имеют смысл, описанный выше. Поскольку в алгоритме gPoE выполняется условие $\sum_j \beta_j = 1$, то значение весового коэффициента j -го эксперта устанавливается следующим образом: $\beta_j = 1/S$.

Полученный при использовании модифицированного алгоритма DIS RTGP результат можно представить выражением

$$p(y_* | x_*, (x, y)) = N(y_* | \mu^{RTGP}, \sigma^{RTGP}),$$

где

$$\begin{aligned} \mu^{RTGP} &= (\sigma^{RTGP})^2 \sum_j \beta_j (\sigma_j^R)^{-2}(x_*) \mu_j^R(x_*); \\ (\sigma^{RTGP})^{-2} &= \sum_j \beta_j (\sigma_j^R)^{-2}(x_*). \end{aligned}$$

Алгоритм DIS RTB

В отличие от метода POE метод BCM, который был предложен Треспом в работе [14], явно включает априорную вероятность $p(y_*)$ при объединении прогнозов. Метод BCM позволяет вычислить целевое распределение вероятностей по формуле (18) как произведение множества конечных плотностей вероятностей в следующем виде:

$$p(y_* | x_*, (x, y)) = \frac{\prod_j p(y_*^{(j)} | x_*^{(j)}, (x^{(j)}, y^{(j)}))}{p^{S-1}(y_* | x_*)}.$$

В модификации алгоритма DIS RTB для каждого j -го эксперта плотность вероятности рассчитывается согласно выражению

$$p(y_*^{(j)} | x_*^{(j)}, (x^{(j)}, y^{(j)})) = N(y_*^{(j)} | \mu_j^R(x_*), \sigma_j^R(x_*)),$$

где $\mu_j^R(*)$, $\sigma_j^R(*)$ имеют смысл, описанный выше.

Полученный при использовании модифицированного алгоритма DIS RTB результат можно представить следующим выражением:

$$p(y_* | x_*, (x, y)) = N(y_* | \mu^{RTB}, \sigma^{RTB}),$$

где

$$\begin{aligned} \mu^{RTB} &= (\sigma^{RTB})^2 \sum_j \beta_j (\sigma_j^R)^{-2}(x_*) \mu_j^R(x_*); \\ (\sigma^{RTB})^{-2} &= \sum_j \beta_j (\sigma_j^R)^{-2}(x_*) + (1 - S) \sigma_{**}^{-2}. \end{aligned}$$

Здесь σ_{**} — дисперсия априорной вероятности $p(y_*)$ алгоритма GP-RTSS.

Алгоритм DIS RTrB

Метод гВСМ является модификацией алгоритма ВСМ с дополнительной функцией, позволяющей увеличить или уменьшить весовой коэффициент эксперта β . По аналогии с методом ВСМ целевое распределение вероятностей для метода гВСМ вычисляется тоже по формуле (18). Тогда соотношение для расчета конечной плотности вероятностей имеет вид

$$p(y_* | x_*, (x, y)) = \frac{\prod_j p^{\beta_j}(y_*^{(j)} | x_*^{(j)}, (x^{(j)}, y^{(j)}))}{p^{-1 + \sum_j \beta_j}(y_* | x_*)}.$$

В модификации алгоритма DIS RTrB для каждого j -го эксперта плотность вероятности вычисляется согласно следующему выражению:

$$p(y_*^{(j)} | x_*^{(j)}, (x^{(j)}, y^{(j)})) = N(y_*^{(j)} | \mu_j^R(x_*), \sigma_j^R(x_*)),$$

где $\mu_j^R(*)$, $\sigma_j^R(*)$ имеют смысл, описанный выше.

Заметим, что в данном алгоритме не требуется соблюдение условия $\sum_j \beta_j = 1$, чтобы обеспечить реализацию вычислительного графа с несколькими слоями. Поэтому весовой коэффициент рассчитывается по формуле

$$\beta_k = \frac{1}{2} (\log \sigma_{**}^2 - \log \sigma_k^2(x_*)),$$

где σ_{**}^2 — дисперсия априорной вероятности $p(y_*)$ алгоритма GP-RTSS.

Полученный при использовании модифицированного алгоритма DIS RTrB результат можно окончательно представить следующим выражением:

$$p(y_* | x_*, (x, y)) = N(y_* | \mu^{RTrB}, \sigma^{RTrB}),$$

где

$$\begin{aligned} \mu^{RTrB} &= (\sigma^{RTrB})^2 \sum_j \beta_j (\sigma_j^R)^{-2}(x_*) \mu_j^R(x_*); \\ (\sigma^{RTrB})^{-2} &= \sum_j \beta_j (\sigma_j^R)^{-2}(x_*) + (1 - \sum_j \beta_j) \sigma_{**}^{-2}. \end{aligned}$$

Результаты моделирования

В данном разделе с помощью моделирования доказываем возможность реализации предлагаемого алгоритма с использованием нескольких наборов тестовых примеров.

Рассмотрим нелинейную стохастическую динамическую систему, которая описывается следующими уравнениями:

$$\begin{aligned} x_{t+1} &= \frac{4}{1 + x^2} + \sin(x) + w_t, \quad w_t \sim N(0; 0, 01^2); \\ y_t &= 3 \sin(x_t) + v_t, \quad v_t \sim N(0; 0, 01^2). \end{aligned}$$

Для сравнительной оценки результатов работоспособности алгоритмов было проведено вычисление средней квадратической ошибки RMSE (root-mean-square error) и отрицательной логарифмической вероятности NLL (negative log-likelihood). Показатель RMSE является показателем согласованности данных, а показатель NLL представляет собой совокупность неопределенности и согласованности.

При расчетах принималось: число данных $N = 1000$; число экспертов S варьировалось в пределах от 1 до 10. Каждый из трех параметров RMSE, NLL и время выполнения вычислений t по конкретному алгоритму рассчитывается как среднее значение результатов 50 независимых вычислений. В табл. 1—3 приведены результаты расчетов параметров RMSE, NLL и t для разных алгоритмов.

Из табл. 1 видно, что значения показателя RMSE, полученные с помощью алгоритма DIS RTSS (DIS RTSS: DIS RTP, DIS RTGP, DIS RTB, DIS RTrB), по сравнению со значением этого же показателя, но вычисленного по алгоритму GP-RTSS, почти одинаковы, с небольшой разницей. Из табл. 2 следует, что показатель NLL, полученный с помощью алгоритма DIS RTSS (DIS RTSS: DIS RTP, DIS RTGP, DIS RTB, DIS RTrB, размером $S = 2$, $S = 4$,

Таблица 1
Table 1Значение показателя RMSE
Value of the indicator RMSE

Число экспертов (S)	GP RTSS	DIS RTP	DIS RTGP	DIS RTB	DIS RTbB
$S = 1$	$0,0260 \pm 0,0081$	$0,0260 \pm 0,1270$	$0,0260 \pm 0,1270$	$0,0260 \pm 0,1270$	$0,0260 \pm 0,1270$
$S = 2$	$0,0240 \pm 0,0186$	$0,0244 \pm 0,1100$	$0,0244 \pm 0,1100$	$0,0244 \pm 0,1100$	$0,0244 \pm 0,1100$
$S = 4$	$0,0247 \pm 0,0134$	$0,0243 \pm 0,0299$	$0,0243 \pm 0,0299$	$0,0243 \pm 0,0299$	$0,0243 \pm 0,0299$
$S = 5$	$0,0220 \pm 0,0106$	$0,0240 \pm 0,0035$	$0,0240 \pm 0,0035$	$0,0240 \pm 0,0035$	$0,0240 \pm 0,0035$
$S = 8$	$0,0238 \pm 0,0150$	$0,0256 \pm 0,0066$	$0,0256 \pm 0,0066$	$0,0256 \pm 0,0066$	$0,0256 \pm 0,0066$
$S = 10$	$0,0252 \pm 0,0104$	$0,0321 \pm 0,0106$	$0,0321 \pm 0,0106$	$0,0321 \pm 0,0106$	$0,0321 \pm 0,0106$

Таблица 2
Table 2Значение показателя NLL
Value of the indicator NLL

Число экспертов (S)	GP RTSS	DIS RTP	DIS RTGP	DIS RTB	DIS RTbB
$S = 1$	$-2,790 \pm 0,140$	$-2,740 \pm 0,070$	$-2,740 \pm 0,070$	$-2,740 \pm 0,070$	$-2,740 \pm 0,070$
$S = 2$	$-2,860 \pm 0,586$	$-2,976 \pm 0,626$	$-2,872 \pm 0,362$	$-2,976 \pm 0,626$	$-2,872 \pm 0,362$
$S = 4$	$-2,910 \pm 0,502$	$-3,244 \pm 0,484$	$-2,956 \pm 0,304$	$-3,244 \pm 0,484$	$-2,956 \pm 0,304$
$S = 5$	$-2,930 \pm 0,502$	$-3,128 \pm 0,618$	$-2,992 \pm 0,358$	$-3,128 \pm 0,618$	$-2,992 \pm 0,358$
$S = 8$	$-3,160 \pm 0,838$	$-3,020 \pm 0,772$	$-3,080 \pm 0,418$	$-3,020 \pm 0,772$	$-3,080 \pm 0,418$
$S = 10$	$-2,900 \pm 0,346$	$-2,791 \pm 0,614$	$-2,840 \pm 0,360$	$-2,791 \pm 0,614$	$-2,840 \pm 0,360$

Таблица 3
Table 3Значение времени выполнения вычислений t
Time of calculations t

Число экспертов (S)	GP RTSS	DIS RTP	DIS RTGP	DIS RTB	DIS RTbB	EKF	UKF
$S = 1$	$52,1 \pm 2,83$	$54 \pm 5,41$	$54 \pm 5,41$	$54 \pm 5,41$	$54 \pm 5,41$	$25,3 \pm 1,42$	$28,3 \pm 1,42$
$S = 2$	$50,6 \pm 2,93$	$24,8 \pm 1,23$	$24,8 \pm 1,24$	$24,8 \pm 1,24$	$24,8 \pm 1,24$	$24,6 \pm 1,46$	$28 \pm 1,46$
$S = 4$	$53,4 \pm 2,75$	$10,46 \pm 0,364$	$10,46 \pm 0,364$	$10,46 \pm 0,364$	$10,46 \pm 0,364$	$26 \pm 1,35$	$29,5 \pm 1,35$
$S = 5$	$51,8 \pm 1,55$	$9,26 \pm 0,308$	$9,24 \pm 0,312$	$9,24 \pm 0,31$	$9,24 \pm 0,308$	$25,2 \pm 0,768$	$29,2 \pm 0,769$
$S = 8$	$53,6 \pm 2,64$	$9,96 \pm 0,115$	$9,92 \pm 0,152$	$9,92 \pm 0,152$	$9,92 \pm 0,152$	$26,11 \pm 1,34$	$31,6 \pm 2,27$
$S = 10$	$51,7 \pm 1,37$	$10,26 \pm 0,185$	$10,26 \pm 0,184$	$10,26 \pm 0,185$	$10,26 \pm 0,184$	$25,2 \pm 0,597$	$30,71 \pm 0,224$

$S = 5$), по сравнению с вычисленным по алгоритму GP-RTSS имеет меньшее значение, что демонстрирует превосходство распределенного метода Гаусса DIS RTSS.

В случае, когда происходит обработка одних и тех же данных, можно заметить, что чем больше число экспертов (S), тем лучше результат значений показателей RMSE и NLL. Когда данные имеют размерность $S = 8$ и $S = 10$, значение RMSE и NLL несколько хуже. Чтобы получить приемлемое значение RMSE и NLL, требуется задать разумное число экспертов.

Из табл. 3 видно, что в случае обработки одних и тех же данных время их обработки с помощью алгоритма DIS RTSS (DIS RTP, DIS RTGP, DIS RTB, DIS RTbB) значительно меньше, чем при использовании алгоритмов GP-RTSS, EKF и UKF. Однако при числе экспертов $S = 8$ и $S = 10$ время расчета по распределенному алгоритму Гаусса DIS RTSS уже не уменьшается. Полученный результат объясняется тем, что чем больше число независимых экспертов, тем больше времени потребуется для слияния их результатов обра-

ботки. Поэтому число экспертов требуется задавать разумно.

На основе результатов проведенного анализа, полученных выше, можно сделать важный вывод о том, что распределенный алгоритм Гаусса DIS RTSS имеет преимущества в показателях согласованности и неопределенности. Самое главное, что метод DIS RTSS может значительно уменьшить число вычислений и позволяет обрабатывать крупномасштабные массивы данных. Если правильно выбрать подходящее число экспертов, то преимущества вычислений с использованием данного алгоритма будут еще более очевидными.

Из рис. 3 (см. третью сторону обложки) видно, что дисперсии алгоритмов DIS RTP и DIS RTB меньше, чем дисперсия у алгоритма GP-RTSS. Дисперсия алгоритма DIS RTGP почти такая же, как дисперсия у алгоритма GP-RTSS. Дисперсия у алгоритма DIS RTrB существенно больше. Поэтому можно сделать вывод о том, что среди рассмотренных четырех алгоритмов алгоритм DIS RTGP является наилучшим выбором с учетом общей эффективности каждого алгоритма.

Выводы

Проблемы обработки больших объемов данных и их фильтрации в реальном режиме времени являются важными задачами для обеспечения позиционирования и навигации с помощью системы SLAM для мобильных роботов, особенно для планетоходов.

В данной статье исследованы четыре новые модификации алгоритма распределенного фильтра Гаусса DIS RTSS: DIS RTP, DIS RTGP, DIS RTB, DIS RTrB, которые позволяют обрабатывать большие объемы данных.

Результаты исследований и моделирование показало, что алгоритм DIS RTSS имеет свои преимущества по таким показателям, как точность, неопределенность и надежность. При его применении не требуется выполнения процедуры линеаризации в процессе разработки модели системы, кроме того, этот алгоритм свободен от явления деградации частиц. Поэтому алгоритм DIS RTSS значительно снижает зависимость расчетов от модели окружающей среды и может применяться для любой динамической системы.

Особенно важно отметить, что при использовании данного алгоритма значительно со-

кращается время вычислений, причем если рационально выбрать необходимое число экспертов, то преимущества при большем объеме вычислений будут еще более очевидными.

Результаты моделирования показали, что среди четырех модификаций алгоритма DIS RTSS алгоритм DIS RTGP является наиболее рациональным решением.

Список литературы

1. **Durrant W. H., Bailey T.** Simultaneous localization and mapping // IEEE robotics & automation magazine. 2006. Vol. 13. P. 99—110.
2. **Barfoot T. D.** State Estimation for Robotics. Cambridge University Press, 2017. University Printing House, Cambridge CB2 8BS, United Kingdom. 2017. P. 1—368.
3. **Ellery A.** Planetary rovers: robotic exploration of the solar system. Springer, 2016. P. 1—702.
4. **Achin J., Truong X. N., Morari M.** Learning and control using Gaussian processes // ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPs). IEEE, 2018. P. 140—149.
5. **Deisenroth M. P., Rasmussen C. E., Fox D.** Learning to control a low-cost manipulator using data-efficient reinforcement learning // Robotics: Science and Systems VII, 2011. P. 57—64.
6. **Kocijan J.** Modelling and control of dynamic systems using Gaussian process models. Springer International Publishing Switzerland, 2016. P. 1—281.
7. **Cao G., Lai E. M. K., Alam F.** Gaussian process model predictive control of unknown non-linear systems // IET Control Theory & Applications, 2017. Vol. 11 (5). P. 703—713.
8. **Deisenroth M. P., Turner R. D., Huber M. F. et al.** Robust filtering and smoothing with Gaussian processes // IEEE Transactions on Automatic Control, 2012. 57(7). P. 1865—1871.
9. **Arasaratnam I., Haykin S.** Cubature Kalman Filters // IEEE Transactions on Automatic Control. 2009. Vol. 54 (6), P. 1254—1269.
10. **Quinonero-Candela J., Rasmussen C. E.** A unifying view of sparse approximate Gaussian process regression // Journal of Machine Learning Research. 2005. Vol. 6 (12), P. 1939—1960.
11. **Фомичев А. В., Ван Гуоянь.** Эффективный алгоритм фильтра Гаусса для метода SLAM // 11-я Российская мультиконференция по проблемам управления: "Управление в аэрокосмических системах" (УАС-2018). СПб.: АО "Концерн "ЦНИИ "Электрон", 2018. С. 260—269.
12. **Nguyen Trung V., Bonilla E. V.** Fast allocation of Gaussian process experts // Proceedings of the International Conference on Machine Learning, 2014. P. 145—153.
13. **Cao Y., Fleet D. J.** Generalized product of experts for automatic and principled fusion of Gaussian process predictions. Preprint arXiv:1410.7827, 2014.
14. **Tresp V.** A Bayesian committee machine. Neural Computation, German, 2000.
15. **Deisenroth P. J., Ng W.** Distributed Gaussian processes. Preprint arXiv:1502.02843, 2015.

Research on Improved Gaussian Smoothing Filters for SLAM Application

Guoyan Wang, guoyan@yandex.ru, A. V. Fomichev, a.v.fomichev@bmstu.ru, Yiran Dy, devon.du@mail.ru, Bauman Moscow State Technical University, Moscow, 105005, Russian Federation

Corresponding author: **Fomichev Alexey V.**, Ph. D., Associate Professor, Bauman Moscow State Technical University, Moscow, 105005, Russian Federation, e-mail: a.v.fomichev@bmstu.ru

Accepted on June 25, 2019

Abstract

To address the navigation issues of the planetary rover and construct a map for the unknown environment as well as the surface of the planets in our solar system, the simultaneous localization and mapping can be seen as an alternative method. In terms of the navigation with the laser sensor, the Kalman filter and its improving algorithms, such as EKF and UKF are widely used in the process of processing information. Nevertheless, these filter algorithms suffer from low accuracy and significant computation expensive. The EKF algorithm has a linearization process, the UKF algorithm is better matched in a nonlinear system than the EKF algorithm, but it has more computational complexity. The GP-RTSS filtering algorithm, based on a Gaussian filter, is significantly superior to the EKF and UKF algorithms regarding the sensor fusion accuracy. The Gaussian Process can be used in different non-linear system, does not need prediction model and linearization. However, the main barrier in the process of implementing the GP-RTSS algorithm is that the Gaussian core function requires a lot of computation. In this paper, an algorithm, so-called DIS RTSS filter under a distributed computation scheme, derived from the GP-RTSS Gaussian smoothing and filter, is proposed. The distributed system can effectively reduce the cost of computation (computation expense and memory). Moreover, four fusion methods for the DIS RTSS filter, i.e., DIS RTP, DIS RTGP, DIS RTB, DIS RTrB are discussed in this paper. The experiments show that among the four algorithms described above, the DIS RTGP algorithm is the most effective solution for practical implementation.

The DIS RTSS filtering algorithm can realize a high processing rate and can theoretically process an infinite number of data samples.

Keywords: Gaussian process, distributed system, filtering algorithm, algorithm DIS RTSS, algorithm DIS RTP, algorithm RT GP, algorithm DIS RTB, algorithm DIS RTrB.

For citation:

Wang Guoyan, Fomichev A. V., Dy Yiran. Research on Improved Gaussian Smoothing Filters for SLAM Application, *Mekhatronika, Avtomatizatsiya, Upravlenie*, 2019, vol. 20, no. 12, pp. 756–764.

DOI: 10.17587/mau.20.756-764

References

1. Durrant W. H., Bailey T. Simultaneous localization and mapping, *IEEE robotics & automation magazine*, 2006, vol. 13, pp. 99–110.
2. Barfoot T. D. State Estimation for Robotics, Cambridge University Press, 2017, pp. 1–368.
3. Ellery A. Planetary rovers: robotic exploration of the solar system, Springer, 2016, pp. 1–702.
4. Achin J., Truong X. N., Morari M. Learning and control using Gaussian processes, *Submitted to ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPs)*. IEEE, 2018, pp. 140–149.
5. Deisenroth M. P., Rasmussen C. E., Fox D. Learning to control a low-cost manipulator using data-efficient reinforcement learning, *Science & Systems*, 2011.
6. Kocijan J. Modelling and control of dynamic systems using Gaussian process models, Springer, 2011, pp. 57–64.
7. Cao G., Lai E. M. K., Alam F. Gaussian process model predictive control of unknown non-linear systems, *IET Control Theory & Applications*, 2017, vol. 11(5), pp. 703–713.
8. Deisenroth M. P., Turner R. D., Huber M. F. et al. Robust filtering and smoothing with Gaussian processes, *IEEE Transactions on Automatic Control*, 2012, 57(7), pp. 1865–1871.
9. Arasaratnam I., Haykin S. Cubature Kalman Filters, *IEEE Transactions on Automatic Control*, 2009, vol. 54, no. 6, pp. 1254–1269.
10. Quinonero-Candela J., Rasmussen C. E. A unifying view of sparse approximate Gaussian process regression, *Journal of Machine Learning Research*, 2005, vol. 6(12), pp. 1939–1960.
11. Fomichev A. V., Wang Guoyan. Jeŝektivnyj algoritŝ filhtra Gaussa dlja metoda SLAM, *11 Rossijskaja mulhtik konferencija po problemam upravlenija "Upravlenie v ajerokosmichskih sistemah" (UAS-2018)*, 2018, pp. 260–269 (in Russian).
12. Nguyen Trung V., Bonilla E. V. Fast allocation of Gaussian process experts, *Proceedings of the International Conference on Machine Learning*, 2014, pp. 145–153.
13. Cao Y., Fleet D. J. Generalized product of experts for automatic and principled fusion of Gaussian process predictions, preprint arXiv:1410.7827, 2014.
14. Tresp V. A Bayesian committee machine, *Neural Computation*, 2000.
15. Deisenroth P. J., Ng W. Distributed Gaussian processes, Preprint arXiv:1502.02843, 2015.