

Тряпицын Валерий Леопольдович

инженер-программист, ООО «Фарватер», Санкт-Петербург, Россия. ORCID: 0009-0009-3866-2572
Электронный адрес: val20063@mail.ru

Valery L. Tryapitsyn

Software Engineer, Farvater LLC, Saint Petersburg, Russia. ORCID: 0009-0009-3866-2572
E-mail address: val20063@mail.ru

ПРЕДСКАЗАНИЕ И УТОЧНЕНИЕ ИОНОСФЕРНЫХ ПОПРАВOK
В РАДИОНАВИГАЦИИ НА ОСНОВЕ МАШИННОГО ОБУЧЕНИЯ
И ФИЛЬТРА КАЛМАНА С ИСПОЛЬЗОВАНИЕМ ДАННЫХ ИЗ МОДЕЛЕЙ
ЭЛЕКТРОННОЙ КОНЦЕНТРАЦИИ И РАДИОНАВИГАЦИОННЫХ
ИЗМЕРЕНИЙ

Аннотация. В статье рассматривается методика предсказания и уточнения ионосферных поправок в радионавигации с использованием машинного обучения и фильтра Калмана. В качестве первого источника данных применяются предсказанные значения вертикальной электронной концентрации ионосферы, полученные с помощью методов машинного обучения, что позволяет повысить точность оценки ионосферных поправок. Второй источник – значения, вычисляемые через накопление радионавигационных измерений в реальном времени, что обеспечивает актуальность и оперативность данных. Предложенная методика позволяет объединить преимущества моделирования и реальных измерений для повышения точности коррекции ионосферных эффектов. Результаты экспериментов демонстрируют эффективность подхода и его потенциал для применения в современных системах спутниковой навигации.

Ключевые слова: радионавигация, ионосферные поправки, машинное обучение, фильтр Калмана, электронная концентрация, моделирование ионосферы, радионавигационные измерения, навигационные системы.

Для цитирования: Тряпицын В.Л. Предсказание и уточнение ионосферных поправок в радионавигации на основе машинного обучения и фильтра Калмана с использованием данных из моделей электронной концентрации и радионавигационных измерений // Вестник Российского нового университета. Серия: Сложные системы: модели, анализ, управление. 2025. № 3. С. 15 – 25. DOI: 10.18137/RNU.V9187.25.03.P.15

PREDICTION AND REFINEMENT OF IONOSPHERIC CORRECTIONS
IN RADIO NAVIGATION BASED ON MACHINE LEARNING AND KALMAN
FILTER USING DATA FROM ELECTRON CONCENTRATION MODELS
AND RADIONAVIGATION MEASUREMENTS

Abstract. The article discusses a methodology for predicting and refining ionospheric corrections in radio navigation using machine learning and the Kalman filter. The first source of data is the predicted values of the vertical electron density of the ionosphere obtained using machine learning methods, which allows for

increasing the accuracy of ionospheric corrections. The second source is the values calculated through the accumulation of radio navigation measurements in real time, which ensures the relevance and efficiency of the data. The proposed methodology allows combining the advantages of modeling and real measurements to improve the accuracy of ionospheric effects correction. The experimental results demonstrate the effectiveness of the approach and its potential for application in modern satellite navigation systems.

Keywords: radio navigation, ionospheric corrections, machine learning, Kalman filter, electron density, ionospheric modeling, radio navigation measurements, navigation systems.

For citation: Tryapitsyn V.L. (2025) Prediction and refinement of ionospheric corrections in radionavigation based on machine learning and Kalman filter using data from electron concentration models and radionavigation measurements. *Vestnik of Russian New University. Series: Complex Systems: Models, analysis, management*. No. 3. Pp. 15 – 25. DOI: 10.18137/RNUV9187.25.03.P.15 (In Russian).

Введение

Современные системы спутниковой навигации, такие как GPS и ГЛОНАСС, широко используются в различных сферах деятельности – от транспорта и геодезии до навигации – в условиях сложной среды. Одним из ключевых факторов, влияющих на точность и надежность навигационных решений, является ионосфера, которая вызывает задержки радиосигналов и приводит к ошибкам в определении положения. Точные оценки ионосферных поправок являются необходимым условием повышения точности навигационных систем, особенно в условиях динамично меняющейся ионосферной среды [1].

На сегодняшний день существует множество методов моделирования и измерения электронной концентрации в ионосфере [2–4], однако их интеграция и использование для повышения точности коррекций остаются актуальными задачами. В последние годы особое внимание уделяется применению методов машинного обучения [5; 6] для предсказания ионосферных параметров, что позволяет получать более точные и своевременные оценки.

Целью данной работы является разработка и внедрение метода, объединяющего предсказания вертикальной электронной концентрации VTEC (Vertical Total Electron Concentration). VTEC показывает количество электронов в вертикальной колонне ионосферы над определенной точкой. Измеряется в единицах TECU (Total Electron Content Units), где $1 \text{ TECU} = 10^{16}$ электронов на квадратный метр. Высокие уровни VTEC могут вызывать значительные задержки и искажения радиосигналов, особенно в системах GPS и других спутниковых навигационных систем. Это происходит из-за преломления и рассеяния электромагнитных волн в ионосфере. Также они влияют на фазу и группу скорости радиоволн, что может приводить к ошибкам в позиционировании и навигации.

Метод машинного обучения очень востребован, поскольку совмещает в себе два подхода – прогнозирование на уровне машинного обучения и вычисление в реальном времени на основе текущих радионавигационных измерений [7]. Метод на основе машинного обучения имеет такие преимущества, как большой объем накопленных данных, вычисление правильных закономерностей и построение точных моделей предсказания [8]. Метод на основе текущих радионавигационных измерений позволяет гибко реагировать на изменения окружающей среды, которые приводят к отхождению от ранее рассчитанной модели, поскольку реальное радионавигационное измерение и накопление данных в реальном времени позволяют оперативно корректировать модели и повышать их точность.

Предсказание и уточнение ионосферных поправок в радионавигации на основе
машинного обучения и фильтра Калмана с использованием данных...

Использование фильтра Калмана в качестве инструмента интеграции этих источников данных позволяет повысить точность и надежность оценки ионосферных задержек [9], что особенно важно для повышения эффективности и безопасности навигационных систем в условиях изменяющейся ионосферной среды, поэтому в данном исследовании со-вмещены оба подхода.

Применение машинного обучения для прогнозирования VTEC

Обозначим основные шаги в фильтрации Калмана. Мы хотим учесть дополнительно предсказанное алгоритмом машинного обучения значение VTEC. Для этого в фильтр Калмана нужно включить его как часть состояния системы и использовать в моделировании динамики и измерений.

Включим расчетное значение VTEC как одну из переменных состояния и $VTEC_{pred}$ (предсказанное алгоритмом машинного обучения) – как другую в векторе состояния. В качестве начального приближения для следующего шага будем использовать предсказанное значение VTEC.

Производим инициализацию матрицы измерений: $x_0 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$ – начальное состояние.

Перед каждым шагом прогноза необходимо обновить состояние внешним предсказанием:

$\bar{x}_{i,k} \leftarrow \begin{bmatrix} VTEC_{pred}^{(k)} \\ \text{предыдущая_скорость} \\ \text{предыдущее_ускорение} \end{bmatrix}$, где i – номер столбца матрицы измерений x .

Инициализация матрицы динамики: $F = \begin{bmatrix} 1 & \Delta t & \frac{1}{2}\Delta t^2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix}$.

Эта матрица моделирует эволюцию системы с учетом равномерного ускорения: первая строка – позиция обновляется с учетом скорости и ускорения, вторая – скорость обновляется с учетом ускорения, третья – ускорение предполагается постоянным. Матрица описывает, как VTEC изменяется со временем, например, с учетом предсказанного значения. Это классическая модель для системы с постоянным ускорением, где Δt – дискретный шаг времени. В нашем случае он будет рассчитываться как разность между опорным временем (временем начала измерений) и временем текущего наблюдения.

Инициализация матрицы наблюдения: $H = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$.

Эта матрица связывает состояние системы с двумя источниками измерений – измерением VTEC и внешним предсказанием VTEC.

Далее для каждого шага времени $k = 1, 2, \dots, N$.

Предсказание состояния: $\hat{x}_{k|k} = F\hat{x}_{k-1|k-1} + u_k$. На этом шаге происходит прогноз следующего состояния системы с использованием модели динамики и предыдущего состояния, а также добавляется внешнее воздействие u_k .

Предсказание ковариационной матрицы ошибок: $P_{k|k-1} = FP_{k-1|k-1}F^T + Q$.

Это обновление оценки неопределенности системы после прогноза, учитывающее модельный шум Q .

$$\text{Получение измерений: } z_k = \begin{bmatrix} VTEC_{meas,k} \\ VTEC_{ext_pred,k} \end{bmatrix}.$$

На этом шаге получаем два источника данных: реальное измерение VTEC и внешнее предсказание VTEC, которые будут использованы для обновления.

$$\text{Предсказание измерений: } \hat{z}_{k|k-1} = H\hat{x}_{k|k-1}.$$

На этом шаге происходит расчет ожидаемых измерений на основе прогнозируемого состояния с использованием матрицы наблюдения H – это функция измерения, связывающая состояние с наблюдаемыми данными.

$$\text{Вычисление инновационной ковариационной матрицы: } S_k = HP_{k|k-1}H^T + R.$$

Обновление: после получения новых измерений необходимо скорректировать предсказанное значение VTEC с помощью фильтра Калмана, чтобы получить более точную оценку.

$$\text{Расчет коэффициента Калмана: } K_k = P_{k|k-1}H^T S_k^{-1}.$$

Этот коэффициент определяет, насколько сильно обновлять состояние, учитывая ошибку измерений и неопределенность модели.

$$\text{Обновление оценки состояния: } \hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k (z_k - \hat{z}_{k|k-1}).$$

Это шаг, в котором прогнозируемое состояние корректируется на основе разницы между фактическими измерениями и их предсказаниями, умноженной на коэффициент Калмана. Этот шаг позволяет системе учитывать новые данные и уменьшать неопределенность оценки.

$$\text{Обновление ковариационной матрицы ошибок: } P_{k|k} = (I - K_k H)P_{k|k-1}.$$

Проведем моделирование алгоритма. Оно нам позволит получить численный и графический результат. Получившуюся характеристику представим на Рисунке 1. На рисунке видно, каким образом в результирующем графике учитывается последовательность значений внешнего предсказания.

Для формирования локальной модели была разработана подпрограмма расчета, основанная на алгоритме, описанном в п. 1. Она написана на языке программирования C++, так как алгоритм накопления измерений работает в микроконтроллерном навигационном устройстве. Все основные математические функции для расчета ионосферных моделей сведены в класс ION. Опишем структуру программы, используя язык программирования.

При старте программы осуществляется вызов функции `ION::InitIonoMode(intperiod)`, которая применяется для старта. В качестве параметра ей передается период, в течение которого будет происходить наблюдение и запись результатов навигационных измерений. Данный параметр задается в настройках навигационного приемника.

Дальнейшая работа программы проходит внутри функции `voidION::IonoProc()`. Сначала происходит считывание значения времени из структуры, заполняемой при приеме данных от приемника, проверка его корректности. Далее при первом запуске происходит создание структуры станции мониторинга ионосферы для данного приемника:

`St0 = new STATION(nChan)`, где `STATION` – дружественный ION класс, `nChan` – номер канала (антенны).

Предсказание и уточнение ионосферных поправок в радионавигации на основе машинного обучения и фильтра Калмана с использованием данных...

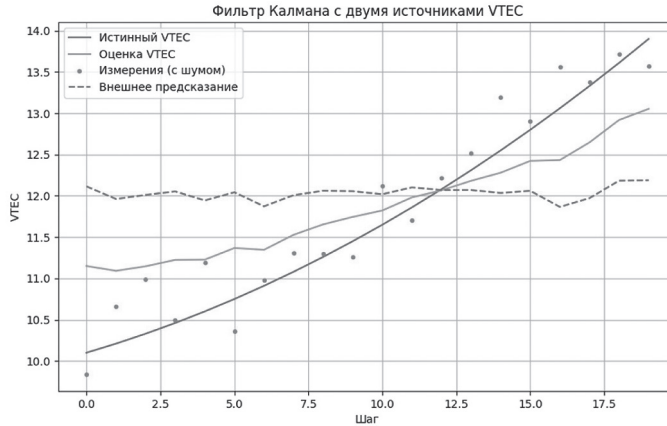


Рисунок 1. Характеристика фильтрации Калмана с двумя источниками VTEC

Источник: здесь и далее рисунки выполнены автором.

Осуществляется вызов функции *StartNavDataPeriod(*tep*)* для текущего момента времени *tep*, в которой фиксируется текущее время как время начала наблюдений, обнуляются маски наличия спутников по всем системам. Дальнейшие вызовы функций происходят уже в постоянном режиме. Происходит инициализация координат приемника для станции мониторинга:

```
St0->InitPecef();
```

Вызов функции измерений:

```
St0->MeasChan(tep);
```

В ней происходит запись измеренных значений *L* (фаза несущей частоты), *C* (измеренная псевдодальность), *D* (измеренный доплер) в структуру *OBSDATA*, созданную для каждого спутника. Данные значения получают посредством функции *GetIonoLCD()*, которая обеспечивает их получение из данных, полученных из приемника при его периодическом опросе. Далее в процедуре обработки новой обсервации *void STATION::MeasProcessing(const OBSDATA&Obs)* происходит запись данной обсервации в список сеансов измерений для данного спутника. Для этого используется *OBSDATA* – это структура данных по одному спутнику за одну эпоху (но по обеим частотам). Из набора этих структур по всем спутникам создается массив *Iep[nsys][NSat]*. Элемент массива – список сеансов измерений для одного спутника:

```
ISEANSseans(Obs,Interval);
```

```
std::list<ISEANS>&iepList = Iep[nsys][NSat];
```

где

ISEANS – класс, который определяет интервал, на котором производится непрерывная оценка параметров ионосферы;

Interval – дискретность измерений в сек.

Далее обрабатываются последующие измерения при помощи добавления их функцией *AddObservation(Obs)*, где *nsys* – номер навигационной системы; *nSat* – номер спутника.

В функции обработки обсервации *bool ISEANS::AddObservation(const OBSDATA&Obs)* проверяются выполнения условий допустимости значений измерений *C*, *D*, *L* и происхо-

дит их интегрирование по одной или двум частотам в структуре накопления АСС. Если условие не выполнено, то измерение идентифицируется как сбойное и происходит сброс интегратора. Вычисляется линейная комбинация одночастотных измерений:

$Z = (ACC.C1 - ACC.L1 * C / FreqL1) / (2.0 * dTacc)$, а если измерения двухчастотные, составляем двухчастотную комбинацию:

$$Z = -C * (ACC.L1 / FreqL1 - ACC.L2 / FreqL2) / ((1 - \text{sqr_freq12}) * dTacc),$$

где

$dTacc$ – время накопления измерений;

$FreqL1, FreqL2$ – первая и вторая несущая частота измерений;

sqr_freq12 – квадрат отношения $F1/F2$.

Это Z и будет являться оценкой ТЕС. ТЕС – это наклонная электронная концентрация, которая определяется вдоль направления на спутник, измерения которого используются. $VTEC$ определяется через нее так: $VTEC = ТЕС / FZ$, где FZ – функция наклона, пропорциональная высоте спутника [1]. Вычисление ТЕС далее описано через коэффициенты разложения локальной модели ионосферы.

Далее производится проверка на возможный срыв слежения или выброс, и если был срыв слежения, то происходит реинициализация фильтра Калмана, и измерения обрабатываются в алгоритме фильтрации Калмана, в которой участвуют следующие параметры:

$Const\ double\ **F$ – матрица прогноза.

Матрица наблюдений определяется как

$Const\ double\ H[Nsrc][SIZE_VIP] = \{\{1,0,0\}\{1,0,0\}\};$ // матрица наблюдений (частных производных),

где

$Nsrc = 2$ – это число источников данных. У нас их два (от радионавигационных измерений и предсказанное алгоритмом машинного обучения);

$SIZE_VIP = 3$ – размерность матрицы процесса (оценка ТЕС, скорость ТЕС и скорость изменения ТЕС);

R – дисперсия оценки.

Для инициализации точек прокола организован цикл по списку $NavDataList$, в котором в функции $boolSTATION::NewIPoint(const NAVDATA *SV)$ в каждую точку прокола из списка всех точек прокола ионосферы заносятся следующие вычисленные значения из $NavDataList$, где параметр SV – вектор состояния спутника:

$Ipp.push_back(IPPIERCE(SV->Id,Isf,Vsf,FZ,TEC,RMS));$

Vsf – скорость перемещения подыоносферной точки;

Igg – географические координаты подыоносферной точки;

Isf – координаты подыоносферной точки в фиксированной относительно Солнца системе координат;

$Angle$ – трехмерный угол направления на спутник;

FZ – фактор угла наклона;

RMS – среднеквадратическое отклонение.

Первоначальное приближение ТЕС в параметрах функции вычисляется как прогноз значения на момент T через начальные значения коэффициентов фильтра Калмана как $TEC = Xkalm[0] + dt * Xkalm[1]$, где dt – промежуток времени с момента последней observations. Далее вызывается процедура создания новой точки прокола, где создаем узлы

Предсказание и уточнение ионосферных поправок в радионавигации на основе
машинного обучения и фильтра Калмана с использованием данных...

в списке точек прокола ионосферы *ListNode*. При этом идет цикл от минимальной долготы *FirstLon* до максимальной *LastLon* с шагом 5° . *FirstLon* и *LastLon* определяются через *FirstObs* и *LastObs* соответственно при подстановке их в функцию *SunFixCoorIp()*. В функции *SunFixCoorIp()* к значению долготы в градусах прибавляется текущее время, переведенное в градусы.

```
FirstLon = SunFixCoorIp(St0->Pgeog,St0->FirstObs).lon;
LastLon = SunFixCoorIp(St0->Pgeog,St0->LastObs).lon;
где St0->Pgeog – географические координаты приемника.
```

После выполнения этих расчетов происходит вызов функции *BuildLIM()* для построения локальной модели ионосферы. Внутри этой функции строятся две модели разного порядка: *lim1* (первого порядка) и *lim2* (второго порядка). Проверяется условие того, что число принятых спутников не меньше достаточного количества для применения модели второго порядка. Если число спутников меньше 8, то понижаем порядок модели. Далее в цикле по всем подионосферным точкам

```
for(ipp = St0->Ipp.begin(); ipp != St0->Ipp.end(); ++ipp) { ... }
```

определяются квазиэкваториальные координаты и скорость точки прокола:

```
param3 D = QuasiEquatCoor(ipp.P, CurrIm.Psun);
param3 V = D - QuasiEquatCoor(ipp.P - ipp.V, CurrIm.Psun);
Здесь param3 – структура вида {X,Y,Z};
```

Далее для этих величин производится вычисление коэффициентов разложения локальной модели ионосферы в ряд Тейлора $H[0]$ - $H[5]$ по методу наименьших квадратов. Эта последовательность коэффициентов H идет в функцию фильтрации Калмана как строка частных производных параметра Z по оцениваемым параметрам.

Оценка ТЕС рассчитывается, таким образом, в соответствии с разложением Тейлора. Для порядка модели, равного 1:

```
if(Order == 1) {
F[] = {1,d.X,d.Y};
tec = Xkalm[0]+Xkalm[1]*F[1]+Xkalm[2]*F[2];
}
```

Для порядка модели, равного 2:

```
if(Order == 2) {
F[] = {1,d.X,d.Y,d.X*d.X,d.Y*d.Y,d.X*d.Y};
Tec= Xkalm[0] +Xkalm[1]*F[1]+ Xkalm[2]*F[2]+ Xkalm[3]*F[3]+ Xkalm[4] *F[4]+
Xkalm[5]*F[5];
}
```

Затем для заключительных этапов алгоритма ТЕС преобразуем в VТЕС, как описано выше.

Далее считываем значение предсказанной алгоритмом машинного обучения оценки VТЕС:

```
z_pred = GetTEC_FromBase(lon,lat, year, day, hour);
```

Здесь функция *GetTEC_FromBase()* считывает по заданным координатам широты, долготы, даты и времени значение VТЕС из сформированного предварительно табличного файла на нужный промежуток времени и участок местности по работе алгоритма предсказания и машинного обучения [2].

В вектор измерений z записываем реальное + внешнее предсказание:

$Z[] = \{Z_m, z_pred\};$

Берем эти измерения Z и делаем фильтрацию по Калману:

$F = GetMatrixForecast(Obs.Ter, Test);$ // матрица прогноз вектора состояния

где $Obs.Ter$ – время начала текущего наблюдения; $Test$ – опорное время для вычисления матрицы состояния фильтра Калмана; ему присваивается время первого наблюдения.

Далее находим R (оценка дисперсии):

$if (fSingleFreq) \{ // признак одночастотных измерений (по частоте L1)$

$interval = (Interval > 1) ? Interval : 1;$

$R = (rms_code) * (rms_code) / (4.0 * (dTacc / (interval)));$

$\} else \{$

$R = 4 / ((1 - \text{sqr_freq12}) * (1 - \text{sqr_freq12})) * (rms_phase) * (rms_phase) / (dTacc / (Interval));$

$\}$

Далее осуществляется фильтрация Калмана:

$Forecast(F).Filtr(Z, H, R);$

Результатом будет $Xkalm[0], Xkalm[1]$.

$Xkalm[0]$ перед следующим шагом присваиваем значение $Z[0]$, то есть это и есть искомая оценка VTEC. Для вычисления более точного значения можно применять формулы разложения локальной ионосферной модели в ряд Тейлора, используя модель первого или второго порядка [1]. Там будут использоваться приращения dX и dY по сетке квазиэкваториальных координат и, соответственно, значения $Xkalm[0], Xkalm[1]$ и $Xkalm[2]$.

Результаты работы

В результате мы получим распределение VTEC в виде географической карты по широте и долготе. Мы ее можем получить при помощи составления глобальной модели ионосферы (GIM) и передать из навигационного контроллера через интерфейс связи с внешним компьютером, используя специально разработанный протокол и информационные пакеты данных, так как мы это делали в работе [1], а на компьютере уже построить карты распределения VTEC.

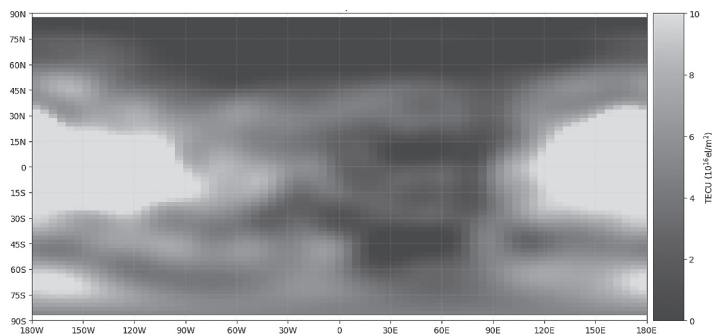


Рисунок 2. Карта распределения ионосферной электронной концентрации VTEC земной поверхности

Теперь мы можем скачать с сайта glonass-ias карту VTEC на выбранную дату и время и принять ее как эталонную для сравнения [2]. Далее вырезаем из этих обеих глобальных

Предсказание и уточнение ионосферных поправок в радионавигации на основе
машинного обучения и фильтра Калмана с использованием данных...

карт наш локальный участок местности, на котором проводилась обработка спутниковых измерений. Этот участок будет ограничен линиями по широте *FirstLat*, *LastLat* и долготы *FirstLon*, *LastLon*. Затем вычисляется корреляцию между этими двумя изображениями с использованием функции библиотеки *numpy* и *ssim*. Скачаем карты распределения VTEC для дат 15-03-2023 и 15-08-2023. Проведем расчеты для сравнения для пяти разных географических районов:

```
# grid_lat, grid_lon – двумерные массивы координат
# VTEC_origin_1, VTEC_res_1 – эталонная и сформированная по измерениям карты
для первой даты
# VTEC_origin_2, VTEC_res_2 – эталонная и сформированная по измерениям карты
для второй даты

# Создаем координатные сетки
latitudes = np.linspace(-90, 90, 180)
longitudes = np.linspace(-180, 180, 360)
grid_lat, grid_lon = np.meshgrid(latitudes, longitudes, indexing='ij')

# районы для анализа
districts = [
    {'FirstLat': -30.0, 'LastLat': 0.0, 'FirstLon': -60.0, 'LastLon': 0.0},
    {'FirstLat': 30.0, 'LastLat': 60.0, 'FirstLon': 60.0, 'LastLon': 120.0},
    {'FirstLat': -60.0, 'LastLat': -30.0, 'FirstLon': 120.0, 'LastLon': 180.0},
    {'FirstLat': 40.0, 'LastLat': 45.0, 'FirstLon': -10.0, 'LastLon': 10.0},
    {'FirstLat': -45.0, 'LastLat': -40.0, 'FirstLon': -120.0, 'LastLon': -115.0},
]
correlation_results = []

for idx, district in enumerate(districts):
    FirstLat = district['FirstLat']
    LastLat = district['LastLat']
    FirstLon = district['FirstLon']
    LastLon = district['LastLon']

# Создаем маски по координатам
lat_mask = (grid_lat >= FirstLat) & (grid_lat <= LastLat)
lon_mask = (grid_lon >= FirstLon) & (grid_lon <= LastLon)
region_mask = lat_mask & lon_mask

# Вырезаем области
VTEC_origin_crop_1 = VTEC_origin_1[region_mask]
VTEC_res_crop_1 = VTEC_res_1[region_mask]
VTEC_origin_crop_2 = VTEC_origin_2[region_mask]
VTEC_res_crop_2 = VTEC_res_2[region_mask]
```

вычисляем корреляцию методом SSIM

$$ssim_score_1 = ssim(VTEC_res_crop_1, VTEC_origin_crop_1)$$
$$ssim_score_2 = ssim(VTEC_res_crop_2, VTEC_origin_crop_2)$$

Результаты корреляции всех вариантов сравнения карт мы получили в диапазоне 0,78 ... 0,9. Это достаточно хороший результат, который показывает точность прогнозирования VTEC при помощи нашего метода.

Заключение

В данной статье предложен комплексный подход к предсказанию и уточнению ионосферных поправок в радионавигации, основанный на использовании методов машинного обучения и фильтра Калмана. Объединение предсказанных значений электронной концентрации, полученных с помощью моделей и алгоритмов машинного обучения, с реальными радионавигационными измерениями позволяет значительно повысить точность и оперативность оценки ионосферных эффектов. Проведенные эксперименты подтвердили эффективность предложенного метода, что делает его перспективным для внедрения в современные навигационные системы, обеспечивая более надежную и точную работу в условиях изменяющейся ионосферной среды.

Литература

1. Klobuchar J.A. Ionospheric Time-Delay Algorithm for Single-Frequency GPS Users // IEEE Transactions on Aerospace and Electronic Systems. 1987. Vol. 23. No. 3. P. 325–331. DOI: 10.1109/TAES.1987.310829
2. Anghel A., Carrano C. Kalman filter-based algorithms for monitoring the ionosphere and plasmasphere with GPS in near-real time // Journal of Atmospheric and Solar-Terrestrial Physics. 2009. Vol. 71. No. 1. P. 158–174. DOI: <https://doi.org/10.1016/j.jastp.2008.10.006>
3. Mannucci A.J., Wilson B.D., Yuan D.N., Ho C.H., Lindqwister U.J., Runge T.F. A global mapping technique for GPS-derived ionospheric total electron content measurements // Radio Science. 1998. Vol. 33. No. 3. P. 565–582. DOI: 10.1029/97RS02707
4. Arıkan F., Nayir H., Sezen U., Arıkan O. Estimation of single station interfrequency receiver bias using GPS-TEC // Radio Science. 2008. Vol. 43. No. 4. RS4004. DOI: 10.1029/2007RS003785
5. Zolesi B., Cander L.R. Total Electron Content Modelling and Mapping // Ionospheric Prediction and Forecasting. Springer Geophysics. Springer, Berlin, Heidelberg, 2014. P. 147–160. DOI: 10.1007/978-3-642-38430-1_7
6. Chen J., Gao Y. A Machine Learning-Based Tropospheric Prediction Approach for High-Precision Real-Time GNSS Positioning // Sensors. 2024. Vol. 24. Article no. 2957. DOI: <https://doi.org/10.3390/s24102957>
7. Tryapitsyn V.L., Dubinko T.Yu. Development and implementation of a method for transferring the parameters of ionospheric models for building a map of the electron concentration of the ionosphere in real time // AIP Conference Proceedings. 2021. Vol. 2402. No. 1. Article no. 020055. DOI: 10.1063/5.0071538
8. Zhang R., Li H., She Y., Yang J., Li W., Zhao D., Hu A. Deep Learning Applications in Ionospheric Modeling: Progress, Challenges, and Opportunities // Remote Sensing. 2025. Vol. 17. No. 1. Article no. 124. DOI: <https://doi.org/10.3390/rs17010124>

Предсказание и уточнение ионосферных поправок в радионавигации на основе
машинного обучения и фильтра Калмана с использованием данных...

9. Лаговский А.Ф., Мышерин А.В., Шагимуратов И.И. Анализ вычислительных трудностей при реализации фильтра Калмана для оценки полного электронного содержания ионосферы // Вестник Балтийского федерального университета им. И. Канта. Серия: Физико-математические и технические науки. 2007. № 10. С. 22–27. EDN IBZGQX.

References

1. Klobuchar J.A. (1987). Ionospheric Time-Delay Algorithm for Single-Frequency GPS Users. In: *IEEE Transactions on Aerospace and Electronic Systems*. Vol. 23. No. 3. Pp. 325–331. DOI: 10.1109/TAES.1987.310829
2. Anghel A., Carrano C. (2009) Kalman filter-based algorithms for monitoring the ionosphere and plasmasphere with GPS in near-real time. *Journal of Atmospheric and Solar-Terrestrial Physics*. Vol. 71. No. 1. Pp. 158–174. DOI: <https://doi.org/10.1016/j.jastp.2008.10.006>
3. Mannucci A.J., Wilson B.D., Yuan D.N., Ho C.H., Lindqwister U.J., Runge T.F. (1998) A global mapping technique for GPS-derived ionospheric total electron content measurements. *Radio Science*. Vol. 33. No. 3. Pp. 565–582. DOI: 10.1029/97RS02707
4. Arikan F., Nayir H., Sezen U., Arikan O. (2008). Estimation of single station interfrequency receiver bias using GPS-TEC. *Radio Science*. Vol. 43. No. 4. RS4004. DOI: 10.1029/2007RS003785
5. Zolesi B., Cander L.R. (2014) Total Electron Content Modelling and Mapping. In: *Ionospheric Prediction and Forecasting*. Springer Geophysics. Springer, Berlin, Heidelberg. Pp. 147–160. DOI: 10.1007/978-3-642-38430-1_7
6. Chen J., Gao Y. (2024) A Machine Learning-Based Tropospheric Prediction Approach for High-Precision Real-Time GNSS Positioning. *Sensors*. Vol. 24. Article no. 2957. DOI: <https://doi.org/10.3390/s24102957>
7. Tryapitsyn V.L., Dubinko T.Yu. (2021) Development and implementation of a method for transferring the parameters of ionospheric models for building a map of the electron concentration of the ionosphere in real time. In: *AIP Conference Proceedings*. Vol. 2402. No. 1. Article no. 020055. DOI: 10.1063/5.0071538
8. Zhang R., Li H., She Y., Yang J., Li W., Zhao D., Hu A. (2025) Deep Learning Applications in Ionospheric Modeling: Progress, Challenges, and Opportunities. *Remote Sensing*. Vol. 17. No. 1. Article no. 124. DOI: <https://doi.org/10.3390/rs17010124>
9. Lagovskii A.F., Mysherin A.V., Shagimuratov I.I. (2007) Analysis of calculative difficulties for application of Kalman filter to estimate electronic content of ionosphere. *Vestnik IKBFU. Physics, Mathematics, and Technology*. No. 10. Pp. 22–27. (In Russian).

Поступила в редакцию: 05.08.2025

Received: 05.08.2025

Поступила после рецензирования: 02.09.2025

Revised: 02.09.2025

Принята к публикации: 15.09.2025

Accepted: 15.09.2025