

Программная реализация нейросетевого алгоритма вычисления координат источника радиоизлучения с учетом ограничений по дальности прямой видимости с использованием библиотеки `alglib`

Э. А. Кирсанов, email: ekir74@mail.ru

Д. А. Ларионов

ВУНЦ ВВС «Военно-воздушная академия имени профессора Н.Е. Жуковского и Ю.А. Гагарина», (г. Воронеж)

***Аннотация.** Рассматривается программная реализация нейросетевого алгоритма вычисления координат источника радиоизлучения с учетом ограничений по дальности прямой видимости, выполненная на языке программирования C++ с использованием библиотеки `alglib` для угломерной системы местоопределения при произвольном числе пеленгаторов. Проводится сравнительный анализ нейросетевого алгоритма вычисления координат с алгоритмом вычисления координат по минимуму суммы квадратов расстояний до линий пеленгов.*

***Ключевые слова:** вычисление координат, многопозиционная радиосистема, нейросетевой алгоритм.*

Введение

При синтезе нейросетевых алгоритмов вычисления координат источника радиоизлучения (ИРИ) необходимо задавать зону ответственности (ЗО) многопозиционной радиосистемы (МРС), для которой генерируется обучающий набор пар «зашумленный вход – желаемый выход» нейронной сети (НС). При этом зашумленным входом является измеренный с ошибками вектор параметров положения, а желаемым выходом – вектор истинных координат ИРИ, размещенного внутри зоны ответственности МРС [1]. Однако при практическом использовании обученной НС на ее вход может поступить вектор параметров положения от ИРИ, расположенного за пределами ЗО МРС. Но соответствующая ему оценка координат ИРИ, вычисленная нейросетевым алгоритмом, ошибочно окажется внутри ЗО МРС. Поэтому необходимо проверять условие попадания значений координат ИРИ в ЗО. Такая проблема не возникает для МРС УКВ-диапазона, зона действия которых ограничена дальностью прямой видимости. В этом

случае в качестве ЗО МРС можно рассматривать зону прямой видимости (ЗПВ).

Априорная информация о ЗПВ учитывается в алгоритме вычисления координат ИРИ по максимуму апостериорной вероятности, практическая реализация которого сопряжена со значительными вычислительными затратами [1]. Поэтому на практике для угломерных МРС чаще всего используются не учитывающие информацию о ЗПВ алгоритмы вычисления координат на основе центра тяжести многоугольника засечек или по минимуму суммы квадратов расстояний (МСКР) до линий пеленгов, что позволяет им получать оценки координат ИРИ за пределами ЗПВ и значительно увеличивает ошибки местоопределения.

Естественный учет априорной информации о зоне прямой видимости возможен в нейросетевых алгоритмах за счет обучения НС по координатам ИРИ только из этой зоны. При этом наиболее мощные возможности по работе с нейронными сетями (в том числе глубокими) в настоящее время реализованы в Matlab и Python, но в них затруднено создание исполняемых exe-файлов.

Поэтому представляет интерес программная реализация нейросетевого алгоритма вычисления координат ИРИ на языке программирования C++ с использованием библиотеки `alglib` [2].

Целью настоящей работы является исследование возможностей программной реализации нейросетевого алгоритма вычисления координат ИРИ с учетом ограничений по дальности прямой видимости на языке программирования C++ с использованием библиотеки `alglib` для угломерной системы местоопределения и сравнительный анализ его эффективности с алгоритмом вычисления координат по минимуму суммы квадратов расстояний до линий пеленгов.

1. Постановка и решение задачи

Ограничимся рассмотрением задачи местоопределения на плоскости. Пусть заданы координаты пеленгаторов $\mathbf{x}_{p_i} = (X_{p_i}, Y_{p_i})^T$, $1 \leq i \leq N$ и среднеквадратичная ошибка (СКО) измерения пеленга σ_θ . Зона прямой видимости системы местоопределения определяется как множество точек с координатами $\mathbf{x} = (X, Y)^T$, расстояние от которых до всех пеленгаторов не превышает дальности прямой видимости R_{pw} , то есть $D = \{(X, Y) \mid \sqrt{(X - X_{p_i})^2 + (Y - Y_{p_i})^2} \leq R_{pw}, 1 \leq i \leq N\}$.

Построение нейросетевого алгоритма вычисления координат неподвижного ИРИ по однократно измеряемым пеленгам заключается в

выборе архитектуры НС, формировании обучающего множества и ее обучении. Общая структура нейросетевых алгоритмов вычисления координат ИРИ может быть получена на основе наиболее часто используемой многослойной сети с прямыми связями и нелинейными активационными функциями нейронов скрытых слоев [1]. Любая функция многих переменных может быть аппроксимирована многослойной нейронной сетью такого типа.

Архитектура НС определяется количеством слоев и их размерностью. Будем строить НС на основе трехслойной сети, что обычно обеспечивает упрощенный характер обучения НС и, в то же время, достаточно для решения любой нелинейной задачи. Первый и второй слой НС содержат по двадцать нейронов с нелинейными активационными функциями. Активационные функции нейронов выходного слоя выбираются линейными, что позволяет обойтись без масштабирования выходов. Архитектура НС представлена на рисунке 1, где каждой связи между входами и нейронами, а также нейронами скрытых слоев приписывается определенный вес (на рисунке веса не показаны).

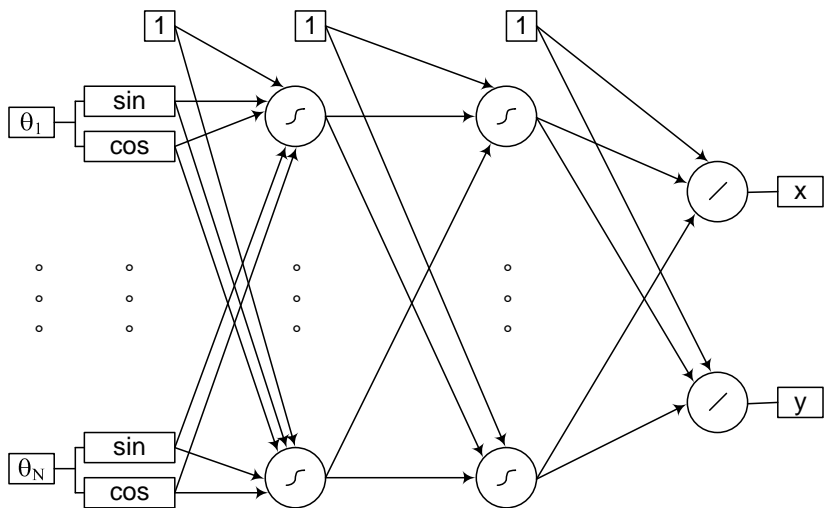


Рис. 1. Архитектура нейронной сети, реализующей алгоритм оценивания координат по измеренным значениям пеленгов

Формирование обучающей выборки сети проводится по следующей методике. Регулярно или случайно формируется заданное

количество точек из области D , затем для каждой такой точки при фиксированном значении СКО измерения пеленга в соответствии с методом имитационного моделирования формируются измеренные значения пеленгов θ_i , $1 \leq i \leq N$. Для преодоления проблемы цикличности на вход НС подаются синусы и косинусы измеренных значений пеленгов. При этом размерность входного слоя будет равна удвоенному значению числа пеленгаторов. Выходом НС является вектор координат ИРИ на плоскости, поэтому размерность выходного слоя равна двум. По сформированным таким образом наборам «вход-выход» осуществляется обучение НС, целью которого является нахождение такого набора весовых коэффициентов сети, при котором обеспечивается требуемая погрешность вычисления координат по обучающей выборке.

Библиотека `alglib` поддерживает предназначенные для решения задач классификации или регрессии нейронные сети перцептронного типа без скрытых слоев, с одним скрытым слоем и с двумя скрытыми слоями [2]. Скрытые слои имеют сигмоидальную функцию активации. Для выходного слоя предусмотрена возможность задания нескольких видов активационной функции, но в рассматриваемой регрессионной задаче она выбирается линейной. В библиотеке `alglib` реализован алгоритм обучения НС `L-BFGS` на основе квазиньютоновского метода Бройдена — Флетчера — Гольдфарба — Шанно с ограниченным использованием памяти.

Фрагмент кода программы, поясняющего процесс объявления и задания параметров НС приведен в листинге 1.

Листинг 1

Объявление и задание параметров НС

```
// параметры НС
int n; // колич. слоев НС (2 или 3)
int ns; // колич. нейронов в скр. слое НС
mlptrainer trn; // объект тренажера
multilayerperceptron net; // многослойный перцептрон
mlpreport rep; // отчет об обучении
real_2d_array xy; // массив обучающих данных НС
real_1d_array x; // массив входов НС
real_1d_array y; // массив выходов НС
double decay; // коэффициент регуляризации
double wstep; // мин. знач. шага прир. весов НС
ae_int_t maxits; // макс. колич. циклов обучения НС
ae_int_t NRest; // колич. перезапусков при обуч. НС

// задание исходных данных
n=3;
ns=20;
```

```

wstep=0.001;
decay=0.001;
maxits=300;
NRest=1;

xy.setlength(Now,2*N+2); // зад. размера массива обуч. данных
x.setlength(2*N);        // задание размера вектора входа НС
y.setlength(2);          // задание размера вектора выхода НС

```

Фрагмент кода программы, поясняющего процесс создания, обучения и тестирования НС приведен в листинге 2.

Листинг 2

Создание, обучение и тестирование НС

```

// создание объекта тренажера
mlpcreatetrainer(2*N,2,trn);
// загрузка тренировочного набора ху в объект тренажера
mlpsetdataset(trn,xy,Now);

// установка параметров обучения
// задание коэффициента регуляризации decay
mlpsetdecay(trn,decay);
// задание мин. значения шага приращения весов НС wstep
// и максимального количества циклов обучения НС maxits
mlpsetcond(trn,wstep,maxits);

if(n==2)
    // создание НС с одним скрытым слоем
    mlpcreate1(2*N,ns,2,net);
else if(n==3)
    // создание НС с двумя скрытыми слоями
    mlpcreate2(2*N,ns,ns,2,net);

// обучение НС
mlptrainnetwork(trn,net,NRest,rep);

for(p=0;p<Nkw;p++) // цикл по контрольной выборке
{
    // заполнение контрольного массива входных данных
    for(k=0;k<2*N;k++) x[k]=Pk[p][k];
    // тестирование НС
    mlpprocess(net,x,y);
    Xns[p]=y[0];
    Yns[p]=y[1];
}

```

В качестве альтернативы нейросетевому рассматривался наиболее просто реализуемый алгоритм местоопределения, в котором оценка координат ИРИ на плоскости ищется как точка, сумма квадратов

расстояний от которой до линий пеленгов минимальна. Согласно указанному алгоритму, оценка координат ИРИ определяется как [1]

$$\begin{aligned}
 x &= \frac{\sum_{i=1}^N B_i^2 \cdot \sum_{i=1}^N A_i C_i - \sum_{i=1}^N A_i B_i \cdot \sum_{i=1}^N B_i C_i}{\sum_{i=1}^N A_i B_i \cdot \sum_{i=1}^N A_i B_i - \sum_{i=1}^N A_i^2 \cdot \sum_{i=1}^N B_i^2}, \\
 y &= \frac{\sum_{i=1}^N A_i^2 \cdot \sum_{i=1}^N B_i C_i - \sum_{i=1}^N A_i B_i \cdot \sum_{i=1}^N A_i C_i}{\sum_{i=1}^N A_i B_i \cdot \sum_{i=1}^N A_i B_i - \sum_{i=1}^N A_i^2 \cdot \sum_{i=1}^N B_i^2},
 \end{aligned} \tag{1}$$

где $A_i = \cos \theta_i$, $B_i = -\sin \theta_i$, $C_i = Y_{P_i} \sin \theta_i - X_{P_i} \cos \theta_i$.

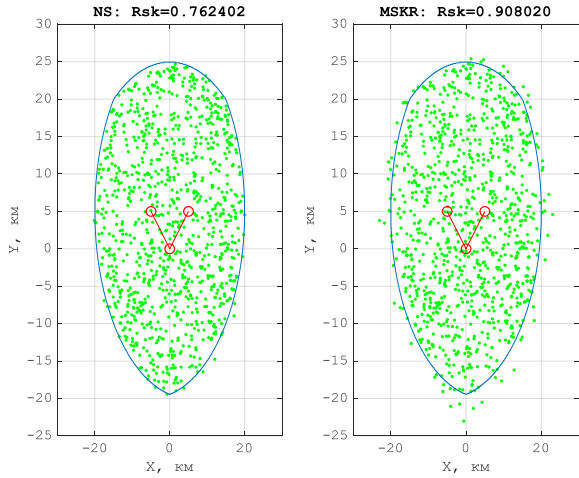
2. Результаты моделирования

Для оценивания эффективности рассмотренных алгоритмов проводилось статистическое моделирование результатов работы угломерной МРС, пеленгаторы которой размещались равномерно на полуокружности радиусом 5 км. Значение дальности прямой видимости задавалось равным $R_{pw} = 25$ км. Обучающая и контрольная целевые

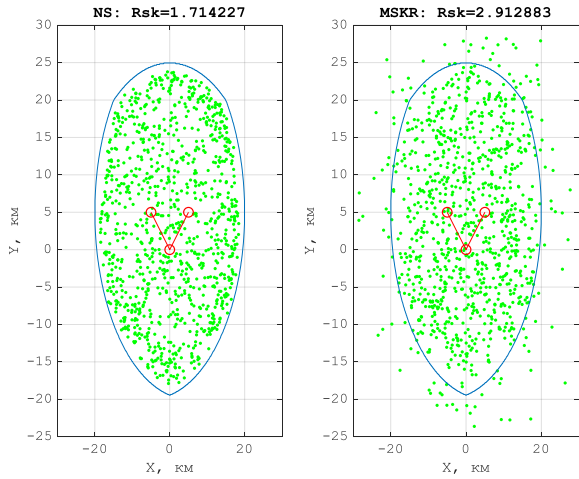
выборки НС объемом соответственно $N_{ow} = 10000$ и $N_{kw} = 1000$ формировалась при генерации координат ИРИ по равномерному закону внутри ЗПВ. В качестве показателя эффективности использовалась усредненная по ЗПВ среднеквадратичная ошибка места.

На рис. 2 и 3 приведен пример решения задачи местоопределения двумя рассмотренными алгоритмами для МРС с количеством пеленгатором соответственно $N = 3$ и $N = 4$ при ошибках измерения пеленга $\sigma_\theta = 1$ градус (а) и $\sigma_\theta = 3$ градуса (б). Рисунки слева соответствуют нейросетевому алгоритму, а справа – алгоритму МСКР. На указанных рисунках пеленгаторы показаны красными кружками, граница ЗПВ – синей линией, а оценки координат – зелеными точками.

Как видно из рисунков, оценки координат ИРИ, соответствующие нейросетевому алгоритму практически не выходят за пределы ЗПВ. Оценки координат ИРИ на основе алгоритма МСКР выходят за пределы ЗПВ. При этом нейросетевой алгоритм обеспечивает меньшую среднеквадратичную ошибку места по сравнению с алгоритмом МСКР. С увеличением количества пеленгатором точность местоопределения увеличивается.

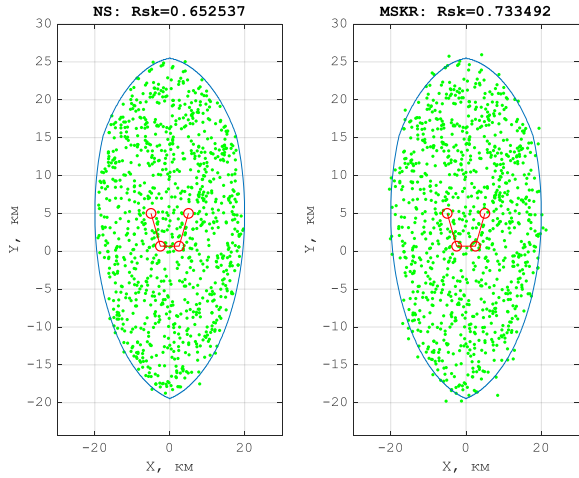


a

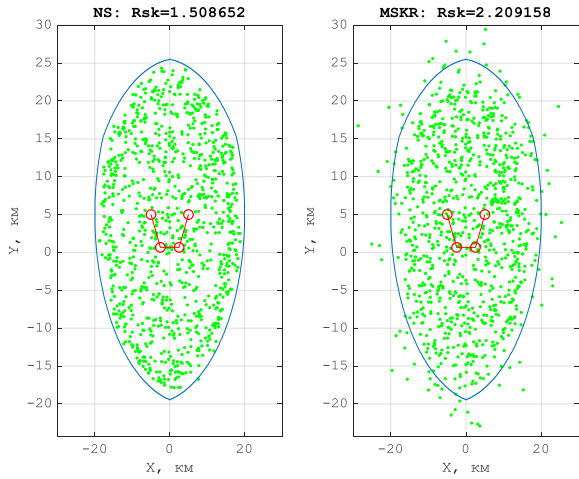


б

Рис. 2. Пример решения задачи местоопределения двумя рассмотренными алгоритмами в трехпозиционной МРС при ошибке измерения пеленга: *а* – $\sigma_\theta = 1$ градус, *б* – $\sigma_\theta = 3$ градуса



a



б

Рис. 3. Пример решения задачи местоопределения двумя рассмотренными алгоритмами в четырехпозиционной МРС при ошибке измерения пеленга: *а* – $\sigma_\theta = 1$ градус, *б* – $\sigma_\theta = 3$ градуса

3. Заключение

Анализ полученных результатов свидетельствует о возможности программной реализации нейросетевого алгоритма вычисления координат ИРИ в многопозиционных радиосистемах с учетом ограничений по дальности прямой видимости с использованием библиотеки `alglib`, несмотря на ограниченные возможности указанной библиотеки по работе с нейронными сетями.

Полученные результаты свидетельствуют о важности учета информации о зоне прямой видимости в алгоритмах местоопределения. Особую роль приобретает учет априорной информации о возможном местоположении ИРИ при больших ошибках пеленгования. Так, как следует из рисунка 2б, при значении ошибки измерения пеленга $\sigma_\theta = 3$ градуса для трехпозиционной МРС учитывающий информацию о зоне прямой видимости нейросетевой алгоритм обеспечивает в 1,7 раза меньшую ошибку местоопределения по сравнению с алгоритмом вычисления координат по минимуму суммы квадратов расстояний до линий пеленгов, не учитывающим указанную информацию.

Список литературы

1. Кирсанов, Э. А. Обработка информации в пространственно-распределенных системах радиомониторинга: статистический и нейросетевой подходы / Э. А. Кирсанов, А. А. Сирота. – М. : ФИЗМАТЛИТ, 2012. – 344 с.
2. Библиотека `alglib` [Электронный ресурс]. – Режим доступа : <https://www.alglib.net>