

18. Балхарет А. А., Падерно П. И. Оценка времени выполнения задачи при изменении напряженности деятельности оператора // Изв. СПбГЭТУ «ЛЭТИ». 2009. № 5. С. 19–23.

19. Burkov E. A., Paderno P. I., Sopina O. P. Analysis and combination of activity algorithms evaluation methods // 2018 Third Intern. Conf. on Human Factors in

Complex Technical Systems and Environments (ERGO) and Environments (ERGO). SPb., 2018. P. 111–114.

20. Назаренко Н. А., Падерно П. И., Сатторов Ф. Э. Способ оценки логической сложности и стереотипности алгоритмов деятельности операторов // Науч.-техн. вестн. информ. технологий, механики и оптики. 2019. Т. 19, № 4. С. 740–746.

---

N. A. Nazarenko, P. I. Paderno  
Saint Petersburg Electrotechnical University

## EVALUATION OF LOGIC COMPLEXITY AND STEREOTYPITY OF CYCLIC STRUCTURES IN OPERATOR ACTIVITY ALGORITHMS

*The analysis of the evaluation of the algorithms of the operators in the process of designing man-machine systems. A new approach to assessing the coefficients of stereotype and logical complexity of activity algorithms having cycles (repeating a series of operations) is presented. The basic typical functional structures and corresponding calculation formulas for calculating the lengths (sizes) and stereotypicity coefficients and the logical complexity of individual fragments with a cyclic structure are shown. The proposed approach, being a tool for developing a human-machine system, greatly simplifies its computational activities, and makes it possible to evaluate the main characteristics of the algorithms of operator activity in stages. In addition, the representation of dependencies proposed within the framework of the approach facilitates understanding of the obtained intermediate and final estimates. The approach is the basis of the designed information module for supporting the activities of the developer in solving problems related to the evaluation of the algorithms of the operators of developed complex man-machine systems and complexes for various purposes.*

**Activity algorithm, logical complexity, stereotype, cyclic structures, operator, quality of activity algorithms, human-machine systems**

---

УДК 004.514.62

М. Г. Пантелеев, А. Ф. Салимов  
Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина)

## Анализ алгоритмов навигации интеллектуального агента в виртуальном футболе

*Проводится анализ различных методов решения навигационной задачи интеллектуальным агентом в среде виртуального футбола Robosoccer на основе зашумленных данных, поступающих от визуального сенсора. Рассмотрены две группы методов вычисления абсолютных координат объектов на основе сенсорных данных о флагах и линиях: тригонометрические методы и методы, основанные на использовании фильтра Калмана и фильтра частиц. Кратко описано инструментальное программное средство, разработанное для проведения экспериментальных исследований и позволяющее произвольно варьировать условия решения навигационной задачи. Представлены экспериментальные результаты сравнительного анализа быстродействия и точности алгоритмов, реализующих различные методы. Полученные результаты позволяют агенту решать навигационную задачу с использованием алгоритмов произвольного времени, разменивая время решения на качество (точность) результата. С учетом полученных результатов определены направления дальнейших исследований по реализации оценки тактической обстановки в виртуальном футболе.*

**Интеллектуальный агент, многоагентная система, виртуальный футбол, RoboCup Soccer, навигационная задача, алгоритмы произвольного времени, фильтр Калмана, фильтр частиц**

Создание интеллектуальных агентов (ИА) и основанных на них многоагентных систем (МАС) является в настоящее время магистральным

направлением развития искусственного интеллекта [1], [2]. Под ИА понимаются автономные системы, воспринимающие окружающую среду и

---

реализующие в этой среде целенаправленное поведение. К важным особенностям ИА относятся их способность действовать группами, в том числе в условиях активного противодействия со стороны других групп агентов.

В последние годы в качестве эталонной платформы для отработки различных подходов к построению ИА реального времени (ИА РВ) и тактик группового поведения МАС в условиях группового противодействия используется виртуальный футбол [3]–[5]. Сообщество специалистов в области ИА и МАС поставлена амбициозная задача – к 2050 г. создать команду автономных роботов-футболистов, способных обыграть команду чемпионов мира.

Одной из важных задач, решаемых ИА, является навигация (определение собственного местоположения) на основе информации, поступающей от визуальных сенсоров. Для решения этой задачи широко используются вероятностные подходы, среди которых наиболее известны фильтр Калмана и фильтр частиц [6], [7]. Выбор наиболее эффективных методов и алгоритмов решения навигационной задачи с учетом особенностей конкретной среды функционирования ИА РВ требует их экспериментального сравнительного анализа.

**Особенности навигационной задачи в среде виртуального футбола.** Среда виртуального футбола (ВФ) RoboCup Soccer Server позволяет в реальном времени моделировать игру в футбол команд, состоящих из 11 автономных ИА. Каждый игрок реализуется как независимая программа, подключающаяся к серверу через UDP-сокеты. Сервер предоставляет виртуальное поле, на кото-

ром имитирует действия игроков в соответствии с поступающими от них командами. В каждом такте моделирования игрок получает от сервера сенсорную информацию и отправляет ему команды о собственных действиях.

Платформа ВФ моделирует важнейшие особенности реальных МАС – ограниченное поле зрения агентов, шумы восприятия (сенсорной информации) ИА и моделей движения игроков и мяча, ограниченная пропускная способность коммуникационных каналов и др.

Агент-футболист имеет три сенсора: визуальный, слуховой и сенсор тела. Основную информацию агент получает от визуального сенсора, которым он может управлять, задавая ширину и направление взгляда. При этом в каждый момент времени агент видит только часть поля. Частота получения визуальной информации от сервера зависит от выбранного агентом режима работы сенсора. На рис. 1 показан общий вид поля с расставленными на нем флагами (отмечены точками).

Флаги – это статические объекты с априори известными координатами, вследствие чего они могут наряду с линиями использоваться в качестве навигационных привязок.

Визуальный сенсор обеспечивает информацию обо всех попавших в зону видимости объектах среды: мяче (b), игроках (p), флагах (f), линиях (l) и воротах (g) [5].

Визуальная информация поступает от сервера в следующем базовом формате:

(see ObjName Distance Direction  
DistChng DirChng BodyDir HeadDir),

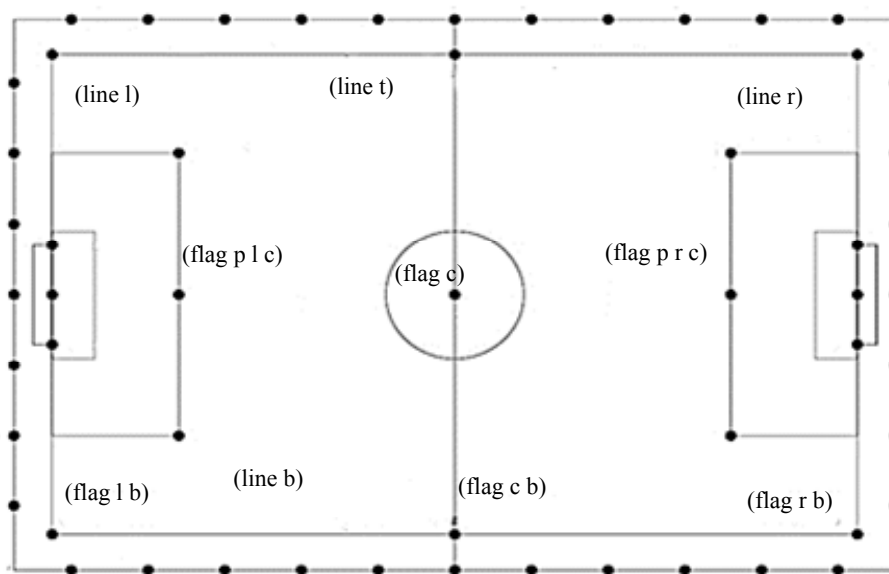


Рис. 1

где ObjName – тип/спецификатор видимого объекта; Distance – расстояние до объекта; Direction – направление на объект; DistChng – изменение расстояния; DirChng – изменение направления; BodyDir – направление тела; HeadDir – направление взгляда (головы).

Спецификаторы характеризуют объекты одного типа и задаются специальными символами после типа объекта. У полевых игроков спецификаторы указывают название команды и уникальный номер игрока.

Спецификаторы флагов определяют их положение на поле. Например, (f c) – флаг в центре поля; (f p l b) – флаг, отмечающий нижний угол (b – bottom) штрафной площади (p – penalty area) левой половины поля (l – left).

Пример кадра визуальной информации, поступающей от сервера, представлен ниже:

```
(see 0 ((f c) 32.4 0 0 0) ((f r t) 45.3 -4)
((f c t) 53.3 40) ((f p l t) 39.7 35) ((f p l c) 36.4 23)
((f t 0) 62.3 -8) ((f t r 10) 59 5) ((f t r 20) 64.7 9)
((f t r 30) 63.3 17) ((f t r 40) 74 19) ((f t r 50)
82.6 28) ((f t l 10) 65.3 -17) ((f t l 20) 67.7 -26)
((f t l 30) 65 -32) ((f t l 40) 74.5 -44) ((f r t 20)
77.6 42) ((f r t 30) 85.6 34) ((b) 24.3 0 0 0)
((p «TeamName» 10) 43.1 32) ((l t) 63.4 66)).
```

Агент получает визуальную информацию о каждом наблюдаемом объекте в полярной системе координат, центром которой является он сам (т. е. азимут и расстояние до объекта). Используя эту относительную информацию и априорные знания координат статических объектов, он может вычислить собственные абсолютные координаты, а затем и абсолютные координаты других игроков.

В различные моменты времени (такты моделирования) в поле зрения агента в зависимости от его положения и направления взгляда оказываются различные навигационные привязки (флаги и линии). При этом количество таких привязок и расстояния до них варьируются в широких пределах. Так, игрок, смотрящий в сторону центра поля, видит намного больше навигационных привязок, чем игрок, стоящий в углу поля, взгляд которого направлен на ближайшую боковую линию.

Существенный фактор при решении навигационной задачи – неточность (зашумленность) получаемой от сервера сенсорной информации. В частности, шумы, накладываемые на расстояние до объекта, вычисляются сервером по следующей формуле:

$$Q\_Distance = Quantize(\exp(Quantize(\ln(distance), quantizeStep)), 0.1),$$

где Q\_Distance – воспринимаемое агентом расстояние до наблюдаемого объекта; distance – истинное расстояние до объекта; quantizeStep – шаг квантования.

Таким образом, для объекта, находящегося на расстоянии 1 м, погрешность определения расстояния может составлять 10 см, а при расстоянии 10 м – даже 1 м.

Задача определения абсолютных координат игроков (себя и других) по наблюдаемой информации может решаться разными методами и алгоритмами, отличающимися затрачиваемым временем и точностью решения.

В [2], [8]–[10] разработан подход к построению ИА РВ, в соответствии с которым агент в проблемных ситуациях динамически определяет имеющийся запас времени и адаптирует к нему процесс обдумывания решения. В рамках данного подхода алгоритмы решения частных задач строятся как *алгоритмы произвольного времени* (АПВ), в которых качество результатов (в частности, точность) зависит от выделяемого данному алгоритму времени. В рамках данного подхода решение навигационной задачи рассматривается с позиций АПВ. Для построения профилей АПВ, фиксирующих зависимость точности результата от выделяемого алгоритму времени, необходим сравнительный экспериментальный анализ различных методов решения навигационной задачи.

**Методы навигации на основе тригонометрических моделей.** На основе анализа литературы [6], [7] были выделены следующие методы решения навигационной задачи в среде ВФ:

- по ближайшему флагу и дальней линии;
- по двум ближайшим флагам;
- на основе фильтра Калмана;
- на основе фильтра частиц.

*Навигация по ближайшему флагу и дальней линии.* В данном случае сначала среди всех видимых объектов определяются ближайший флаг и дальняя линия. Направление игрока к линии используется для нахождения угла поворота головы агента. Пусть  $l_b$  – биссектриса угла обзора игрока, проходящая через его абсолютные координаты. Если в поле зрения игрока попала линия  $l$ , полученное им визуальное сообщение будет содержать расстояние  $d$  до точки пересечения  $l_b$  с  $l$  и угол  $\alpha$  между  $l_b$  и  $l$  (рис. 2).

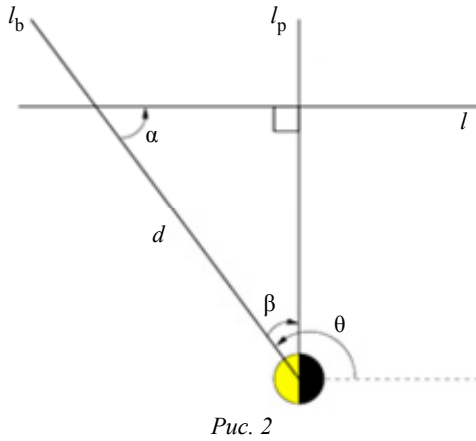


Рис. 2

При этом угол  $\alpha$  будет равен углу поворота  $l_b$  до совпадения с  $l$ , имеющему положительное значение при вращении по часовой стрелке и отрицательное – в противном случае.

Для получения угла поворота головы  $\theta$  игрока необходимо вычислить угол  $\beta$  между линиями  $l_b$  и  $l_p$ , перпендикулярной к  $l$ . Угол  $\beta$  вычисляется по следующей формуле:

$$\beta = -\text{sign}(\alpha)(90 - |\alpha|).$$

Вычислив угол  $\theta$ , можно перейти к обработке информации о флаге. Сенсорная информация о флаге  $f$  включает направление  $f_\phi$  на него относительно угла поворота головы игрока и расстояние  $f_r$  до него. Поскольку агенту априори известны абсолютные координаты  $(f_x, f_y)$  всех флагов, он может с учетом относительной сенсорной информации вычислить свои абсолютные координаты  $(p_x, p_y)$ . Для этого необходимо сначала выполнить поворот на угол  $\theta$  в полярной системе координат. Абсолютные координаты игрока  $(p_x, p_y)$  можно получить преобразованием полярных координат в декартовы:

$$(p_x, p_y) = (f_x, f_y) - \pi(f_r, f_\phi + \theta),$$

где  $\pi$  – функция преобразования полярных координат в декартовы:

$$(x, y) = \pi(r, \phi) = (r \cos(\phi), r \sin(\phi)).$$

*Навигация по двум ближайшим флагам и дальней линии.* В данном алгоритме сначала определяются два ближайших флага среди всех видимых. Знание абсолютных координат флага и расстояния до него задает круг возможных позиций агента с центром в точке расположения флага и радиусом, определяемым расстоянием до него. Аналогично определяет круг возможных позиций агента и вто-

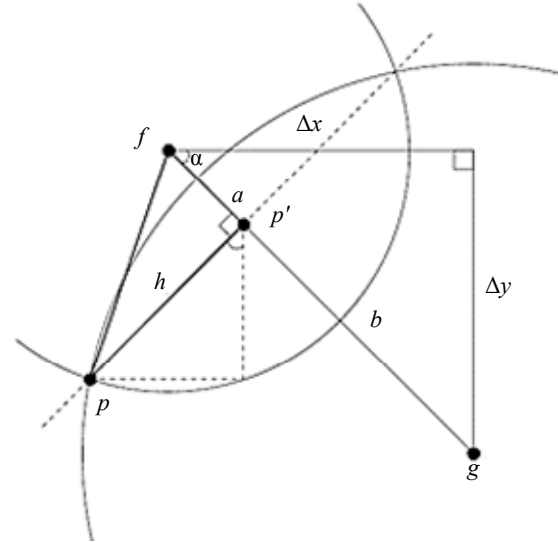


Рис. 3

рой флаг. Положение агента, очевидно, определяется пересечением этих окружностей (рис. 3).

Для однозначного определения точки необходимо вычислить расстояние  $d$  между флагами  $f$  и  $g$ , используя их абсолютные координаты:

$$d = \sqrt{(g_x - f_x)^2 + (g_y - f_y)^2}.$$

Расстояние  $d$ , как видно на рис. 3, складывается из отрезков  $[f; p']$  и  $[p'; g]$ . На рис. 3 обозначены:  $a$  – расстояние от флага  $f$  до точки  $p'$ ;  $b$  – расстояние от точки  $p'$  до флага  $g$ ;  $h$  – расстояние от точки  $p$  до точки  $p'$ . Для треугольников  $gpp'$  и  $fpp'$  можно записать:

$$\begin{cases} f_r^2 = a^2 + h^2; \\ g_r^2 = b^2 + h^2. \end{cases} \quad (1)$$

Учитывая соотношение  $b = (d - a)$ , можно вычислить значение  $a$ :

$$a = \frac{f_r^2 - g_r^2 + d^2}{2d}.$$

Тогда абсолютные координаты точки  $p'$  определяются следующим образом:

$$(p'_x, p'_y) = (f_x + a \cos(\alpha), f_y + a \sin(\alpha)).$$

Значения  $\cos(\alpha)$  и  $\sin(\alpha)$  определяются из соотношений:

$$\sin(\alpha) = \frac{\Delta y}{d}, \quad \cos(\alpha) = \frac{\Delta x}{d}.$$

Исходя из этого, абсолютные координаты игрока  $p$  будут равны:

$$(p_x, p_y) = (p'_x - h \operatorname{sign} \sin(\alpha), p'_y + h \operatorname{sign} \cos(\alpha)),$$

где, исходя из (1),  $h = \sqrt{f_r^2 - a^2}$ .

Истинное местоположение игрока можно определить с помощью знака  $\operatorname{sign}$ : при положительном значении разности  $(g_\varphi - f_\varphi)$  знак  $\operatorname{sign} = +1$ , в противном случае  $\operatorname{sign} = -1$ .

Использование только тригонометрических соотношений может приводить к большим ошибкам, значительно варьирующимся в различных ситуациях. Более точные решения навигационной задачи можно получить с применением подходов, основанных на фильтрации.

**Методы навигации с использованием фильтрации.** При определении абсолютных координат агента на футбольном поле тригонометрическими методами используется визуальная информация, полученная только в текущем такте. Вместе с тем, информацию о некотором объекте, получаемую в разных сенсорных тактах, можно рассматривать как последовательные наблюдения динамического процесса.

В подходах, использующих идеи фильтрации, состояния (координаты) агента рассматриваются как управляемый марковский процесс со скрытыми состояниями  $x_t \in X \subset R_q$  и временным шагом  $t$  [11].

Обозначим  $x_0$  – начальное состояние игрока,  $P(x_0)$  – первоначальное распределение в момент времени  $t = 0$ . Тогда динамика игрока, реализуемая сервером, может рассматриваться как стохастическая модель переходов  $P(x_{t+1}|x_t, a_t)$ , в которой агент своими действиями  $a_t$ , совершаемыми в момент времени  $t$ , меняет свое состояние  $x_t$  на состояние  $x_{t+1}$ . Будем полагать сенсорные данные  $y_t$ , отправляемые сервером в каждом такте, условно независимыми относительно  $x_t$ . Тогда задачу навигации можно свести к оценке апостериорной плотности вероятности  $P(x_t|y_t)$  в пространстве состояний  $X$ , описывающем текущее состояние агента в момент времени  $t$ . По правилу Байеса можно записать:

$$P(x_{t+1} | y_{t+1}) \propto P(y_{t+1} | x_{t+1}) P(x_{t+1}), \quad (2)$$

где априорная функция плотности вероятности  $P(x_{t+1}|y_{t+1})$  соответствует апостериорной функции от последнего шага времени. Для  $P(x_{t+1})$  можно записать

$$P(x_{t+1}) = \int P(x_{t+1} | x_t) P(x_t | y_t) dx_t. \quad (3)$$

В данной формуле используется марковское предположение о независимости текущих значений от предыдущих временных шагов.

Уравнения (2) и (3) позволяют построить итерационную схему байесовской фильтрации, в которой для аналитического определения апостериорной вероятности необходимо вычислять интеграл согласно (3). Результат вычисления необходимо домножить на значения плотности вероятности  $P(y_{t+1}|x_{t+1})$ , после чего нормировать плотность вероятности  $P(x_{t+1}|y_{t+1})$ . Апостериорные вероятности могут быть вычислены аналитически, если модели наблюдения и перехода линейно гауссовы. Поскольку среда моделирования Robocup Soccer не удовлетворяет этому требованию, необходимо использовать аппроксимации.

Эффективным методом вычисления апостериорных распределений в байесовской фильтрации является фильтр частиц [7]. Он основан на дискретной аппроксимации непрерывной функции апостериорной плотности (2) с использованием набора частиц  $x_t^i$  с соответствующими весами  $\pi_t^i$ , где  $i = 1, \dots, N$ . Эмпирическая апостериорная оценка имеет вид

$$P(x_t | y_t) = \sum_{i=1}^N \pi_t^i \delta(x_t - x_t^i). \quad (4)$$

Используя (4), интеграл для вычисления (3) можно заменить суммированием:

$$P(x_{t+1}) = \sum_{i=1}^N \pi_t^i P(x_{t+1} | x_t^i).$$

После замены всех интегралов на суммы и всех непрерывных функций плотности на дискретные выражения нормирование фиксированной апостериорной функции сведется к нормированию дискретных значений на единицу суммы:

$$P(x_{t+1} | y_{t+1}) \propto P(y_{t+1} | x_{t+1}) \sum_{i=1}^N \pi_t^i P(x_{t+1} | x_t^i).$$

*Навигация с использованием фильтра Калмана.* Фильтр Калмана позволяет получать оценку вектора состояния объекта (в данном случае – координат игрока) на основе серии зашумленных измерений.

Решение навигационной задачи в ВФ с использованием фильтра Калмана укрупнено включает следующие шаги:

1. Разбор и анализ принятой визуальной информации с целью получения списка видимых игроку флагов и их относительных координат.

2. Циклическая обработка всевозможных пар флагов. В каждом цикле выбираются два вида флагов и вычисляются абсолютные координаты агента по этим флагам.

3. Вычисление дисперсии ошибки сенсора (качество оценки дисперсии определяет качество работы фильтра Калмана):

$$\sigma_{i+1}^2 = \frac{(r_{\max} - r_{\min})^2}{12},$$

где  $r_{\max}$  и  $r_{\min}$  – максимально и минимально возможные значения расстояния;  $\sigma^2$  – дисперсия ошибки.

4. Обновление значения коэффициента усиления Калмана ( $K$ ) с учетом полученной дисперсии. Значение коэффициента должно обеспечивать максимальную близость вычисляемых оптимальных значений абсолютных координат к их истинным значениям:

$$K = \frac{\hat{\sigma}_i^2}{\hat{\sigma}_i^2 + \hat{\sigma}_{i+1}^2}.$$

5. Коррекция с помощью коэффициента Калмана оценочного значения абсолютных координат агента в данной итерации:

$$\hat{x}_{i+1} = \hat{x}_i + K(y_i - \hat{x}_i).$$

*Навигация с использованием фильтра частиц.* Фильтр частиц является методом определения абсолютных координат агента, в соответствии с которым для оценки координат создается множество гипотез (частиц) об их текущих значениях.

Алгоритм определения абсолютных координат на основе фильтра частиц включает пять шагов: *инициализация, прогноз, коррекция* (вычисление весовых коэффициентов и ресэмплинг), *оценка состояния*.

Инициализация фильтра частиц выполняется в момент приема первого сенсорного кадра и сводится к генерации частиц случайным образом вокруг полученной точки. В результате в окрестности точки генерируется  $N$  частиц с изначально равными весами.

Положение каждой частицы определяется на основе полученной от визуального сенсора информации о траектории и расстоянии. На этапе прогнозирования учитывается диапазон возможных значений расстояния до агента. Если расстояние между частицей и агентом находится вне этого диапазона значений, частица удаляется из набора. Таким образом после этапа прогнози-

рования с учетом информации о диапазоне значений из начального набора из  $N$  частиц остается  $K$  частиц. Возможны случаи, когда из общего набора могут быть удалены все частицы ( $K = 0$ ). Такие ситуации возникают при резкой смене направления движения игрока, слишком большом интервале между измерениями и некоторых других факторах. В таких случаях сетка частиц инициализируется заново.

Процедура коррекции выполняется каждый раз при получении новой сенсорной информации и включает два шага: определение диапазонов значений и *ресэмплинг*.

Перед вычислением диапазона значений некоторые частицы могут быть удалены с использованием калмановской фильтрации. Каждый раз при получении новой информации определяются верхняя и нижняя границы расстояний между частицами и агентом. Частицы, оказавшиеся по результатам прогнозирования вне диапазона, удаляются из общего набора частиц.

Это позволяет отбросить заведомо ложные гипотезы о возможном положении объекта и тем самым сократить вычислительные затраты и повысить точность результата. После нескольких шагов процедуры коррекции у большинства частиц, соответствующих ошибочным гипотезам, веса могут стать близкими к нулю. Такие частицы практически не вносят вклад в финальную оценку вектора состояния, однако на них тратятся вычислительные ресурсы. Процедура ресэмплинга позволяет перераспределить вычислительные ресурсы, отбрасывая частицы с малым весом и дублируя частицы, имеющие большой вес.

На последнем этапе оценки состояния абсолютные координаты агента рассчитываются как взвешенная сумма состояний всех частиц.

**Подход и результаты экспериментальной оценки алгоритмов навигации.** Для экспериментального исследования и сравнительного анализа эффективности различных алгоритмов навигации разработана инструментальная программа на языке Java, позволяющая:

- произвольным образом позиционировать агента на поле и задавать направление и режим работы его визуального сенсора;

- решать навигационную задачу разными методами и оценивать точность решения и затраченное время.

Пользовательский интерфейс разработанной программы представлен на рис. 4.

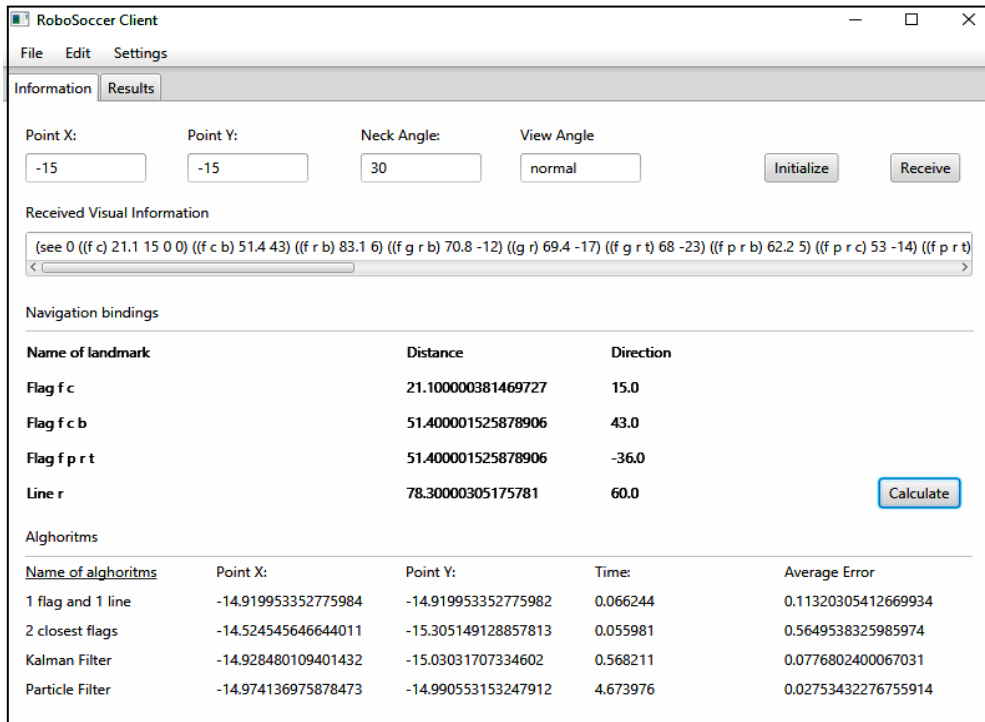


Рис. 4

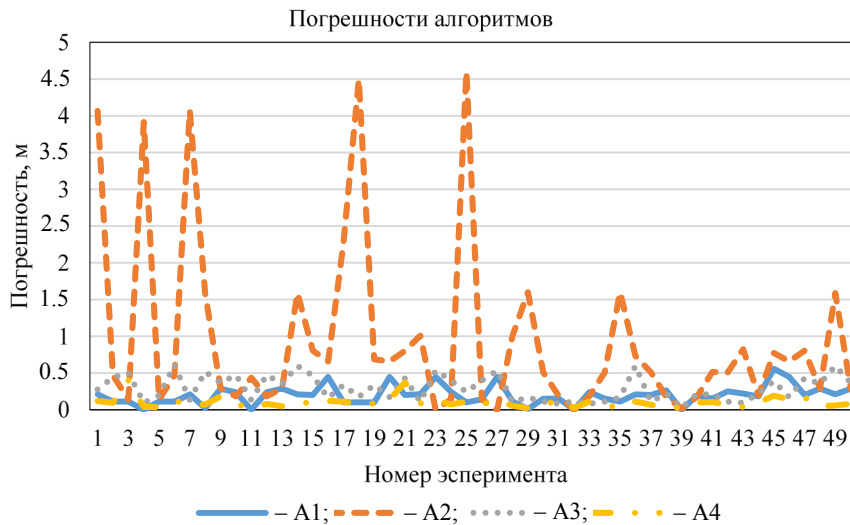


Рис. 5

Поскольку количество навигационных привязок (флагов), попадающих в поле зрения агента, существенно зависит от угла обзора, эксперименты проводились отдельно для разных углов обзора: узкого (*narrow*), нормального (*normal*) и широкого (*wide*). Результаты исследования при узком угле обзора, когда агент видит мало флагов, представлены на рис. 5.

Погрешности алгоритмов вычислялись при различных 50 абсолютных координатах. Графики на рисунке соответствуют следующим алгоритмам вычисления абсолютных координат агента: A1 – по ближайшему флагу и дальней линии; A2 – по двум ближайшим флагам; A3 – с использо-

ванием фильтра Калмана; A4 – с использованием фильтра частиц.

Как видно из приведенных зависимостей, наибольшую ошибку дает алгоритм определения координат по двум ближайшим флагам. Наиболее точен метод на основе фильтра частиц, поскольку он использует информацию обо всех видимых флагах с учетом последовательности наблюдений.

Результаты экспериментальной оценки времени реализации алгоритмов представлены на рис. 6. Эксперименты проводились на ПЭВМ со следующими характеристиками: 2 ядра, 4 логических процессора, частота 2.2 ГГц.

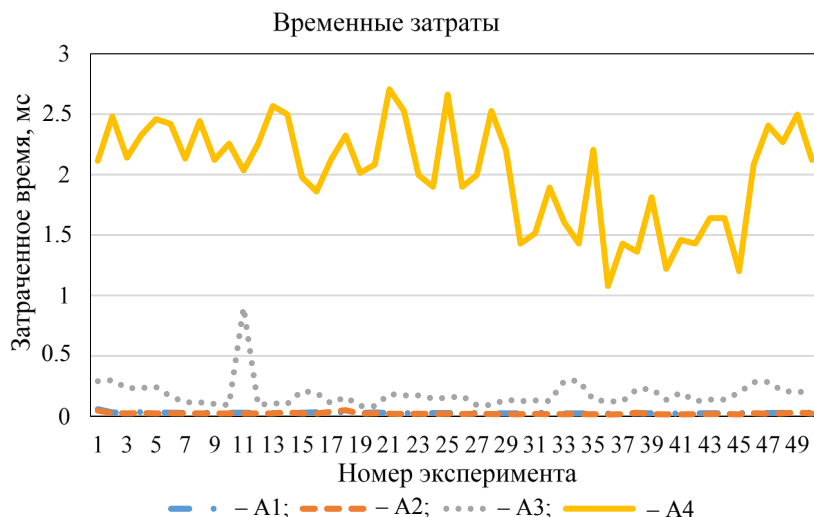


Рис. 6

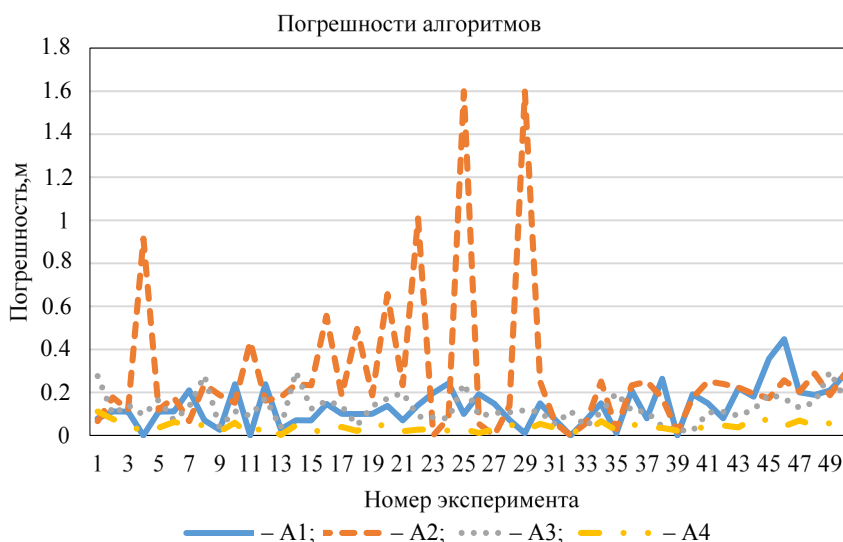


Рис. 7

Как показал эксперимент, алгоритм с использованием фильтра частиц работает в 7...15 раз медленнее других алгоритмов, что обусловлено большим числом операций обработки.

Усредненные значения ошибки определения абсолютных координат агента и затрат времени для разных алгоритмов при узком угле обзора представлены в сводной табл. 1.

Таблица 1

Алгоритм определения абсолютных координат агента	Средняя ошибка, м	Время выполнения, мс
По ближайшему флагу и дальней линии	0.2482	0.0274
По двум ближайшим флагам	1.0154	0.0222
На основе фильтра Калмана	0.2901	0.1826
На основе фильтра частиц	0.0974	2.0154

Анализ полученных результатов показывает, что по совокупности двух характеристик – точности и времени – предпочтителен алгоритм определения абсолютных координат агента по ближайшему флагу и дальней линии. Данный алгоритм по затратам времени незначительно уступает наиболее быстродействующему алгоритму определения координат по двум ближайшим флагам и, вместе с тем, дает ошибку, сопоставимую с алгоритмом на основе фильтра Калмана.

Основным режимом работы визуального сенсора агентов, установленным в среде ВФ по умолчанию, является режим нормального угла обзора (normal). При данном режиме в поле зрения агента попадает больше навигационных привязок, чем в режиме narrow.

Результаты экспериментальных исследований погрешностей алгоритмов навигации в режиме визуального сенсора normal представлены на



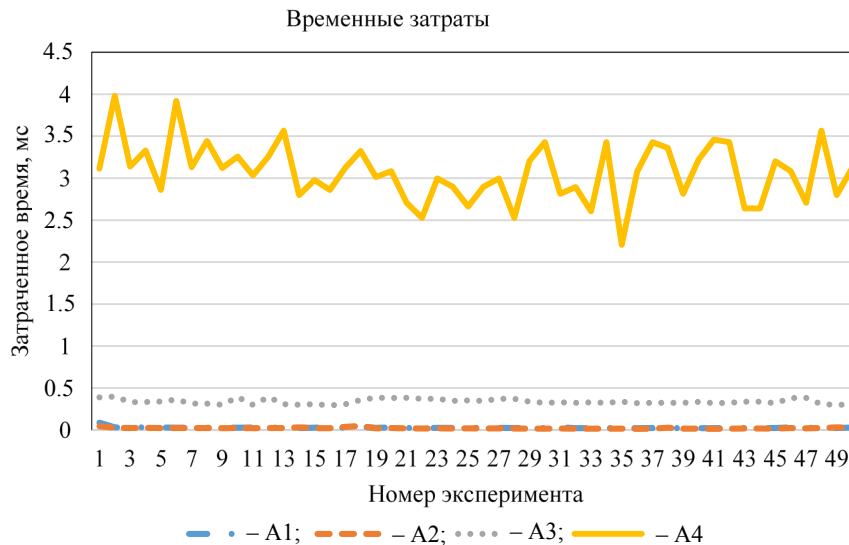


Рис. 8

рис. 7. Как видно из графиков, погрешность алгоритма определения координат по двум ближайшим флагам существенно (в два и более раз) снизилась, однако в отдельных случаях сохраняются резкие выбросы значений ошибки. Как показал анализ, такие выбросы соответствуют ситуации, когда два ближайших флага имеют близкий азимут, т. е. угол между ними мал (близок к нулю).

Результаты измерений времени работы навигационных алгоритмов при нормальном угле обзора представлены на рис. 8. Усредненные значения ошибки определения абсолютных координат и затрат времени для разных алгоритмов при данном режиме работы визуального сенсора представлены в сводной табл. 2.

Таблица 2

Алгоритм определения абсолютных координат агента	Средняя ошибка, м	Время выполнения, мс
По ближайшему флагу и дальней линии	0.1851	0.0280
По двум ближайшим флагам	0.2991	0.0264
На основе фильтра Калмана	0.1474	0.3398
На основе фильтра частиц	0.0403	3.0734

Анализ результатов показывает, что алгоритмы определения координат по ближайшему флагу и дальней линии и по двум ближайшим флагам имеют практически одинаковую временную сложность. При этом первый из них обеспечивает точность результата более чем в 1.5 раза выше, чем алгоритм с использованием двух ближайших флагов.

Вместе с тем, у алгоритма на основе фильтра частиц результаты более чем в 4 раза точнее, чем

у алгоритма на основе ближайшего флага и дальней линии. Однако данный алгоритм – наиболее медленный и тратит примерно на порядок больше времени, чем алгоритм на основе фильтра Калмана.

Сопоставление табл. 1 и 2 показывает, что при нормальном угле обзора точность алгоритмов в 1.8–2.2 раза выше, чем при узком угле обзора. При этом время вычисления координат на основе тригонометрических методов практически не изменилось. Вместе с тем, времена работы алгоритмов, основанных на фильтрации, вследствие увеличения количества видимых флагов выросли более чем в 1.5 раза.

В режиме широкого угла обзора (wide) в область видимости агента попадает максимальное число флагов и линий. Результаты экспериментальной оценки погрешности различных алгоритмов определения абсолютных координат агента для данного режима визуального сенсора представлены на рис. 9.

График погрешностей алгоритма определения координат на основе двух ближайших флагов на данном рисунке сглажен – удалены резкие выбросы, обусловленные близостью флагов по азимуту. Как видно из графиков, минимальную погрешность обеспечивает алгоритм на основе фильтра частиц, при этом он характеризуется минимальной дисперсией ошибки.

Результаты оценки времени работы навигационных алгоритмов при широком угле обзора агента представлены на рис. 10.

В табл. 3 приведены усредненные значения ошибок определения абсолютных координат и затрат времени для разных алгоритмов при дан-

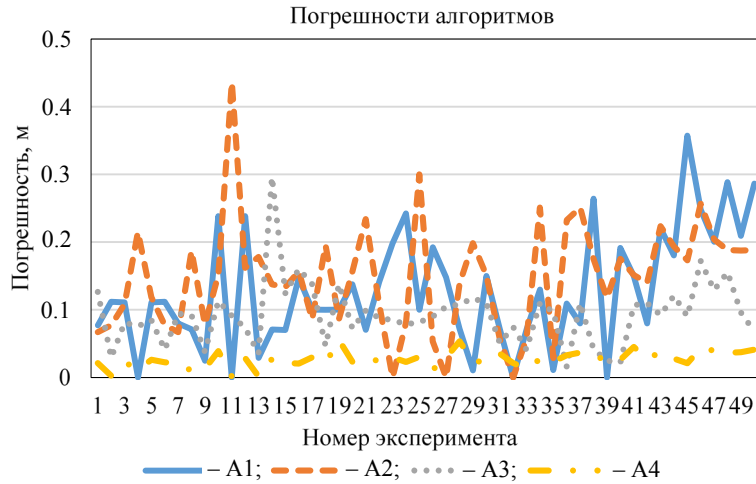


Рис. 9

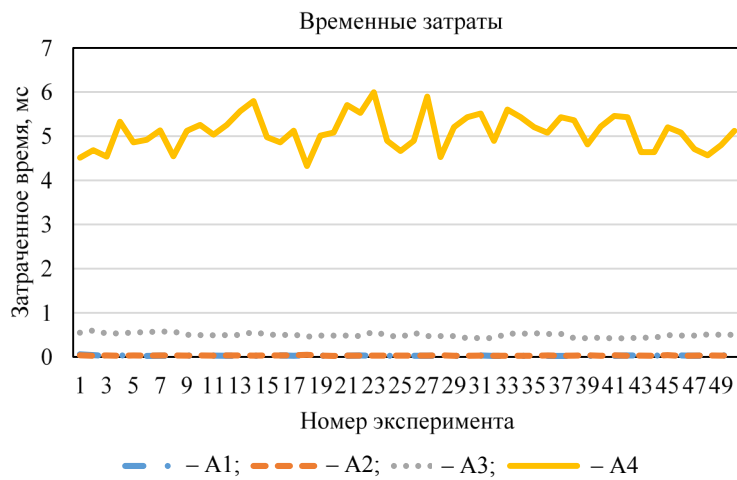


Рис. 10

ном режиме работы визуального сенсора. Как видно из сравнения данных результатов с результатами, полученными для нормального угла обзора, погрешность вычисления координат при широком угле обзора уменьшается на 20–40 %. При этом время выполнения алгоритмов увеличивается для разных методов от 5 до 60 %.

Таблица 3

Алгоритм определения абсолютных координат агента	Средняя ошибка, м	Время выполнения, мс
По ближайшему флагу и дальней линии	0.1680	0.0288
По двум ближайшим флагам	0.2472	0.0309
На основе фильтра Калмана	0.0927	0.4954
На основе фильтра частиц	0.0263	5.0974

По критерию точности решения навигационной задачи в ВФ лучшим является алгоритм, основанный на фильтре частиц. Однако он требует максимальных временных затрат (в среднем

4 мс). Тригонометрические алгоритмы определения абсолютных координат быстрые, но имеют значительно более низкую точность, существенно зависящую от количества видимых флагов и выбранного качества работы визуального сенсора. Кроме того, данные методы могут давать аномально высокие ошибки при близких азимутах направления на ближайшие флаги.

Наличие различных алгоритмов решения навигационной задачи, отличающихся временем работы и точностью результата, позволяет рассматривать их с позиций АПВ и адаптировать процесс обдумывания агентом решения с учетом динамически (ситуативно) меняющихся временных ограничений. Таким образом, направлениями дальнейших исследований являются:

- построение уточненных профилей АПВ решения навигационной задачи и их использование в архитектуре ИА РВ для среды ВФ;
- использование вычисленных абсолютных координат игроков для определения тактических расстановок команд в процессе игры.

## СПИСОК ЛИТЕРАТУРЫ

1. Рассел С., Норвиг П. Искусственный интеллект: современный подход. 2-е изд. М.: Издат. дом «Вильямс», 2006. 1408 с.
2. Пантелеев М. Г., Пузанков Д. В. Интеллектуальные агенты и многоагентные системы. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2015. 215 с.
3. Официальный сайт турнира Robocup. URL: <http://www.robocup.org> (дата обращения 10.09.2019).
4. Футбольный сервер Robocup. URL: <https://github.com/rcsoccersim/rcssserver/> (дата обращения 10.09.2019).
5. Руководство пользователя Robocup Soccer. URL: <https://rcsoccersim.github.io/manual/> (дата обращения 10.04.2019).
6. Thrun S., Wolfram B., Fox D. Probabilistic Robotics (Intelligent Robotics and Autonomous Agents). The MIT Press, 2005. 672 p.
7. Московский А. Д. Неоднозначные модели в задаче локализации мобильного робота // 5-й Всерос. науч.-практ. семинар «Беспилотные транспортные средства с элементами искусственного интеллекта» (БТС-ИИ-2019): Тр. семинара. Переславль-Залесский: РАИИ, 2019. С. 9–18.
8. Пантелеев М. Г. Концепция построения интеллектуальных агентов реального времени на основе модели опережающего итеративного планирования // Тр. 13-й Нац. конф. по ИИ с междунар. участием КИИ-2012. Т 3. Белгород: Изд-во БГТУ, 2012. С. 25–33.
9. Пантелеев М. Г. Формальная модель опережающего итеративного планирования действий интеллектуальных агентов реального времени // Тр. 14-й Нац. конф. по ИИ с междунар. участием КИИ-2014. Т. 1. Казань: Изд-во РИЦ «Школа». 2014. С. 323–333.
10. Panteleev M. G. Advanced Iterative Action Planning for Intelligent Real-Time Agents // Procedia Computer Science. 2019. Vol. 150. P. 244–252.
11. Kok J. R., de Boer R. The Incremental Development of a Synthetic Multi-Agent System: The UvA Trilearn 2001 Robotic Soccer Simulation Team, M. S. Master thesis. Netherlands: University of Amsterdam, 2002. 217 с.

---

M. G. Panteleyev, A. F. Salimov  
*Saint Petersburg Electrotechnical University*

### ANALYSIS OF INTELLIGENT AGENT'S NAVIGATION ALGORITHMS IN A VIRTUAL FOOTBALL

*The analysis of various methods of solving the navigation problem by an intelligent agent in the environment of virtual football Robosoccer on the basis of noisy data coming from a visual sensor. Two groups of methods for calculating the absolute coordinates of objects based on sensory data about flags and lines are considered: trigonometric methods and methods based on the use of the Kalman filter and the particle filter. The instrumental software tool developed for carrying out experimental researches and allowing to vary arbitrarily conditions of the decision of a navigation problem is briefly described. Experimental results of the comparative analysis of the speed and accuracy of algorithms implementing various methods are presented. The obtained results allow the agent to solve the navigation problem using algorithms of arbitrary time, exchanging the solution time for the quality (accuracy) of the result. Taking into account the obtained results, the directions of further research on the implementation of the assessment of the tactical situation in virtual football are determined.*

**Intelligent agent, multi-agent system, virtual soccer, RoboCup Soccer, navigation task, arbitrary time algorithms, Kalman filter, particle filter**

---