

ПРИНЦИПЫ РЕАКТИВНОГО УПРАВЛЕНИЯ МОБИЛЬНЫМ РОБОТОМ

Рассматривается система для реактивного управления мобильным роботом. Она базируется на использовании нейросетевых технологий. Такая система позволяет роботу успешно взаимодействовать с неизвестной окружающей средой без столкновений с препятствиями.

Введение

Способность нейронных сетей к обучению создает предпосылки к созданию на их базе систем управления мобильными роботами, которые способны адаптироваться к окружающей среде. Это позволяет обеспечить робастное управление мобильным роботом при неточной информации от сенсорных устройств. Реактивное управление предполагает взаимодействие робота с неизвестной окружающей средой. В этом случае известны только координаты целевой точки движения. Используя информацию от сенсорных устройств, робот должен достичь целевой точки, избегая столкновений с препятствиями. В настоящее время существуют два основных метода обучения таких систем: обучение с учителем [1,2], при котором известна эталонная выходная информация, и подкрепляющее обучение [1,3,4].

В данной работе описывается нейросетевая система, использующая обучение с учителем. Показан простой способ формирования обучающей выборки. Описываемая система была экспериментально проверена в лаборатории робототехники (Германия), используя мобильные роботы «Walter» и «Pioneer».

1. Общие принципы обхода препятствий

Рассмотрим основные принципы построения реактивной системы для управления мобильным роботом. В этом случае известны только координаты целевой точки. Задача робота состоит в том, чтобы достичь конечного пункта движения в неизвестном пространстве с препятствиями. Из этого следует, что знания в реактивной системе представляются в виде «ситуация—реакция». При этом реакция соответствует текущему направлению движения робота. Задача робота состоит в нахо-

ждении кратчайшего пути между начальной и целевой точкой движения при отсутствии знаний относительно расположения препятствий.

1.1. Входная информация

Входной информацией реактивной системы являются данные от различных сенсорных устройств. В результате обработки таких данных (data fusion) генерируется локальная карта местности, которая соответствует интеграционной картине окружающей обстановки. Такая карта характеризует в определенном радиусе обзора и угловом диапазоне 180° расположение препятствий и расстояния до них:

$$OG = \{S(i), -90^\circ \leq i \leq 90^\circ\},$$

где $S(i)$ – расстояние до препятствия, когда угол между текущим направлением робота и препятствием равен i° .

Такой подход был использован при экспериментах с мобильным роботом “Walter” [4]. Этот робот имеет 7 ультразвуковых датчиков и инфракрасный сканер. Эти датчики позволяют определять препятствия в радиусе 2,4 м и в диапазоне 180° . В результате обработки этих данных формируется локальная карта местности.

1.2. Определение подходящего интервала движения

Первой задачей реактивной системы является определение оптимального интервала движения. Такой интервал является ближайшим к цели и характеризуется линейными (R_L, R_R) и угловыми (W_L, W_R) расстояниями до препятствий (рис. 1).

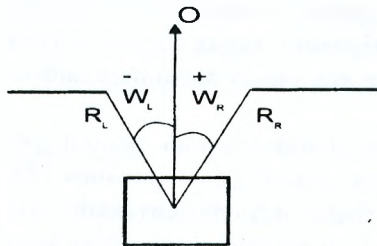


Рис. 1. Линейные и угловые характеристики интервала:
O – текущее направление

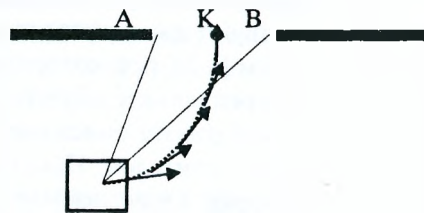


Рис. 2. Траектория движения

Для выбора подходящего интервала движения используется динамическая нейронная сеть с фиксированными весовыми коэффициентами, а также аналитический метод. В случае если свободный интервал движения не найден, происходит поворот робота на 90° , если это возможно, и снова поиск свободного интервала движения.

Структура и алгоритм функционирования данного модуля описаны в [5].

1.3. Определение оптимального направления движения

Следующая задача реактивной системы – определение подходящего направления движения в выделенном интервале. Оптимальным направлением движения является такое направление, которое обеспечивает минимальное угловое расстояние до цели в выбранном интервале движения без столкновения с препятствиями. Определение такого интервала осуществляется путем анализа выбранного интервала движения. Для этого используется аналитический и нейросетевой подходы.

1.3.1. Аналитический подход

Аналитический подход применяется в случае достаточно широких интервалов движения, когда $R_d > 2d$, где R_d – ширина выбранного интервала, а d – ширина робота. В этом случае анализируются следующие данные: угол α между текущим направлением и целью; угловые (W_L , W_R) и линейные (R_L , R_R) характеристики интервала движения. В результате формируется оптимальное направление движения, которое соответствует минимальному расстоянию до целевой точки. Аналитический модуль может работать в двух режимах. Первый из них предназначен для движения в открытом пространстве или в коридоре без препятствий. В этом случае робот движется по прямой линии. Второй используется в пространстве с препятствиями, например при движении робота через дверной проем. В этом случае направление движения должно быть перпендикулярным линии (AB), характеризующей расположение интервала движения (рис. 2). Поэтому траекторией движения робота должна быть дуга эллипса, которая проходит через определенную точку K в выделенном интервале. Позиция точк/и K выбирается исходя из обеспечения кратчайшего расстояния до цели. Структура и алгоритм функционирования этого модуля описаны в [5].

1.3.2. Нейросетевой подход

Во время движения робот должен избегать столкновения с препятствиями. Это сложная задача, если выделенный интервал движения имеет небольшие размеры. В этом случае неточная карта местности может привести к столкновению с препятствием. Нейросетевой подход используется для робастного управления роботом на узких интервалах движения, когда $R_d < 2d$. Если обучить нейронную сеть правильным выходным данным, она будет обеспечивать робастное управление роботом при неточной карте местности. Для этих целей используются многослойные нейронные сети. Структура нейросетевого модуля представлена на рис. 3.

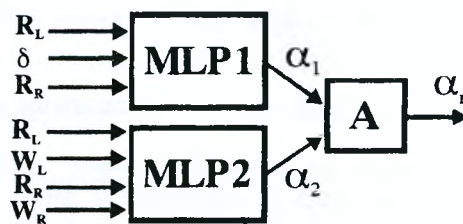


Рис. 3. Архитектура нейросетевого модуля

Этот модуль состоит из двух многослойных нейронных сетей MLP_1 и MLP_2 . Робастное направление движения робота формируется арбитром по правилу:

$$\alpha_r = \begin{cases} \alpha_1, & \text{если } (R_L \vee R_R) \leq g_1 R_i; \\ \alpha_2, & \text{если } (R_L \wedge R_R) > g_1 R_i, \end{cases} \quad (1)$$

где $0 < g_1 \leq 1$ – постоянный коэффициент; R_i – порог видимости системы.

Исходя из (1) $\alpha_r = \alpha_1$, если робот движется в среде с различными препятствиями (например, дверной проем и др.), и $\alpha_r = \alpha_2$, когда робот движется в коридоре. Блок MLP_1 формирует траекторию движения в виде дуги эллипса, что уменьшает вероятность столкновения с боковыми препятствиями. В результате достигается стабильное прохождение робота через дверные проемы. Блок MLP_2 формирует траекторию в виде прямой линии, что позволяет достичь стабильного движения в открытом пространстве и в коридорах. Рассмотрим структуру этих блоков.

Блок MLP_1 представляет собой 3-слойную нейронную сеть, которая состоит из 3 входных, 8 скрытых и 1 выходного элемента. В качестве входной информации используются линейные (R_L и R_R) и угловые (δ) характеристики интервала движения. Здесь $\delta = W_L + W_R$. В качестве функции активации используется сигмоидная функция. Блок MLP_2 представляет собой также 3-слойную нейронную сеть. Она состоит из 4 входных, 6 скрытых и одного выходного нейрона. В качестве входной информации используются линейные (R_L и R_R) и угловые (W_L , W_R) характеристики интервала движения. Для обучения этих нейронных сетей используется алгоритм обратного распространения ошибки с адаптивным шагом обучения [6].

2. Формирование обучающей выборки

Для обучения блока нейронных сетей необходимо сформировать обучающую выборку. Каждый обучающий набор представляется в числовой форме и состоит из нескольких входных и одного выходного значения. Блок нейронных сетей используется для управления роботом

на узких интервалах движения, ширина которых меньше, чем $2d$. Пусть R_i – радиус видимости робота. Тогда необходимо сформировать тренировочные наборы в следующей области V :

$$V \in \begin{cases} R_d \leq 2d; \\ R_L \leq R_i; \\ R_R \leq R_i. \end{cases} \quad (2)$$

В качестве траектории движения блок MLP_1 формирует дугу окружности, которая проходит через центр робота и определенную точку K в выделенном интервале движения. Зная координаты точки K и координаты интервала движения (X_A, Y_A, X_B, Y_B), можно определить траекторию движения робота и направление движения в каждой точке (см. рис. 2). В результате для одного положения робота относительно выделенного промежутка движения получается совокупность тренировочных наборов.

Производя вращение выделенного интервала движения $[A B]$ и точки K относительно центра робота, можно получить различные тренировочные наборы. Изменяя в области V расположение робота относительно интервала движения и выполняя перечисленные выше операции, можно получить обучающую выборку, которая состоит из множества тренировочных наборов. Аналогичный подход применяется для блока MLP_2 .

Для обеспечения робастного управления роботом при неточной карте местности необходимо соответствующим образом выбирать положение точки K в выделенном интервале движения. Тогда, если обучить нейронную сеть правильным выходным данным, то она сможет обеспечить устойчивое управление роботом при неточной карте местности.

Представленный подход характеризуется минимальным набором экспериментальных данных. Достаточно определить только положение точки K и характеристики интервала движения. Было проведено компьютерное моделирование модуля нейронных сетей. Размер обучающей выборки для этого модуля составил 120 наборов. После обучения робот успешно проходил узкие участки движения с различных позиций.

3. Модуль точного управления

Недостатком описанных выше подходов является то, что они не учитывают расстояния с боковых сторон робота до препятствий. В результате при выполнении маневров возможны столкновения робота с препятствиями.

Для того чтобы избежать столкновений в таких ситуациях, необходимо, чтобы боковое расстояние от робота до препятствия было больше, чем радиус окружности, описанной вокруг робота:

$$S > \frac{d}{2}\sqrt{2}, \quad (3)$$

где d – ширина робота.

Если условие (3) не выполняется, то управление роботом переходит к модулю точного управления. В этом случае угол поворота робота в любом направлении является постоянной величиной и равен 1° .

Структура модуля точного управления изображена на рис. 4. Входной информацией для данного модуля является текущее направление движения робота α_a , которое поступает от аналитического модуля или модуля нейронных сетей, а также локальная карта местности $\{S(-18)...S(0)...S(18)\}$.

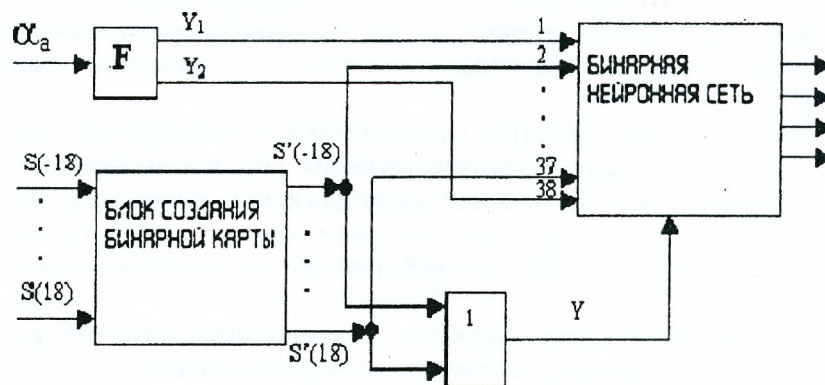


Рис. 4. Модуль точного управления

Блок преобразования F предназначен для преобразования углового направления движения α_a в двоичный вид. Это необходимо для управления бинарной нейронной сетью. Блок F выполняет следующие функции:

$$Y_1 = \begin{cases} 1, & \text{если } \alpha_a > 0; \\ 0, & \text{если } \alpha_a \leq 0. \end{cases} \quad Y_2 = \begin{cases} 1, & \text{если } \alpha_a < 0; \\ 0, & \text{если } \alpha_a \geq 0. \end{cases} \quad (4)$$

Блок создания бинарной карты местности необходим для формирования локальной карты местности заданной конфигурации (рис. 5) и формирования сигнала Y активации бинарной нейронной сети. Такая карта нужна для управления роботом в ситуациях, когда препятствия расположены слишком близко (на расстоянии меньшем чем Δ) от края робота.

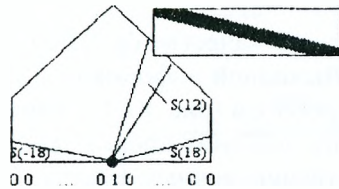


Рис. 5. Конфигурация карты местности

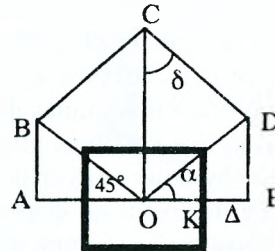


Рис. 6. Пример создания бинарной карты местности

Треугольная форма выбрана для обеспечения плавных маневров при наличии препятствий по фронту робота. Технология преобразования состоит в том, что если препятствие находится в области $ABCDE$, то соответствующие элементы $S'(p)$ устанавливаются в единичные значения, а иначе – в нулевые (рис. 6).

В результате получится бинарный массив, который характеризует наличие препятствий в соответствующей области. Блок создания бинарной карты состоит из одного слоя пороговых нейронных элементов (рис. 7), каждый из которых соответствует определенному сектору карты местности.

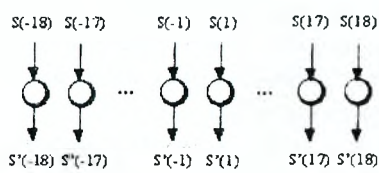


Рис. 7. Создание бинарной карты местности

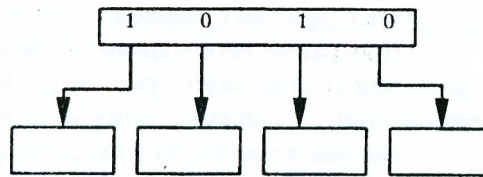


Рис. 8. Команды управления роботом

Нейронные элементы выполняют следующие функции:

$$S'(p) = \begin{cases} 1, & \text{если } S(p) \leq T(p); \\ 0, & \text{если } S(p) > T(p). \end{cases} \quad (5)$$

Здесь $T(p)$ – порог данного нейрона. Для создания бинарной карты местности в данной области необходимо соответствующим образом определить значения порогов нейронов.

Восемнадцать нейронов ($S'(-18)...S'(-10)$, $S'(18)...S'(10)$), которые соответствуют области OAB и ODE , участвуют в формировании сигнала возбуждения бинарной нейронной сети Y (см. рис. 4). В результате, если препятствие находится в этих областях, то управление роботом производит бинарная нейронная сеть.

Эта сеть состоит из трех слоев. В качестве функции активации используется сигмоидная функция. Выходной информацией нейронной сети являются команды управления роботом (рис. 8). Бинарная нейронная сеть функционирует по принципу огибания препятствий. Для обучения ее необходимо формировать тренировочные наборы. Для создания обучающей выборки используется логический подход. Как показали эксперименты, необходимый объем обучающей выборки составляет 40 наборов.

4. Результаты экспериментов

В мае-июне 2000 г. были проведены эксперименты с мобильными роботами в лаборатории робототехники в Fachhochschule Ravensburg-Weingarten (Германия).

Была разработана программная система для управления мобильным роботом, функционирующая в системе Solaris на машинах класса Sun Sparcstation. Она была протестирована на мобильных роботах Pioneer (ActivMedia) и Walter (LabMate).

Система была создана с использованием языка C/C++, библиотеки PAI (Pioneer Application Interface), обеспечивающей связь с мобильным роботом Pioneer, и стандартной библиотеки X-Windows. Робот соединялся с рабочей станцией с помощью последовательного кабеля.

Программе необходимо указать начальную и конечную позицию движения и стартовать (рис. 9, на карте блоками показаны реальные препятствия, а точками – показания датчиков робота).

На рис.9 видно, что существует проблема обработки неточных показаний датчиков. Кроме того, робот Pioneer имеет сравнительно небольшое количество датчиков, поэтому было реализовано простое запоминание глобальной карты местности (т.е. показания датчиков сохраняются и используются при формировании локальной карты местности в последующем).

Вначале робот должен был сформировать обучающую выборку способом, указанным в разд. 2. После этого производилось обучение нейронных сетей методом обратного распространения ошибки с адаптивным шагом обучения [6].

Проведенные эксперименты показали способность робота двигаться в неизвестной окружающей среде с препятствиями без столкновений, проходя через узкие интервалы движения (такие, как дверной проем).

4. Golovko V., Schilling K., Roth H., Sadykhov R., Albertos P., Dimakov V. The architecture of the neural system for control of a mobile robot // Proceedings of the ICNNAI'99, Brest, Belarus, 1999. – P. 57-61.

5. Golovko V. and Dimakov V. Architecture of Neural System for Control of Autonomous Vehicles // Preprints of the 3rd IFAC Symposium of Intelligent Autonomous Vehicles, Madrid, Spain, 1998. – Oxford UK: Elsevier Science Ltd. – 1998. – V. 1.

6. Golovko V., Savitsky Y. New approach of the recurrent neural network training // Proceedings of the ICNNAI'99, Brest, Belarus. – 1999. – V.1. – P.32-35.

*Институт технической
кибернетики НАН Беларуси*