Обучение систем с дискретным управлением

Голиков К.А.

В докладе изложена работа по созданию алгоритма обучения системы с дискретным управлением действовать и достигать целей. Обучение происходит на основе проб и ошибок. Весь опыт системы сохраняется в Базе Данных. Оптимизация алгоритма производится по двум критериям: точность достижения поставленных целей и максимальное сокращение времени обучения. Сокращение времени обучения реализуется, главным образом, уменьшением количества пробных действий с помощью методов прогнозирования и интерполяции по опытным данным.

Ключевые слова: позиционирование, алгоритм обучения, робот, интерполяция, аппроксимация.

Пусть есть некоторый **Алгоритм** – обучаемый субъект, принимающий решения, который автономно исследует оптимальное поведение с помощью попыток и ошибок, взаимодействует с устройством для решения проблемы управления.

Пусть есть **Система** – это то, с чем может взаимодействовать алгоритм в процессе обучения. Предполагается, что система – это некоторое физическое устройство или виртуальная модель устройства (через систему можно опосредованно взаимодействовать с иными системами и окружающей средой).

Алгоритм работает в дискретном времени: каждый такт времени он может узнавать часть данных о состоянии системы, а также выбирать действие, которое будет производиться системой, на основе полученных данных и предположений: любые произведённые системой действия сохраняются в БД, на основе этих данных делаются прогнозы.

У системы есть **приводы**, обеспечивающие действия системы по своим внутренним неизвестным законам, зависящим от внутреннего состояния системы и изменяющихся во времени. У системы есть **входы**, алгоритм подаёт на них дискретные значения - указания приводам действовать. У системы есть **выходы**, с них алгоритм может считывать

вещественные значения, получать частичные данные о состоянии - *обратную связь*. У системы есть фиксированное **начальное положение**. Из этого начального положения алгоритм учится достигать *целей*. Цели характеризуются системой и задачей, которую система предназначена решать.

Мы изначально *отказываемся от возможности аналитически построить точную или приближённую физическую модель системы, полагаемся только на опыты и обратную связь*. Это принципиальное ограничение нашего исследования.

АВТОМАТНАЯ МОДЕЛЬ СИСТЕМЫ

Будем рассматривать **систему**, как чёрный ящик, который работает по тактам.

Входы – m штук (m-чётно) бинарных значений (1 или 0), означают включён ли привод в текущий такт или выключен

+ аппаратная кнопка $reset$ (1 или 0) - возвращение системы в начальное положение.

Выходы – y штук вещественных чисел, описывают известную часть состояния системы

- функция выходов $f(c_1, \ldots, c_m, reset) = (x_1, ..., x_y)$
- функция перехода $g(c_1, \ldots, c_m, reset, q_1, \ldots, q_i, ...)$ – состояние внутренних переменных, их количество и структура связей неизвестны.

Кнопка reset возвращает систему в известное начальное положение (абсолютно точно) из любого состояния.

Количество приводов чётно, потому что любое действие должно быть обратимо. В систему приводы добавляются по два антагониста, работающих в разных направлениях. *Известно, как нумеруются входы приводов, для каждого управления можно получить обратное.* Если произвести действие системой, подавая на входы C до некоторого конечного состояния, тогда, выполняя инверсное управление C', из этого конечного состояния можно вернуться в начальное положение почти точно. Всегда точно вернуться обратно (без reset) нельзя, потому что внутренние законы функционирования приводов системы зависят от состояния системы и времени - со временем они гладко изменяются в небольших пределах. Таким образом, от действия к действию конечное состояние системы может смещаться. Следовательно, запомненное в базе данных действие, успешно достигшее цели, на практике не всегда её достигает (корректирующие операции будут обсуждены в конце статьи).

Каждый такт на входы системы можно подавать значения включены или выключены конкретные приводы, т.е. усилие привода - одна из неизвестных внутренних переменных, напрямую управлять ей нельзя. Можно варьировать *время запуска и длительность работы каждого привода*. Т.к. в приводах и внутренних механизмах системы есть определённое трение и прочие лаги, то для того, чтобы активация входов привела к фактическому изменению состояния системы, приводы должны быть активны в течение какого-то времени. Одного такта активации недостаточно, чтобы привод развил силу, способную преодолеть величину трения, сдвинуть систему из одного состояния в другое. Единицы нужно группировать в последовательности, такие входы приводят к результатам.

Кроме того, самым дорогим ресурсом при обучении - является *время работы системы*. Чтобы производить как можно больше действий, нужно чтобы сами действия были бы короткими. Поэтому вводится n – *максимальное число тактов времени действия системы*. Назовём **эпизодом** обучения изменение состояния системы от известного начального положения в некоторое *конечное положение* с последующим возвратом в начальное положение (обычно с помощью кнопки reset) за время меньшее или равное n.

Определим понятие **управление** - это матрица, у которой число строк m(число приводов) и число столбцов n(максимальная длительность эксперимента), эта матрица определяется последовательностями времени работы приводов в течение отдельного эпизода обучения. При этом наложим дополнительное ограничение на управление - общее число последовательностей единиц во всех строках матрицы должно быть не больше k, где $k << n$. В таком случае движение получится не очень длинное, более прямолинейное – сложнее сгенерировать управление, ведущее систему к цели окольными путями.

Решения задачи будем искать в виде **управлений**.

ИНТЕРПРЕТАЦИЯ ДЛЯ КОНКРЕТНОЙ ЗАДАЧИ

Обучение алгоритма отрабатывается в *виртуальной среде* для задачи **позиционирования разных роботов**. Применяя алгоритм для разных систем показываем широту возможностей алгоритма и достаточность описанных выше ограничений для успешного решения задачи.

В рамках задачи позиционирования количество приводов m, для которого исследуем поведение системы маленькое от 4 до 10. k для управлений - в пределах 4-8 одновременно активных приводов, а число тактов

эпизода $n = 10000$. Выходов у системы ровно 4, понимаются как координаты *базовой точки* системы (x,y) и вектор *моментальной скорости* этой точки $v = (v_x, v_y)$.

Цели интерпретируются как точки, в которые нужно привести базовую точку системы с заданной наперёд *точностью* быстрейшим образом. А *конечным положением* для эпизода является обязательно статичное положение системы - полная остановка базовой точки. Система пришла в целевую точку (выполнила задание), если её базовая точка находится в малой окрестности целевой точки (в соответствии с точностью), и скорость базовой позиции $v = (0., 0.)$. Цели выбираются произвольным образом, они больше нужны для обучения, чем для работы системы. Пробуя попадать в конкретные цели, система в итоге учится попадать в любые точки. При этом, чем больше вокруг новой точки, в которую хочется спозиционироваться, располагается выученных конечных положений, тем точнее действие будет произведено обученной системой.

Для уменьшения времени обучения целевыми точками лучше покрывать область, в которой в дальнейшем будет производиться полезная работа. Например, если система - это робот манипулятор, который должен научиться закручивать шурупы в отверстия на определённых координатах. Нужно, во-первых, расположить робота так, чтобы все отверстия лежали в рабочем пространстве манипулятора, во-вторых, закрепить банку с шурупами в начальном позиции робота, в-третьих, целевые точки задать так, чтобы они покрывали область детали с отверстиями, в которые нужно вкручивать шурупы.

ПРИМЕРЫ СИСТЕМ

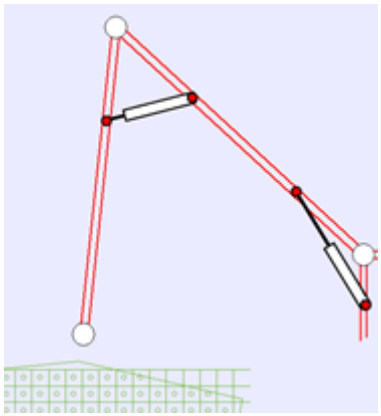

Рис. 1.

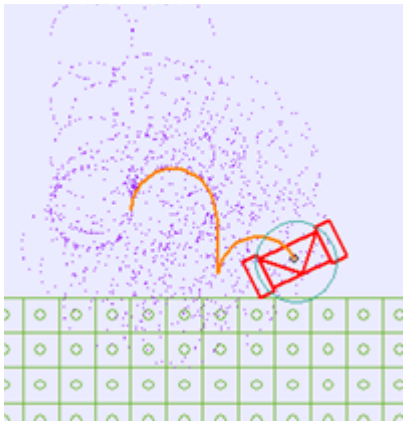

Рис. 2.

1. **Манипулятор** с 2-3мя вращательными и призматическими сочленениями (рис.1). Базовая точка – точка центра захвата. Привод – действие: либо выдвигающее, либо сдвигающее пневмо-цилиндр сочленения.

2. **Вездеход** с 2мя гусеницами (рис.2). Базовая точка – центр робота. Привод - вращение гусеницы либо вперёд, либо назад.

СЛОЖНОСТИ В РЕШЕНИИ ЗАДАЧИ

Подчеркнём отдельно те сложности, которые необходимо преодолеть при решении задачи в описанных выше ограничениях:

1. За минимальное время произвести системой *ценные действия*, отражающие возможности и принципы функционирования неизученной системы. Получить управления, хорошо подходящие для интерполяции и предсказаний поведения системы. В рамках задачи позиционирования: получить равномерную плотную решётку конечных положений покрывающую целевые точки.

2. Получить метод интерполяции по сохранённым в БД управлениям, достаточно точный, а также метод выбора действий, которые ценно будет произвести для уточнения алгоритма интерполяции, чтобы минимизировать возможность промахов.

3. В условиях отсутствия времени на переобучение при изменяющихся законах функционирования приводов системы построить алгоритм адаптации к этим изменениям.

1. РАВНОМЕРНАЯ РЕШЁТКА КОНЕЧНЫХ ПОЛОЖЕНИЙ

Случайная генерация управлений не даёт выборки действий, по которым может получиться хорошая интерполяция. Получается много конечных положений системы рядом с начальным положением и совсем мало в остальном рабочем пространстве робота. Кроме того, эти сгенерированные управления – описывают неоптимальные траектории (рис.3), сильно отличаются по содержащимся последовательностям активных приводов, по ним нельзя построить достаточно точную интерполяцию за адекватное время обучения.

Написан двухэтапный алгоритм построения равномерной решётки конечных положений в нужной области, он обеспечивает возможность сравнимости и усреднения управлений. На первом этапе варьированием числа активных приводов и их длительностей получается *разряжённая решётка действий системы, конечные положения которых покрывают всё рабочее пространство робота.* Это даёт понимание того, на что в принципе способна система, где располагаются *цели в координатах*

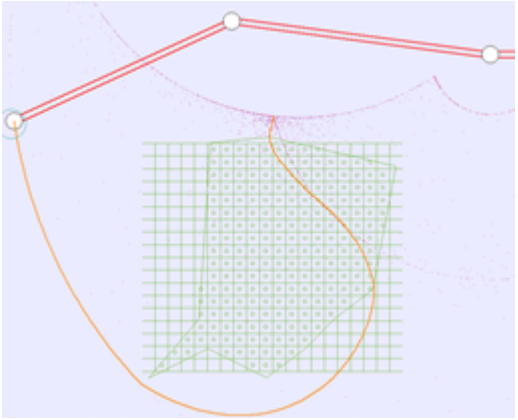

Рис. 3.

управлений. Второй этап - построение плотной по возможности равномерной решётки, покрывающей все целевые точки. Идея в следующем: длительности работы мускулов увеличиваются с некоторым шагом, одновременно производятся смещения моментов времени старта последовательностей работы приводов относительно друг друга в создаваемом управлении. Подбор управлений - это подбор значений аргументов неизвестной функции, чтобы выдерживать равные дистанции между её значениями. "Адаптивный"подбор изменений управлений осуществляется методами Монте-Карло и МНК на основе имеющиеся в БД управлений выполненных действий.

2. ИНТЕРПОЛЯЦИЯ ПО ПРОИЗВЕДЁННЫМ ДЕЙСТВИЯМ

После построения плотной решётки конечных положений. Приближение к целевым или желаемым точкам осуществляется методом Стохастического Градиентного спуска. Не всякие два управления подходят для градиентного спуска. Если управления очень непохожи друг на друга, то среднее арифметическое не существует – всё разное. Если есть среднее арифметическое, значит разные лишь длительности работы одних и тех же приводов с сохранением их порядка активации.

На рис.4 показана ситуация после окончания работы алгоритма получения плотной решётки. На нём видно два кластера управлений. Внутри кластеров получение промежуточных управлений возможно, а управления из разных кластеров между собой усреднить нельзя - результат усреднения не приведёт в конечное положение между выбранными.

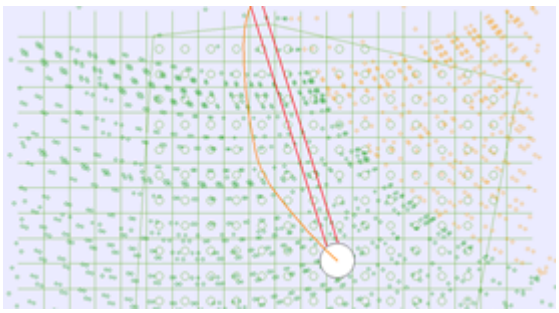

Рис. 4.

Интерполяции строятся локальные по ближним управлениям для ближних конечных положений, попадающие в один кластер. В этом случае построение интерполяции не очень затратная по вычислительной мощности операция, а точность приближения у неё довольно высокая.

3. АДАПТАЦИЯ К ИЗМЕНЕНИЯМ В ДЕЙСВИИ ПРИВОДОВ

В реальном мире при функционировании системы в среде могут возникать

• *системные изменения* – изменения в приводах происходят постепенно и медленно, вносят ограниченные смещения, действуют длительно в течение всего эпизода переобучения (атмосферное давление, износ подшипников и пр.), к ним нужно *уметь приспособляться*,

• *мгновенные изменения* – сильное смещение, действует один такт времени, неповторяется (порыв ветра, задевание недопустимого объекта), нужно *уметь отфильтровывать этот шум*.

Если бы условия менялись сильно и непредсказуемым образом, действовали бы длительно, то, очевидно, прогностические методы к такому окружению были бы неприменимы. Законы движения и изменения в приводах задаются разностными дифференциальными уравнениями, проводя эксперименты с переобучением управления системой важно выяснить границы применимости адаптации для разных видов уравнений.

Параметры внутреннего состояния системы делятся на три вида:

• задаваемые (вес груза, угол установки системы, . . .),

• наблюдаемые (температура помещения, износ подшипников, . . .),

• ненаблюдаемые (изменение вязкости рабочего тела пневматики, трение, влажность воздуха, плотность среды, . . .)

Внутренние параметры переходятся в тот или иной вид, исходя из технической возможности задать параметры, предусмотреть и установить необходимые датчики на физическое устройство и пр.

Итак, *память обширная*, предоставляет возможности по построению адекватных предсказаний, *но её данные со временем устаревают и перестают верно отображать обстановку вещей*. Например, у манипулятора, из-за засорения локтевого подшипника, появляется неодинаковое для любой точки рабочей области смещение влево на несколько миллиметров. Подавая управление, ранее точно приводившее захват в цель, сейчас приводит его в точку левее цели.

ОБУЧЕНИЕ АДАПТАЦИИ

Для того, чтобы переобучаться времени нет, уточнение действий производятся "на лету". После успешной фазы обучения происходит эксплуатация обученной системы. База Данных постоянно дополняется. Так как записи действий в БД сохраняются навсегда, важно не запутаться в одинаковых действиях со смещёнными конечными положениями. Для этого выделяется первое *эталонное действие с определённым управлением*, при повторениях в указанную запись добавляются *фактические смещения базовой точки* системы.

Процедура «Удержание на траектории»

Во время повторения запомненной траектории в каждый момент времени производится слежение за величиной и направлением отклонения от эталонной траектории, в управление подмешивается сначала выбранная наугад инверсная часть, с каждым тактом она подбирается точнее, чтобы компенсировать выявленное смещение.

Данная процедура позволяет уменьшить величину конечного отклонения насколько это возможно для необученного алгоритма с подкреплением. Для того чтобы полностью нивелировать смещение, необходимо использовать статистические методы.

Процедура «Изучение смещений»

1. Разделить область позиционирования (конечных положений) на несколько небольших регионов.

2. Для каждого региона регистрировать направление отклонений, коллекционировать их, похожие собирать в кластеры, задавать вес больше для актуальных смещений.

3. По данным кластера построить обратное преобразование: по смещению найти инверсное управление, которое нужно подмешать к эталонному управлению, чтобы вернуться на траекторию и попасть в итоге в цель.

4. Для каждого исполняемого действия, как можно раньше выявлять самый похожий кластер.

Функцию аппроксимации полного эпизода - возвращающую управление по начальной и конечной позициям системы - обучить дольше и затратнее[4], чем несколько простых функций аппроксимаций, каждая из которых решает простую подзадачу, а все работают вместе, дополняя друг друга.

Список литературы

[1] Яблонский С.В. Введение в дискретную математику. — М.: Высшая школа, 2006.

[2] Саттон Р.С., Барто Э.Г. Обучение с подкреплением — 2-е изд. — М.:БИНОМ.Лаборатория знаний, 2014.

[3] Богачёв К.Ю. Практикум на ЭВМ. Методы приближения функций — М.: Мех.-мат. МГУ, 1998.

[4] Окуловский Ю.С. Интеллектуальные алгоритмы калибровки робототехнических систем — [Электронный ресурс] // Екатеринбург: УрГУ, 2010. URL: http://elar.urfu.ru/bitstream/10995/21944/1/Okulovskii_GK_P1047.pdf (дата обращения: 13.11.2018).

Learning systems with descrete control

Golikov K.A.

The report outlines the work to create a discrete-control system learning algorithm for acting and achieving goals. Learning is based on trials and misses. The entire experience of the system is stored in the Database. The algorithm is optimized by two criteria: the accuracy of achieving the goals and the maximum reduction in training time. The reduction in training time is implemented mainly by reducing the number of trials using prediction methods and interpolation by experimental data.

Keywords: positioning, learning algorithm, robot, interpolation, approximation.