Содержание

Введение 7

1 Обучение с подкреплением 8

1.1 Обзор обучения с подкреплением (RL) 8

1.2 Элемент обучения с подкреплением 9

1.3 Марковский процесс принятия решений (МППР) 10

1.3.1 Задачи оптимального последовательного решения 11

1.3.2 Обратная рекурсия для значения 12

1.3.3 Обзор адаптивного динамического программирования 12

1.3.4 Уравнение Беллмана и уравнение оптимальности Беллмана 13

1.4 Итерация по стратегии и итерация по критерию 15

1.4.1 Итерация по стратегии алгоритм 15

1.4.2 Итерации по критерию 15

1.4.3 Сравнение итерации по стратегии и итерации по критерию 16

2 Регулятора для линейных систем 17

2.1 Оптимальный регулятор 17

2.2 Адаптивный регулятор 17

2.3 Оптимальный адаптивный регулятор 18

3. Задача оптимального управления для нелинейных систем 21

3.1 Оптимальное управление и уравнение Гамильтона Якоби – Беллмана 21

3.2 Адаптивный алгоритм оптимального управления на основе итераций по стратегии 23

3.3 Реализация алгоритма итерации по стратегии в реальном времени 24

4 Результаты моделирование 26

4.1 Регулятор для линейных систем 26

4.2 Исследование влияния времени выборки *T* на оптимальный адаптивный регулятор. 29

4.3 Регулятор для нелинейных систем 33

4.4 Регулятор линейной системы для перевернутого маятника и тележки 35

4.5 Моделирование для маятника 38

5 Заключение 42

Список использованных источников 43

Введение

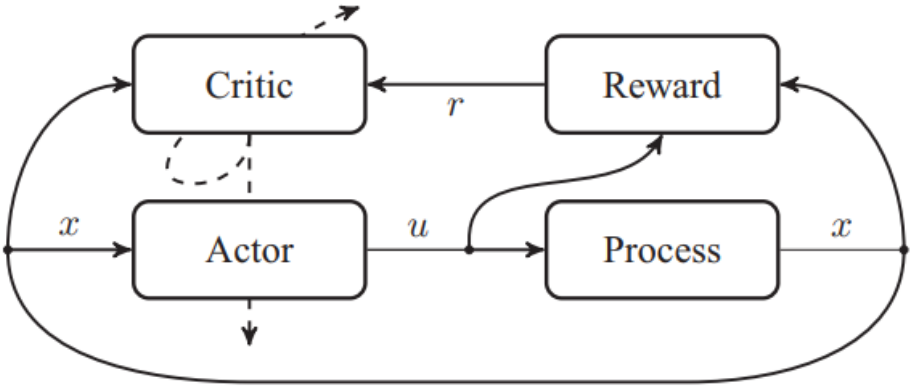
Оптимальные регуляторы обычно разрабатываются в автономном режиме путем решения уравнений Гамильтона-Якоби-Беллмана (HJB), например, уравнения Риккати, с использованием полных знаний о динамике системы. Такой регулятор имеет множество применений в науке, технике и исследованиях. Для расширения применимости оптимального управления необходима разработка алгоритмов оптимального управления, способных адаптироваться к изменению динамических свойств системы. Адаптивные регуляторы учатся в режиме реального времени управлять системами с неизвестными параметрами, используя данные, измеряемые в моменте вдоль траекторий системы. Однако адаптивные регуляторы обычно не сходятся к оптимальным решениям. Метод обучения с подкреплением (RL) и адаптивное динамическое программирование стали прочной основой для разработки адаптивных оптимальных регуляторов. Такой метод предполагает причинно-следственную связь между действиями и вознаграждением или наказанием. Агент взаимодействует со своей средой посредством действия, и за этим действием следует вознаграждение (положительный сигнал RL), что дает снижение затрат на управление, а наказание (отрицательный сигнал RL) - увеличение затрат на управление. Действуя таким образом, алгоритм RL со временем обучается оптимальной стратегии. Алгоритм применяется для решения множества задач оптимального управления, включая обеспечение устойчивости, подавление шумов, оптимальное слежение за траекторией и т. д., без необходимости решения уравнений Гамильтона - Якоби - Беллмана, что позволяет решать задачу управления в условиях неполной информации о динамике объекта.

1 Обучение с подкреплением

1.1 Обзор обучения с подкреплением (RL)

Обучение с подкреплением — это одно из машинного обучения. Речь идет о принятии правильных действий для получения максимальной выгоды в конкретной ситуации. Обучение с подкреплением (RL) — это наука о принятии решений. Речь идет об обучении оптимальному поведению в окружающей среде для достижения максимального вознаграждения. В области теории управления обучение с подкреплением относится к методу, который позволяет разрабатывать адаптивные регуляторы в реальном времени для решения определяемых пользователем задач оптимального управления.

Обучение с подкреплением подразумевает наличие причинно-следственной связи между действиями и вознаграждением или наказанием. Это структура, в которой агент (или регулятор) оптимизирует свое поведение, взаимодействуя с окружающей средой. Совершив действие в каком-то состоянии, агент получает скалярное вознаграждение или наказание от среды, которое дает агенту представление о качестве этого действия. Алгоритмы RL исходят из идеи, что успешное поведение (приносящее высокие награды) будет запоминаться в том смысле, что оно имеет тенденцию использоваться повторно в последующих случаях [1]. Идея RL возникла из экспериментов обучения в биологии. RL теоретически тесно связан прямо и косвенно с адаптивными методами оптимального управления.



Рисунка 1. Схематический обзор алгоритма актер-критик [2]

Одной из популярных структур RL является структура «Актер-критик» [Barto, Sutton, Anderson 1983], Актерско-критические методы сочетают в себе преимущества методов "только актер" и "только критик" [3], в которой компонент «Актер» выполняет действия (стратегия управления), влияющие на окружающую среду, а секция «Критик» оценивает это действие. На основе этой оценки используется множество методов для калибровки или улучшения действия, чтобы новое действие создавало большую ценность (награду), чем предыдущее. Таким образом, структура «Актер-Критик» состоит из двух этапов: оценка поведения и улучшение поведения. Поведенческая оценка

проводится путем наблюдения за результатами, полученными из окружающей среды после выполнения определенного поведения.

Метод «Актер-критик» обычно обладают хорошей сходимостью, в отличие от методов, основанных только на критике [4].

1.2 Элемент обучения с подкреплением

В этом разделе представлены некоторые элементы и термины обучения с подкреплением. Можно также рассмотреть некоторые элементы обучения с подкреплением, представленные в [5].

В обучении с подкреплением есть термин «агент», который означает субъект, который взаимодействует с окружающей средой посредством действий.

Среда - это окружающее агента пространство, в котором он существует и взаимодействует.

Действия - это методы агента, которые позволяют ему взаимодействовать с окружающей средой и изменять ее. Основываясь на состоянии S(t) текущей среды, агент будет выполнять действия a(t)

Стратегия — это то, что определяет, как агент действует в данный момент времени. Другими словами, стратегия — это отображение состояний среды на действия, которые будут выполняться в этих состояниях. Политика является ядром агента при определении поведения. В некоторых случаях политика может быть простой функцией или таблицей поиска. В других случаях политика может включать в себя обширные вычисления, например процесс поиска.

Вознаграждение - За каждое действие среда посылает агенту определенное вознаграждение. Цель агента - максимизировать общее вознаграждение, которое он получает в течение длительного периода времени. Сигнал о вознаграждении помогает определить, какие события являются хорошими, а какие плохими для агента, а также является основной базой для изменения стратегии. Если действие, выбранное в соответствии с стратегией, приносит низкое вознаграждение, стратегия может быть изменена. В будущем агент будет выбирать другие действия в аналогичных ситуациях.

Исследование и использование - Одна из проблем, возникающих в обучении с подкреплением — это компромисс между использованием и исследованием. Чтобы получить большее вознаграждение, агент должен отдавать предпочтение тем действиям, которые он уже пробовал в прошлом и которые помогли ему достичь вознаграждения. Агент рассматривает все возможные действия для данного состояния, а затем выбирает действия, основываясь на максимальном значении этих действий. Это называется использованием, поскольку для принятия решения мы используем доступную информацию.

Кроме того, вместо выбора действий, основанных на максимальном будущем вознаграждении, агент может выбирать действия случайным образом. Случайные действия важны, поскольку они позволяют агенту исследовать и обнаруживать новые состояния, которые не были выбраны во время операции. Агент должен использовать то, что он испытал, для получения вознаграждения,

но он также должен исследовать, чтобы сделать лучший выбор в отношении будущих действий.

1.3 Марковский процесс принятия решений (МППР)

Обучение с подкреплением можно выразить с помощью марковского процесса принятия решений (МППР), как показано на рисунке 2. Каждое окружение представлено состоянием, которое отражает происходящее в окружении. Агент RL совершает действия в среде, которые вызывают изменение текущего состояния среды, порождая новое состояние, и получает вознаграждение в зависимости от результатов. Агент получает положительное вознаграждение за хорошие действия и отрицательное - за плохие, что помогает ему оценить выполненное действие в данном состоянии и учиться на опыте. Идея МППР представлена в статье [7].

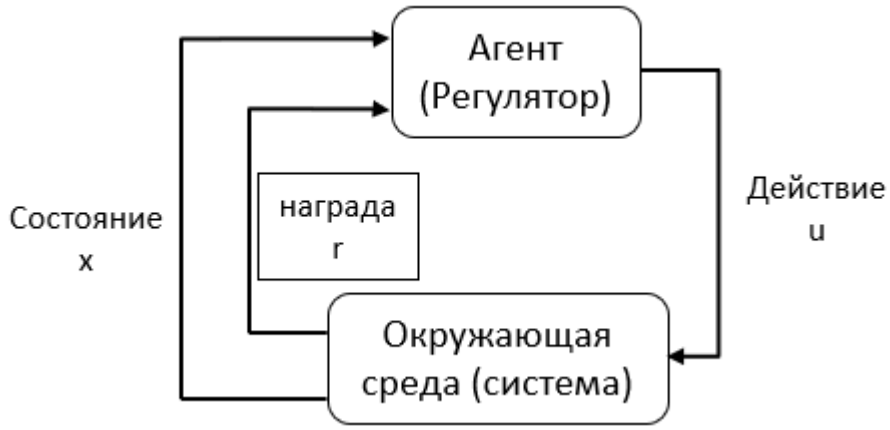


Рисунок 2.1 - Взаимодействие агента и среды в марковском процессе принятия решений.

Рассмотрим МППР *(X, U, P, R)*, где X — набор состояний, а U — набор действий или элементов управления. P - вероятность перехода между состояниями. Значение P находится в интервале [0;1], эта вероятность перехода описывает условную вероятность $P^{u}_{x,x'} = Pr\{x' \mid x, u\}$ перехода из состояния x в состояние x', когда вы находитесь в состоянии x и выбираете действие u. Функция ценности $R: X \times U \times X \to R$ это ожидаемая непосредственная стоимость $R^{u}_{x,x'}$ уплаченная после перехода в состояние $x' \in X$ учитывая, что МППР запускается в состоянии $x \in X$ и выполняет действие u. Вероятность перехода $P^{u}_{x,x'}$ совершенно не зависит от того, как МППР достигла этого состояния, она зависит только от текущего состояния x.

Проблема МППР заключается в нахождении отображения $\pi: X \times U \to [0,1]$, которое для каждого состояния x и действия u дает условную вероятность $\pi(x,u) = Pr(u \mid x)$ выполнения действия u при условии, что МППР находится в состоянии x. Такое отображение называется управлением с обратной связью или стратегией действий или стратегией. Стратегия $\pi(x,u) = \Pr\{u|x\}$ называется стохастической или смешанной, если существует ненулевая вероятность выбора более одного элемента управления в состоянии x.

Смешанные стратегии можно рассматривать как векторы распределения вероятностей, имеющие в качестве компонента i вероятность выбора i-го управляющего воздействия в состоянии $x \in X$. Если отображение $\pi(x,u) = \Pr\{u|x\}$ допускает только одно управление с вероятностью единица, то в каждом состоянии x отображение называется детерминированной стратегией. Тогда $\pi(x,u) = \Pr\{u|x\}$ соответствует функции, отображающей состояния в управления $\mu(x): X \to U$.

МППР, которые имеют конечные пространства состояний и действий, называются конечными МППР.

1.3.1 Задачи оптимального последовательного решения

Обозначим значения состояния и действия в момент времени k через x_k, u_k. Зачастую желательно, чтобы системы, спроектированные человеком, были оптимальными с точки зрения экономии ресурсов, таких как стоимость, время, топливо и энергия.

Определение ценности этапа во время k по формуле:

$$r_k = r_k(x_k, u_k, x_{k+1}) \quad (1)$$

Тогда функция ценности:

$$R^u_{x,x'} = E\{r_k | x_k = x, u_k = u, x_{k+1} = x'\} \quad (2)$$

с $E\{.\}$ - оператором ожидаемого значения.

Определение индекса производительности как сумму будущих затрат за интервал времени $[k, k+T]$:

$$J_{k,T} = \sum_{i=0}^{T} \gamma^i r_{k+1} = \sum_{i=k}^{k+T} \gamma^{i-k} r_i \quad (3)$$

где γ - коэффициент дисконтирования, уменьшающий вес затрат, понесенных в дальнейшем.

Предположим, что агент выбирает стратегию управления $\pi_k(x_k, u_k)$, которая используется на каждом этапе k МППР. Нас в первую очередь интересуют стационарные стратегия, где условные вероятности $\pi_k(x_k, u_k)$ не зависят от k. Тогда условные вероятности $\pi_k(x,u) = \pi(x,u) = \Pr\{u|\, x\}$ для всех k. Нестационарная детерминированная стратегии имеет вид $\pi = \{\mu_0, \mu_1, \mu_2, \dots\}$, где каждая запись представляет собой функцию $\mu_k(x): X \to U;\ k = 0, 1, \dots$. Стационарные детерминированные стратегия не зависят от времени, т. е. имеют вид $\pi = \{\mu, \mu, \dots\}$.

Выберите фиксированную стационарную стратегию $\pi(x,u) = \Pr\{u|\, x\}$. Тогда "замкнутая" МППР сводится к цепи Маркова с пространством состояний X. То есть вероятности переходов между состояниями фиксированы, и дальнейшая свобода выбора действий отсутствует. Вероятности переходов этой цепи Маркова задаются следующим образом:

$$p_{x,x'} \equiv P^{\pi}_{x,x'} = \sum_u \Pr\{x'|x,u\} \Pr\{u|x\} = \sum_u \pi(x,u) P^u_{x,x'} \quad (4)$$

где используется тождество Чепмена-Колмогорова [3].

Ценность стратегии определяется как условное ожидаемое значение будущих затрат при запуске в состоянии x в момент k и после этого следовании стратегии $\pi(x,u)$:

$$V_k^\pi(x) = E_\pi\{J_{k,T}|x_k = x\} = E_\pi\left\{\sum_{i=k}^{k+T}\gamma^{i-k}r_i \mid x_k = x\right\} \quad (5)$$

где $E_\pi\{.\}$ - ожидаемое значение при условии, что агент придерживается стратегии $\pi(x,u)$, и $V^\pi(x)$ называется функцией ценности для стратегии $\pi(x,u)$, что является ценностью пребывания в состоянии x с учетом того, что является ценностью пребывания в состоянии x, учитывая, что стратегия $\pi(x,u)$.

Основная цель МППР — определить стратегию $\pi(x,u)$, позволяющую минимизировать ожидаемые будущие затраты

$$\pi^*(x,u) = argmin\, V_k^\pi(s)$$

$$= \underset{\pi}{\operatorname{argmin}}\, E_\pi\left\{\sum_{i=k}^{k+T}\gamma^{i-k}r_i \mid x_k = x\right\} \quad (6)$$

Такая стратегия называется оптимальной, а соответствующее оптимальное значение задается как

$$V_k^*(x) = \min_\pi V_k^\pi(x) = \min_\pi E_\pi\left\{\sum_{i=k}^{k+T}\gamma^{i-k}r_i \mid x_k = x\right\} \quad (7)$$

1.3.2 Обратная рекурсия для значения

Используя тождество Чепмена-Колмогорова и свойство Маркова, значение стратегии $\pi(x,u)$ можно записать как

$$V_k^\pi(x) = E_\pi\{J_{k,T}|x_k = x\} = E_\pi\left\{\sum_{i=k}^{k+T}\gamma^{i-k}r_i \mid x_k = x\right\}$$

$$V_k^\pi(x) = E_\pi\left\{r_k + \gamma\sum_{i=k+1}^{k+T}\gamma^{i-(k+1)}r_i \mid x_k = x\right\},$$

$$V_k^\pi(x) = \sum_u \pi(x,u)\sum_{x\prime} P_{xx\prime}^u\left[R_{xx\prime}^u + \gamma E_\pi\left\{\sum_{i=k+1}^{k+T}\gamma^{i-(k+1)}r_i \mid x_{k+1} = x''\right\}\right] \quad (8)$$

Поэтому функция ценности для стратегии $\pi(x,u)$ удовлетворяет:

$$V_k^\pi(x) = \sum_u \pi(x,u)\sum_{x\prime} P_{xx\prime}^u\left[R_{xx\prime}^u + \gamma V_{k+1}^\pi(x')\right] \quad (9)$$

Это уравнение обеспечивает обратную рекурсию для значения в момент времени k в терминах значения в момент времени $k+1$.

1.3.3 Обзор адаптивного динамического программирования

Оптимальную стоимость можно записать как

$$V_k^*(x) = \min_\pi V_k^\pi(x) = \min_\pi \sum_u \pi(x,u)\sum_{x\prime} P_{xx\prime}^u\left[R_{xx\prime}^u + \gamma V_{k+1}^\pi(x')\right] \quad (10)$$

Принцип оптимальности Беллмана утверждает, что "Оптимальная стратегия обладает свойством того, что независимо от предыдущих действий управления оставшиеся действия образуют оптимальную стратегию относительно состояния, возникающего в результате этих предыдущих действий". Поэтому мы можем записать (10) в следующем образом

$$V_k^*(x) = \min_{\pi} \sum_{u} \pi(x,u) \sum_{x'} P_{xx'}^{u} \left[R_{xx'}^{u} + \gamma V_{k+1}^{\pi}(x') \right] \quad (11)$$

Предположим, что произвольное управление u применяется в момент времени k, а оптимальная стратегия применяется с момента $k+1$ и далее. Тогда принцип оптимальности Беллмана указывает на то, что оптимальное управление в момент времени k определяется следующим образом:

$$\pi^*(x_k = x, u) = \operatorname*{argmin}_{\pi} \sum_{u} \pi(x,u) \sum_{x'} P_{xx'}^{u} \left[R_{xx'}^{u} + \gamma V_{k+1}^{\pi}(x') \right] \quad (12)$$

В предположении, что цепь Маркова, соответствующая каждой стратегии, с вероятностями перехода, является эргодической, каждая МППР имеет стационарную детерминированную оптимальную стратегию. Тогда мы можем эквивалентно минимизировать условное ожидание по всем действиям u в состоянии x. Поэтому:

$$V_k^*(x) = \min_{u} \sum_{x'} P_{xx'}^{u} \left[R_{xx'}^{u} + \gamma V_{k+1}^{*}(x') \right] \quad (13)$$

$$u_k^* = \operatorname*{argmin}_{u} \sum_{x'} P_{xx'}^{u} \left[R_{xx'}^{u} + \gamma V_{k+1}^{*}(x') \right] \quad (14)$$

Обратная рекурсия (11), (13) составляет основу динамического программирования (ДП), которое дает автономные методы для работы в обратном направлении во времени для определения оптимальной стратегии. ДП - это автономная процедура поиска оптимального значения и оптимальных стратегий, которая требует знания полной динамики системы в виде вероятностей перехода $P_{x,x'}^{u} = \Pr\{x' | x, u\}$ и ожидаемых затрат $R_{xx'}^{u} = E\{r_k | x_k = x, u_k = u, x_{k+1} = x'\}$.

1.3.4 Уравнение Беллмана и уравнение оптимальности Беллмана

Обучение с подкреплением включает в себя определение решений для управления в реальном времени и с упреждением во времени. Ключом к этой проблеме является уравнение Беллмана, которое будет представлено ниже.

Динамическое программирование — это метод поиска оптимального значения и стратегия в обратном времени. В отличие от этого, в обучении с подкреплением направлено на поиск оптимальной стратегии на основе каузального опыта путем последовательного принятия решений, направленных на улучшение управляющих действий на основе наблюдаемых результатов использования текущей стратегии. Для этого необходимо разработать методы нахождения оптимальных значений и оптимальных стратегий, которые могут быть применены в прямом времени. Уравнение Беллмана является ключом к этому проблему.

Чтобы получить методы, направленные в прямом времени, для поиска оптимальных значений и оптимальных стратегий, устанавливается временной горизонт T равным бесконечности и определяется стоимость бесконечного горизонта

$$J_k = \sum_{i=0}^{\infty} \gamma^i r_{k+1} = \sum_{i=k}^{\infty} \gamma^{i-k} r_i \quad (15)$$

Бесконечная функция ценности связанного горизонта для стратегии $\pi(x,u)$ равна

$$V^{\pi}(x) = E_{\pi}\{J_k | x_k = x\} = E_{\pi}\left\{\sum_{i=k}^{\infty} \gamma^{i-k} r_i | x_k = x\right\} \quad (16)$$

Используя (4) с $T = \infty$, можно увидеть, что функция ценности стратегии $\pi(x,u)$ удовлетворяет уравнению Беллмана

$$V^{\pi}(x) = \sum_{u} \pi(x,u) \sum_{x\prime} P^{u}_{xx\prime} \left[R^{u}_{xx\prime} + \gamma V^{\pi}_{k+1}(x')\right] \quad (17)$$

Уравнение Беллмана (17) можно интерпретировать как уравнение согласованности, которому должна удовлетворять функция стоимости на каждом временном этапе. Оно выражает связь между текущей ценностью нахождения в состоянии x и ценностью нахождения в следующем состоянии x' при условии, что используется стратегия $\pi(x,u)$.

Если МППР конечен и имеет N состояний, то уравнение Беллмана (17) представляет собой систему N одновременных линейных уравнений для значения $V^{\pi}(x)$ пребывания в каждом состоянии x при текущей стратегии $\pi(x,u)$.

Оптимальное значение бесконечного горизонта удовлетворяет

$$V^{*}(x) = \min_{\pi} V^{\pi}(x) = \min_{\pi} \sum_{u} \pi(x,u) \sum_{x\prime} P^{u}_{xx\prime} \left[R^{u}_{xx\prime} + \gamma V^{\pi}(x')\right] \quad (18)$$

Тогда принцип оптимальности Беллмана приводит к уравнению оптимальности Беллмана

$$V^{*}(x) = \min_{\pi} V^{\pi}(x) = \min_{\pi} \sum_{u} \pi(x,u) \sum_{x\prime} P^{u}_{xx\prime} \left[R^{u}_{xx\prime} + \gamma V^{*}(x')\right] \quad (19)$$

Эквивалентно, при условии эргодичности цепей Маркова, соответствующих каждой стратегии, уравнение оптимальности Беллмана можно записать как

$$V^{*}(x) = \min_{u} \sum_{x\prime} P^{u}_{xx\prime} \left[R^{u}_{xx\prime} + \gamma V^{*}(x')\right] \quad (20)$$

Это уравнение известно как уравнение Гамильтона – Якоби – Беллмана (HJB) в системах управления. Если МППР конечен и имеет N состояний, то уравнение оптимальности Беллмана представляет собой систему N нелинейных уравнений для оптимального значения $V^{*}(x)$ пребывания в каждом состоянии. Оптимальное управление определяется выражением

$$u^* = \underset{u}{arg\min} \sum_{x'} P^u_{xx'} \left[R^u_{xx'} + \gamma V^*(x') \right] \quad (21)$$

1.4 Итерация по стратегии и итерация по критерию

Эти два метода являются методами автономной оптимизации в обучении с подкреплением. Чтобы иметь возможность реализовать методы онлайн-оптимизации, необходимо объединить их с методами приближенного динамического программирования. Это то, что нам нужно для построения алгоритма оптимального адаптивного регулятор в реальном времени.

1.4.1 Итерация по стратегии алгоритм

Чтобы понять алгоритм итерации по стратегии, нам сначала нужно обобщить функцию значения.

$$V^\pi(x) = \sum_u \pi(x,u) \sum_{x'} P^u_{xx'} \left[R^u_{xx'} + \gamma V^\pi(x') \right] \quad (22)$$

Это равенство называется равенством Беллмана и упрощает вычисление функции ценности за счет использования динамического программирования.

Алгоритм итерации по стратегии представлен следующим образом:

Выберите начальную стратегию $\pi_0(x,u)$. Начиная с j = 0, повторите по j до сходимости

Оценка стратегии (обновление ценности)

$$V_j(x) = \sum_u \pi_j(x,u) \sum_{x'} P^u_{xx'} \left[R^u_{xx'} + \gamma V^\pi_j(x') \right], \forall x \in X \quad (23)$$

Улучшение стратегии (обновление стратегии)

$$\pi_{j+1}(x,u) = \underset{\pi}{arg\min} \sum_{x'} P^u_{xx'} \left[R^u_{xx'} + \gamma V^\pi_j(x') \right], \forall x \in X \quad (24)$$

На каждой итерации j, алгоритм итерации по стратегии определяет решение уравнения Беллмана для определения значения функции ценности $V_j(x)$, соответствующей текущей стратегии $\pi_j(x,u)$. Тогда улучшение стратегии осуществляется благодаря уравнению (24). Эти шаги повторяются до тех пор, пока и функция ценности, и стратегия придут к своим оптимальным значениям. Чтобы алгоритм работал корректно, нам нужно обратить внимание на начальную инициализацию стратегию $\pi_0(x,u)$ и начальное начальное значение $V_1 \leq V_0$.

1.4.2 Итерации по критерию

Алгоритм итерация по критерию основан на чрезвычайно простой, но очень эффективной идее. Рассматривая алгоритм, итерация по стратегии, мы видим, что он одновременно поддерживает и стратегию, и функции ценности, что делает вычисления громоздкими и трудоемкими. Алгоритм итерация по

критерию начинает с того, что сначала пытается оптимизировать функцию ценности, затем стратегия, соответствующая оптимальной функции ценности, конечно же, также будет оптимальной стратегией.

Алгоритм итерации по стратегии представлен следующим образом:

Выберите начальную стратегию $\pi_0(x,u)$. Начиная с j = 0, итерации по j до сходимости

Обновление ценности

$$V_{j+1}(x) = \sum_u \pi_j(x,u) \sum_{x'} P^u_{xx'} \left[R^u_{xx'} + \gamma V_j{}^{\pi}(x')\right], \forall x \in S_j \in X \quad (25)$$

Улучшение стратегии

$$\pi_{j+1}(x,u) = \underset{\pi}{arg\min} \sum_{x'} P^u_{xx'} \left[R^u_{xx'} + \gamma V^{\pi}_{j+1}(x')\right], \forall x \in S_j \in X \quad (26)$$

В каждом цикле j алгоритм итерация по критерию оценивает значение функции $V_{j+1}(x)$, соответствующее стратегии управления $\pi_j(x,u)$, на основе значения функции ценности на предыдущем этапе . Обновление стратегии производится по формуле (26). Повторяйте шаги алгоритма до сходимости.

1.4.3 Сравнение итерации по стратегии и итерации по критерию

В то время как значение $V_j(x')$ в алгоритме итерации по стратегии представляет собой фактическое значение текущей стратегии π_j, в алгоритме итерации по критерию его можно рассматривать как оценку стоимости перехода из состояния x в будущее состояние x'.

В алгоритме итерации по критерию в каждом цикле мы обновляем значение функции ценности $V(x)$, и когда $V(x)$ достигает оптимального значения, стратегия автоматически становится оптимальной. В отличие от итерации по критерию, итерация по стратегии пытается определить эффективность текущей стратегии с помощью оценки стратегии, а затем обновляет новую стратегию, после чего стратегия и значение функции ценности постоянно обновляются до тех пор, пока не будет достигнуто оптимальное значение.

Итерации по стратегии обычно сходятся быстрее, чем итерации по критерию. Поэтому во многих случаях он предпочтительнее. При определенных условиях гарантируется сходимость алгоритма итерации по стратегии за ограниченное число шагов. Однако количество шагов, необходимых для сходимости алгоритм итерации по критерию, не обязательно ограничено.

2 Регулятора для линейных систем

Рассмотрим непрерывную линейную и инвариантную систему во времени систему с моделью в пространстве состояний.

$$\dot{x} = Ax(t) + Bu(t), \quad (2.1)$$

Где $B \in \mathbb{R}^{n\times m}$, $A \in \mathbb{R}^{n\times n}$, $x(t) \in \mathbb{R}^n$ - состояния системы, $u(t) \in \mathbb{R}^n$ – входной сигнал управления, пара матрицы (A, B) – стабилизируемость.

Задачи включается себе в том, что нужно разрабатывать стратегию управления с обратной связью $u = -Kx$ так, чтобы

$$\lim_{t\to\infty} |x(t)| = 0 \quad (2.2)$$

2.1 Оптимальный регулятор

Одним из наиболее известных типов оптимальных регуляторов является линейный квадратичный регулятор (LQR). Для системы (2.1), задача LQR с обратной связью по состоянию $u = -Kx$ состоит в поиске оптимального регулятора u^*, который минимизирует функцию стоимости $J(x,u)$ на бесконечном горизонте, связанную с системой:

$$J(x,u) = \int_0^\infty \left(x^T(t)Qx(t) + u^T(t)Ru(t)\right)dt \quad (2.3)$$

Здесь можно показать, что оптимальное управление является линейной функцией от x

$$u = -Kx = R^{-1}B^T P\, x \quad (2.4)$$

где P — определенное положительное решение алгебраического уравнения Риккати

$$A^T P + PA - PBR^{-1}B^T P + Q = 0 \quad (2.5)$$

Если $Q \geq 0$, $R > 0$, (A, B) стабилизируемость и (Q, A) обнаруживаемость, то алгебраическое уравнение Риккати имеет единственное симметричное положительное решение и соответствующий ему регулятор делает систему (3.1) асимптотически устойчивой.

2.2 Адаптивный регулятор

В случае адаптивного регулятора для системы (2.1) матрица A считается неизвестной. Полагая $A = A_0 + b\theta^T$, уравнение системы (2.1) принимает вид

$$\dot{x} = A_0 x + b(\theta^T x + u) \quad (2.6)$$

Где θ - неизвестный параметр и A_0 – эталонная матрица Гурвица.

Сигнал управления имеет следующую формулу

$$u = -\hat{\theta}^T x \quad (2.7)$$

Где $\hat{\theta}$ - оценка вектора θ, динамическая модель системы будет иметь вид

$$\dot{x} = A_0 x + b\theta^T x - b\hat{\theta}^T x = A_0 x + b\tilde{\theta}^T x \quad (2.8)$$

Где $\tilde{\theta} = \theta - \hat{\theta}$ - вектор параметрических ошибок.

Алгоритм адаптивной стабилизации объекта (2.1) имеет вид

$$\dot{\hat{\theta}} = \gamma x x^T P b = \gamma x b^T P x \quad (2.9)$$

Где симметрическая положительно определенная матрица P является решением уравнения Ляпунова

$$A_0^T P + P A_0 + Q = 0, Q = Q^T > 0 \quad (2.10)$$

2.3 Оптимальный адаптивный регулятор

Для системы (2.1) задача включается в том, что, нужно найти оптимальный регулятор u^*, который минимизирует квадратичную функцию стоимости бесконечного горизонта, связанную с системой.

$$V(x(t_0), t_0) = \int_{t_0}^{\infty} \left(x^T(\tau) Q x(\tau) + u^T(\tau) R u(\tau)\right) d\tau \quad (2.11)$$

Где для системы (2.1) матрица A неизвестна, $Q \geq 0, R \geq 0$ и $\left(\sqrt{Q}, A\right)$ – обнаруживаемость.

Тогда оптимальный регулятор записывается следующим образом:

$$u^*(t) = arg\min_{u(t)} V\left(t_0, x(t_0), u(t)\right), t_0 \leq t \leq \infty \quad (2.12)$$

Решение этой задачи оптимального управления, определенное в соответствии с принципом оптимизации Беллмана, дается с помощью обратной связи по состоянию $u(t) = -Kx = -R^{-1}B^T P x$.

Пусть K - стабилизирующий коэффициент обратной связи по состоянию для (2.1) такой, что $\dot{x} = (A - BK)$ представляет собой устойчивую замкнутую систему. Тогда квадратичная бесконечная стоимость или ценность на бесконечном горизонте определяется выражением

$$V\big(x(t)\big) = \int_0^\infty (x^T(\tau)Qx(\tau) + u^T(\tau)Ru)\, d(\tau)\,, u = -Kx$$

$$V(x(t)) = \int_0^\infty x^T(\tau)(Q + K^T RK)x^T(\tau)d(\tau) = x^T(\tau)Px(t) \quad (2.13)$$

Где P- симметричная положительно определенная матрица, является решением матричного уравнения Ляпунова

$$(A - BK)^T P + P(A - BK) = -(K^T RK + Q) \quad (2.14)$$

Тогда $V(x(t))$ служит функцией Ляпунова для (2.1) с коэффициентом усиления регулятора K. Функция ценности (2.8) можно записать в следующем виде

$$V\big(x(t)\big) = \int_t^{t+T} x^T(\tau)(Q + k^T RK)x(\tau)d(\tau) + V(x(t+T)) \quad (2.15)$$

Это уравнение Беллмана для задачи LQR. Используя уравнение Беллмана IRL, можно обойти отмеченные в разделе (1.3.2) проблемы применения обучения с подкреплением к системам с непрерывным временем.

Интегральное армирование

$$\rho(x(t), t, T) = \int_t^{t+T} x^T(\tau)(Q + K^T RK)x(\tau)d(\tau) \quad (2.16)$$

Формула (2.15) называется уравнением Беллмана обучения с интегральным подкреплением (IRL).

Далее показано, как найти оптимальный регулятор, применяя интегральное обучение с подкреплением к линейным системам (итерации по стратегии) (2.1).

$$W_k^T[\phi(x(t)) - \phi(x(t+T))] = \int_t^{t+T} x^T(\tau)(Q + K_k^T R K_k)\,\mathrm{d}\tau.$$

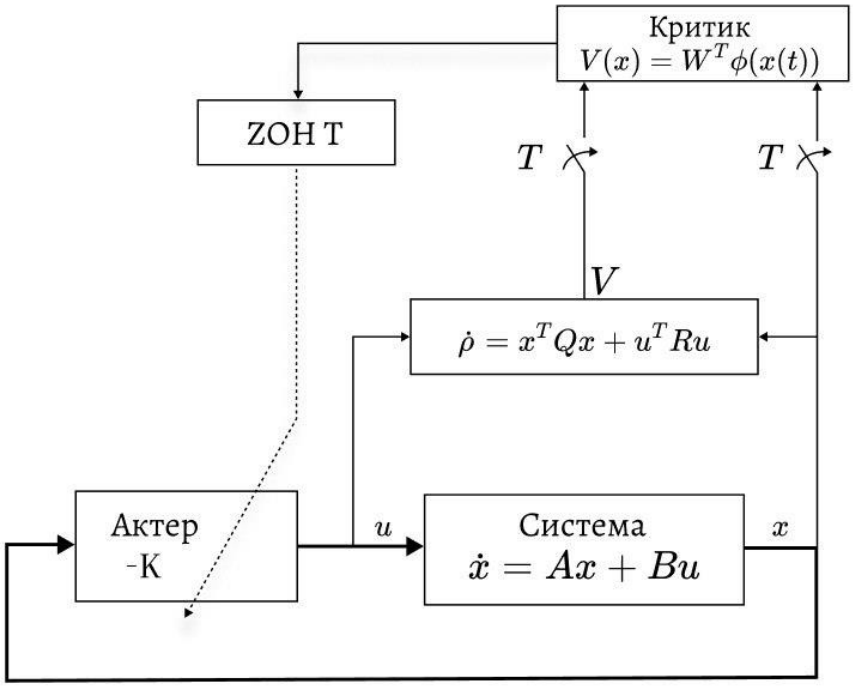


Рисунок 2.1 Структура системы с оптимальным адаптивным регулятором для линейных систем.

На основе интегрального обучения с подкреплением (2.15) можно написать следующий алгоритм RL:

$$V(x(t)) = \int_t^{t+T} x_\tau^T\big(Q + K_i^T R K_i\big)x_\tau d(\tau) + x_{t+T}^T P x_{t+T} \quad (2.16)$$

Параметры функции ценности $V_i\big(x(t)\big) = x^T(t)Px(t)$ на итерации i IRL являются элементами симметричной ядерной матрицы P_i. Они должны быть найдены на каждой итерации i путем измерения данных $(x(t), x(t+T)$, u(t)) в моменты времени t вдоль траекторий системы. Для вычисления этих параметров термин $x^T(t)P_i x(t)$ записывается как

$$x^T(t)Px(t) = W^T\phi(x(t)) \quad (2.17)$$

Где $\phi\big(x(t)\big)$ – вектор функции активации. $\phi\big(x(t)\big)$ обозначает базисный вектор квадратичного полинома произведения Кронекера

$$\phi\big(x(t)\big) = [x_1^2 \quad x_1x_2 \quad \ldots \quad x_1x_n \quad x_2^2 \quad x_2x_3 \quad \ldots \quad x_n^2]^T \quad (2.18)$$

Применяя формулу аппроксимации значения функции (2.17), можно записать (2.15) следующим образом

$$W_{j+1}(\phi(x(t)) - \phi(x(t+T))) = \int_t^{t+T} x_\tau^T (Q + K_i^T R K_i) x_\tau \, d\tau = d(\phi(x(t), K_i), \quad (2.17)$$

Применяя к этой проблеме структуру «Актер-Критик», Критик использует уравнение (2.17) для вычисления матрицы P_i. Когда Критик сходится к решению P_i , Актер рассчитает новое значение управления для управления системой на основе P_i по формуле:

$$u_{i+1} = -R^{-1} B^T P_i x = K_{i+1} x \quad (2.16)$$

С помощью использования алгоритма итерации по стратегии мы можем найти оптимальный регулятор без необходимости знания матрицы для линейной системы.

3. Задача оптимального управления для нелинейных систем

К непрерывным системам обучение с подкреплением применить сложнее, чем к дискретным, и оно дает меньше результатов. Один из методов, описанных в [13], [14], называется обучением с интегральным подкреплением. Этот метод может быть использован для разработки методов оптимального управления непрерывными системами, не требуя полного понимания динамики системы. Этот метод называется методом оптимального адаптивного управления. Ниже изучается и рассматривается оптимальный адаптивный регулятор для нелинейных систем.

3.1 Оптимальное управление и уравнение Гамильтона Якоби – Беллмана

Рассмотрим нелинейную динамическую систему с непрерывным временем

$$\dot{x} = f(x) + g(x)u \quad (3.1)$$

Где $x(t) \in R^n$ – состояние системы, входное управление $u(t) \in R^m$, точка равновесии $x = 0$, $f(0) = 0$ и $f(x) + g(x)u$ Липшиц (Lipschitz) на съемочной площадке $\Omega \in R^n$. Предположим, что система стабилизируема на Ω, т. е. существует непрерывная функция управления $u(t)$ такая, что замкнутая система асимптотически устойчива на Ω.

Определите меру эффективности или функцию ценности, которая имеет значение, связанное со стратегией управления с обратной связью $u = \mu(x)$, заданное как

$$V^\mu(x(t)) = \int_t^\infty r(x(\tau), u(\tau)) d\tau \quad (3.2)$$

Где $r(x,u) = Q(x) + u^T Ru$, положительно определенный $Q(x)$, т. е. $Q(x) > 0$ для всех x и $x = 0 \Rightarrow Q(x) = 0$ и положительно определенная матрица $R = R^T \in R^{m\times m}$.

Для LQR с непрерывным временем приведенные выше выражения имеют вид как показан в пункте 2.

$$\dot{x} = Ax + Bu \quad (3.3)$$

$$V^{\mu}\big(x(t)\big) = \frac{1}{2}\int_t^{\infty}(x^T Qx + u^T Ru)d\tau. \quad (3.4)$$

Определение 1. Стратегия управления $\mu(x)$ определяется как допустимая относительно (3.2) на Ω, обозначаемая $u \in \Psi(\Omega)$, если $\mu(x)$ непрерывна на Ω, $u(0) = 0$, $\mu(x)$ стабилизирует (3.1) на Ω и $V(x_0)$ конечно $\forall x_0 \in \Omega$.

Для любой допустимой стратегии управления $\mu \in \Psi(\Omega)$, если соответствующая функция стоимости $V(x_0)$ равна C^1, то бесконечно малым эквивалентом (3.2) является уравнение Беллмана

$$0 = r(x,\mu(x)) + (\nabla V^{\mu})^T(f(x) + g(x)\mu(x)), V^{\mu}(0) = 0 \quad (3.5)$$

Где ∇V^{μ}, взятый здесь как вектор-столбец, обозначает вектор градиента функции ценности V^{μ} по x. Учитывая допустимый регулятор $\mu(x) \in \Psi(\Omega)$, уравнение Беллмана можно решить через соответствующую функцию ценности $V^{\mu}(x)$. В линейном случае оно становится уравнением Ляпунова. Учитывая, что $\mu(x)$ является допустимой стратегии управления, если $V^{\mu}(x)$ удовлетворяет (3.5) с $r(x,\mu(x)) \geq 0$, то можно доказать, что $V^{\mu}(x)$ является функцией Ляпунова для системы (1) со стратегией управления $\mu(x)$.

Теперь можно сформулировать задачу оптимального управления: для непрерывной системы (3.1), множества $u \in \Psi(\Omega)$ допустимых стратегий управления и функции ценности на бесконечном горизонте (3.2) найти допустимую стратегию управления такую, что индекс ценности (3.2), связанная с системой (3.1), минимизируется.

Определите гамильтониан

$$H(x,\mu(x),\nabla V^{\mu}) = r\big(x,\mu(x)\big) + (\nabla V^{\mu})^T(f(x) + g(x)u) \quad (3.6)$$

Тогда оптимальная функция ценности $V^*(x)$ удовлетворяет уравнению HJB

$$0 = \min_{\mu} H(x,\mu(x),\nabla V^*) \quad (3.7)$$

и удовлетворяющие оптимальному управлению

$$\mu^* = arg\min_{\mu} H(x, \mu(x), \nabla V^*) \quad (3.8)$$

$$\mu^* = -\frac{1}{2} R^{-1} g^T(x) \nabla \mathrm{V}^*(x) \quad (3.9)$$

Подставляя эту стратегию оптимального управления в гамильтониан, мы получаем формулировку уравнения HJB в терминах $V^*(x)$:

$$0 = Q(x) + \nabla V^{*T}(x) f(x) - \frac{1}{4} \nabla V^{*T}(x) g(x) R^{-1} g^T(x) \nabla V^*(x), \quad (3.10)$$

$$V^*(0) = 0$$

Это достаточное условие функции оптимального значения [10]. В случае линейной системы с квадратичной функцией стоимости это уравнение HJB становится уравнением Риккати.

Чтобы найти оптимальное управляющее решение задачи, нужно всего лишь решить HJB (3.10) для функции ценности, а затем подставить решение в (3.9) для получения оптимального управления. Однако решение уравнения HJB, как правило, затруднено, поскольку оно представляет собой нелинейное дифференциальное уравнение второго порядка, следующее за градиентом функции стоимости, и его решение также требует полного знания системы системной динамики (т. е. динамики системы, описываемой уравнением функции $f(x)$ и $g(x)$ должны быть известны). Более того, глобально гладких решений может не существовать.

3.2 Адаптивный алгоритм оптимального управления на основе итераций по стратегии

Итерация по стратегии (PI) — это итерационный метод обучения с подкреплением [4, 12] для решения (3.10) на основе простых уравнений. PI состоит из двух этапов: улучшение стратегии и оценка стратегии. Говорят, что итерация по стратегии имеет структуру обучения с подкреплением «Актер-Критик», где улучшение стратегии выполняется актером, Решение уравнения Беллмана выполняется критиком для получения функции ценности, этот процесс называется оценкой стратегии. Алгоритм итерация по стратегии задается следующим образом.

Пусть $u(x)$ — допустимая стратегия для (3.1), такая что замкнутая система асимптотически устойчива на Ω. Тогда стоимость бесконечного

горизонта для любого $x(t) \in \Omega$ определяется формулой (3.2), а $V(x(t))$ служит функцией Ляпунова для (3.1). Функцию стоимости (3.2) можно записать в виде

$$V^{\mu}\big(x(t)\big) = \int_t^{t+T} r(x(\tau), \mu\big(x(\tau)\big)d\tau + V^{\mu}\big(x(t+T)\big) \quad (3.11)$$

На основании (3.11) и (3.7), учитывая начальную допустимую стратегию управления $u^{(0)}(x)$, можно вывести следующую схему итерации по стратегии

1. Решить для $V^{\mu^{(i)}}(x)$ с использованием

$$V^{\mu^{(i)}}\big(x(t)\big) = \int_t^{t+T} r\left(x(\tau), u^{(i)}\big(x(\tau)\big)\right) d\tau + V^{\mu}(x(t+T) \quad (3.12)$$

2. Обновить стратегию управления с помощью

$$\mu^{(i+1)}(x) = \underset{\mu}{\operatorname{argmin}}\{H(x, \mu, V_x^{u^{(i)}}\} \quad (3.13)$$

Где

$$\mu^{(i+1)}(x) = -\frac{1}{2}R^{-1}g^T(x)\nabla V_x^{u^{(i)}} \quad (3.14)$$

Уравнения (3.13) и (3.14) формулируют новый алгоритм итерации по стратегии для поиска оптимального управления без использования каких-либо знаний о внутренней динамике системы $f(x)$.

3.3 Реализация алгоритма итерации по стратегии в реальном времени

Применяя аппроксимационные возможности параллельной нейронной сети для уменьшения объема вычислений по сравнению со структурой «Актор-критик», этот алгоритм использует нейронную сеть для аппроксимации оптимальной функции стоимости $V(x)$ с помощью $x \in \Omega$ следующим образом:

$$\hat{V}(x) = \widehat{W}^T\phi(x) \quad (3.15)$$

Где $\widehat{W}^T \in R^N$ – неизвестная веса, N - количество нейронов, $\phi(x)$ – вектор функции активации.

Используя нейронную сеть для аппроксимации оптимальной функции ценности, подставьте формулу (3.15) в формулу (3.12), чтобы получить:

$$\widehat{W}^T\phi(x(t)) = \int_t^{t+T} r\left(x(\tau), u^{(i)}\big(x(\tau)\big)\right) d\tau + \widehat{W}^T\phi(x(t+T)) \quad (3.16)$$

Появляется ошибка $e(t)$, которая является ошибкой аппроксимации функции Беллмана.

$$e(x(t),T) = \widehat{W}^T\left(\phi\big(x(t+T)\big) - \phi\big(x(t)\big)\right) = -\int_t^{t+T} r\left(x(\tau), u^{(i)}\big(x(\tau)\big)\right) d\tau \quad (3.17)$$

Обозначим что

$$h(t) = \phi\big(x(t+T)\big) - \phi\big(x(t)\big) \quad (3.18)$$

$$y(t) = \int_t^{t+T} r\left(x(\tau), u^{(i)}\big(x(\tau)\big)\right) d\tau \quad (3.19)$$

Уравнение (3.16) можно написать в следующий вид

$$e(t) = \widehat{W}^T h(t) + y(t) \quad (3.20)$$

Где $H = [h(t_1), \quad \dots \quad , h(t_N)]$, $Y = [y(t_1), \quad \dots \quad , y(t_N)]^T$ (3.21)

Уравнение (3.20) является линейной функцией по параметру $\widehat{W}$. Следовательно, мы можем применить алгоритм наименьших квадратов ошибки, чтобы найти оптимальное значение для $\widehat{W}$.

Данные системы собирается из N различных выборок за период времени T, поэтому мы вычисляем (3.19) в n точках от $t_1 \rightarrow t_N$, чтобы получить следующие функции:

$$H = [h(t_1), \quad \dots \quad , h(t_N)] \quad (3.22)$$

$$Y = [y(t_1), \quad \dots \quad , y(t_N)] \quad (3.23)$$

Чтобы определить веса W нейронной сети, аппроксимирующей функцию V, что приводит к минимизации следующей целевой функции

$$S = \int_\Omega e(x,T)e(x,T)\, dx \quad (3.24)$$

Произведение $< f, g >= \int_\Omega f g\, dx$ интеграла Лебега можно записать:

$$< \frac{de(x,T)}{d\widehat{W}}, e(x,T) >_\Omega = 0 \quad (3.25)$$

Использовать уравнение (3.20) для уравнения (3.25), получили

$$H\big[H\widehat{W} + Y\big] = 0 \quad (3.26)$$

$$\widehat{W} = -(HH^T)^{-1}HY \quad (3.27)$$

Алгоритм онлайн обучения с интегральным подкреплением использует нейронные сети представлен следующим образом

Шаг 1. $\forall x \in \Omega_x$, инициализировать допустимый закон управления $u(x) \in \psi(\Omega)$. Поместите в систему управляющий сигнал $u^{(0)}$ и соберите необходимую данные системы о состоянии и управляющих сигналах на N различных выборках за период времени T. Назначьте i ←0, инициализируйте ε_w

Шаг 2. Используйте данные, собранную о системе, для расчета H и Y. Определить W из уравнения (3.27)

Шаг 3: обновите закон управления для следующего цикла

$$u^{i+1}(x) = -\frac{1}{2}R^{-1}g(x)^T\nabla V_x\widehat{W}^{(i)} \quad (3.28)$$

Если критерий сходимости удовлетворяется так, что $||\widehat{W}^{(i+1)} - \widehat{W}^{(i)}|| < \varepsilon_w$, алгоритм завершается. Если не удовлетворено, присвойте $i \leftarrow i+1$, подайте сигнал $u^{(i)}$ в систему и соберите необходимую информацию системы о состоянии и управляющих сигналах в N различных выборках за период T, затем вернитесь к шагу 2.

4 Результаты моделирование

В этом разделе для применения и исследования алгоритмов, представленных ранее в разделах 2 и 3, была выбрана система перевернутого маятника и тележки [17][18].

4.1 Регулятор для линейных систем

Рассмотрим непрерывную инвариантную во времени линейная система

$$\dot{x} = Ax + Bu, \quad (4.1)$$

$$A = \begin{bmatrix} 0 & 1 \\ 10 & -10 \end{bmatrix}, B = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Где $x \in \mathbb{R}^2$ – состояние системы, $u \in \mathbb{R}^1$ – сигнал управления системы

Задача оптимального управления состоит в оптимизации следующей функции

$$J(x,u) = \int_0^\infty \left(x^T(\tau)Qx(\tau) + u^T(\tau)Ru(\tau)\right)d\tau \quad (4.2)$$

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, R = 1$$

В случае оптимального регулятора (LQR), решая уравнение Риккати

$$A^T P + PA - PBR^{-1}B^T P + Q = 0$$

Мы получали матрицу $P = \begin{bmatrix} 2.0739 & 0.211 \\ 0.211 & 0.0672 \end{bmatrix}$, следовательно обратная матрица K вычислена

$$K = R^{-1}B^T P = [2.284 \quad 0.278]$$

Параметры для адаптивного регулятора

$$\gamma = 1, A_0 = A - BK = \begin{bmatrix} -2.284 & 0.7217 \\ 7.7151 & -10.2783 \end{bmatrix}, P_a = P = \begin{bmatrix} 2.0739 & 0.211 \\ 0.211 & 0.0672 \end{bmatrix}$$

Далее представлены результаты моделирования

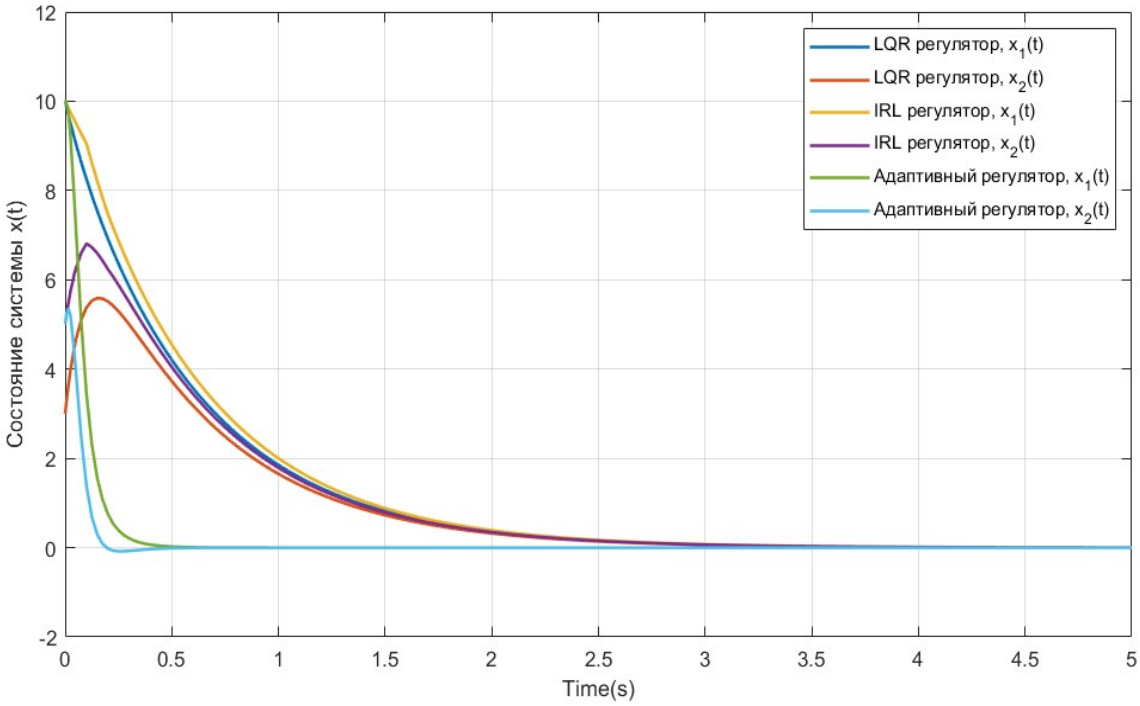


Рисунок 4.1 - Графика вектор состояния $x(t)$ для оптимального регулятора, адаптивный регулятор и оптимального адаптивного регулятора.

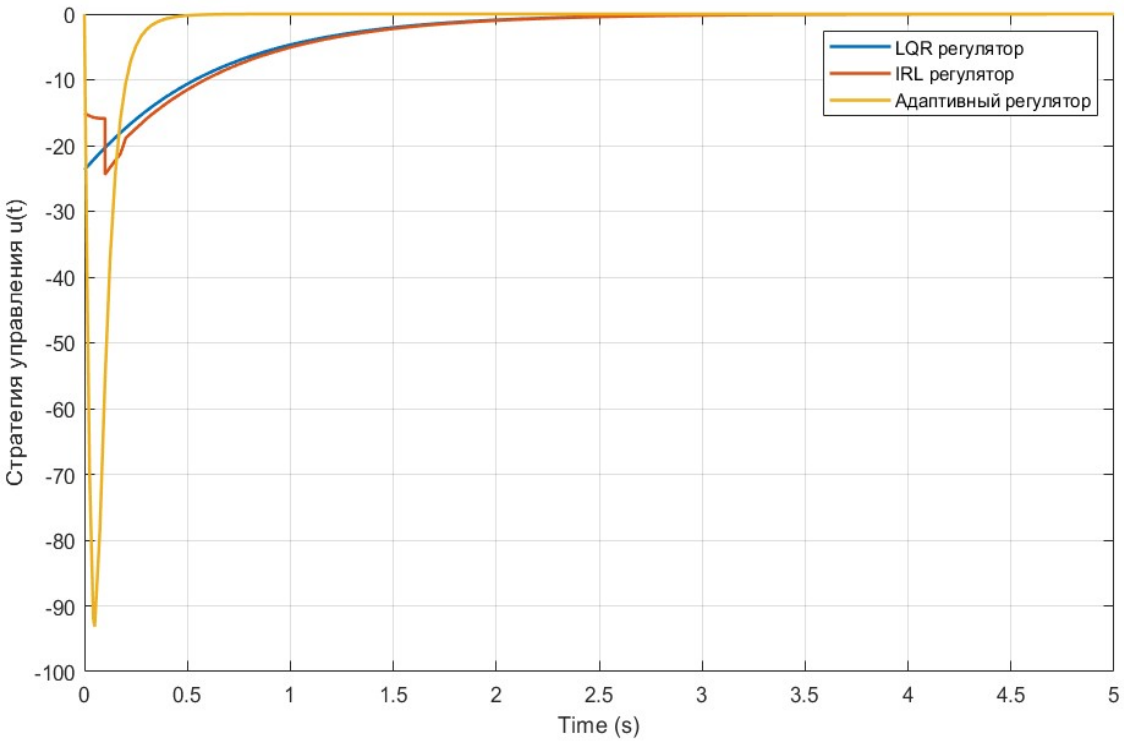


Рисунок 4.2 - Графика сигнал управления для оптимального регулятора, адаптивный регулятор и оптимального адаптивного регулятора (IRL)

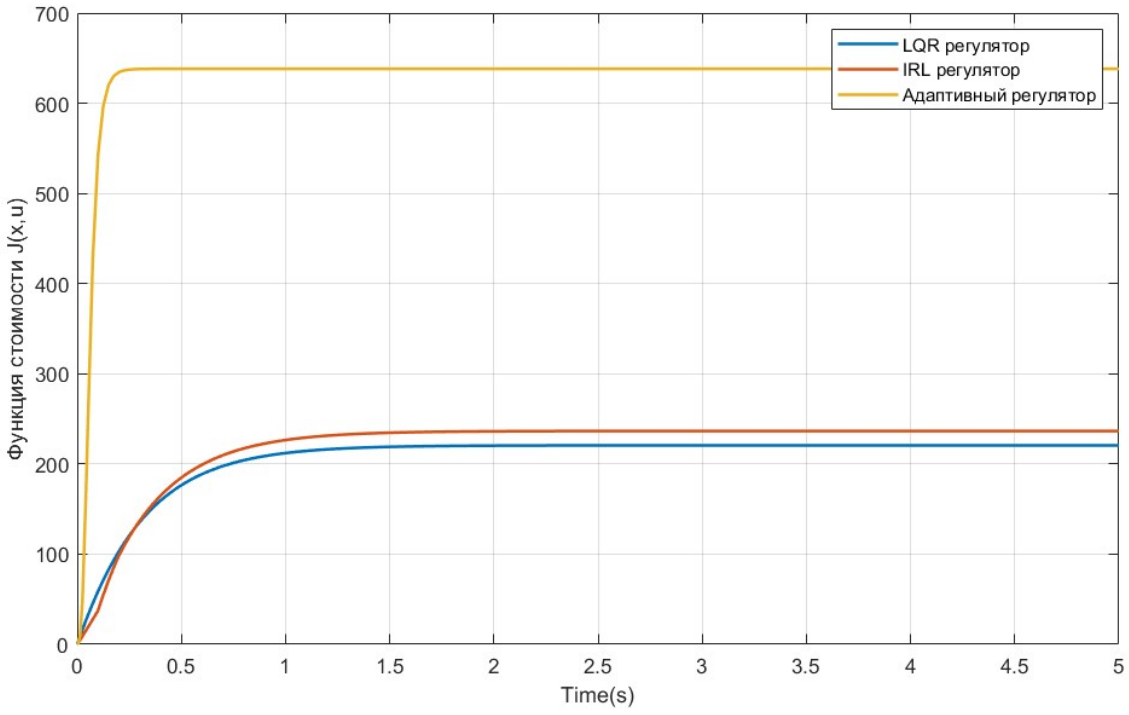


Рисунок 4.3 - Значение функции стоимости для оптимального регулятора, адаптивный регулятор и оптимального адаптивного регулятора (IRL)

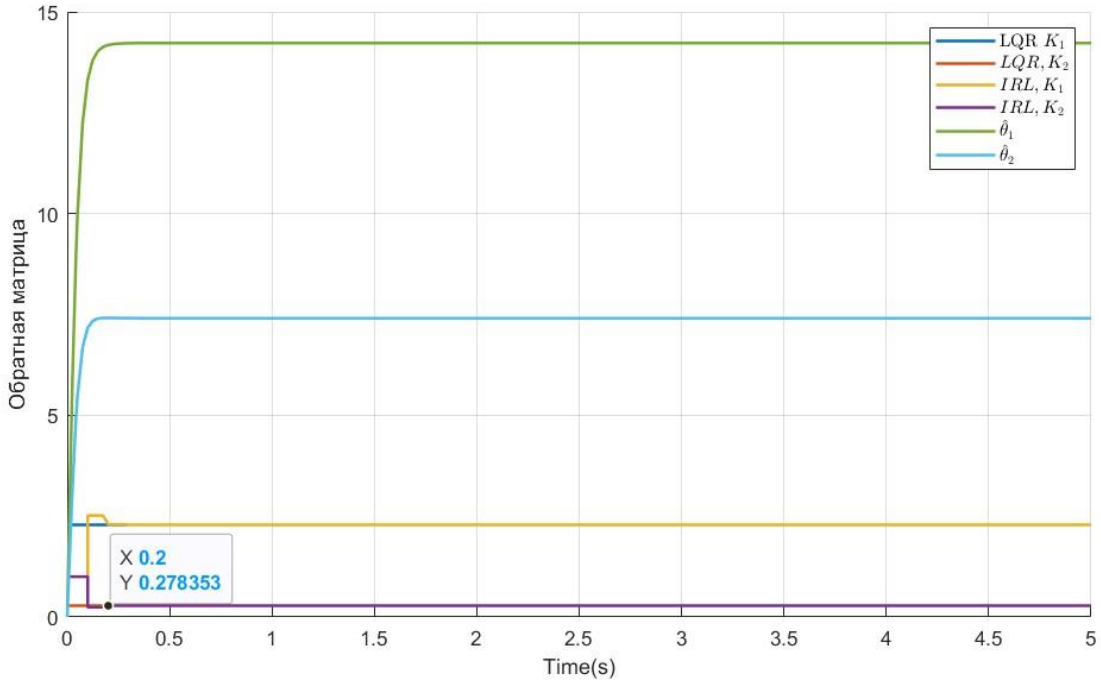


Рисунок 4.4 – Графика матрицы регулятора K(LQR), параметры $\hat{\theta}$ для адаптивного регулятора и сходимости значения обратной матрицы оптимального регулятора (IRL)

Приведенные выше результаты моделирования показывают, что с помощью алгоритма итерации по стратегии обучения с подкреплением оптимальный адаптивный регулятор сошелся к асимптоте оптимального управления через 0.2 секунды. При этом адаптивный регулятор не сходится к оптимальному управлению.

Кроме того, производительность адаптивного регулятора зависит от выбора параметров γ и P_a для системы. Поэтому при изменении значений матрицы Q и R нам приходится заново выбирать параметры оптимального регулятора. Для оптимального адаптивного регулятора нам не нужно настраивать параметры при изменении Q и R.

4.2 Исследование влияния времени выборки T на оптимальный адаптивный регулятор.

При применении алгоритма обучения с интегральным подкреплением к оптимальному адаптивному регулятору нам нужно точно аппроксимировать оптимальную матрицу P, используя данные по траектории системы на интервалах времени T. Графики влияния времени выборки T к производимости оптимального адаптивного регулятора представлены ниже.

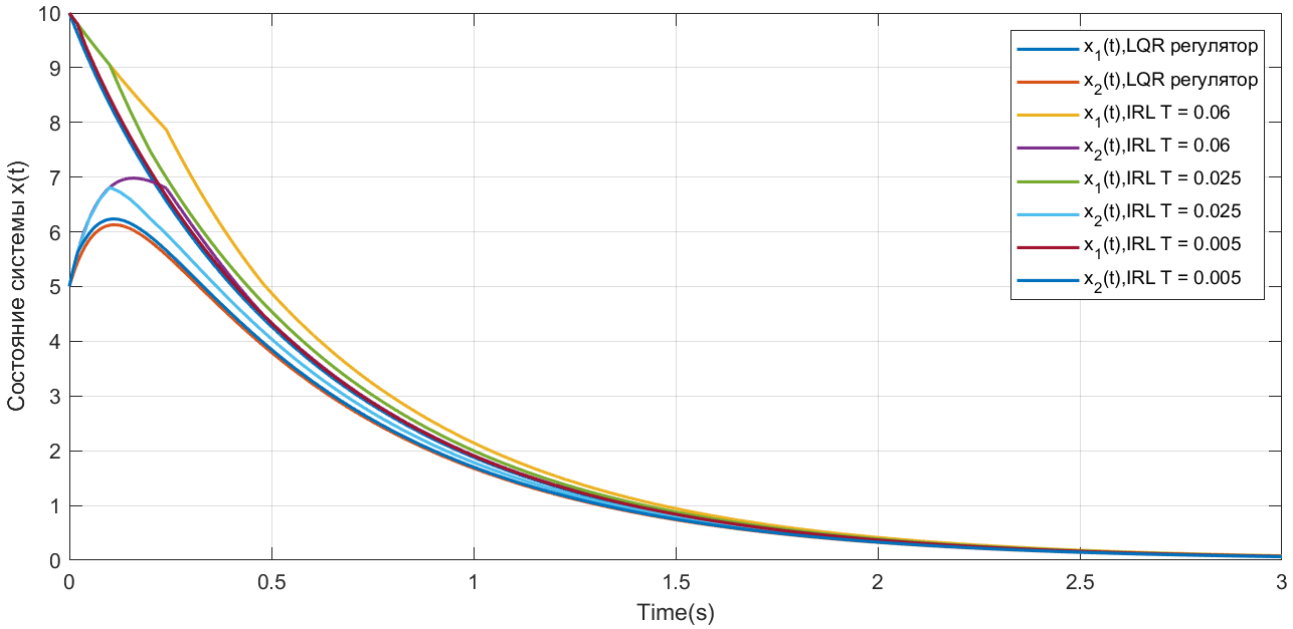


Рисунок 4.5 – Графика состояния системы $x(t)$ оптимального регулятора и оптимального адаптивного регулятора при некоторых значениях T

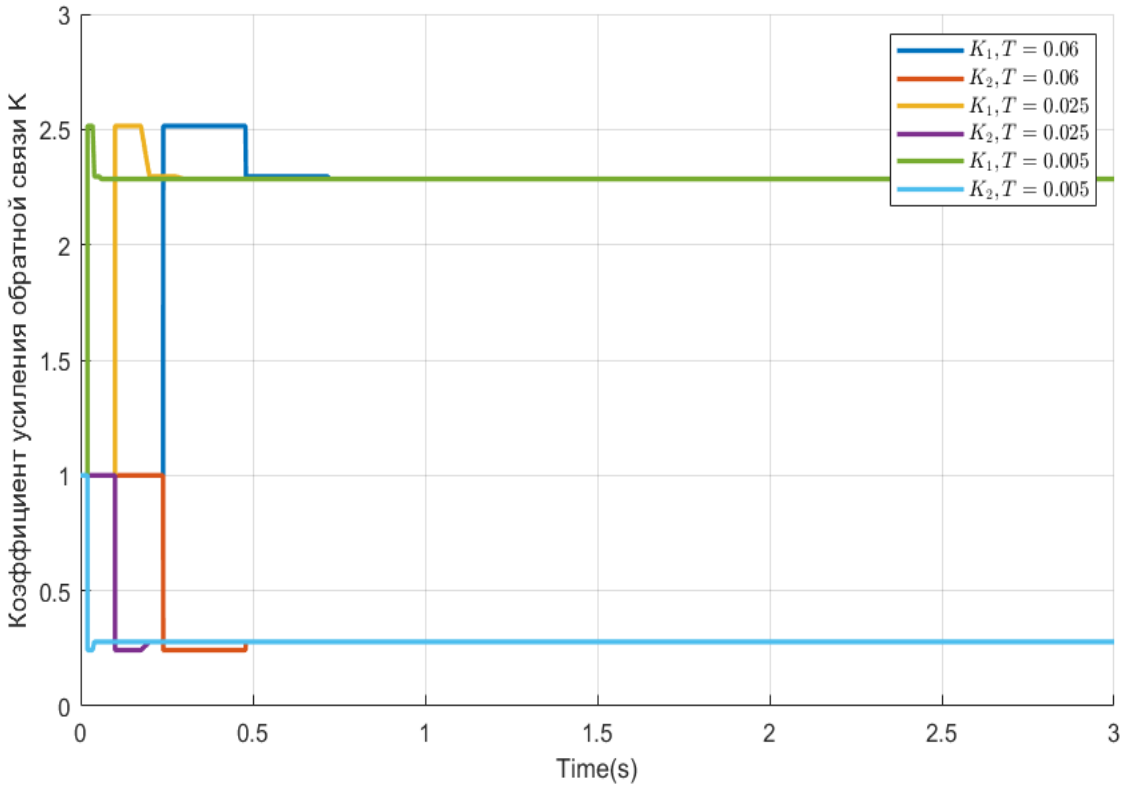


Рисунок 4.6 – Графики коэффициента усиления обратной связи K оптимального адаптивного регулятора для разных значений T

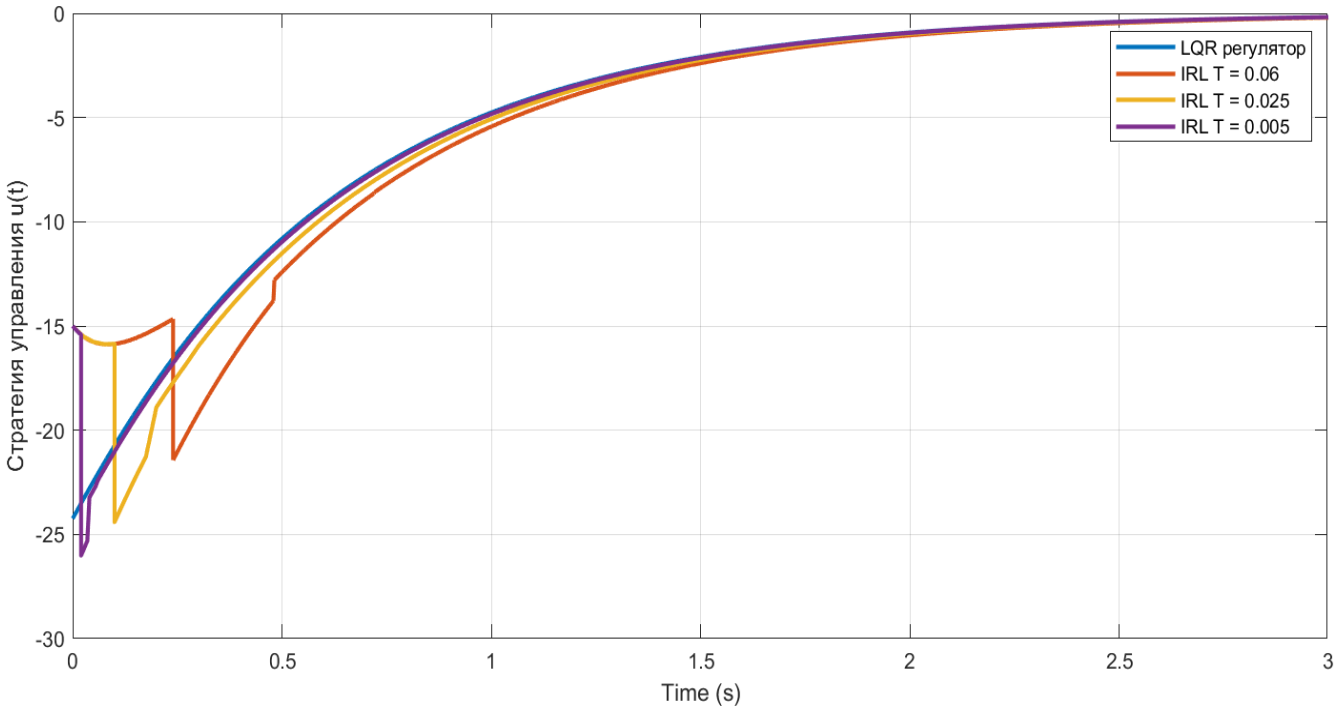


Рисунок 4.7 – График, представляющий управляющий сигнал оптимального регулятора LQR и оптимального адаптивного регулятора с различными значениями T.

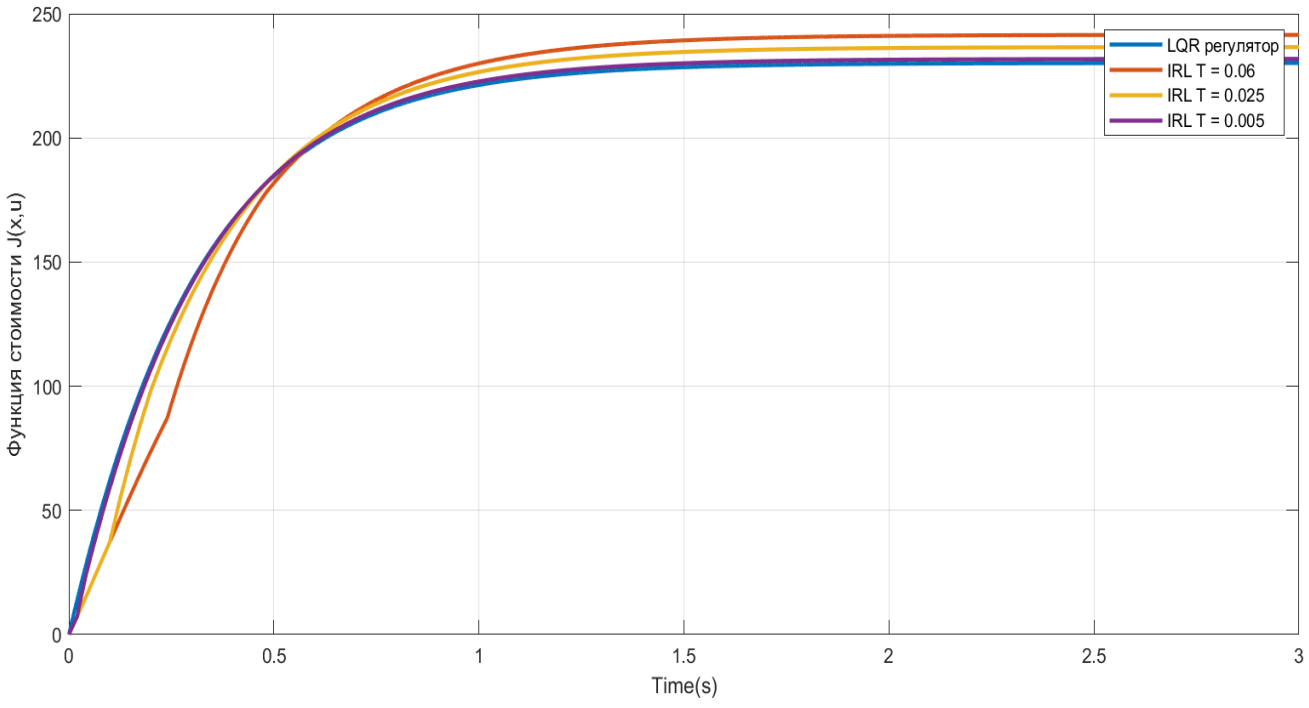


Рисунок 4.8 - График, отражающий значение функции стоимости для оптимального регулятора и оптимального адаптивного регулятора с различными значениями T.

Поскольку алгоритм использует данные, полученные из траектории системы, система не будет устойчивой, когда состояние системы x(t) равно примерно 0, и в этот момент алгоритм будет работать неправильно, что приведет к тому, что управление перестанет быть оптимальным. Поэтому мы должны выключить алгоритм, когда стратегия управления приблизится к оптимальному значению. В этом случае нам необходимо отключить алгоритм, когда коэффициент матрицы обратной связи K приблизится к оптимальному значению. Еще один случай, на который следует обратить внимание, заключается в том, что мы применяем управляющий сигнал u_0 , чтобы сделать систему устойчивой. Поэтому может случиться так, что система сойдётся до того, как будет найдено оптимальное значение управления. Поэтому нам нужно выбрать подходящее время выборки T.

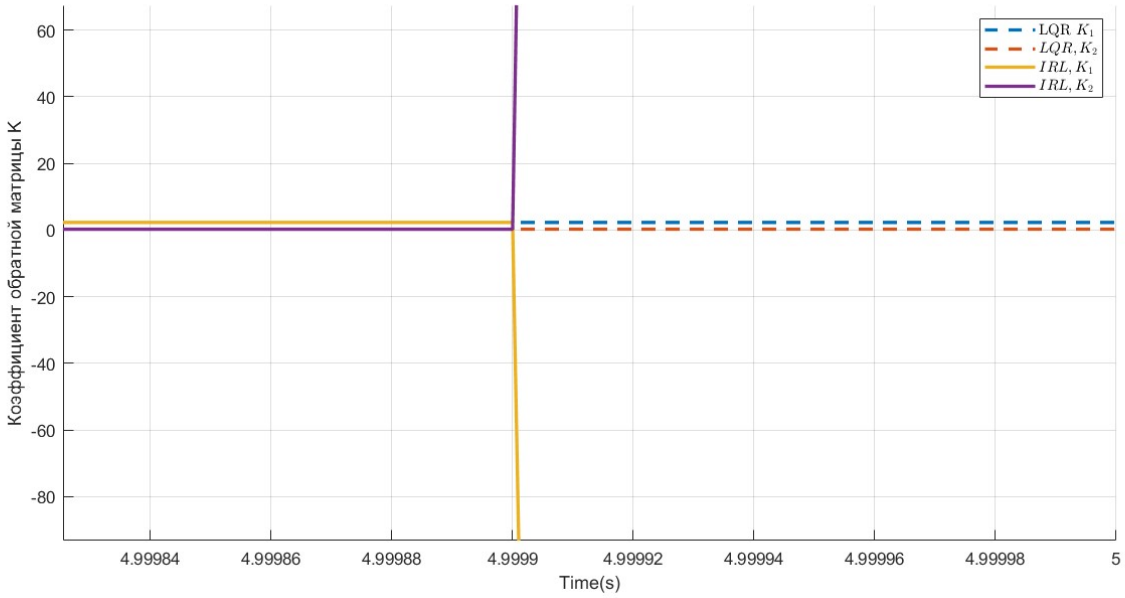


Рисунок 4.9 - Графики коэффициента усиления матрицы обратной связи K оптимального регулятора и оптимального адаптивного регулятора при $T = 0.25$

На графике выше мы видим, что значение коэффициента матрицы обратной связи K приблизилось к оптимальному значению, как и в случае использования LQR. Но когда состояние системы сходится к нулю, алгоритм не отключается, в результате чего значение коэффициента матрицы обратной связи K быстро расходится.

При применении обучении с интегральным подкреплением и приближенного динамического программирования для линейных систем мы часто используем квадратичную функцию стоимости. Пусть n - размер состояния системы, $x \in \mathbb{R}^n$. Тогда объем данных, который необходимо собрать

для вычисления новой стратегии управления, равен $\frac{n(n+1)}{2}$. Таким образом, для квадратичной системы $(n = 2)$ необходимо собрать не менее 3 точек данных, а для квадратичной системы $(n = 4)$ необходимо собрать не менее 10 точек данных, чтобы аппроксимировать параметры матрицы P. То есть, собрав необходимое количество точек данных, алгоритм будет аппроксимировать параметры матрицы P. Алгоритм продолжается до тех пор, пока значение P не сойдется с оптимальным значением.

4.3 Регулятор для нелинейных систем

Рассмотрим нелинейных систем

$$\dot{x} = \begin{bmatrix} -x_1 + x_2 \\ -0.5x_1 - 0.5x_2(1 - (\cos(2x_1) + 2)^2) \end{bmatrix} + \begin{bmatrix} 0 \\ \cos(2x_1) + 2 \end{bmatrix} u \quad (4.3)$$

Функция ценности:

$$J = \int_0^\infty (x^T Q x + u^T R u) dt \quad (4.4)$$

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, R = 1$$

Решить уравнение HJB мы получили функцию Беллмана и сигнал оптимального управления

$$V^*(x) = w_1^* x_1^2 + 2w_{12}^* x_1 x_2 + w_2^* x_2^2 = 0.5x_1^2 + x_2^2 \quad (4.5)$$

$$u^*(x) = -(\cos(2x_1) + 2)x_2^2 \quad (4.6)$$

Функция вектора активации выбрана

$$\phi(x) = [x_1^2 \quad x_1 x_2 \quad x_2^2] \quad (4.7)$$

Инициализировать начальный вес $W = [-2 \quad 0 \quad 5]^T$ и $x_0 = [1 \quad 2]^T$

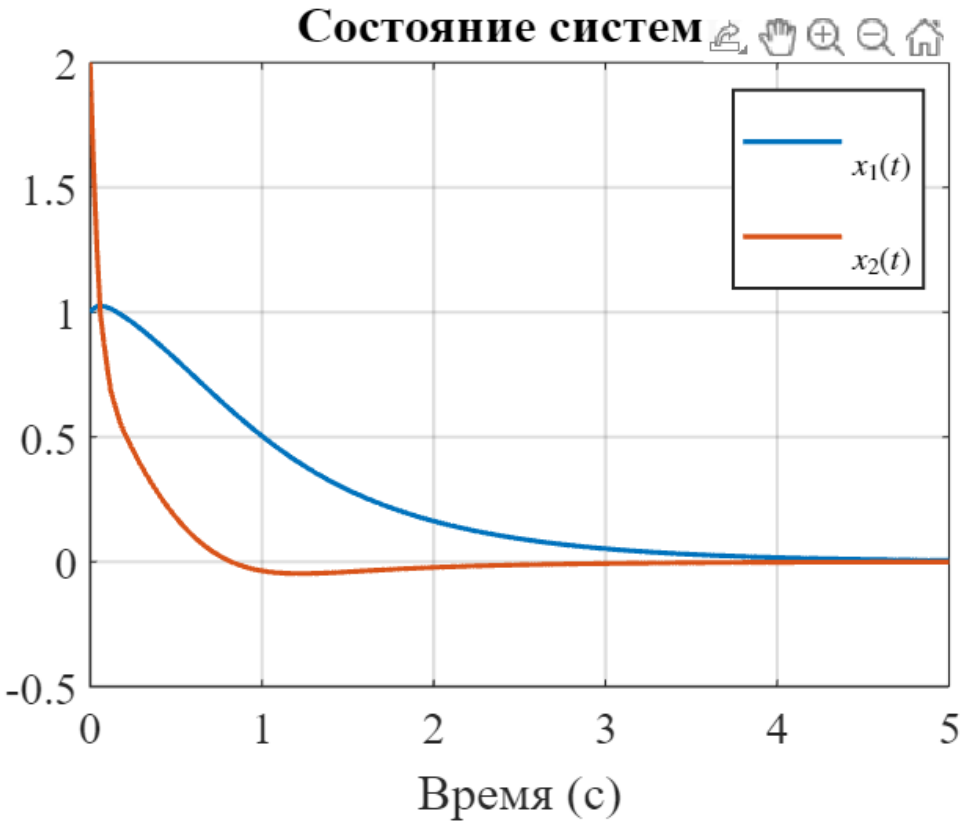


Рисунок 4.10 – График состояния системы с использованием онлайна обучения с интегральным подкреплением

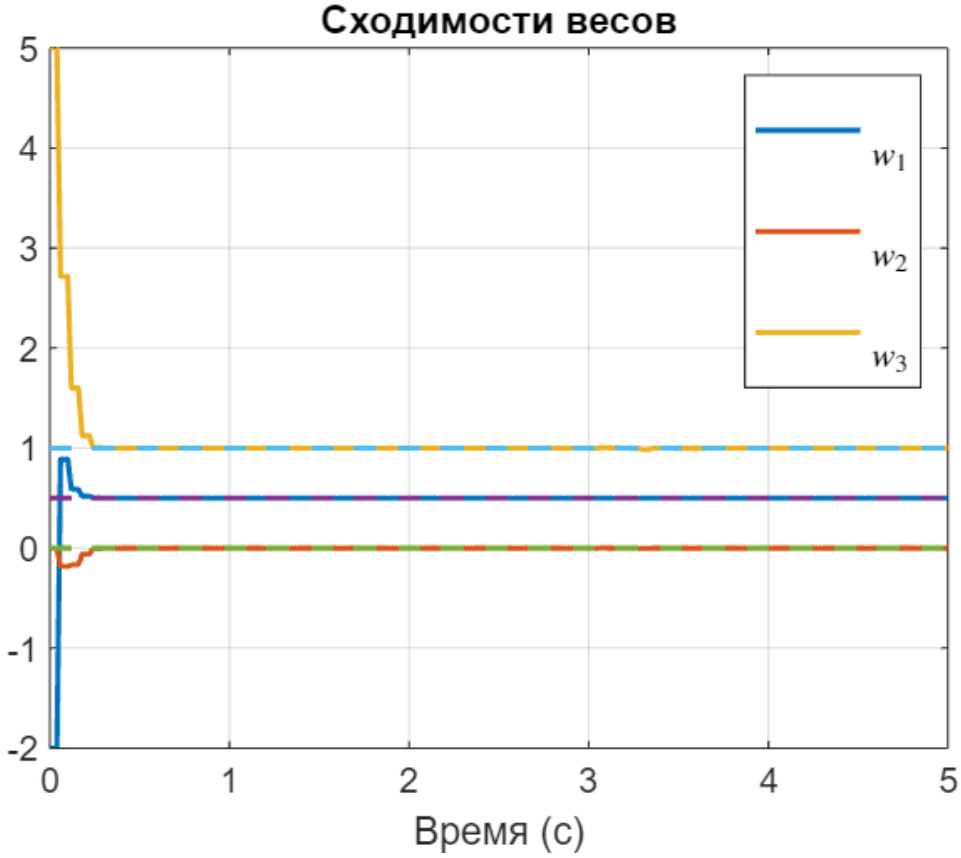


Рисунок 4.11 Сходимость весов W

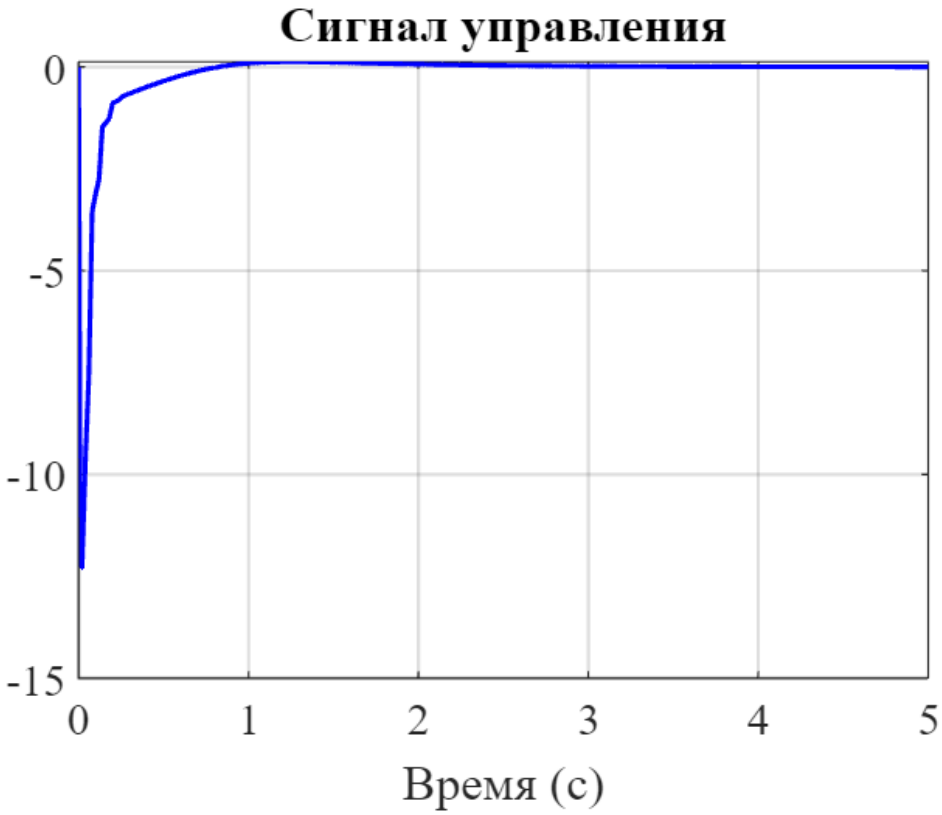


Рисунок 4.12 – Сигнал управления системой с алгоритмом обучения с подкреплением

Как видно из графика, вес W сходится точно к своему оптимальному значению через 0.24 секунды, в то время как сигнал управления по-прежнему помогает системе стабилизироваться на достаточно хорошей скорости. Тем самым показывая корректность алгоритма.

4.4 Регулятор линейной системы для перевернутого маятника и тележки

Данная система

$$\dot{x} = Ax + Bu,$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{(M+m)g}{ML} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{mg}{M} & 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ -\frac{1}{ML} \\ 0 \\ \frac{1}{M} \end{bmatrix}$$

Где $x = \begin{bmatrix} \theta \\ \dot{\theta} \\ x \\ \dot{x} \end{bmatrix}$ – состояние системы, $u \in \mathbb{R}^1$ – сигнал управления системы

Параметры для моделирования из [18]

$$M = 2.4 \text{ кг}, m = 0.23 \text{ кг}, g = 9.81 \frac{\text{м}}{\text{с}^2}, L = 0.46 \text{ м}$$

Задача оптимального управления состоит в оптимизации следующей функции

$$J(x,u) = \int_0^\infty \left(x^T(\tau)Qx(\tau) + u^T(\tau)Ru(\tau)\right)d\tau$$

$$Q = \begin{bmatrix} 1 & 0.1 & 0 & 0 \\ 0.1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.1 \\ 0 & 0 & 0.1 & 1 \end{bmatrix}, R = 1$$

Матрица оптимального регулятора (LQR)

$$K = R^{-1}B^T P = [-62.7 \quad -13.1 \quad -1.0 \quad -2.92]$$

Результаты моделирования

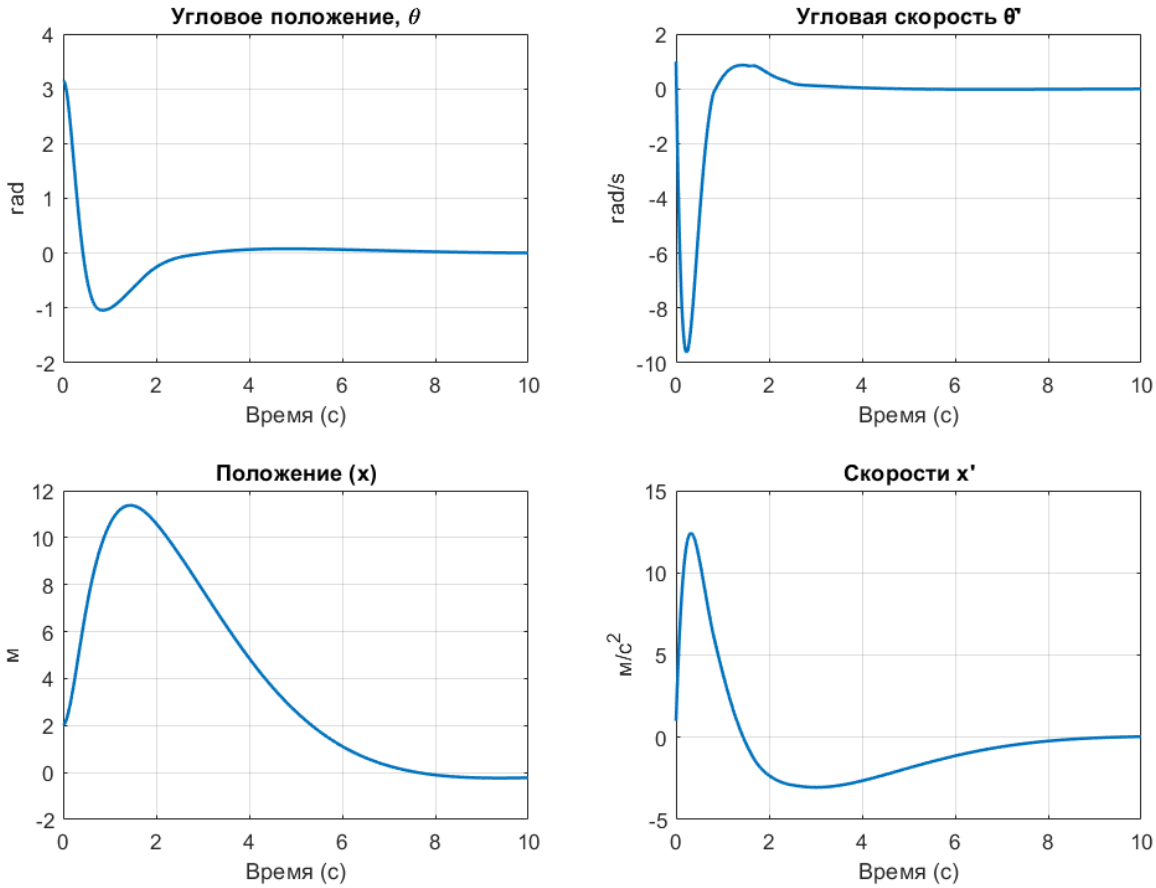


Рисунок 4.13 – Графика состояния системы $x(t)$

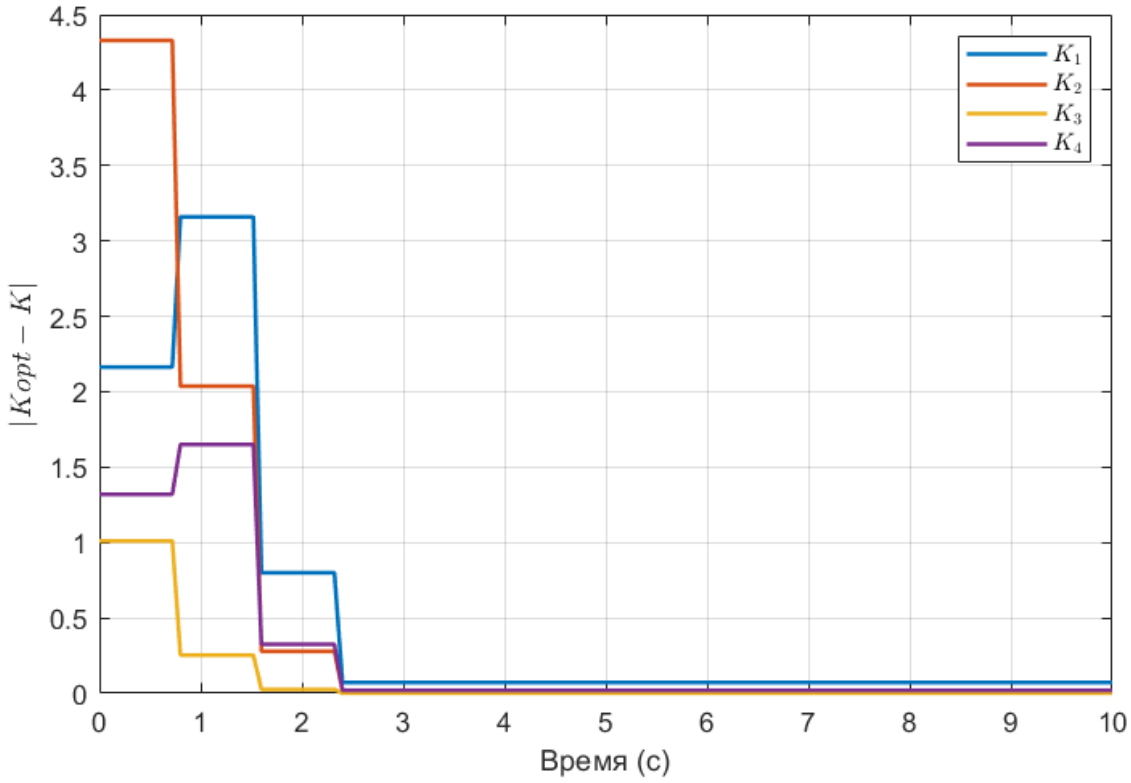


Рисунок 4.14 – График отклонения коэффициентов усиления управления K от их оптимальных значений K_{opt}

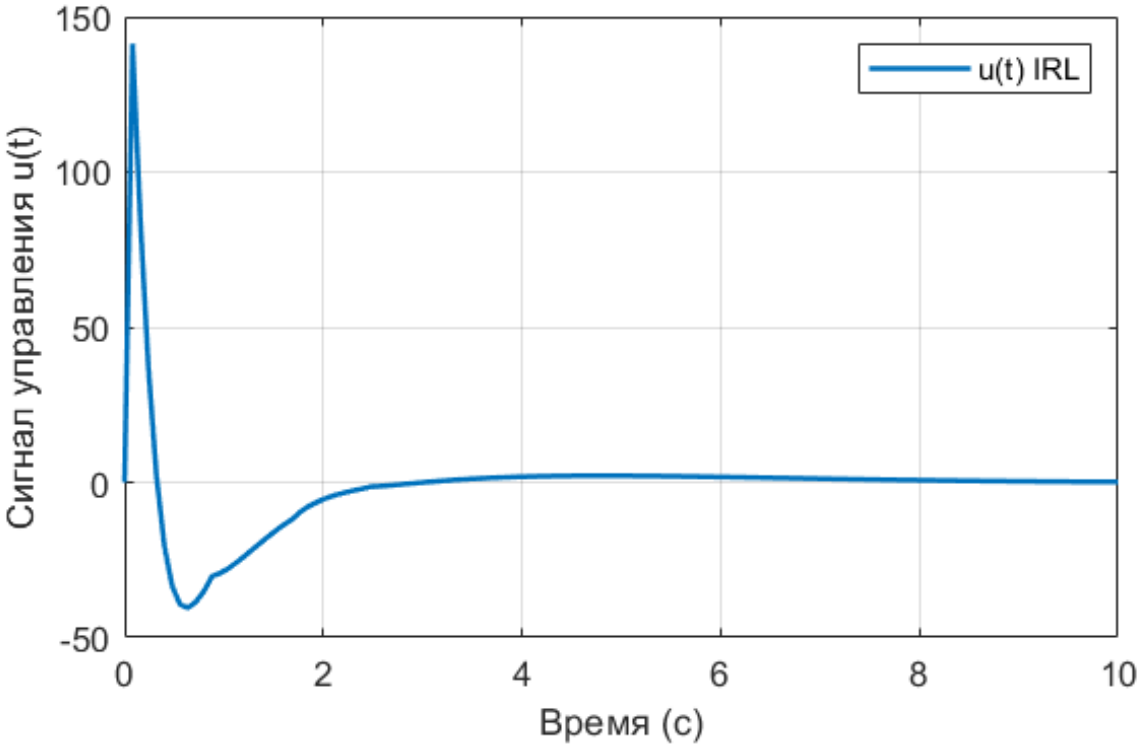


Рисунок 4.15 – График сигнал управления системы $u(t)$

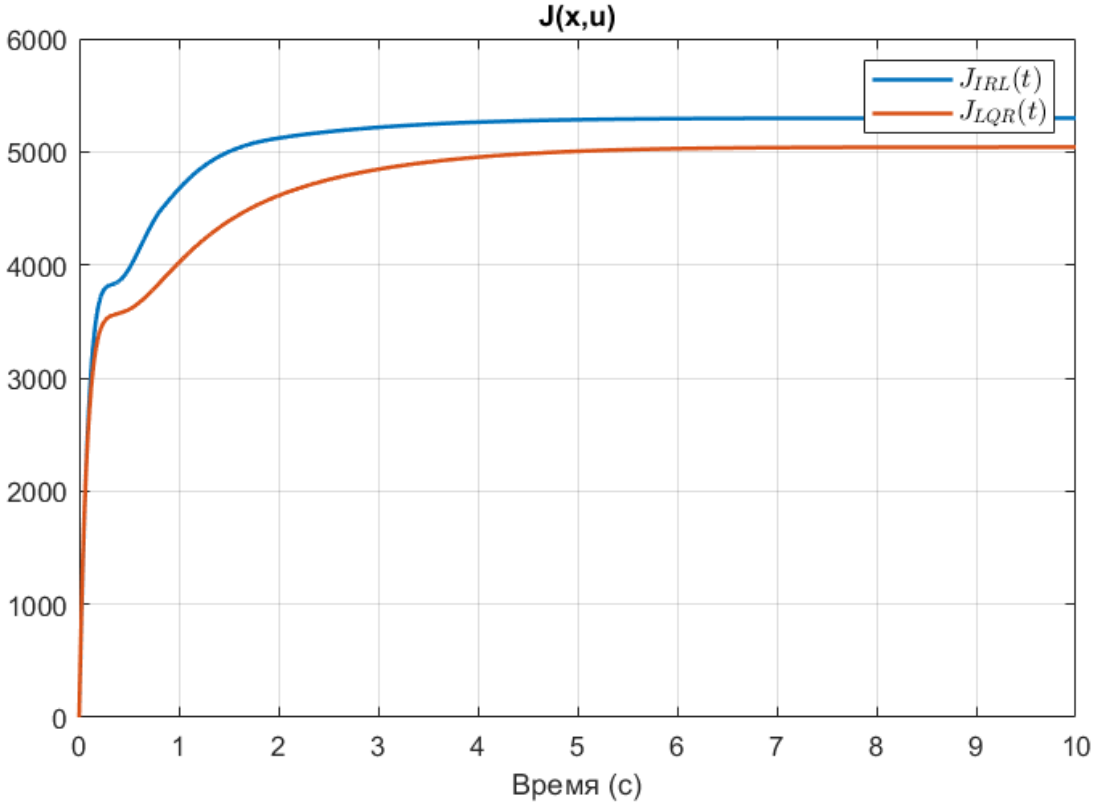


Рисунок 4.16 – График, отражающий значение функции стоимости оптимального регулятора и оптимального адаптивного регулятора

4.5 Моделирование для маятника

Рассмотрим систему маятника

$$\dot{x} = \begin{bmatrix} \dot{\theta} \\ \frac{-mglsin(theta)-B\dot{\theta}}{ml^2} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{ml^2} \end{bmatrix} u \quad (4.8)$$

Где $x = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix}$- состояние системы, u – сигнал управление

Функция ценности:

$$J = \int_0^\infty (x^T Q x + u^T R u) dt \quad (4.4)$$

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, R = 1$$

Для моделирования, функции вектора активации выбирается следующим образом

$$\phi(x) = [x_1^2 \quad x_1 x_2 \quad x_2^2]$$

Инициализировать начальный вес $W = [0 \quad 2 \quad 0]^T$ и $x_0 = \begin{bmatrix} \frac{\pi}{6} & 0 \end{bmatrix}^T$

Результаты моделирования

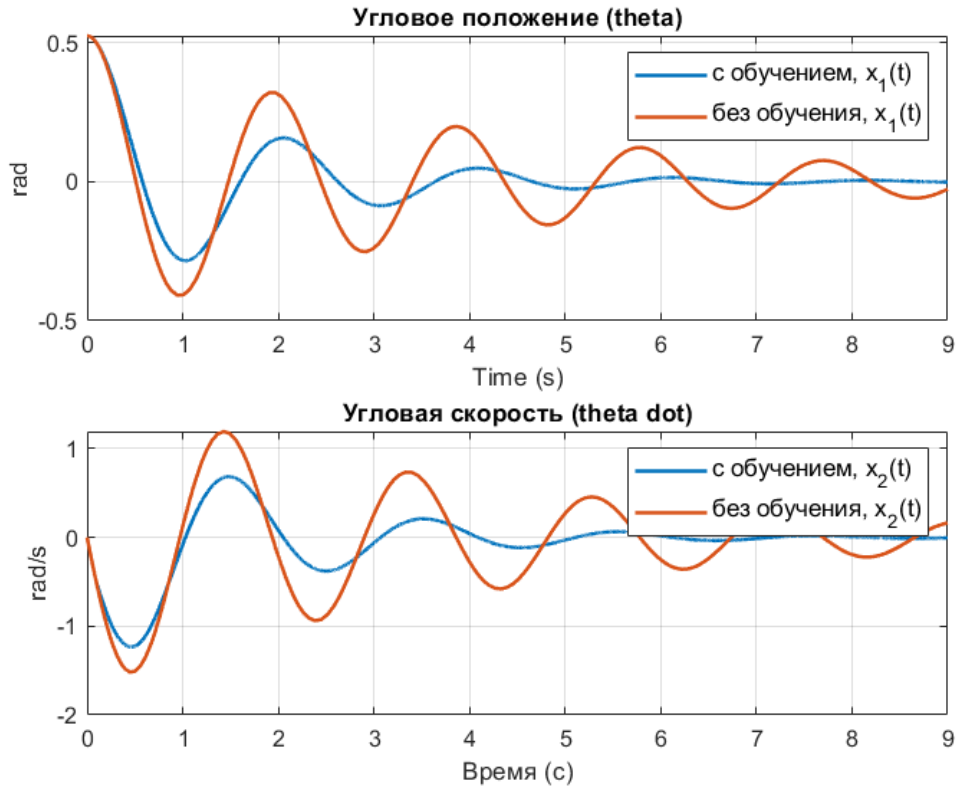


Рисунок 4.17 – Состояние с обучением и без обучения

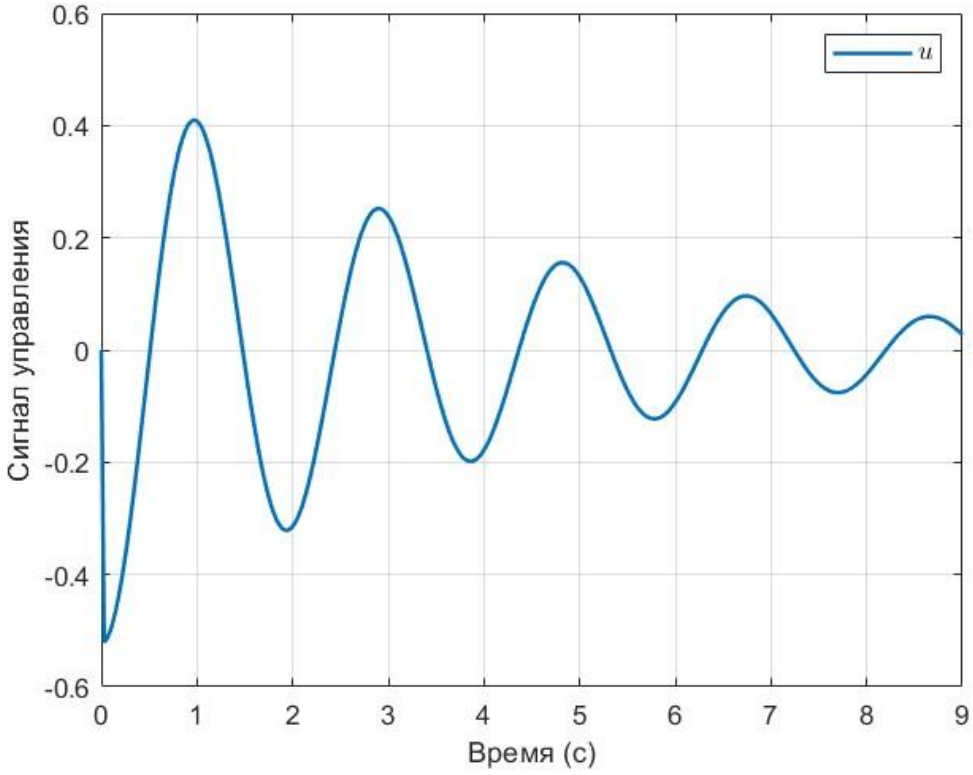


Рисунок 4.18 – Сигнал управления при без обучения

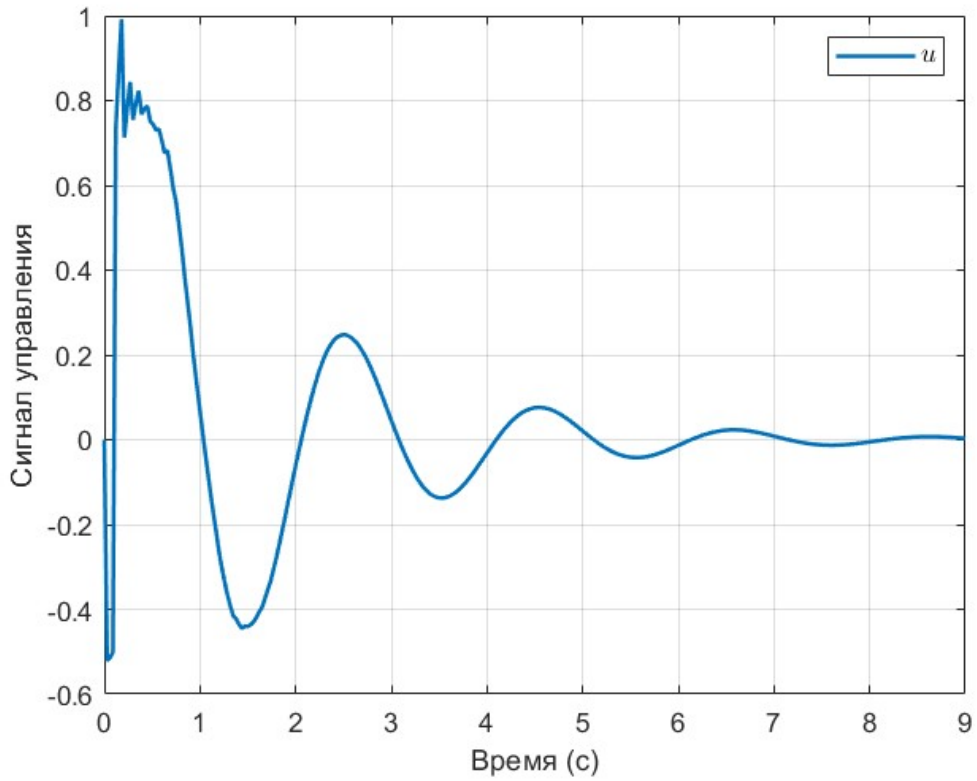


Рисунок 4.19 – Сигнал управления u с обучением

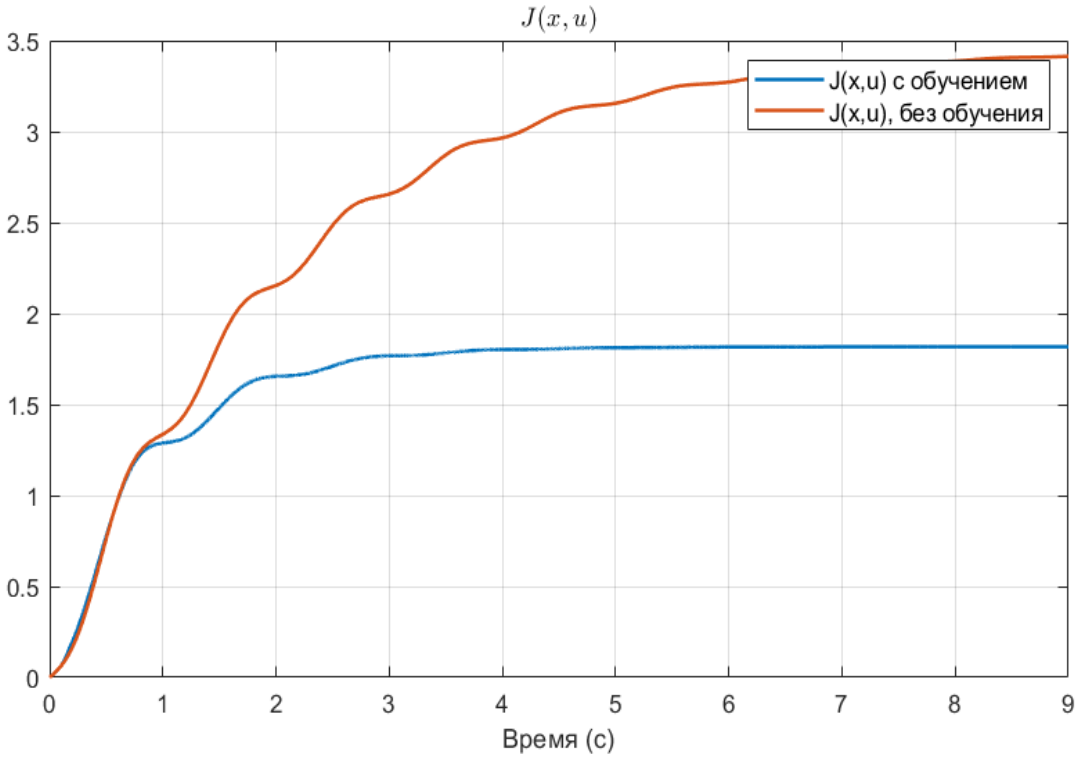


Рисунок 4.20 – Значение функции стоимости $J(x,u)$ с обучением и без обучения

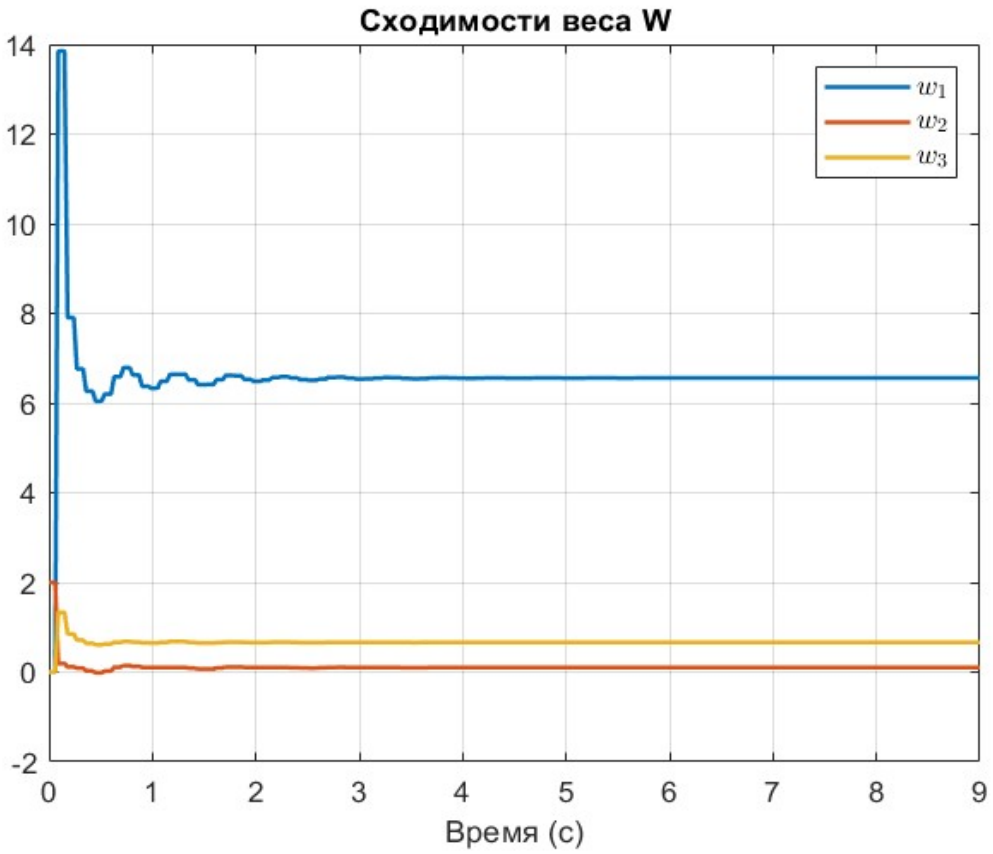


Рисунок 4.21 – Сходимости весов при применении обучения с интегральным подкреплением

Мы видим, что состояние системы сходилось лучше в случае с обучением. Значение функции стоимости также минимизировано, видно, что оно меньше в случае без обучения. Таким образом, при использовании обучение с интегральным подкреплением, регулятор улучшает исходный управляющий сигнал u_0 и минимизирует функцию стоимости $J(x,u)$. Для этой системы сложно решить задачу HJB для сравнения с результатами системы. Но в целом алгоритм IRL сошелся к стратегии управления, которая лучше, чем изначально инициализированный сигнал управления. Пример в разделе 4.3 более наглядно показывает сходимость алгоритма к оптимальному значению управления для нелинейной системы.

5 Заключение

В данной работе были изучены основные концепции и идеи, лежащие в основе алгоритма обучения с подкреплением. Были изучены два популярных алгоритма в обучении с подкреплением — это алгоритмы итерации по стратегии и по критерию, предназначенные для решения проблемы поиска оптимальных стратегии.

Представлены и исследованы математические регуляторы, таким как адаптивный регулятор, оптимальное регулятор (LQR) и оптимальный адаптивный регулятор, основанные на обучении с интегральным подкреплением.

Моделирование выполняется в MATLAB. Для линейных систем был реализован анализ и исследование эффективности оптимального адаптивного регулятора на основе обучения с интегральным подкреплением. Проведено сравнение эффективности оптимального адаптивного регулятора с эффективностью оптимального регулятора и адаптивного регулятора. В результате оптимальный адаптивный регулятор имеет эффективность, примерно равную эффективность оптимального регулятора. При этом адаптивный регулятор не сходится к оптимальному управлению. Для нелинейных систем корректность оптимального адаптивного регулятора доказана, когда веса функции аппроксимируют сходящееся значение к решению уравнения Гамильтона Якоби Беллмана.

Оптимальный адаптивный регулятор, основанный на обучении с интегральным подкреплением, строится на основе данных, собранных по траектории движения системы. Поэтому проанализировано и изучено влияние времени собрки данных T на сходимость оптимального управления оптимального адаптивного регулятора. В результате, чем меньше время T, тем лучше система приближается к оптимальному управлению. Выбор подходящего времени сборки данных был представлен. Если регулятор сходится к оптимальному значению, мы должны прекратить обновление стратегии управления системой, подробности представлены в разделе 4.2.

Таким образом, был успешно исследован оптимальный адаптивный регулятор на основе интегрального обучения с подкреплением. Преимущество этого регулятора в том, что мы получаем оптимальный управляющий сигнал без необходимости полного понимания динамики системы (без знания матрицы A для линейных систем и без знания $f(x)$ для нелинейных систем).

Список использованных источников

1. Frank Lewis and Draguna Vrabie // Reinforcement learning and adaptive dynamic programming for feedback control. Circuits and Systems Magazine, IEEE, 9:32 – 50, 01 2009. http://dx.doi.org/10.1109/MCAS.2009.933854

2. Ivo Grondman, Lucian Bus¸oniu, Gabriel A. D. Lopes, and Robert Babuska // A Survey of Actor-Critic Reinforcement Learning: Standard and Natural Policy Gradients. https://doi.org/10.1109/TSMCC.2012.2218595

3. Ivo Grondman, Lucian Bus¸oniu, Gabriel A.D. Lopes and Robert Babuska // A Survey of Actor-Critic Reinforcement Learning: Standard and Natural Policy Gradients

4. V. R. Konda and J. N. Tsitsiklis // “On Actor-Critic Algorithm”, SIAM Journal on Control and Optimization, vol. 42, No. 4, pp. 1143–1166, 2003.

5. Richard S. Sutton and Andrew G. Barto // Reinforcement Learning: An Introduction, Second Edition, MIT Press, Cambridge, MA, 2018

6. Fadi AlMahamid, and Katarina Grolinger, Department of Electrical and Computer Engineering Western University // Reinforcement Learning Algorithms: An Overview and Classification

7. A. Geramifard, T. J. Walsh, S. Tellex, G. Chowdhary, N. Roy, and J. P. How// A Tutorial on Linear Function Approximators for Dynamic Programming and Reinforcement Learning, Vol. 6, No. 4 (2013) 375–454, 2013

8. Beakcheol Jang, Myeonghwi Kim, Gaspard Harerimana, and Jong Wook Kim // Q-Learning Algorithms: A Comprehensive Classification and Applications, Department of Computer Science, Sangmyung University, Seoul 03016, South Korea

9. Draguna Vrabie, Kyriakos G. Vamvoudakis and Frank L. Lewis // Optimal Adaptive Control and Differential Games by Reinforcement Learning Principles. http://dx.doi.org/10.1049/PBCE081E

10. Frank L. Lewis, Draguna L. Vrabie, Vassilis L. Syrmos // Optimal control-3rd ed, 2012.

11. Bahare Kiumarsi, Member, IEEE, Kyriakos G. Vamvoudakis, Senior Member, IEEE, Hamidreza Modares, Member, IEEE, and Frank L. Lewis, Fellow, IEEE, Optimal and Autonomous Control Using Reinforcement Learning: A Survey, 2018.

12. Kyriakos G. Vamvoudakis, Draguna Vrabie, Frank L. Lewis, Online adaptive learning of optimal control solutions using integral reinforcement learning, 2011. https://doi.org/10.1109/ADPRL.2011.5967359

13. D. Vrabie, O. Pastravanu, M. Abu-Khalaf, and F. L. Lewis, "Adaptive optimal control for continuous-time linear systems based on policy iteration," Automatica, vol. 45, no. 2, pp. 477–484, 2009. https://doi.org/10.1016/j.automatica.2008.08.017

14. D. Vrabie and F. L. Lewis, "Neural network approach to continuoustime direct adaptive optimal control for partially-unknown nonlinear systems," Neural Netw., vol. 22, no. 3, pp. 237–246, Apr. 2009.

15. Vamvoudakis KG, Lewis FL. Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem. Automatica 2010. https://doi.org/10.1109/TAC.1968.1098829

16. Kleinman D. On an iterative technique for Riccati equation computations. IEEE Transactions on Automatic Control 1968;

17. Uğur Yıldıran, Yildiz Technical University, Adaptive Control of an Inverted Pendulum by a Reinforcement Learning based LQR Method.

18. Vinayak Kumar, Ruchi Agarwal, Modeling and Control of Inverted Pendulum cart system using PID-LQR based Modern Controller, July 01-03, 2022, Prayagraj, India.