

Трёхконтурная система обучения как альтернатива единой целевой функции в архитектуре автономных агентов

Аннотация

Современные методы обучения автономных систем, включая обучение с подкреплением (RL) и обучение на демонстрациях (imitation learning), как правило, опираются на единую целевую функцию. Такой подход неизбежно приводит к конфликтам приоритетов в сложных средах, где агент должен одновременно обеспечивать безопасность, достигать цели и взаимодействовать с другими агентами. Компромиссное решение часто оказывается неоптимальным или опасным. В данной работе предлагается альтернативная архитектура, основанная на трёх параллельных, конкурирующих каналах валентности: витальном (V_h), эпистемическом (V_e) и социальном (V_s). Каждый канал имеет собственную функцию оценки и динамически конкурирует за управление через механизм Winner-Takes-All с гистерезисом. Показано, что такая архитектура не требует ручного взвешивания приоритетов и способна разрешать конфликты целей без компромиссов. Приведены примеры применения в задачах автономной навигации и управления транспортом.

Автор: Максим Тимошенко
Дата: 25.04.2026

1. Введение: проблема единой целевой функции

1.1. Как устроены современные автономные системы

подавляющее большинство современных автономных агентов — от роботов-пылесосов до беспилотных автомобилей — обучаются через оптимизацию **единой целевой функции**. В случае обучения с подкреплением (RL) это reward-функция, которая назначает числовую награду за желательное поведение и штраф за нежелательное. В случае imitation learning это функция потерь, которая минимизирует расхождение между действиями агента и демонстрациями эксперта.

Такой подход хорошо работает в простых, изолированных средах — например, в играх (Atari, Go, шахматы) или в симуляциях с чётко определёнными правилами. Однако по мере усложнения среды и увеличения числа одновременно действующих факторов единая целевая функция начинает давать сбои.

1.2. Проблема конфликта приоритетов

Реальный мир не сводится к одной метрике. Автономный автомобиль должен одновременно:

- **Обеспечивать безопасность** пассажиров и окружающих (избегать столкновений).
- **Достигать цели** (следовать маршруту, соблюдать ПДД).
- **Взаимодействовать с другими агентами** (пропускать спецтранспорт, понимать жесты регулировщика, реагировать на команды водителя).

Эти цели не просто различны — они могут вступать в прямой конфликт. Например, соблюдение ПДД требует остановиться на красный свет, а безопасность может потребовать проехать перекрёсток, чтобы избежать столкновения с потерявшим управление автомобилем сзади.

Классический подход к разрешению таких конфликтов — **ручное взвешивание** компонентов целевой функции:

text

$$R_{\text{total}} = \alpha * R_{\text{safety}} + \beta * R_{\text{navigation}} + \gamma * R_{\text{social}}$$

где коэффициенты α , β , γ подбираются разработчиком вручную. Этот подход имеет фундаментальные недостатки:

1. **Комбинаторный взрыв.** Невозможно предусмотреть все возможные сочетания факторов и подобрать для каждого правильные веса.

2. **Компромисс вместо приоритета.** Усреднение сигналов даёт «среднее» решение, которое может оказаться опасным в критической ситуации. Если R_safety требует тормозить, а $R_navigation$ требует ехать, компромиссное решение может означать «притормозить, но не остановиться» — что приведёт к аварии.
3. **Непрозрачность.** При возникновении аварии трудно понять, почему агент принял то или иное решение — веса были подобраны эмпирически, а не на основе явных архитектурных принципов.

1.3. Альтернативный подход: конкуренция вместо усреднения

В данной работе предлагается принципиально иной подход. Вместо того чтобы закладывать все цели в одну функцию и усреднять их, мы предлагаем **разделить их на независимые контуры** и позволить им **конкурировать** за управление.

Этот подход навеян функциональной архитектурой биологического мозга, где различные мотивационные системы (выживание, познание, социальность) конкурируют за доступ к моторному выходу через механизмы латерального ингибирования. Победитель получает полный контроль, а не усреднённый компромисс.

2. Трёхконтурная архитектура

2.1. Общее описание

Предлагаемая архитектура состоит из трёх независимых каналов валентности — V_h (витальный), V_e (эпистемический) и V_s (социальный). Каждый канал:

- Имеет собственные входные данные (сенсоры, внутреннее состояние).
- Имеет собственную функцию оценки, которая вычисляет числовой приоритет P_i в диапазоне 0–100.
- Не зависит от других каналов и не требует информации об их состоянии.

В каждый момент времени управление агентом получает тот канал, чей приоритет максимален.

2.2. Витальный контур (V_h)

Назначение: обеспечение физической целостности агента и его гомеостаза.

Входные данные:

- Сенсоры, непосредственно связанные с безопасностью (лидар, радар, акселерометр, датчики удара).
- Параметры внутреннего состояния (уровень энергии, температура, давление масла и т.д.).

Функция приоритета P_{vh} :

Приоритет V_h вычисляется как сумма вкладов от различных факторов угрозы. Основной фактор — время до столкновения (ТТС). При $TTC < 0.5$ секунды приоритет достигает 100, что вызывает безусловное прерывание. Другие факторы включают: обнаружение пешехода, критический перегрев двигателя, падение давления масла, аквапланирование, гололёд.

Ключевая особенность: V_h имеет **безусловный приоритет**. При достижении критического порога ($P_{vh} \geq 90$) он перехватывает управление немедленно, игнорируя все остальные контуры и механизм гистерезиса.

2.3. Эпистемический контур (V_e)

Назначение: познание среды, навигация, достижение целей.

Входные данные:

- Данные позиционирования (GPS/ГЛОНАСС, IMU, колёсные одометры).
- Карта (дорожный граф, разметка, знаки, светофоры).
- Данные о трафике и погоде.

Функция приоритета P_{ve} :

Базовый приоритет V_e составляет 40. Это означает, что в нормальном режиме движения V_e всегда активен и управляет агентом. Приоритет повышается при отклонении от маршрута, приближении к сложному перекрёстку, ухудшении погодных условий, снижении эффективности движения (затор).

Максимальный приоритет V_e ограничен значением 80. Это гарантирует, что навигационные задачи никогда не смогут перевесить критический сигнал безопасности от V_h (90+).

2.4. Социальный контур (V_s)

Назначение: взаимодействие с другими агентами и оператором.

Входные данные:

- Команды оператора (голос, жесты, нажатия кнопок).
- Сигналы от других агентов (V2X-сообщения).
- Данные о спецтранспорте (сирены, проблесковые маячки).
- Данные о регулировщике.

Функция приоритета P_{vs} :

Приоритет V_s варьируется от 0 (социальные сигналы отсутствуют) до 75. Наивысший приоритет (70) назначается команде регулировщика. Команды водителя имеют переменный приоритет: от 25 (пожелание «поехали быстрее») до 65 (экстренное требование остановки).

Максимальный приоритет V_s (75) ниже критического порога V_h (90), что гарантирует: даже регулировщик не может заставить автомобиль совершить опасный манёвр.

3. Механизм конкуренции

3.1. Winner-Takes-All (WTA)

В каждый момент времени все три контура вычисляют свои приоритеты P_{vh} , P_{ve} , P_{vs} . Управление получает контур с максимальным приоритетом:

text

$$\text{winner} = \operatorname{argmax}(P_{vh}, P_{ve}, P_{vs})$$

Проигравшие контуры полностью отстраняются от управления, но продолжают мониторинг и пересчёт своих приоритетов.

3.2. Гистерезис

Для предотвращения «дребезга» — хаотичного переключения между контурами при близких значениях приоритетов — используется гистерезис с параметром $\Delta = 15$. Переключение на новый контур происходит только если его приоритет превышает приоритет текущего активного контура как минимум на 15 единиц:

text

```
if P_new > P_current + 15:
```

```
    switch_to(new_contour)
```

Это означает, что новый контур должен быть «заметно увереннее» текущего. Например, если V_e имеет приоритет 40, а V_s — 50, разница составляет 10, чего недостаточно для переключения. V_e сохраняет управление. Только когда V_s поднимет свой приоритет до 55 или выше, он сможет перехватить управление.

3.3. Безусловное прерывание

Из правила гистерезиса есть одно исключение: критический порог V_h , равный 90. Если P_{vh} достигает 90 или выше, гистерезис игнорируется, и управление передаётся V_h немедленно, на следующем же такте (через 10 миллисекунд при частоте арбитра 100 Гц).

Это реализует фундаментальный принцип: **безопасность важнее всего**. Никакие навигационные задачи, никакие команды водителя или регулировщика не могут перевесить прямую угрозу жизни.

3.4. Сравнение с классическим взвешиванием

Параметр	Классическое взвешивание	Трёхконтурная WTA
Принцип	Усреднение сигналов	Конкуренция за управление
Критические ситуации	Компромисс (опасно)	Безусловный приоритет V_h
Конфликт ПДД и безопасности	Ручное взвешивание	Динамическое переключение
Масштабируемость	Комбинаторный взрыв	Независимые контуры

Параметр	Классическое взвешивание	Трёхконтурная WTA
Объяснимость	«Весы так подобраны»	«Победил V_h с приоритетом 95 из-за $TTC < 0.5$ »

4. Аппаратная реализация

В отличие от классических нейросетевых подходов, требующих мощных GPU и облачных вычислений, трёхконтурная архитектура может быть реализована на дешёвых микроконтроллерах с NPU (например, WCH CH32V307, ~\$1.5). Для гарантированной реакции на критические события витальный контур V_h рекомендуется выносить на отдельный MCU (WCH CH32V003, ~\$0.3), подключенный к исполнительным механизмам через мультиплексор.

Общая стоимость дополнительного оборудования менее \$100 для автомобилей, что делает архитектуру доступной для массового применения.

5. Примеры применения

5.1. Автономный автомобиль: проезд перекрёстка с регулировщиком

Этот сценарий демонстрирует ключевое преимущество трёхконтурной архитектуры — способность разрешать конфликты без компромиссов.

Исходная ситуация: автомобиль подъезжает к перекрёстку. Светофор красный.

Шаг 1: V_e (навигация) имеет приоритет 45 (40 базовый + 5 за приближение к перекрёстку). Он готов остановиться на красный.

Шаг 2: V_s обнаруживает регулировщика, жест которого означает «проезжайте». V_s выставляет приоритет 70. Поскольку $70 > 45 + 15$, V_s перехватывает управление. Автомобиль начинает движение на красный.

Шаг 3: V_h обнаруживает пешехода, перебегающего дорогу. $TTC = 0.3$ секунды. V_h выставляет приоритет 100 — безусловное прерывание. Автомобиль экстренно тормозит, несмотря на приказ регулировщика.

Шаг 4: Пешеход ушёл. Приоритет V_h падает до 10. V_s снова получает управление и завершает манёвр.

Шаг 5: Перекрёсток пройден. V_s теряет приоритет (регулировщик остался позади). Управление возвращается V_e .

Этот сценарий невозможно реализовать с помощью единой целевой функции без риска либо проигнорировать регулировщика, либо задавить пешехода.

5.2. Автономный дрон: потеря видимости в тумане

Архитектура также была проработана для задачи автономной навигации БПЛА в условиях отсутствия GPS и радиосвязи, включая обход опасных зон, потерю видимости в тумане, восстановление позиции по рельефу и другие сценарии. (Подробности — у автора.)

6. Границы применимости

Трёхконтурная архитектура не является универсальным решением для всех задач ИИ. Она наиболее эффективна в следующих условиях:

1. **Наличие нескольких конкурирующих целей**, которые не могут быть сведены к одной метрике без потери качества.
2. **Наличие критических ситуаций**, требующих безусловного приоритета безопасности.
3. **Работа в реальном времени**, где компромиссное решение может быть опасным.

Для задач, где цели не конфликтуют (например, игра в шахматы), классический RL с единой целевой функцией остаётся предпочтительным.

7. Заключение

В данной работе предложена трёхконтурная архитектура обучения автономных агентов, основанная на принципе динамической конкуренции независимых каналов валентности. Показано, что такой подход решает фундаментальную проблему конфликта приоритетов, с которой сталкиваются классические методы обучения с единой целевой функцией.

Ключевые преимущества архитектуры:

- Безусловный приоритет безопасности через механизм критического прерывания V_h .
- Отсутствие необходимости в ручном взвешивании компонентов целевой функции.
- Прозрачность и объяснимость решений.
- Возможность реализации на дешёвом оборудовании.

Архитектура верифицирована на задаче управления беспилотным автомобилем. Также проработано применение к автономной навигации БПЛА (подробности — у автора). Дальнейшие направления исследований включают: формальную верификацию механизма конкуренции, оптимизацию параметров гистерезиса и критического порога, а также применение архитектуры к другим классам автономных систем (промышленные роботы, умные дома, медицинские устройства).

Ссылки:

1. Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning: An Introduction* (2nd ed.). MIT Press.
2. Silver, D., et al. (2021). Reward is enough. *Artificial Intelligence*, 299, 103535.
3. LeCun, Y. (2022). A Path Towards Autonomous Machine Intelligence. *OpenReview*.
4. Тимошенко М. (2026). *Становление субъекта: архитектура, этика и дорожная карта субъективного ИИ*.

GitHub: <https://github.com/maksim-timoshenko/AI-consciousness-continuum>

5. Тимошенко М. (2026). *Функциональная архитектура мозга: три контура мотивации и поведения.*
6. Тимошенко М. (2026). *Трёхконтурный автопилот для автономного наземного транспорта (эскизный проект).*