

# Распределенное моделирование гибридных систем

## Введение

В наши дни компьютерное моделирование является мощным и широко применяемым инструментом прогнозирования и анализа поведения и свойств существующих или разрабатываемых систем в самых различных областях деятельности человека. Моделирование позволяет предсказывать последствия внесения изменений в систему и проводить оптимизацию параметров системы в соответствии с различными критериями, тем самым повышая эффективность принимаемых и снижая риск принятия ошибочных решений. С ростом объема и сложности моделей систем, а также появлением специфических требований к самим моделям и процессу моделирования, возникла необходимость в распределенном моделировании, когда исполнение различных компонент системы проводится на многопроцессорной системе или на различных компьютерах сети. Распределенное моделирование позволяет добиться повышения производительности, достичь инкапсуляции внутренней логики работы компонент, а также создает предпосылки для повторного использования уже созданных компонент в моделях новых систем и возможности совместной работы компонент, созданных различными производителями.

Цель данной работы состояла в изучении возможностей распределенного моделирования гибридных систем, т.е. систем, в которых существенную роль играют как дискретные, так и непрерывные переменные (скорость, координата, стоимость и т.п.).

На начальном этапе работы были изучены возможности существующих стандартов распределенного моделирования: HLA, ALSF, DIS. Все эти стандарты разработаны для моделирования только дискретных событийных систем. Стандарт HLA (High Level Architecture) является наиболее перспективным и расширяемым стандартом, потому он был выбран в качестве основы для изучения проблем распределенного моделирования гибридных систем. Затем профессиональная среда моделирования гибридных систем AnyLogic ([www.anylogic.com](http://www.anylogic.com)) была расширена программными модулями, поддерживающими распределенное моделирование в стандарте HLA. После этих подготовительных этапов стало возможным приступить собственно к научной части работы: к анализу возможности распределенного моделирования гибридных систем в рамках стандарта HLA. В следующих разделах излагаются результаты этой работы.

## 1 Проблемы распределенного моделирования непрерывных и гибридных систем

Стандарт HLA является стандартом для систем с дискретным поведением. В случае необходимости какому-либо объекту демонстрировать непрерывное поведение, наиболее простым и очевидным решением является дискретизация значений непрерывной переменной по времени. Однако такое решение проблемы не является удовлетворительным в случае наличия сложных зависимостей между распределенными объектами по непрерывным переменным, что может привести к демонстрации моделью неадекватного поведения (неустойчивость изначально стабильных систем, запаздывание реакции, ...). Особенно сильно эти проблемы проявляются при распределенной симуляции гибридных систем, где дискретное и непрерывное поведение зависят друг от друга. Например, система из трех компонент [Lin, K., "Simulation of Distributed Dynamic Systems with Time Delays", 1997], уравнения динамики которых (ЛДУ второго порядка) связаны по непрерывным переменным, при «локальном» (не распределенном) моделировании демонстрирует устойчивое поведение (рис. 1), в то время как при распределенном моделировании при равномерной по времени дискретизации значений непрерывных переменных начиная с определенных значений шага дискретизации, та же система начинает демонстрировать неустойчивое поведение, расходится (рис. 2).

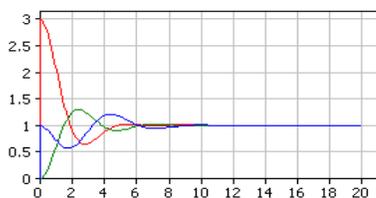


рис. 1 «Локальное» моделирование системы

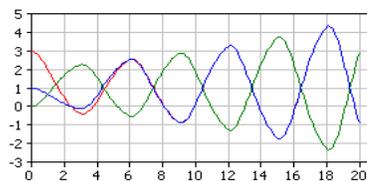


рис. 2 Распределенное моделирование системы

Причиной получения такого результата послужила принципиальная невозможность совместного решения системы дифференциальных уравнений для всех компонент системы при распределенном моделировании, когда каждое из уравнений решается независимо от других, а значения переменных обновляются в дискретные моменты времени.

## 2 Разрыв связи компонент по непрерывным переменным

В процессе построения модели системы происходит определение ее структуры (декомпозиция), выделение компонент и их описание на выбранном языке представления. Для обмена информацией между компонентами устанавливаются связи. В случае моделирования гибридной системы часть связей может соединять компоненты по непрерывным переменным (например, координатам в пространстве), когда один компонент экспонирует свою непрерывную переменную, а другой – отслеживает ее изменения. Отслеживающий компонент может использовать значение переменной для описания собственного непрерывного поведения (например, переменная будет присутствовать в правой части дифференциальных уравнений, описывающих поведение отслеживающего компонента); для принятия решения (отслеживание момента выполнения некоторого условия с участием этой переменной) или других целей. При моделировании всей системы на одной моделирующей машине (локальное моделирование) для получения корректных результатов в первом случае уравнения, задающие поведение компонент, должны быть объединены в единую систему уравнений и решаться совместно. Для отслеживания с необходимой точностью условия (предиката), заданного над непрерывной переменной, требуется встраивание механизмов проверки таких предикатов в алгоритмы вычисления значений непрерывной переменной (например, интеграция проверки условий с алгоритмами численных методов решения дифференциальных уравнений). Примеры связей компонент по непрерывным переменным для первого и второго случаев приведены на рис. 3 и рис. 4 соответственно.

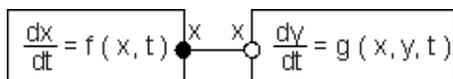


рис. 3 Связь по непрерывной переменной первого типа

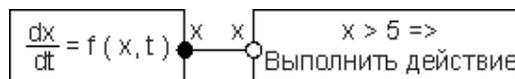


рис. 4 Связь по непрерывной переменной второго типа

В процессе разбиения модели на распределенные компоненты происходит нарушение «непосредственных» связей между компонентами. Взаимодействие осуществляется посредством инфраструктуры поддержки распределенного исполнения моделей (рис. 5), которая может работать поперек относительно медленных каналов передачи данных.

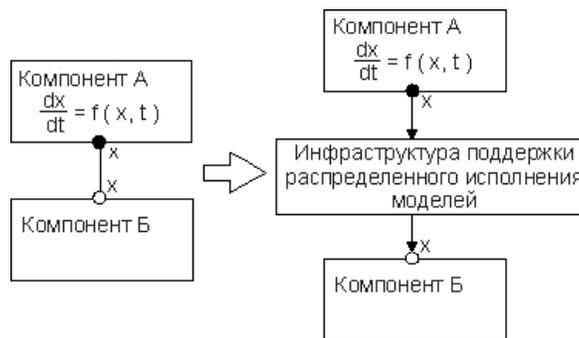


рис. 5 Нарушение связи по непрерывным переменным

Связи по дискретным переменным состояния естественным образом представляются в виде сообщений, посылаемых лишь в моменты изменения значений переменных. В этом случае

не происходит потери информации, так как наблюдающий компонент имеет всю информацию о дискретном состоянии наблюдаемого компонента. При разрыве связи по непрерывным переменным возникает проблема выбора метода представления непрерывно меняющегося значения переменной для удаленного компонента. Могут быть предложены следующие решения:

- Обновление значений с заданной периодичностью. Достоинство – простота реализации. Недостатки – существенные потери информации, точности, увеличение вероятности проявления проблемы чувствительности (поведение компоненты между двумя последовательными обновлениями неизвестно для наблюдающего, который может пропустить интересующее его событие). Достижение заданной точности может потребовать большой частоты обновлений, передаваемых по сети, что быстро сведет скорость моделирования к нулю.
- Обновление значений с заданной точностью. Моделирующий компонент посылает обновление наблюдающему компоненту только в случае, если отклонение фактического значения переменной от предыдущего значения, которое было передано ранее, превысит некоторый порог. Это значительно более гибкий подход, позволяющий гарантировать заранее заданную точность. Недостатки – достижение заданной точности может потребовать большой частоты обновлений.
- Предсказание поведения. Компонент делает предположение о дальнейшем поведении переменной и передает наблюдающему компоненту параметры аппроксимирующей функции. Обновление параметров происходит при достижении максимально допустимого отклонения предсказания от фактического значения переменной. Достоинства – более точное представление поведения. Недостатки – частота обновлений зависит от адекватности выбора аппроксимирующей функции, дополнительная вычислительная нагрузка на компоненты зависит от вида критерия допустимого отклонения и сложности вычисления аппроксимирующей функции, ошибка определения момента выполнения условия над наблюдаемой переменной может быть существенна.

Однако рассмотренные выше методы, к сожалению, в ряде случаев не могут обеспечить необходимую точность при моделировании гибридных систем при приемлемом количестве обновлений. Поэтому в ходе проведенного исследования был предложен свой метод, который полностью решает проблему для ограниченного, но важного класса распределенных гибридных моделей:

- **Удаленная обработка предикатов.** В гибридных системах непрерывное и дискретное поведение взаимосвязаны и могут влиять друг на друга. При распределенном моделировании представление дискретных событий и взаимодействий не вызывает особых затруднений. Однако, отслеживание влияния непрерывного поведения на дискретное (рис. 4), требует достаточно точного представления непрерывного поведения для отслеживающего компонента. В распределенном случае применение периодического обновления либо предсказания с помощью аппроксимирующей функции может привести к неприемлемому запаздыванию или пропуску события при определении момента выполнения условия, заданного над непрерывной переменной. Как правило, форма таких условий известна на этапе построения модели (например,  $h > C$ , где  $h$  – непрерывная переменная,  $C$  – константа). Идея удаленной обработки предикатов заключается в перенесении процесса вычисления условий из наблюдающего компонента в компонент, моделирующий соответствующую переменную. Наблюдающий компонент устанавливает значение параметров. Моделирующий компонент вычисляет значения переменной, проверяет выполнение условия и сообщает о факте его выполнения наблюдающему компоненту. Определение момента времени наступления события в этом случае происходит с максимально возможной точностью, что может быть немаловажным при моделировании процессов управления и принятия решений. При этом решается задача уменьшения объема передаваемой в процессе моделирования информации между распределенными компонентами.

Предлагаемый метод удаленной обработки предикатов при распределенном моделировании гибридной системы имеет свои ограничения. В случае необходимости проверки сложного условия вида  $P(x,y)$ , где  $x$  и  $y$  – непрерывные переменные, моделируемые различными

распределенными компонентами, организация удаленной обработки такого условия возможна лишь в случае, когда  $P(x,y)$  может быть представлена в виде  $F(P(x),P(y))$ , то есть когда условия на  $x$  и  $y$  являются разделяемыми. В противном случае может быть рекомендовано пересмотреть способ разбиения модели системы на распределенные компоненты таким образом, чтобы компоненты, жестко связанные по непрерывным переменным исполнялись в рамках одного распределенного компонента (с использованием одной моделирующей машины и ее внутренних методов реализации связи компонент по непрерывным переменным). Альтернативой может служить применение методов периодического обновления значений либо предсказания поведения непрерывной переменной с обязательным контролем приемлемости результирующей потери точности (см. выше).

### 3 Организация взаимодействия распределенных компонент на основе агентной технологии

Описанные выше методы организации взаимодействия распределенных компонент были опробованы на системе AnyLogic, в которую на первых этапах работы были встроены средства поддержки стандарта распределенного моделирования HLA.

Для реализации описанных выше методов была построена специальная агентная система, ядро которой тесно взаимодействует с ядром компонента распределенной модели (см. рис. 6). Эта агентная система и была включена в систему моделирования AnyLogic.

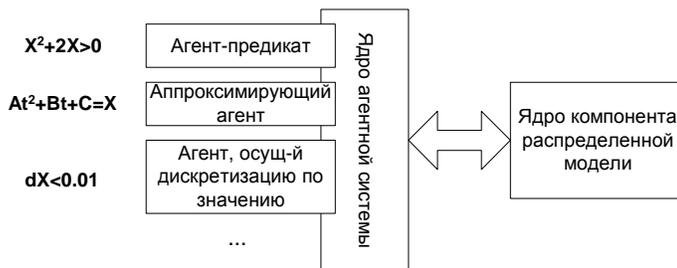


рис. 6 Агентная система внутри компонента распределенной модели

Суть предложенного подхода в следующем. На компоненте распределенной модели подключается специальный программный модуль – ядро агентной системы. Данный модуль осуществляет взаимодействие с ядром модели и имеет доступ ко всем переменным компонента с высокой точностью (определяемой используемым численным методом). К этому модулю по сети через специально разработанный интерфейс подключаются программы-агенты, которые посылаются другими компонентами распределенной модели, если они заинтересованы в мониторинге какой-либо переменной [переменных] данного компонента.

Программа-агент может представлять собой: программу, контролирующую частоту обновлений (дискретизация по значению); аппроксимирующую программу; программу-предикат и др. Будучи засланными, исполняющиеся программы-агенты осуществляют взаимодействие с «родительским» компонентом по сети. Так, например, агент-предикат в момент срабатывания предиката, высылает уведомление о произошедшем событии. Агент-аппроксиматор периодически высылает вектор параметров аппроксимирующей функции и так далее.

Так как агенты фактически подключаются к ядру системы моделирования, это позволяет им с большой точностью отслеживать изменения переменных, производить аппроксимацию и вычисление моментов срабатываний предикатов.

#### 3.1 Предопределенные типы агентов

В общем случае пользователь системы может написать своего агента фактически «с нуля» (единственное условие: агент обязательно должен удовлетворять требованиям интерфейса системы). Однако, в ходе работы были созданы несколько предопределенных типов агентов, которые могут быть использованы в большинстве случаев.

##### 3.1.1 Агент-линейный аппроксиматор

Данный агент осуществляет мониторинг заданной переменной и осуществляет аппроксимацию поведения этой переменной с помощью полиномов первой степени.

### 3.1.2 Агент-квадратичный аппроксиматор

Данный агент осуществляет мониторинг заданной переменной и осуществляет аппроксимацию поведения этой переменной с помощью полиномов второй степени.

### 3.1.3 Агент-вычислитель предиката

Данный предопределенный тип агента предусматривает вычисление заданного предиката на переменных удаленного компонента. Агент информирует компонент, инициировавший его загрузку, о моменте наступления данного события.

## 3.2 Преимущества разработанного решения

Разработанное решение позволяет значительно ускорить распределенное моделирование гибридной системы, увеличить точность моделирования, уменьшить сетевой трафик. Рассмотрим подробнее свойства этого решения:

**Универсальность.** Разработанная агентная система поддерживает четыре метода организации взаимодействия распределенных компонент, которые могут быть использованы в зависимости от требований конкретной модели.

**Динамическая загрузка.** Код агента может быть подгружен по сети в любой момент, когда в нем возникнет необходимость. Агент так же может отключиться от мониторинга переменных в любой момент.

**Поддержка обратной связи.** В интерфейс агентов включена возможность обратной связи с «родительским» компонентом. Агент может общаться с компонентом, инициировавшим его загрузку посредством специальных сообщений. Таким образом, возможно постоянное согласование параметров мониторинга и их модификация.

**Гибкость.** Данное решение позволяет использовать в качестве агента практически любой код. Т.е. возможно использование не только предикатов, аппроксимаций, но и других, более сложных конструкций. Например, внутри агента возможно использование некоторой предыстории изменения переменных, возможен анализ сложных переменных-объектов (например, текстовых строк) и так далее.

## 3.3 Пример работы агента-квадратичного аппроксиматора

На компонент, осуществляющий моделирование поведения маятника был послан аппроксимирующий агент, целью которого был мониторинг ординаты маятника с абсолютной точностью = 0.01. На рис. 7 представлен график координаты аппроксимируемой функции (абсцисса).

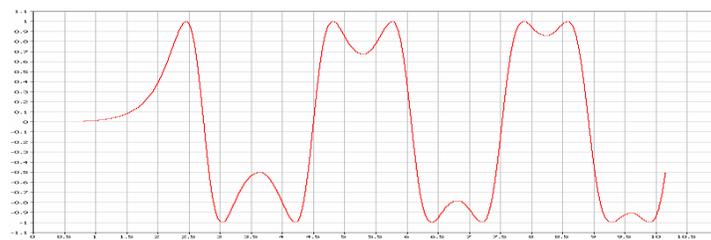


рис. 7 Реальное поведение

На рис. 8 представлены: график функции, построенной по результатам работы агента, осуществляющего аппроксимацию данной переменной полиномами второй степени; и график погрешности относительно функции-оригинала.

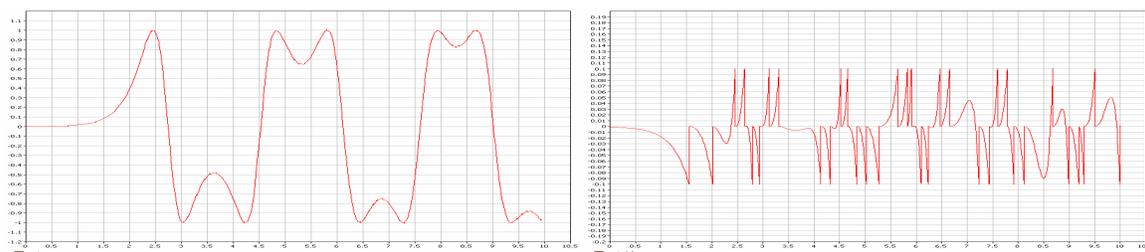


рис. 8 Аппроксимированное поведение (слева) и график погрешности аппроксимации

Точность аппроксимации – 0.01. При этом число сообщений (коэффициенты аппроксимирующего полинома), переданных по сети, составило 81. Для сравнения для достижения той же точности при использовании дискретизации по значению потребовалось 1315 обновлений значения переменной, при использовании дискретизации по времени – более 3000.

### 3.4 Работа агента-предиката. Модель «Защита от воздушной атаки»

Модель представляет собой стратегическую игру, в результате которой оценивается эффективность противовоздушной обороны по линии фронта. На ней расположены три ракетно-пусковых установки с ограниченными радиусами действия радаров. Атакующие самолеты стремятся преодолеть линию фронта и пролететь к мирным сооружениям. Пусковые установки обнаруживают и пытаются нейтрализовать самолеты противника. На рис. 9 показан пример работающей модели воздушной атаки.

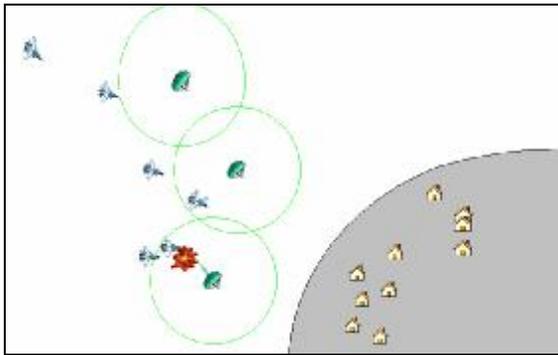


рис. 9: Защита от воздушной атаки

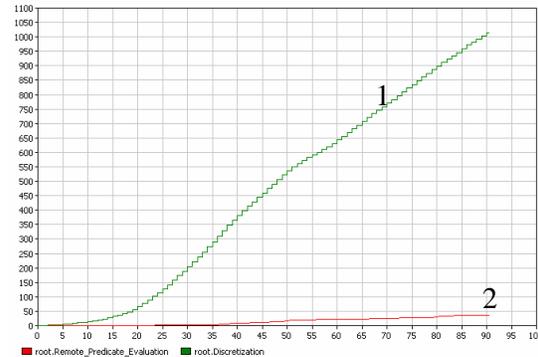


рис. 10 Число посланных по сети сообщений при использовании дискретизации по времени(1) и технологии удаленной оценки предикатов(2)

Для создания распределенных версий симуляционной модели использовалось следующее разбиение модели на компоненты:

1. компонент №1: атакующие самолеты;
2. компонент №2: радары и пусковые установки;
3. компонент №3: компонент, осуществляющий визуализацию (Viewer).

В результате были построены две версии данной модели: распределенная версия, использующая дискретизацию по времени и распределенная версия, использующая технологию удаленной оценки предикатов. В последнем случае в качестве предикатов выступало математическое описание радиусов действий радаров. При пересечении каким-либо из самолетов зоны действия радаров, высылалось подтверждение этого события. Т.о. имитировалось событие «обнаружения» самолета радарными установками.

Версия модели с использованием технологии удаленной оценки предикатов показала уменьшение сетевой загрузки более чем на 95% по сравнению с версией, использующей дискретизацию по времени (обновление раз в 1 сек. модельного времени) (см. рис. 10). При этом точность определения момента пересечения самолётом зоны действия радарной установки значительно возросла (определяется точностью численных методов).

## 4 Заключение

Цель данной работы состояла в изучении возможностей распределенного моделирования гибридных систем. В ходе проведения работ по адаптации стандарта распределенного моделирования для моделирования гибридных систем были достигнуты следующие результаты:

- Изучены существующие стандарты распределенного моделирования, и выбран стандарт HLA для адаптации к моделированию гибридных систем
- Осуществлена интеграция стандарта в систему моделирования AnyLogic, что потребовало модификацию ядра системы;
- Рассмотрено несколько подходов к организации взаимодействия распределенных компонент с точки зрения возможности их применения к моделированию гибридных систем;

- Предложен свой метод организации взаимодействия компонент распределенной гибридной модели: удаленная обработка предикатов;
- Создана система на основе агентной технологии, которая предоставляет четыре способа организации взаимодействия компонент распределенной гибридной модели;
- Построены модели раскрывающие возможности созданной системы.