

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ АВТОМАТИЗИРОВАННЫХ ФОРМОВОЧНЫХ ЛИНИЙ ДЛЯ ИЗГОТОВЛЕНИЯ ОТЛИВОК

Ю.Г. Дреус (МИФИ),

М.В. Зенькович, А.С. Любченко (ЗАО "Литаформ")

Излагаются основные принципы создания имитационных моделей автоматизированных формовочных линий (АФЛ) для изготовления отливок. Описание приводится на примере участка заливки-охлаждения АФЛ.

Рассматриваются вопросы создания имитационных моделей, применяемых для анализа различных компоновок АФЛ на стадии проектирования, выбора наилучшей структуры АФЛ в зависимости от технологических требований и разработки алгоритмов управления оборудованием, входящим в состав АФЛ. Рассмотрим принципы построения имитационной модели АФЛ на примере участка заливки-охлаждения АФЛ для изготовления отливок.

Перед разработчиками была поставлена задача создания имитационной модели, позволяющей эксперту оценивать работу участка заливки-охлаждения АФЛ, а также реализующую:

- отображение состояния участка в любой момент времени;
- сбор статистики о состоянии системы;
- возможность изменения параметров функционирования участка в любой момент времени;
- имитацию работы участка согласно техническим характеристикам его элементов;
- возможность изменения структуры и/или алгоритма функционирования моделируемого участка;
- возможность использования модели в качестве промышленного тренажера для обучения персонала АФЛ.

Описание объекта моделирования

Типовой участок заливки-охлаждения (рис. 1) состоит из следующих стандартных устройств: заливочная ветка на n мест для форм, ветка охлаждения форм на n мест, заливочная машина, которая вместе с одной или несколькими заливочными ветками размещается в непосредственной близости от плавильных печей, и передаточная тележка. Все ветки в зависимости от планировки могут быть как однонаправленные, так и двунаправленные.

Заливочная ветка на n мест — это очередь из незалитых форм длины n , на которой происходит заливка незалитых форм. Ветка охлаждения на n мест — это очередь из залитых форм длины n , на которой проис-

ходит охлаждение залитых форм. Передаточная тележка — механизм, с помощью которого происходит перемещение форм от одной ветки участка к другой. Передаточные тележки движутся по рельсовым путям, расположенным по краям заливочных и охлаждающих веток. Когда с одной стороны около ветки останавливается тележка с установленной на ней формой, а с другой стороны останавливается пустая тележка — можно произвести смещение очереди форм на одну позицию. Форма, находящаяся на тележке, сталкивается на ветку, а форма, находящаяся с другой стороны ветки, попадает на пустую тележку.

Заливочная машина служит для заливки незалитых форм, находящихся на заливочной ветке. Заливочная машина состоит из транспортного механизма и ковша, установленного на транспортный механизм.

Из внешней среды на участок заливки-охлаждения поступают незалитые формы с участка формовки и расплавленный металл в ковшах с участка плавки. С участка во внешнюю среду (на участок выбивки) поступают залитые охлажденные формы.

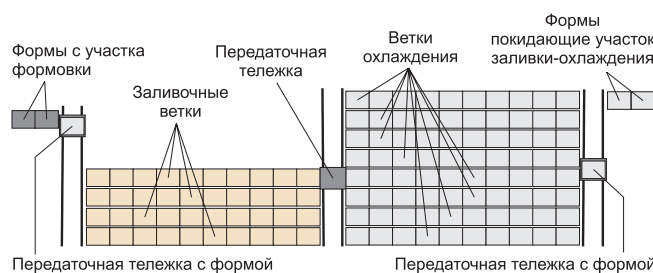
Понятия и определения, используемые в описании модели:

- модель — программа, имитирующая процесс функционирования системы в РВ;
- элемент модели — составная часть модели, не подлежащая дальнейшему делению. Элементами модели являются модели составных частей реального объекта (заливочные ветки, ковш, передаточная тележка и т.п.);
- программа-модель — конкретная реализация имитационной модели в программном коде.

Реализация

Для решения поставленной задачи использовалось объектно-ориентированное моделирование. В настоящее время существует множество универсальных и проблемно-ориентированных сред моделирования, однако для создания модели был выбран язык программирования С++ по причине того, что высокоуровневая среда моделирования является "черным ящиком" для пользователя, он не знает, по каким алгоритмам происходит моделирование. При создании модели на языке программирования пользователь свободен от этого ограничения. Для реализации возможности изменения структурной модели и алгоритмов взаимодействия элементов модели структурная модель, описание алгоритмов взаимодействия и интерфейс пользователя реализованы в различных модулях (рис. 2).

В связи с необходимостью отразить функционирование системы во времени для построения модели



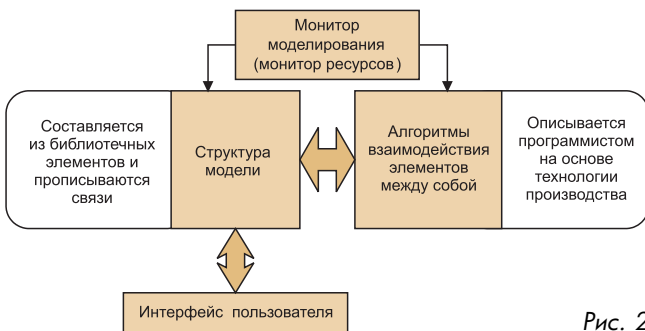


Рис. 2

был использован дискретно-событийный метод моделирования [1]. В качестве механизма продвижения времени был использован механизм продвижения времени с постоянным шагом.

Структурная модель

Модуль программы "структурная модель" является подпрограммой, описывающей состояние моделируемого объекта. Эта программа описывает каждый элемент моделируемого участка: его поведение, взаимосвязи с другими элементами. По своей сути, это структурная модель участка, в которой состоянию каждого элемента участка заливки-охлаждения АФЛ соответствует состояние соответствующих объектов программы.

Все объекты модели описаны в виде классов в понимании языка C++. Разработана библиотека этих классов [2], что дает возможность легко внедрять новые элементы в модель. Все элементы библиотеки являются наследниками базового класса или наследниками наследников базового класса.

Полями базового класса являются общие параметры устройств: "занят/свободен"; "работает/не работает", ссылка на модельное время, внутренний счетчик времени. Методами базового класса являются все методы, связанные с инкапсуляцией общих параметров, а также методы передачи временного среза параметров, т.е. методы, передающие переменные состояния в интерфейс пользователя.

Наследниками базового класса являются классы, описывающие реальные устройства участка (такие как передаточная тележка, заливочная ветка, ветка охлаждения, заливочный ковш и т.д.). Полями этих классов кроме полей, унаследованных от базового класса, являются поля, описывающие собственные параметры устройства и поля состояния устройств (эти поля входят в блок атрибутов). Собственные параметры устройств (к этим параметрам относятся: объем, число позиций, скорость, масса и т.п.) могут быть заданы пользователем или быть жестко прописаны в программе. В число этих параметров входят ссылки на другие элементы. Например, ковш не может лить металл в пустоту, поэтому у него должна быть ссылка на объект класса "форма". Поля состояния могут быть определены как собственные параметры, однако они выделяются отдельно, так как их главное отличие состоит в том, что эти поля объект меняет, тем самым определяя свое состояние. Например, для ковша это позиция, количест-

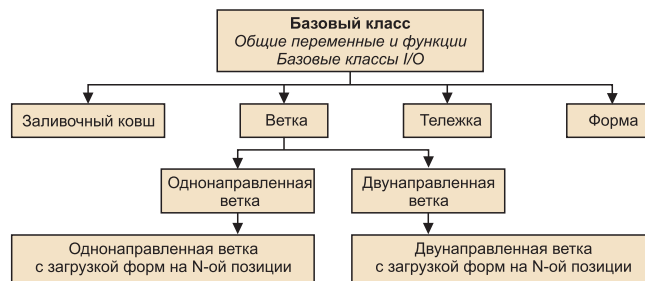


Рис. 3

во металла в данный момент и т.п. Методами класса являются собственные методы, описывающие действия устройств (например, у заливочного ковша – залить форму) и общие методы, унаследованные от родительских классов, а также методы передачи временного среза параметров, то есть методы, передающие переменные состояния в интерфейс пользователя. Была разработана библиотека классов, необходимых для построения структурной модели описываемого участка. Каждый класс библиотеки описывает поведение соответствующего элемента моделируемого производственного участка с требуемой детализацией. Например, поведение металла в ковше описано до расчета скорости истечения металла в зависимости от диаметра стаканчика и оставшегося количества металла в ковше, а поведение веток охлаждения – на уровне транспортировки форм. Также есть возможность описать более детально элементы определенного класса и прописать внутренние законы функционирования, если это понадобится для анализа функционирования системы. Структура библиотечных классов показана на рис. 3.

На базе имеющихся классов созданы элементы системы. Для реализации структурной модели участка необходимо было решить задачу описания взаимодействия элементов. Задача решена при помощи указателей на классы и методы. Каждый элемент системы имеет атрибут или атрибуты типа "указатель на класс". Имеется метод или методы, использующие эти указатели. Так в классе "Заливочный ковш" есть указатель на класс "Форма". Внутренняя функция (метод класса) "Залить металл" использует этот указатель для изменения состояния элемента, на который указатель ссылался. Каждый элемент системы имеет необходимые связи, реализованные по средствам указателей. При помощи этих связей изменяются состояния элементов, на которые ссылается рассматриваемый элемент в зависимости от запускаемой внутренней функции данного элемента.

Таким образом, получаем структурную модель участка заливки-охлаждения АФЛ. Такой подход к построению структурной модели решает проблему отслеживания распределения ресурсов во времени среди элементов системы. За распределение ресурсов отвечают методы соответствующих классов. Например, двунаправленная ветка имеет два указателя на класс "тележка" (правая и левая тележки) и метод "сдвиг влево". При запуске этого метода внутри него идет проверка состояния элементов, которые привязаны к соответствующим

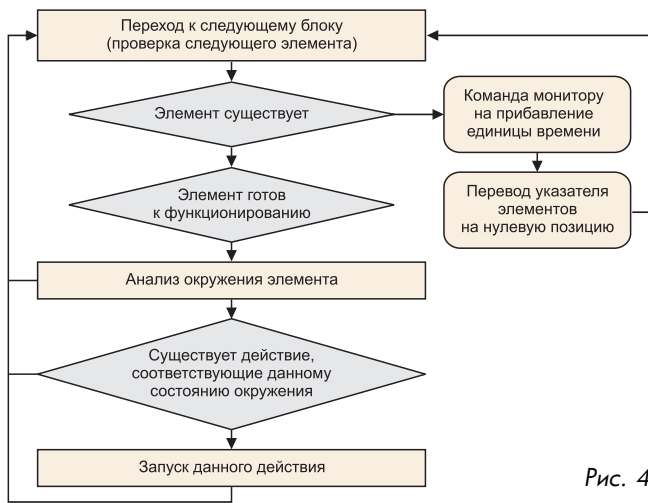


Рис. 4

указателям. Если на правой тележке нет формы или на левой нет свободного места, или хотя бы один указатель нулевой, то функция возвращает флаг невозможности операции. Однако при положительной проверке метод "сдвиг влево" меняет соответственно состояния элементов "тележка". Тем самым реализуется распределение ресурса типа "форма" между линией и тележками; то же происходит с другими ресурсами.

Модуль алгоритмов взаимодействия между элементами

Модуль алгоритмов взаимодействия между элементами является подпрограммой, изменяющей состояние модуля программы "Структурная модель" системы в зависимости от его текущего состояния. В этом модуле прописаны все триггерные состояния системы, которые приводят к запуску определенных функций в элементах системы.

Структурная модель участка позволяет получать срез состояния моделируемого объекта. Теперь необходимо решить задачу управления этой моделью. Имеем стандартную задачу управления объектом по заданной циклограмме. Циклограмма штатного функционирования системы задается технологами и конструкторами линии. Модуль алгоритмов взаимодействия между элементами системы накладывается на структурную модель, запуская определенные атрибуты элементов в зависимости от циклограммы функционирования участка заливки-охлаждения АФЛ. Для описания циклограммы используется условно-событийный метод. Поясним принцип метода. Как было сказано в описании модуля "Структурная модель", у каждого элемента системы есть атрибуты, моделирующие действие соответствующего объекта участка заливки-охлаждения АФЛ. Проиндексируем все элементы структурной модели. Возьмем первый элемент. Если этот элемент готов к действию, то есть не находится в состоянии "занят", "сбой" или другом состоянии, не связанным с ожиданием команды, то анализируется его окружение. Под окружением понимается замкнутое множество элементов системы, которому принадлежит этот элемент (множество элементов, на которое ссылается данный элемент и все элементы из этого множества). Для каждого атри-

бута, связанного с активностью элемента в системе, распишем условия выполнения, то есть состояния окружения, при которых запускается этот метод. Так проходим по всем методам этого объекта. Запущенные методы приведут к изменению состояния окружения этого объекта. Возьмем второй элемент системы и проведем ту же операцию.

Приоритетность владения ресурсами при таком подходе определяется простой расстановкой очередности проверки условий выполнения методов. Первый проверяемый метод – наивысший приоритет.

Такой подход к реализации управления решает одновременно несколько задач и позволяет:

- легко модифицировать алгоритм функционирования системы;
- моделировать системные отказы, остановки, любые нештатные ситуации. При этом не меняется алгоритм функционирования системы, а лишь меняется состояние самой структурной модели в некоторый момент времени;
- легко внедрять новые элементы в структурную часть модели, не затрагивая уже отработанного алгоритма функционирования.

Блок-схема алгоритма функционирования модуля "Управление" приведена на рис. 4.

В сравнении с жестким описанием циклограммы работы участка появляется возможность экспертам оценить функционирование системы во внештатных ситуациях; это одно из значимых преимуществ данного подхода.

Программа принимает блочный вид, каждый блок которой отвечает за поведение элемента в системе. Следовательно, для изменения структуры моделируемого участка достаточно добавить в структурную модель новый элемент и написать условия выполнения его методов в рассматриваемом модуле. Удаление блоков также не повлияет на алгоритм управления, так как просто сократит окружение некоторых элементов.

Монитор моделирования

Под монитором моделирования понимается механизм, управляющий процессами и ресурсами в модели. Можно выделить основные блоки, которые имеют в том или ином виде все мониторы моделирования [3]: "Управление процессами", "Работа с ресурсами типа "устройство", "Работа с ресурсами типа "память". В этих блоках решаются проблемы генерации очереди (или очередей) запросов (заявок) с привязкой к модельному времени и распределения ресурсов в зависимости от приоритета. Другие функции монитора моделирования, такие как сбор статистики, анализ данных, связь с интерфейсом пользователя и т.п. – альтернативные. Различают мониторы с постоянным и переменным шагом моделирования ΔT .

В силу специфики задачи классический монитор моделирования не удобен. Централизованный механизм генерации заявок и распределения ресурсов сильно усложнял систему и делал ее менее гибкой.



Рис. 5

Также этот механизм трудно реализовать при выбранном методе управления структурной моделью.

Был разработан и предложен вариант распределенного монитора моделирования. Его суть заключается в том, чтобы каждый элемент системы имел собственный механизм контроля ресурсов.

В данной модели ресурсы можно поделить, как и в классическом варианте, на два типа:

- "память" (делимый распределенный в пространстве между элементами ресурс): металл;
- "устройство" (неделимый ресурс): заливочная ветка, ветка охлаждения, передаточная тележка, форма, заливочный ковш.

Механизм распределения ресурсов реализован следующим образом. Как было сказано ранее в описании модуля алгоритмов взаимодействия между элементами, каждый элемент имеет окружение, на которое он ссылается. При запуске метода, который обрабатывает действие какого-либо элемента, происходит передача ресурса связанного с ним элемента. Если это ресурс типа "память", то передача идет с расчетной интенсивностью. Если типа "устройство", то происходит передача за один шаг модели. Получается, что ресурсы передаются по средствам описания алгоритмов их передачи от элемента к элементу внутри методов всех классов.

Созданная модель относится к классу моделей с постоянным шагом ΔT . Такой выбор сделан в силу необходимости моделирования процесса в РВ. Можно реализовать механизм приращения модельного времени в соответствии с реальным, тем самым, решив задачу проигрывания процессов во временном масштабе. Однако механизм отслеживания времени не реализован стандартным методом генерации очереди заявок, а схож с ранее описанным механизмом распределения ресурсов. Механизм реализован следующим образом. В программе есть глобальная переменная ($T_{глоб}$), которая есть счетчик модельного времени. Единица модельного времени соответствует 1 с моделируемого времени. В базовом классе есть указатель на глобальное время, который приравнивается к адресу переменной $T_{глоб}$ при создании нового элемента системы, и переменная $T_{лок}$, соответствующая времени освобождения элемента. Возьмем переменную Q_3 , отвечающую за состояние элемента "занят". Тогда:

$$Q_3 = \begin{cases} 0, & T_{глоб} > T_{лок} \\ 1, & T_{глоб} \leq T_{лок} \end{cases}$$

При запуске какого-либо метода элемента он изменяет $T_{лок}$ на величину, соответствующую времени выполнения задачи $T_{лок} = T_{глоб} + \Delta T$. Структурное представление любого класса из созданной библиотеки изображено на рис. 5.

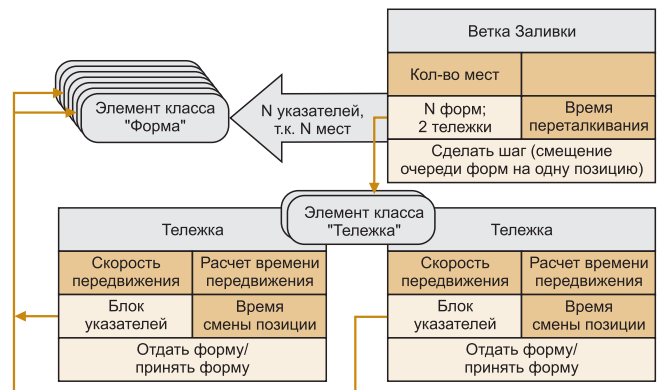


Рис. 6

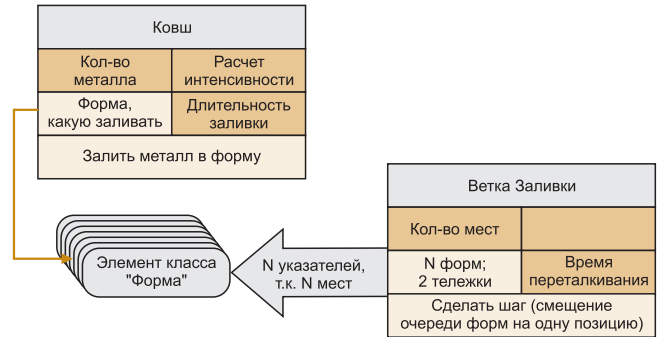


Рис. 7

Рассмотрим примеры связей элементов и работы мониторинговой части каждого элемента.

Пусть необходимо промоделировать смещение очереди форм, находящихся на заливочной ветке, на одну позицию (рис. 6). Запускается метод смещения очереди форм на одну позицию. Проводится проверка допустимости выполнения операции. Если все условия выполнены, то переходим к выполнению операции. Происходит сдвиг адресов *элементов типа форма*, принадлежащих заливочной ветке, на одну позицию влево (элемент $q_i = q_{i+1}$), с добавлением пустого адреса в конец. Адрес первой формы (уходящей с ветки) присваивается *указателю на элемент форма* пустой тележки. Добавленный последний адрес приравнивается значению адреса тележки, с которой уходит форма, а ее *указатель на элемент форма*, приравнивается значению NULL.

Для смещения необходимо 12 с. Следовательно, $T_{лок} = T_{глоб} + 12$ и $Q_3 = 1$ до тех пор, пока переменная $T_{глоб} \leq T_{лок}$. При этом ветка считается занятой.

Так происходит, если рассматриваются методы, работающие с ресурсами типа "устройство". Если же передается ресурс типа "память" (например, моделируется заливка металла в форму – рис. 7), то происходит то же самое, только $T_{лок} = T_{глоб} + 1$. Передаваемое количество ресурса рассчитывается как:

$$W = \int_{T_{глоб}}^{T_{глоб}+1} F_{int}(t) dt,$$

где $F_{int}(t)$ – функция интенсивности передачи ресурса.

Схемы стандартного монитора моделирования и предложенного варианта "распределенного" монитора моделирования представлены на рис. 8.

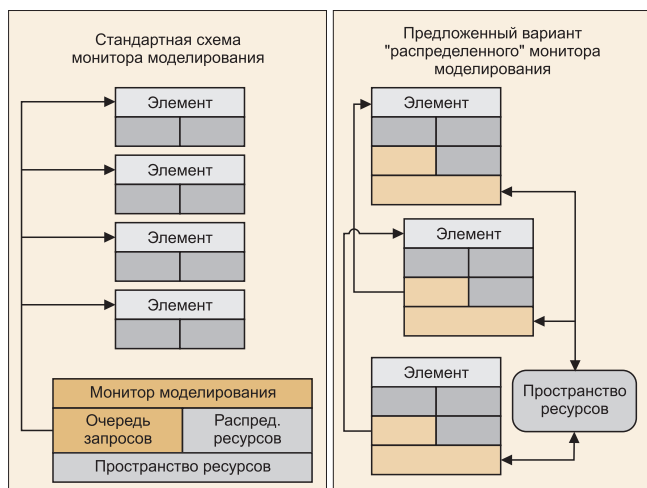


Рис. 8

Такой подход имеет ряд преимуществ.

- Можно моделировать отказ элемента в РВ по желанию оператора. Для этого на каждом шаге с приращением $T_{глоб}$ на единицу происходит приращение $T_{лок}$ всех элементов, помеченных как отказные или стоящие на паузе.

- Можно моделировать работу системы с заданным масштабом времени (ограничение накладывается производительностью ЭВМ на максимально быстрое проигрывание процесса). Тем самым можно ускорять не интересующие этапы и замедлять более важные.

- Нет необходимости генерации очереди запросов на каждом шаге модели.

Из недостатков такого подхода можно выделить только традиционный недостаток моделей с постоянным шагом ΔT – невозможность моделирования процессов с различной точностью во времени.

Интерфейс пользователя

Реализовав структурную модель, модуль алгоритмов взаимодействия между элементами и монитор моделирования, получаем модель, способную автономно функционировать. Такая автономность дает полную свободу выбора средств визуализации и сбора статистики.

Было принято решение делать интерфейсную часть системы при помощи SCADA-системы, а архивирование данных при помощи БД семейства SQL. Выбор обосновывается дальнейшим развитием системы, возможным ее внедрением на производстве, специализацией SCADA-систем под процессы РВ в промышленных областях и удобством создания ЧМИ и основанных на них промышленных тренажерах. Безусловно, можно было бы реализовать интерфейсную часть модели на C++, однако это решение сильно ограничило бы применимость наработок в дальнейшем, так как интерфейс был бы жестко специализирован под данную модель.

Обмен информацией между SCADA, SQL и моделью был сделан, используя реализованные в Windows протоколы обмена данными между процессами.

В данной модели использовался стандартный протокол DDE. Выбор пал на DDE в силу того, что многие SCADA-системы используют именно его для обмена данными с системами управления. А также существуют БД семейства SQL, с реализованным DDE клиентом. В системе использовались SCADA InTouch (WonderWare) и InSQL (WonderWare на базе MSSQL).

Модель отображает результаты моделирования в виде таблиц, гистограмм, числовых значений параметров. Кроме вывода статистических сведений о проведенном эксперименте, существует возможность включить анимацию работы участка заливки-охлаждения. Анимация, реализованная в данной модели, относится к типу совместной анимации (совместная анимация – анимация, которая осуществляется во время прогона имитационной модели). Анимация полезна: 1) при отладке модели; 2) для лучшего понимания технологами и руководством сути процесса моделирования; 3) для наглядного представления различных вариантов работы и компоновки проектируемой системы; 4) при обучении персонала смоделированной системы.

За последнее время возрос интерес руководства промышленных предприятий к промышленным тренажерам, предназначенным для подготовки обслуживающего персонала производственных систем с целью снижения влияния человеческого фактора. Широкое распространение получили системы компьютерного тренинга операторов АСУТП, это стало возможным в связи с бурным развитием информационных технологий, к их числу относятся системы диспетчерского управления и сбора данных (SCADA-системы).

Эффективность оперативного управления процессом зависит от квалификации персонала и его адекватных действий при возникновении нештатных ситуаций, ошибки которого могут привести к материальным потерям, связанным с выпуском некачественной продукции, порчей оборудования и длительным задержкам в ходе производства. Для повышения эффективности работы персонала участка заливки-охлаждения АФЛ на базе описанной выше модели создан промышленный тренажер, предназначенный для обучения персонала эффективным приемам управления и приобретения навыков поведения в нештатных ситуациях. Отображение информации о процессе производится в виде, приближенном к интерфейсам системы управления АФЛ. Обучаемый учится своевременно и адекватно реагировать на нештатные ситуации (например, при задержке в работе участка формовки вовремя переключить участок заливки-охлаждения в аварийный режим работы или при возникновении бракованных форм по вине формовки своевременно пометить их, чтобы они не попадали на участок заливки). Инструктор имеет возможность просмотреть журнал обучения, оценить результаты тренировки и выдать обучающемуся рекомендации.

Заключение

В статье приведены основные принципы создания имитационной модели АФЛ на примере участка заливки-охлаждения АФЛ. Описана структурная модель участка, модуль алгоритмов взаимодействия между элементами, построенный на условно-событийном принципе, предложен вариант "распределенного" монитора моделирования и описаны возможности SCADA-систем для создания интерфейса пользователя, анимации работы участка заливки-охлаждения и построения промышленных тренажеров.

Имитационные модели такого класса применяются ЗАО "Литаформ" для анализа различных компоновок АФЛ на стадии проектирования, выбора наилучшей

структуры АФЛ в зависимости от технологических требований, разработки алгоритмов управления оборудованием, входящим в состав АФЛ и создания промышленных тренажеров.

Список литературы

1. Лоу А., Кельтон В. Имитационное моделирование. СПб: Питер, 2004.
2. Зенькович М.В., Любченко А.С., Древис Ю.Г. Объектно-ориентированное моделирование промышленных участков// Научная сессия МИФИ – 2008. Сб. научных трудов в 15 томах. Т.13. Автоматизированные системы обработки информации и управления. Электронные измерительные системы. М.: МИФИ 2008.
3. Древис Ю.Г., Золотарев В.В. Введение в имитационное моделирование. М.: МИФИ 2002.

*Древис Юрий Георгиевич – д-р техн. наук, проф. кафедры "Управляющие интеллектуальные системы" Московского инженерно-физического института (государственного университета),
Зенькович Михаил Валерьевич – инженер, Любченко Александр Сергеевич – инженер ЗАО "Литаформ".
Контактный телефон (495) 324-34-54. E-mail: ydrevis@ya.ru zmv@litaform.ru*

Открытая модульная платформа с кроссплатой топологии Single Star для протоколов PCI-Express, Gigabit Ethernet и SRIO

Компания Schroff совместно с партнерами разработала компактную модульную систему MicroTCA, в которой реализовано ноу-хау в области механики, топологии кроссплат, термоуправления и моделирования. Выбор конструкции упрощает концепцию охлаждения. Корпус высотой 3U, глубиной 250 мм и шириной 156 мм оснащен четырьмя слотами для одиночных полноразмерных модулей AdvancedMC с вертикальным размещением. При использовании механических адаптеров можно устанавливать среднеразмерные модули AdvancedMC. Для промышленных приложений возможен монтаж системы в вертикальном или горизонтальном положении либо в качестве настольного варианта. Вентиляцию обеспечивают два осевых вентилятора без резервирования и системы управления. Концепция вентиляции оптимальна, так как модули AdvancedMC устанавливаются не в ряд, и каждый отдельный модуль AdvancedMC вентилируется отдельно. Блок питания переменного тока мощностью 150 Вт, устанавливаемый за кроссплатой, обеспечивает электропитание и тепловую мощность 30 Вт для каждого слота AdvancedMC и концентратора MCH. Дополнительно можно приобрести блок питания мощностью 250 Вт (50 Вт на каждый слот).

В системе устанавливаются стандартизированные концентраторы MicroTCA Carrier Hub (MCH), однако модуль питания MicroTCA (PM) не используется. Дополнительно возможно под-

ключение питающего напряжения 12 В с задержкой по времени, чтобы концентратор MCH мог предварительно проверить конфигурацию системы и с помощью электронного ключа коммутировать с модулями AdvancedMC. Дополнительно предлагается простая система управления питанием по шине I2C.

Концентратор MCH объединяет коммутаторы GbE, PCIe или SRIO и синхронизаторы PCIe. Кроссплата с топологией Single Star обеспечивает соединения между модулями AdvancedMC и концентратором MCH. Коммутатор GbE соединяется с нулевыми портами всех модулей AdvancedMC. Коммутатор PCIe с помощью четырех каналов PCIe соединяется с каждым слотом AdvancedMC (порты 4-7) и обеспечивает тем самым достаточную пропускную способность даже для высокоскоростных плат ввода/вывода. Из-за коммутатора PCIe каналы PCIe разъединены друг с другом. Дополнительно можно использовать в системе каналы PCIe с коммутатором Serial Rapid IO. Кроме того, кроссплата оснащается каналами SAS/SATA для подключения жесткого диска AdvancedMC к двум слотам AdvancedMC (AMC, порт 2). Все слоты AdvancedMC этой системы полностью соответствуют спецификации PIGMG AMC.0 R2.0 (базовой спецификации передовой мезонинной платы). В системе используются стандартизированные концентраторы MCH, и поэтому она может применяться в качестве базового решения для приложений MicroTCA.

Более надежная защита и эффективное кондиционирование

При установке оборудования вне помещений компания Schroff предлагает новое поколение шкафов с улучшенной системой кондиционирования стандартных конструкций Modular и Unibody. Основой для версии Modular служит стойка из алюминиевого профиля, позволяющая добиться большой гибкости в отношении габаритов шкафа, а модульная внутренняя обшивка обеспечивает удобный доступ со всех сторон. Основой для недорогой версии Unibody служит цельный алюминиевый внутренний корпус. Такая конструкция идеально подходит для шкафов малых размеров при монтаже на стенах, мачтах и полу.

Шкафы обеих версий изготавливаются с двойными стенками. Детали наружной обшивки помогают уменьшить выделение тепла под воздействием солнечных лучей, а плотно подогнанные детали внутренней обшивки (IP 55) не допускают проникновения влаги, грязи и мелких животных. Воздух в зазоре между стенками движется без принудительной вентиляции за счет прорезей в наружной обшивке. По заказу внутренние стенки шкафов Modular могут поставляться в исполнении "меандр". При этом поверхности, которые имеют большую площадь, отводят большее количество тепла из внутреннего пространства в вентилируемый зазор между двойными стенками. Такая кон-

цепция позволяет более надежно и бесшумно отводить тепло из внутреннего пространства и не требует дополнительного технического обслуживания и затрат энергии. Дополнительные вентиляторы в пространстве между двойными стенками повышают эффективность охлаждения.

Ввод охлаждающего воздуха во внутреннее пространство корпуса еще более расширяет возможности охлаждения. Специальные плоские фильтры с водоотталкивающей поверхностью позволяют достичь степени защиты IP 55. Если этих мер по охлаждению недостаточно, возможна установка теплообменников.

Оба варианта исполнения шкафов обеспечивают высокую гибкость, позволяя устанавливать 19-дюймовые компоненты и нестандартное дополнительное оборудование. Части обшивки поставляются с устойчивым к атмосферным воздействиям порошковым покрытием, а также с покрытием "анти-граффити". Шкафы соответствуют второму классу механической защиты (WK2), съемная крышка обеспечивает легкую установку, а кабельные вводы защищены от проникновения грызунов и насекомых через пол. Эти шкафы, соответствующие IEC 61 969, прошли испытания на ударную и вибрационную стойкость согласно IEC 68-2-6 и -27 и на сейсмическую стойкость согласно IEC 68-2-57.

[Http://www.schroff.ru](http://www.schroff.ru)