

## ИМИТАЦИОННАЯ МОДЕЛЬ ГИДРОАГРЕГАТА ДЛЯ ТЕСТИРОВАНИЯ АЛГОРИТМОВ АСУТП

В.Е. Захарченко (ООО "Сенсоры. Модули. Системы")

Предложено использовать имитационную модель объекта для тестирования алгоритмов системы управления. С целью проверки предложенной методики разработана действующая модель гидроагрегата. Проведена параметризация модели на основании реальных исторических данных SCADA-системы с использованием генетического алгоритма. Реализация имитационной модели позволила проводить изменение и испытание алгоритмов системы управления без реального объекта.

Согласно планам РАО ЕЭС по развитию энергетики в России необходимо решить задачи реконструкции всех систем управления, внедрения новых современных решений на основе ПЛК и система сбора, архивации и отображения данных. Для настройки системы управления и отладки алгоритмов ее работы планируется использовать имитационные модели, на основе которых можно проводить стажировку и обучение персонала. При этом имитационная модель должна полностью имитировать работу системы управления основного оборудования ГЭС [1]. В связи с этим была поставлена задача реализовать имитацию ТП для проверки алгоритмов АСУТП ГЭС при помощи средств моделирования.

На сегодняшний день имитационное моделирование систем занимает одну из ключевых позиций в науке и технике. Под имитационным моделированием традиционно понимается разработка и выполнение на компьютере программной системы, отражающей поведение и структуру моделируемого объекта [2,3,4].

Тестирование алгоритмов систем управления является значительной частью построения АСУТП [5]. Проверка алгоритмов непосредственно на объекте управления зачастую очень сложна из-за технологических или технических ограничений. В этой связи зародилась идея замены реального объекта на упрощенный информационный аналог, который некоторым образом имитирует основные параметры объекта управления, в то время как система управления реагирует на поведение модели, как на поведение реального объекта.

Рассмотрим возможность построения гибридной модели гидроагрегата, поскольку гибридные модели являются эволюцией непрерывного и дискретного подходов.

**Гибридная модель функционирования гидроагрегата**

Гидравлическая турбина преобразует энергию водного потока в механическую энергию вращения ее рабочего колеса. Эта энергия преобразуется гидрогенератором в электрическую. Водный поток ограничивается направляющим аппаратом [5]. Задача: во всех режимах работы гидроагрегата симитировать изменение значений основных параметров управления в интервалах, близких к реальным. К основным параметрам управления относятся частота вращения и положение штока направляющего аппарата.

Моделирование гидроагрегата в общем виде является задачей нетривиальная и сложной, поэтому будем рассматривать исключительно информационное взаимодействие гидроагрегата и системы управления. За

основу функционирования гидроагрегата возьмем алгоритмы системы управления гидроагрегатом Жигулевской ГЭС. В работе гидроагрегата выделены следующие состояния: готов к пуску, пуск, в работе, разгрузка, останов, аварийный останов от защит, разгон, аварийный останов от разгона. Формальное описание модели представляется картой состояний [2, 6, 7, 8].

Основные имитируемые параметры: напор, шток направляющего аппарата, частота вращения, активная мощность, температура подшипника/подпятника, расход воды на смазку турбинного подпятника/подшипника.

Моделирование осуществлялось в среде AnyLogic 5.3.1 (<http://www.xjtek.com/>), разработанной российской компанией "Экс Джей Текнолоджис". Взаимосвязь параметров представлена на рис. 1, где слева показано дерево проекта, состоящее из модели и нескольких экспериментов, в центре в виде направленного графа отображается схема взаимосвязей основных параметров модели и, наконец, в правой части экрана размещается окно со свойствами выбранного элемента (P – активная мощность гидроагрегата).

Обобщенная схема состояний представлена на рис. 2, где в окне свойств отображены формулы для одного из участков режима пуска гидроагрегата.

**Параметризация и проверка адекватности модели**

Параметризация модели – это очень важный этап в моделировании систем. Необходимо привести параметры модели к диапазону изменения реальных параметров объекта; скорости изменения параметров модели должны соответствовать скорости жизни реального объекта.

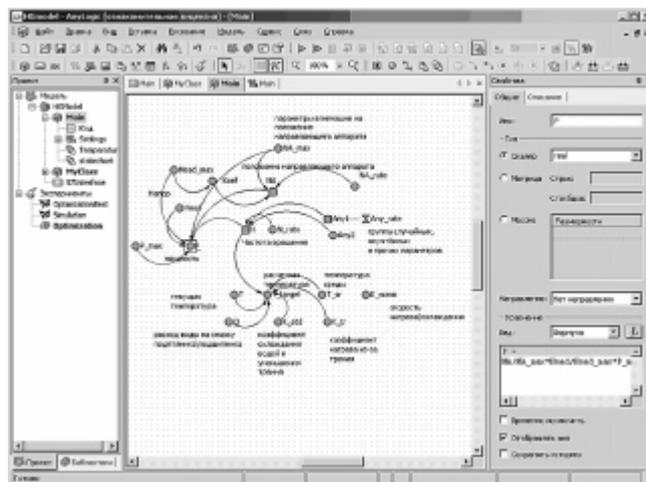


Рис. 1. Взаимосвязь имитируемых параметров

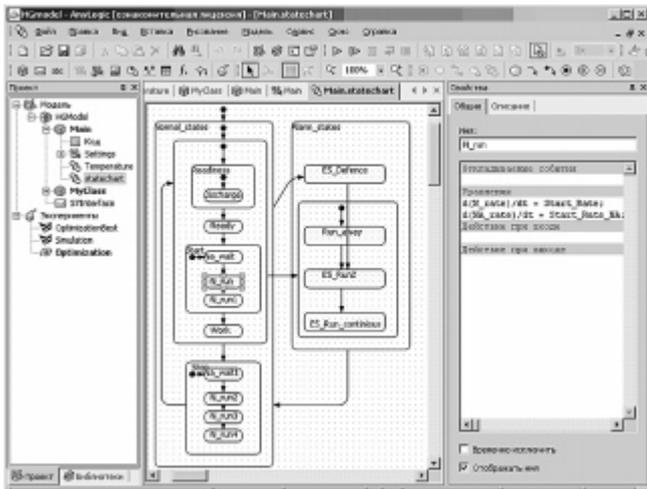


Рис. 2. Обобщенная схема состояний

Параметризация модели проводилась по историческим трендам автоматизированной системы отображения, контроля и управления гидроагрегатами Жигулевской ГЭС.

Во многих случаях для калибровки модели недостаточно визуального соответствия, для более точной настройки решаются оптимизационные задачи.

Оптимизационная задача может быть поставлена различными способами. Рассмотрим оптимизацию по одному параметру. Известно, что на временном интервале  $T$  некоторый параметр изменялся по неизвестному закону  $y(t)$ , но известны значения функции  $y(t)$  в моменты времени  $(t_0, \dots, t_n) \in T$ . Исследователь предполагает, что  $y(t)$  может быть заменена некоторой функцией  $y'(t, x)$ , где  $x$  – вектор коэффициентов. Тогда ошибка (погрешность) имитации параметра на временном интервале  $T$  выражается некоторой функцией  $\overline{f_{err}}$ , основанной на разнице  $y(t)$  и  $y'(t, x)$ . Например, согласно среднеквадратичному критерию ошибка будет представлена функцией

$$\overline{f_{err}} = \sum_{i=0}^n (Y(t_i) - Y'(t_i, x))^2.$$

Назовем  $x^*$  такое решение, при котором отклонение функции имитируемого параметра от функции  $y(t)$  минимально.

Таким образом, получаем задачу вида:

$$\min \overline{f_{err}} \text{ при ограничениях } \begin{cases} G(y'(t_0, x) - y(t_0)) \leq \varepsilon_0 \\ \dots \\ G(y'(t_n, x) - y(t_n)) \leq \varepsilon_n, \end{cases}$$

где  $y'(t_i, x)$  – значение имитируемого параметра в момент времени  $t_i$ ,  $i = \overline{0, n}$ ,  $x$  – набор коэффициентов,  $n$  – число точек, по которым осуществляется моделирование;  $y(t_i)$  – фактическое значение параметра в момент времени  $t_i$ , полученное из архива исторических данных,  $i = \overline{0, n}$ ;  $\varepsilon_i$  – предельное отклонение параметра в момент времени  $t_i$ ;  $\overline{f_{err}}$  – функция выражающая, общую ошибку параметра в рассматриваемом режиме.

Если в ограничениях перенести  $\varepsilon_i$  в левую часть, то получится классическая модель нелинейного программирования.

Для решения задач оптимизации при наличии ограничений используются различные итерационные методы. Методом итерации называют численный метод, который последовательно, шаг за шагом уточняет первоначальное, грубое значение целевой функции [9]. В частности, подобные задачи могут быть решены с помощью метода возможных направлений, метода проекции градиента, метода штрафных функций и др. [10]. Автор решал задачи указанного вида при помощи генетического алгоритма [10], для чего было разработано специальное ПО.

Если оптимальное решение  $x^*$  не может быть достигнуто при заданных ограничениях, необходимо либо сделать новое предположение о виде функции  $y(t)$ , либо разбить исходный временной интервал  $T$  на подынтервалы.

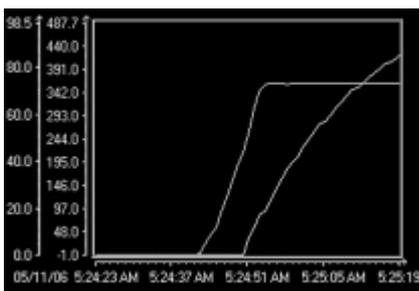
### Проверка адекватности модели

Адекватность модели напрямую зависит от объема исходных данных. Модель признается адекватной, если по всем параметрам, во всех режимах работы выбраны аппроксимирующие функции с такими коэффициентами, что в течение всего времени модельного эксперимента общая ошибка (отклонение от исходных данных) всех параметров не превышает некоторой наперед заданной величины. В противном случае необходимо увеличить число состояний (режимов) модели, после чего вновь провести параметризацию.

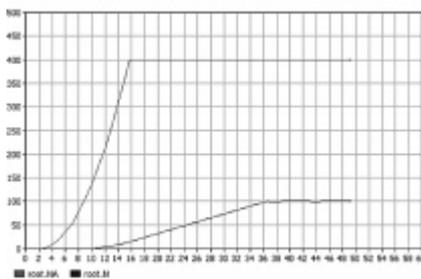
На рис. 3 приведены снимки экранов в режиме "Пуск": а) из архива SCADA-системы; б) в режиме исполнения модели; в) наборы данных из модели перенесены в Excel. На графиках б) и в) по осям абсцисс откладывается модельное время, шаг которого соответствует 1 секунде.

### Модель в режиме исполнения

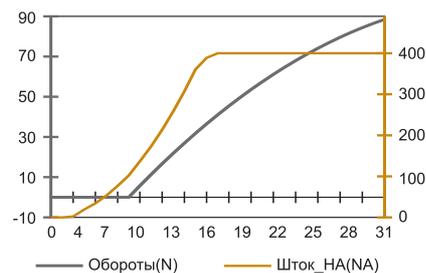
Основное окно визуализации работы модели содержит основные параметры, по которым можно составить представление о ходе ТП (рис. 4). Окно разделено на две части: в одной расположены параметры для настройки ключевых параметров, в другой – непосредственно параметры имитируемого объекта – гидроагрегата. К параметрам настройки относятся коэффициенты, характеризующие скорость нарастания/убывания оборотов турбины, скорость изменения положения штока направляющего аппарата, расход воды на смазку подшипника, скорости изменения температуры подпятника/подшипника. Область "Процесс" содержит такие параметры, как обороты турбины, положение штока направляющего аппарата, активная мощность, усредненная температура подпятника/подшипника. Кроме того, в этой области представлены три дискретных параметра, которые участвуют в алгоритмах системы управления: выключатель, реле гашения поля и тормоз.



a)



b)



c)

Рис. 3. Параметризация модели

Управление моделью осуществляется событийно. Источником события может быть оператор модели или система управления гидроагрегатом. Оператор модели в режиме исполнения может изменять численное значение параметров, корректировать коэффициенты функций, а также нажимать на кнопки управления, размещенные в области настроек. Система управления гидроагрегатом создает событие для модели, изменяя состояние выходов.

На рис. 4 симитирован режим пуска гидроагрегата: происходит открытие направляющего аппарата, вода попадает на лопасти турбины и она плавно раскручивается, вместе с этим происходит нарастание активной мощности, при этом выключатель и реле гашения поля включены, а тормоз выключен. Отдельно рассчитывается температура подпятника/подшипника. С увеличением оборотов температура начинает расти и через некоторое время достигнет точки  $T_{стремления}$ . При этом расход на смазку равен нулю.

На рис. 5 представлен снимок экрана в процессе останова модели гидроагрегата. На текущий момент "гидроагрегат" почти остановлен, выключатель и реле гашения поля отключены, работают тормозные колодки. Температура начинает медленно снижаться. Как только скорость изменения оборотов станет равной нулю и "гидроагрегат" остановится, расчетная температура  $T_{стремления}$  будет равна температуре окружающей среды.

**Взаимодействие с системой управления гидроагрегатом**

Система контроля и управления гидроагрегатом на Жигулевской ГЭС реализована на базе ПЛК SIMATIC S7 400 фирмы Siemens. В системе контролируются около 250 параметров. Данные со всех 20 гидроагрегатов и других систем архивируются и отображаются в SCADA-системе WinCC 6.0.

В программе ПЛК реализована возможность подмены значения с датчика на заданное оператором (или другой системой) по любому параметру.

Среда имитационного моделирования AnyLogic полностью реализована на Java и позволяет подключать динамические библиотеки, написанные на других языках программирования. Для связи с контроллером фирмы Siemens S7-400 была написана динамическая библиотека на языке C++, использующая SIMATIC PRODAVE. Функции динамической библиотечки вызываются из модели периодически, они осуществляют запись в контроллер системы управления имитируемых параметров, а также считывают состояние управляющих сигналов контроллера и транслируют их в имитационную модель.

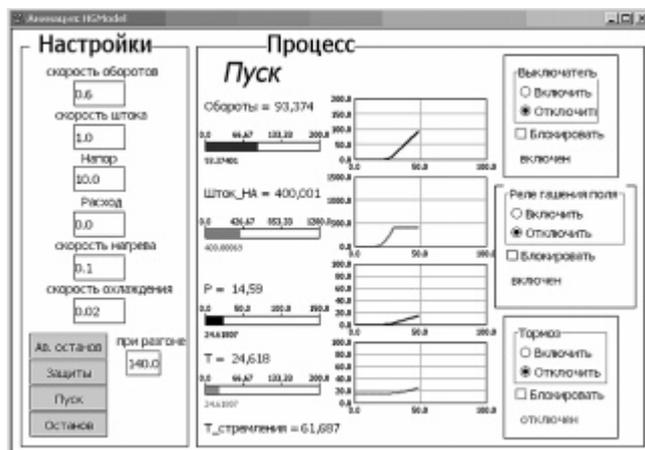


Рис. 4. Основное окно визуализации исполнения модели

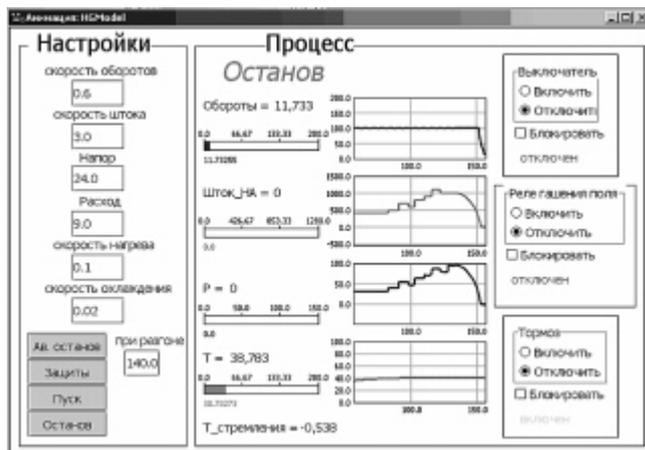


Рис. 5. Основное окно в режиме "Останов"

**Заключение**

На основе средств имитационного моделирования разработана действующая модель гидроагрегата. Проведена параметризация модели на основании реальных исторических данных SCADA-системы WinCC с использованием генетического алгоритма. Для проверки идентифицированной модели на адекватность были адаптированы классические методы имитационного

моделирования. На примере показана работа методики параметризации и проверки модели на адекватность. Реализовано взаимодействие разработанной модели с системой управления гидроагрегатом. Изменение параметров разработанной модели соответствует скорости жизни реального объекта (гидроагрегата), что позволяет проверять алгоритмы системы управления.

Проверка алгоритмов системы управления с помощью имитационной модели осуществляется при помощи графического интерфейса: достаточно выбрать режим и задать коэффициенты. При этом состояние модели объекта наглядно отображается на экране компьютера.

Преимущества проверки алгоритмов с помощью имитационной модели: инвариантность по отношению к платформе и средствам реализации системы управления (ОС РВ, ПЛК); наглядность и простота.

Реализация имитационной модели позволила проводить изменение и испытание алгоритмов системы управления без реального объекта. Кроме того, открывается широкий спектр перспектив: на основе разработанной модели можно не только проверять алгоритмы управления, но и оценивать их качество. Учитывая оценки качества управления в различных ситуациях можно наметить участки модернизации системы управления. В дополнение ко всему перечисленному имитационная модель может использоваться для обучения персонала. Достоинства обучения персонала с помощью имитационной модели: доступность средств, простота, высокая скорость обучения.

#### Список литературы

1. *Самхарадзе Р.Г.* Реконструкция системы управления на действующих ГЭС // Новое в российской электроэнергетике. 2004. № 9.
2. *Карпов Ю.Г.* Имитационное моделирование систем. Введение в моделирование с AnyLogic 5. СПб.: БХВ-Петербург, 2006.
3. *Бенькович Е.С., Колесов Ю.Б., Сениченков Ю.Б.* Практическое моделирование динамических систем. СПб.: "БХВ-Петербург". 2002.
4. *Колесов Ю.Б., Сениченков Ю.Б.* Имитационное моделирование сложных динамических систем [http://www.exponenta.ru/soft/others/mvs/ds\\_sim.asp](http://www.exponenta.ru/soft/others/mvs/ds_sim.asp)
5. Советов Б.Я., Яковлев С.А. Моделирование систем. М.: Высш. Шк., 1985.
6. Кожевников Н.Н. Устройство и эксплуатация оборудования гидроэлектростанций. М.: Высш. шк., 1985.
7. *D.Harel* Statecharts: A Visual Formalism for Complex Systems. Science of Computer Programming, North-Holland, Vol.8, No.3, 1987.
8. *Booch G., Jacobson I. and Rumbaugh J.* The Unified Modeling Language for Object-Oriented Development, Documentation Set Version 1.1, September 1997.
9. *Браун Селик (Bran Selic), Джим Румбаух (Jim Rumbaugh)* Использование UML при моделировании сложных систем РВ. <http://ooad.asf.ru/standarts/uml/umlsystem/index.asp>
10. *Коварцев А.Н.* Численные методы. Уч. пособие. Самарский муниципальный комплекс непрерывного образования. Самара. 1998.
11. *Лю Б.* Теория и практика неопределенного программирования. М.: БИНОМ. Лаборатория знаний, 2005. (Адаптивные и интеллектуальные системы).

*Захарченко Виталий Евгеньевич – инженер ООО "Сенсоры. Модули. Системы".  
Контактный телефон (846)269-15-20. E-mail: zevs@sms-samara.ru*