

УДК 621.311.1:681.51

СОПРЯЖЕНИЕ ПРОГРАММНЫХ СРЕД В ЗАДАЧАХ МОДЕЛИРОВАНИЯ И ТЕСТИРОВАНИЯ СИСТЕМ УПРАВЛЕНИЯ ЭНЕРГЕТИЧЕСКИМИ ГАЗОТУРБИНЫМИ УСТАНОВКАМИ

И. А. Шмидт,

канд. техн. наук, доцент

Б. В. Кавалеров,

канд. техн. наук, доцент

К. А. Один,

программист

Пермский государственный технический университет

А. А. Шигапов,

начальник отдела автоматизации ЗАО «Энергокомплект-Пермь»

Приводится решение задачи сопряжения программных сред для программной реализации разработанной математической модели газотурбинной мини-электростанции. Математическая модель предназначена для тестирования систем управления энергетическими газотурбинными установками.

Ключевые слова — электроэнергетическая система, газотурбинная установка, автоматическое управление.

Введение

Предприятия отечественного авиационного двигателестроения, решая задачи конверсии и диверсификации производства, уделяют значительное и возрастающее внимание разработке газотурбинных установок (ГТУ) наземного применения. В частности, только для нужд электроэнергетики на предприятиях Российской Федерации сегодня разрабатывается и эксплуатируется до 20 типоразмеров малых ГТУ мощностью от 1 до 25 МВт [1]. Новые, резко отличные от полетных, условия функционирования ГТУ предъявляют повышенные и специфические требования к их системам автоматического управления (САУ), которые, тем не менее, до настоящего времени в основном продолжают строиться на базе чисто авиационных прототипов без учета особенностей и характеристик электрической нагрузки. Как результат снижается эффективность принимаемых на этапах проектирования решений, что, в свою очередь, приводит к серьезным препятствиям в обеспечении удовлетворительных эксплуатационных характеристик конечных из-

делий. Основную причину следует искать в отсутствии адекватных программно-алгоритмических средств, обеспечивающих решение задач комплексной автоматизации проектирования и отладки САУ наземных энергетических ГТУ. В статье обсуждаются возможности решения данной проблемы на основе создания гибких компьютерных средств тестирования и наладки алгоритмов САУ ГТУ. С этой целью анализируется математическое описание и исследуется решение задачи сопряжения программных сред разработки алгоритмического аппарата.

Программно-моделирующие комплексы для тестирования и отладки алгоритмического обеспечения САУ ГТУ

Современные энергетические ГТУ представляют собой достаточно сложные для управления объекты. Во-первых, сегодня для наземного применения используются главным образом ГТУ, выполненные по двухвальтовой схеме, что в сравнении с одновальными ГТУ обеспечивает повышенную экономичность и позволяет свободно выби-

рять частоту вращения отдельных блоков [2]. Однако в то же время данная кинематическая схема не способствует улучшению динамических характеристик. Основными причинами затягивания переходных процессов по частоте, например при резких изменениях электрических нагрузок, являются наличие газовой связи между валами ГТУ и их малый момент инерции. Во-вторых, для обеспечения требуемого качества вырабатываемой электроэнергии, в значительной мере определяемого характером переходных процессов установки, требуется дальнейшее совершенствование систем и алгоритмов управления ГТУ. При решении этой задачи ГТУ должны рассматриваться во взаимодействии с другими подсистемами и узлами, для энергетических ГТУ — это, прежде всего, электрогенератор и электроэнергетическая система (ЭЭС) в целом. К сожалению, до настоящего времени разработка и проектирование САУ ГТУ, САУ синхронного генератора производятся в значительной степени независимо, и их объединение в рамках единой мини-электростанции происходит лишь на завершающих стадиях проектирования. При этом при проектировании и отладке САУ ГТУ не учитывается или учитывается крайне приближенно влияние ЭЭС на процессы регулирования ГТУ, то же самое следует сказать и о проектировании САУ генератора. В результате на практике часто возникают аварийные режимы по причине отсутствия согласования алгоритмов функционирования локальных САУ. Для решения указанной проблемы на самых ранних этапах проектирования разработчик должен иметь возможность исследовать поведение ГТУ в качестве модуля единой сложной взаимосвязанной системы. Подобный подход обеспечивает условия согласования характеристик подсистем и локальных САУ и прежде всего САУ ГТУ и САУ синхронного генератора. По этим же причинам разработка, тестирование и отладка алгоритмов САУ ГТУ должны выполняться на основе математического моделирования основных элементов и взаимодействующих подсистем для последующей поэтапной проверки правильности принимаемых решений.

Таким образом, для проектирования, тестирования и отладки алгоритмического обеспечения САУ ГТУ необходимо создать адекватную математическую модель, обеспечивающую моделирование ГТУ во взаимодействии с синхронным генератором и другими структурными элементами ЭЭС. Задачи моделирования ГТУ, структурных элементов ЭЭС, САУ, а также задачи настройки и коррекции алгоритмов САУ целесообразно объединять для совместного исполнения в рамках программного моделирующего комплекса [3]. Подобные гибкие программные моделирующие ком-

плексы являются многофункциональными, они предназначены для комплексного решения задач моделирования, проектирования, тестирования и отладки алгоритмического обеспечения САУ энергетических ГТУ при учете динамических процессов не только собственно в ГТУ, но и в электрогенераторах и прочих элементах ЭЭС. Предложенная концепция совместного моделирования ГТУ и ЭЭС в задачах тестирования и отладки алгоритмов САУ ГТУ хорошо согласуется с современными подходами к управлению энергетическими объектами, обуславливает возможность применения современных методов синтеза и оптимизации САУ [4, 5].

Особенности математического моделирования ГТУ в составе программно-вычислительного комплекса

В рамках программного моделирующего комплекса энергетические ГТУ моделируются совместно с разнообразными элементами ЭЭС. При этом имеют место следующие главные проблемы.

1. Необходимость моделирования различных конфигураций газотурбинных электростанций, различающихся по числу энергетических установок и вариантам возможных схем электроснабжения.

2. Необходимость сопряжения программных сред моделирования, что обусловлено объективными причинами. Математическая модель ГТУ является достаточно сложной, она учитывает геометрию основных сечений установки, вследствие чего ее реализация возможна только при связном воспроизведении всех выходных координат ГТУ, используемых в контурах регулирования. Модель, обеспечивающая такое связное воспроизведение всех координат, строится путем объединения моделей, описывающих различные узлы двигателя или различные процессы в нем. Именно такая модель применяется для доводки и отладки специализированного программного обеспечения САУ [6]. Указанные математические модели ГТУ достаточно давно разработаны и сертифицированы отечественными предприятиями авиационного двигателестроения, вследствие чего в настоящее время актуальна задача сопряжения программных сред создания и исполнения математических моделей ГТУ в составе гибких программно-вычислительных комплексов.

Для разрешения первой проблемы необходима универсальная методика построения сложной математической модели ЭЭС с возможностями ее реконфигурации. Математическая модель ЭЭС включает в себя большое количество взаимосвязанных линейных и нелинейных дифференциальных и алгебраических уравнений; она также

должна учитывать изменения в топологии, в направлении передачи энергии и отвечать ряду других разнообразных, зачастую противоречивых требований.

Сложная математическая модель состоит из моделей структурных элементов и модели их взаимодействия [7]. Для того чтобы представить математические модели всех структурных электрически-взаимодействующих элементов в единой обобщенной форме, представим математическую модель каждого структурного элемента в координатах Парка—Горева, а затем преобразуем математические модели элементов к единообразной векторно-матричной форме записи.

Математические модели всех структурных элементов ЭЭС представим в виде следующего векторно-матричного уравнения:

$$p\mathbf{I} = \pm \mathbf{A}\mathbf{U} - \mathbf{B}\mathbf{I} - \mathbf{H}, \quad (1)$$

где p — символ дифференцирования; \mathbf{I} , $p\mathbf{I}$ — вектор токов и вектор производных токов элемента; знак « \pm » означает, что структурный элемент является источником электрической мощности (генератором), знак « $+$ » — потребителем; \mathbf{U} — вектор напряжений, приложенных между внешними зажимами элемента; \mathbf{A} , \mathbf{B} — матрицы, размерность которых зависит от системы координат, в которых моделируется структурный элемент, а также от того, полные это уравнения или упрощенные; \mathbf{H} — вектор, определяющий воздействие на элемент со стороны средств регулирования электрических параметров. Компоненты вектора \mathbf{H} , связанные с внешним воздействием на элемент, для пассивных элементов электрической системы нулевые, для синхронных машин меняются на каждом шаге расчета.

При реализации динамической модели уравнения (1) решаются на каждом шаге расчета методом численного интегрирования относительно внешних токов структурного элемента. При этом следует отметить, что для пассивных элементов выражение (1) содержит всю систему дифференциальных уравнений элемента. Для электрических машин уравнения (1) следует интегрировать совместно с уравнениями роторных контуров; уравнениями, описывающими механическое состояние элемента, и уравнениями, описывающими действие систем автоматического регулирования. Знаки при слагаемых уравнения (1) соответствуют нормальному направлению токов в электрических машинах (от генератора к узлу). Математическую модель для расчета установившегося режима получаем из уравнений (1), принимая $p = 0$.

В качестве примера рассмотрим процесс получения уравнения вида (1) для наиболее ответ-

ственного элемента ЭСС — синхронного генератора. Система дифференциальных уравнений синхронного генератора представлена в относительных единицах в системе вращающихся координат Парка—Горева (в d -, q -осях) с учетом общепринятых допущений в следующем виде:

$$\left\{ \begin{array}{l} U_d = -\Psi_q \omega - d\Psi_d / dt - I_d r \\ U_q = \Psi_d \omega - d\Psi_q / dt - I_q r \\ U_f = d\Psi_f / dt + I_f r_f \\ 0 = d\Psi_D / dt + I_D r_D \\ 0 = d\Psi_Q / dt + I_Q r_Q \\ d\omega / dt = \frac{1}{T_j} (M_T - M) \\ d\gamma / dt = \omega \\ d\delta / dt = \omega - \omega_0 \\ M = \Psi_d I_q - \Psi_q I_d \end{array} \right. , \quad (2)$$

где U_d , U_q — напряжение на обмотке статора по продольной и поперечной осям; Ψ_d , Ψ_q — потокосцепления статора по продольной и поперечной осям; ω — угловая частота вращения ротора; I_d , I_q — токи статора по продольной и поперечной осям; r — активное сопротивление обмотки статора; U_f — напряжение обмотки возбуждения; Ψ_f , I_f , r_f — потокосцепление, ток и сопротивление обмотки возбуждения; Ψ_D , Ψ_Q — потокосцепления демпферных контуров по продольной и поперечной осям; I_D , I_Q — токи демпферных контуров по продольной и поперечной осям; r_D , r_Q — активные сопротивления демпферных контуров по продольной и поперечной осям; T_j — постоянная времени механической инерции электрической машины; M_T — момент турбины; M — электромагнитный момент генератора; γ — угол поворота вращающихся осей d , q по отношению к неподвижным осям; δ — внутренний угол генератора (угол нагрузки); ω_0 — угловая частота вращения поля статора.

При переходе к относительным единицам в качестве базовых приняты номинальные значения соответствующих величин. Используя известные нелинейные соотношения между потокосцеплениями и токами, после ряда громоздких, но несложных преобразований получаем описание синхронного генератора в форме (1), где векторы формируются следующим образом:

$$p\mathbf{I} = \begin{pmatrix} dI_d / dt \\ dI_q / dt \end{pmatrix}$$

— вектор производных статорных токов;

$$\mathbf{I} = \begin{pmatrix} I_d \\ I_q \\ I_f \\ I_D \\ I_Q \end{pmatrix} \text{ — вектор токов; } \mathbf{U} = \begin{pmatrix} U_d \\ U_q \end{pmatrix} \text{ — вектор статорных напряжений; } \mathbf{H} = \begin{pmatrix} x_D x_{ad} - x_{ad}^2 U_f \\ x_f x_D - x_{ad}^2 \\ 0 \end{pmatrix},$$

где x_f, x_{ad}, x_D — индуктивные сопротивления обмотки возбуждения, генератора по продольной оси и демпферной обмотки по продольной оси.

Вслед за математическими моделями отдельных элементов необходимо построить математическую модель их взаимодействия, которая должна отражать состав элементов и топологию ЭЭС. Математическая модель взаимодействия структурных элементов использует матрицу инцидентности, отражающую топологию рассматриваемой локальной системы электроснабжения. Представим взаимодействие элементов в виде следующего векторно-матричного уравнения для определения узловых напряжений на каждом шаге расчета:

$$\mathbf{MGM}^T \mathbf{U} = -\mathbf{M}\mathbf{W} - \mathbf{M}'\mathbf{I}, \quad (3)$$

где \mathbf{M} — клеточная матрица инцидентности, клетками матрицы являются единичные, нулевые матрицы или матрицы преобразований координат; \mathbf{G} — блочная матрица проводимостей ветвей (элементов), образующих систему; \mathbf{M}^T — транспонированная матрица инцидентности; \mathbf{U} — вектор искомых напряжений узлов; \mathbf{W} — вектор, полученный из правых частей уравнений элементов в форме (1); \mathbf{M}' — матрица, элементами которой являются нулевые клетки или клетки производных элементов матриц преобразования координат.

Матрица \mathbf{G} составляется из матриц \mathbf{A}_i всех элементов моделируемой системы электроснабжения, записанных в форме (1), матрица \mathbf{W} составляется из правых частей уравнений элементов в форме (1):

$$\mathbf{G} = \begin{pmatrix} \mathbf{A}_1 & 0 & \dots & 0 \\ 0 & \mathbf{A}_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{A}_i \end{pmatrix}; \quad \mathbf{W} = \begin{pmatrix} -\mathbf{B}_1 \mathbf{I}_1 - \mathbf{H}_1 \\ -\mathbf{B}_2 \mathbf{I}_2 - \mathbf{H}_2 \\ \vdots \\ -\mathbf{B}_i \mathbf{I}_i - \mathbf{H}_i \end{pmatrix}.$$

Полученная математическая модель обеспечивает расчет всех узловых напряжений системы электроснабжения по векторно-матричному уравнению (3) на текущем шаге расчета. Вслед за этим вновь решаются дифференциальные уравнения элементов (1) и находятся мгновенные значения токов каждого элемента. Процесс повторяется

требуемое число шагов в течение всего расчета. В результате рассчитываются переходные процессы всех режимных параметров ЭЭС.

Рассмотрим функциональность представленной методики на примере топологически простой системы (рис. 1), где нет задающего узла и два генератора работают на общую линию связи. Положительное направление тока для каждой линии показано стрелками.

Уравнения элементов: генератор 1:

$$p\mathbf{I}_1 = -\mathbf{A}_1 \mathbf{U}_1 - \mathbf{B}_1 \mathbf{I}_1 - \mathbf{H}_1; \quad (4)$$

генератор 2:

$$p\mathbf{I}_2 = -\mathbf{A}_2 \mathbf{U}_2 - \mathbf{B}_2 \mathbf{I}_2 - \mathbf{H}_2; \quad (5)$$

линия связи:

$$p\mathbf{I}_л = \mathbf{A}_л \mathbf{U}_{12} - \mathbf{B}_л \mathbf{I}_л. \quad (6)$$

Уравнение (3) в этом случае будет выглядеть следующим образом:

$$\begin{pmatrix} -1 & 0 & \mathbf{C}_{12} \\ 0 & -1 & -1 \end{pmatrix} \begin{pmatrix} \mathbf{A}_1 & 0 & 0 \\ 0 & \mathbf{A}_2 & 0 \\ 0 & 0 & \mathbf{A}_л \end{pmatrix} \begin{pmatrix} -1 & 0 \\ 0 & -1 \\ \mathbf{C}_{21} & -1 \end{pmatrix} \begin{pmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \end{pmatrix} = - \begin{pmatrix} -1 & 0 & \mathbf{C}_{12} \\ 0 & -1 & -1 \end{pmatrix} \begin{pmatrix} -\mathbf{B}_1 \mathbf{I}_1 - \mathbf{H}_1 \\ -\mathbf{B}_2 \mathbf{I}_2 - \mathbf{H}_2 \\ -\mathbf{B}_л \mathbf{I}_л \end{pmatrix} + \begin{pmatrix} 0 & 0 & \dot{\mathbf{C}}_{12} \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{I}_1 \\ \mathbf{I}_2 \\ \mathbf{I}_л \end{pmatrix}, \quad (7)$$

где $\mathbf{1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ — единичная матрица; $\mathbf{0} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$ — нулевая матрица; матрица преобразования координат

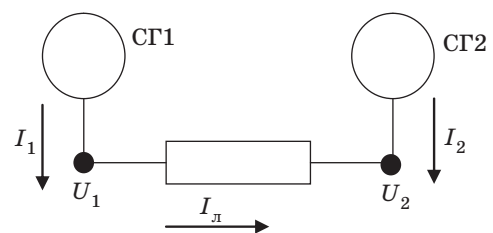
$\mathbf{C}_{12} = \begin{pmatrix} \cos \delta_{12} & -\sin \delta_{12} \\ \sin \delta_{12} & \cos \delta_{12} \end{pmatrix}$; $\dot{\mathbf{C}}_{ij}$ — матрица производных по времени элементов матрицы \mathbf{C}_{ij} :

$$\dot{\mathbf{C}}_{12} = \begin{pmatrix} -\dot{\delta}_{12} \sin \delta_{12} & -\dot{\delta}_{12} \cos \delta_{12} \\ \dot{\delta}_{12} \cos \delta_{12} & -\dot{\delta}_{12} \sin \delta_{12} \end{pmatrix}; \quad \delta_{12} \text{ — угол между } d\text{-, } q\text{-осями первого и второго генераторов.}$$

Алгоритм расчета для примера на рис. 1 следующий.

1. Задаются параметры и начальные значения переменных всех элементов системы — генераторов и линии связи.

1. Задаются параметры и начальные значения переменных всех элементов системы — генераторов и линии связи.



■ Рис. 1. Два генератора работают на общую линию: СГ — синхронный генератор

2. По выражению (7) за один шаг рассчитываются векторы U_1 и U_2 — напряжения первого и второго узлов ЭЭС.

3. Решением дифференциальных уравнений элементов (4)–(6) определяются значения токов, угловых скоростей и других переменных на первом шаге расчета.

4. По вычисленным значениям токов по уравнению (7) рассчитываются новые значения векторов U_1 и U_2 для следующего шага расчета.

5. Процедура повторяется в течение заданного числа шагов.

Конфигурация моделируемой системы электроснабжения модифицируется за счет внесения изменений в матрицу инцидентности M уравнения (3), благодаря чему пользователь имеет возможность формировать различные конфигурации моделируемых ЭЭС.

Рассмотренный выше алгоритмический аппарат может быть проиллюстрирован структурной схемой (рис. 2). На схеме обозначено: САУ ЭС — система автоматического управления электростанцией; $P_{CG_1} \dots P_{CG_n}$ — механические мощности синхронных генераторов (входные переменные моделей ГТУ); $\omega_1 \dots \omega_n$ — угловые скорости свободных турбин (выходные переменные ГТУ). Математическая модель взаимодействия структурных элементов использует для вычисления узловых напряжений токи элементов, подсчитанные на предыдущем шаге расчета. Узловые напряжения являются входными переменными для математических моделей элементов ЭЭС (синхронные генераторы, трансформаторы, линии связи, статические нагрузки и т. п.), которые вычисляют токи для следующего шага расчета. Дополнительные входы представляют собой дополнительные входные переменные элементов ЭЭС, например, для синхронных генераторов — это напряжения возбуждения, вырабатываемые системами возбуж-

дения и автоматическими регуляторами возбуждения (АРВ). Поскольку электростанция может включать в себя несколько энергоблоков (несколько ГТУ), необходимо предусмотреть отдельную САУ ЭС, выполняющую общие координирующие и согласующие функции. Как следует из представленной схемы, математические модели ГТУ должны поддерживать непрерывный обмен данными с остальными компонентами системы моделирования. И здесь разработчик сталкивается с необходимостью разрешения второй проблемы — проблемы сопряжения программных сред моделирования.

Решение задачи сопряжения программных сред

Как правило, используемые в настоящее время многорежимные математические модели ГТУ различного класса сложности созданы с помощью языка программирования Fortran, что обусловлено ориентацией этого языка на вычисления. С другой стороны, программное обеспечение программно-вычислительного комплекса целесообразно разрабатывать на основе современных программных средств, например, используя Delphi. Необходимость включать модели ГТУ в состав программно-вычислительного комплекса потребовала разработать специальную методику сопряжения программных сред. Рассмотрим суть предлагаемой методики.

Использование модулей, написанных на языке Fortran, из приложения, написанного на Delphi, осуществляется в два шага. Каждый из этих шагов рассмотрим детально.

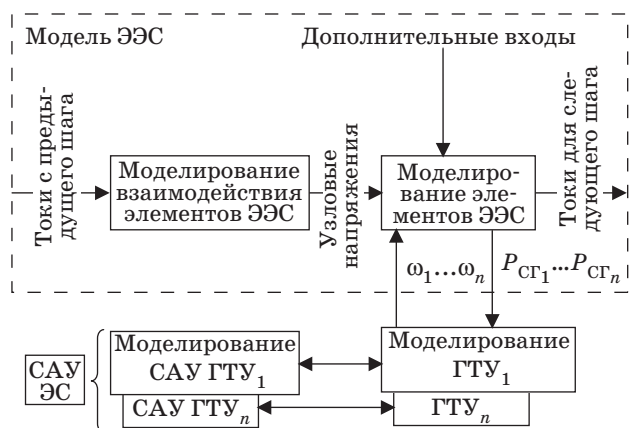
Первый шаг — правила оформления функции на языке Fortran в виде, допускающем их использование.

Рассмотрим вызов фортрановской функции SUBROUTINE GTD(INI)

Единственный способ вызывать функцию, написанную на языке Fortran, это оформить ее как DLL. Для совместимости с программной средой Delphi необходимо выполнить следующие действия.

1. Явно указать соглашение о вызовах и экспортировать имя функции
`cDEC$ ATTRIBUTES DLLEXPORT::GTD`

2. Для всех параметров и переменных указать их тип, совместимый с типами Delphi. Типы Integer совпадают в обоих языках, а типу Double в Delphi будет соответствовать REAL*8 в Fortran. Для того чтобы гарантировать правильный способ описания всех переменных, рекомендуется в начале модуля вставить строчку
`IMPLICIT NONE`



■ Рис. 2. Совместное математическое моделирование ГТУ и ЭЭС

3. Принять меры для доступа к COMMON-блокам. Для этого необходимо, во-первых, сделать их имена доступными, т. е. экспортировать их имена так же, как имя функции:

```
cDEC$ ATTRIBUTES DLLEXPORT:: /D/
cDEC$ ATTRIBUTES DLLEXPORT:: /REZ1/
```

во-вторых, описать явно переменные, входящие в COMMON-блоки. Так, вместо описания

```
COMMON/D/ DG,DGF,DF,DH,DM,DP,DDP,DT,DDT,DFK,J1,J2
COMMON/REZ1/ CP1(16),CT1(16),CT2(16),FMH(60),REG(40),CA(100),
*CN1(16),CN2(16)
```

получим

```
COMMON/D/ DG,DGF,DF,DH,DM,DP,DDP,DT,DDT,DFK,J1,J2
COMMON/REZ1/ CP1,CT1,CT2,FMH,REG,CA,CN1,CN2
REAL*8 DG,DGF,DF,DH,DM,DP,DDP,DT,DDT,DFK
INTEGER J1,J2
REAL*8, DIMENSION(16) :: CP1,CT1,CT2
REAL*8, DIMENSION(60) :: FMH
REAL*8, DIMENSION(40) :: REG
REAL*8, DIMENSION(100) :: CA
REAL*8, DIMENSION(16) :: CN1,CN2
```

4. Те же самые правила нужно применить для всех функций и процедур, вызываемых из головной функции.

5. Особые преобразования нужно применять, если в исходном тексте использовалась встроенная функция, параметрами которой являются числа с плавающей точкой, отличной от REAL*8. Например, если в исходном тексте использовалась функция ALOG, параметром и результатом которой является REAL*4, необходимо проделать следующие преобразования. Пусть имеем исходный текст

```
I2=I2*ALOG(T2/T1),
переменные T2, T1 описываются по умолчанию.
После преобразования получаем следующий фрагмент:
REAL*8 T2, T1
REAL*4 TMP
TMP = T2/T1
I2=I2*ALOG(TMP)
```

После трансляции и построения библиотеки DLL ее нужно разместить в том же каталоге, что приложение на Delphi, которое будет ее использовать. В рассматриваемом случае имя библиотеки «_GTD.dll»

Второй шаг — вызов процедур и функций, написанных на языке Fortran из Delphi-программы.

Для выполнения функции, находящейся в DLL, нужно, во-первых, импортировать эту функцию, описав ее прототип:

```
procedure GTD(var INI: Integer); cdecl; external '_GTD.dll';
```

обратим внимание, что параметры необходимо описывать как параметры-переменные, что совпадает с соглашениями Fortran.

Второй этап более неочевиден — следует обеспечить доступ из программы к переменным,

описанным в общих блоках библиотек. Для этого требуется совершить следующие действия.

1. Импортировать имена общих блоков. Единственный способ импортировать такие имена — это описать их как функции, тем самым «обмануть» транслятор и получить фактически адрес общего блока внутри dll. Для общих блоков D и REZ1 получим описание

```
function D(i :Integer):Integer; cdecl; external '_GTD.dll';
function REZ1(i :Integer):Integer; cdecl; external '_GTD.dll'.
```

2. Для доступа к переменным, входящим в общий блок, нужно подготовить структуры, соответствующие общим блокам. Для этого опишем типы, соответствующие импортированным общим блокам (типы, соответствующие блокам D и REZ1, будут иметь названия T_D и T_REZ1):

```
type
T_D = record
    DG,DGF,DF,DH,DM,DP,DDP,DT,DDT,DFK : Double;
    J1,J2: Integer;
end;
T_REZ1 = record
    CP1,CT1,CT2:ARRAY [1..16] of Double;
    FMHA: ARRAY [1..60] of Double;
    REG:ARRAY [1..40] of Double;
    CA: ARRAY [1..100] of Double;
    CN1,CN2:ARRAY [1..16] of Double;
end;
```

3. Зарезервировать переменные, которые впоследствии будут использованы для доступа к переменным, располагающимся в общих блоках:

```
var
    PD : ^T_D;
    PREZ1 : ^T_REZ1;
```

4. Подготовить функцию для получения адреса общего блока, описанного как функция

```
type
TFnCOMMON = function (i :Integer) :Integer; cdecl;
function GetAdrCOMMON(fn :TFnCOMMON) : Pointer;
var
    Pn1,Pn2,Pn3 : Pointer;
begin
    Pn1 := Addr(fn); //адрес инструкции вызова fn
    Pn2 := Pointer(Pointer(Integer(Pn1) + 2)*); // адрес перехода
    Pn3 := Pointer(Pn2*);// адрес общего блока
    result := Pn3;
end;
```

Поясним, что именно помещается в адрес перехода Pn2, проиллюстрировав на примере. Инструкция вызова фиктивной функции D выглядит следующим образом (рис. 3).

```
FF25C0F74600 jmp dword ptr [$0046F7C0]
```

■ Рис. 3. Инструкция вызова фиктивной функции

```
function D(I :Integer):Integer; cdecl; external 'GTD.dll';
```

В результате вызова функции **GetAdrCOMMON** мы получаем адрес общего блока, фактически располагаемого в памяти, выделенной **dll**.

5. И последнее — при инициализации модуля получить указатели на описанные общие блоки:

```
PD := GetAdrCOMMON(D);
PREZ1 := GetAdrCOMMON(REZ1);
```

Теперь, чтобы получить доступ, например, к переменной **DT** внутри блока **D**, нужно воспользоваться следующим фрагментом:

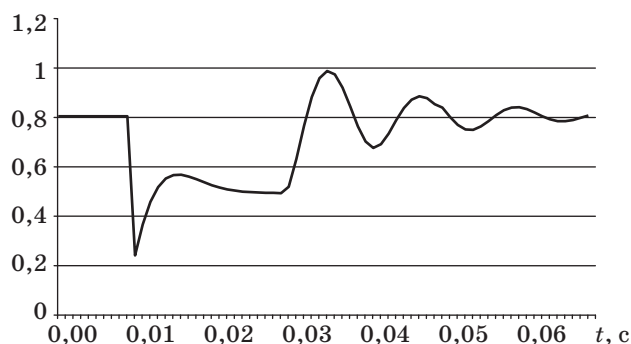
```
@(PD^.DDT);
```

Если переменная внутри блока является массивом, то для доступа к *J*-му элементу массива воспользуемся следующим фрагментом (показываем на примере массива **CP1** из **REZ1**):

```
@(PREZ1^.CP1)^[ J].
```

Вычислительная эффективность математической модели, построенной с использованием описанной методики сопряжения, была проверена экспериментально.

Для большинства существенных переходных процессов максимально допустимый шаг расчета для модели ГТУ составляет 1 мс, для ЭЭС — 10 мкс. Например, характерный переходный процесс продолжительностью 1 с для системы из 6 синхронных генераторов и распределенной комплексной электрической нагрузки рассчитывается с помощью одноточечного метода Рунге—Кутты четвертого порядка за время 14,6 мин на базе Pentium IV, 3 ГГц персонального компьютера. Применение многоточечных методов (Адамса—Башфорта, Адамса—Моултона) позволяет сократить время расчета, для чего можно применить так называемое «распределенное моделирование» на нескольких компьютерах [8]. Основа-



■ Рис. 4. Пример рассчитанного переходного процесса

нием для выбора метода и шага расчета является вид возмущения, налагаемого на энергосистему. Требования к вычислительным ресурсам для разработанных программно-вычислительных комплексов: Microsoft Windows XP, Windows Vista; Pentium IV, 3 ГГц; ОЗУ 512 МБ. График характерного переходного процесса представлен на рис. 4.

Переходный процесс отражает изменение напряжения на внешних зажимах генератора при удаленном коротком замыкании до и после ($t = 0,03$ с) включения в работу САУ. Напряжение измеряется в относительных единицах, время — в секундах. Время расчета на базе Pentium IV, 3 ГГц персонального компьютера составляет 0,6 мин.

Вывод

Предложенные методики математического моделирования и сопряжения программных сред обеспечивают непосредственное включение математических моделей энергетических ГТУ в состав программно-вычислительных комплексов, предназначенных для проектирования и тестирования САУ ГТУ.

Литература

1. Желокова М. З., Максимова И. Р. Прогноз мощностных показателей высокооборотных генераторов с предельной степенью использования для малой энергетики // Изв. РАН. Энергетика. 2008. № 6. С. 127–131.
2. Кириллов И. И. Автоматическое регулирование паровых и газовых турбин и газотурбинных установок. — Л.: Машиностроение, 1988. — 447 с.
3. Винокур В. М., Кавалеров Б. В., Петровичев А. Б. Программный комплекс для математического моделирования автономных мини-электростанций // Электричество. 2007. № 3. С. 2–7.
4. Leung J. S. Global power system control using generator excitation, PSS, FACTS devices and capacitor

- switching // International journal of electrical power & energy systems. 2005. Bd. 27. N 5–6. S. 448 – 464.
5. De Luca A. Power System Simulators for Design and Operations: Two Different Approaches? // ESA SP. 2002. Bd. 502. S. 201–208.
6. Ахмедзянов А. М., Дубравский Н. Г., Тунаков А. П. Диагностика состояния ВРД по термогазодинамическим параметрам. — М.: Машиностроение, 1983. — 20 с.
7. Бусленко Н. П. Моделирование сложных систем. — М.: Наука, 1978. — 440 с.
8. Distributed simulation. A new modeling technique is applied to a HALE UAV power system using detailed subsystem simulations of the turbine engine, generators, and loads // Aerospace engineering. 2004. Bd. 24. S. 24–27.