УДК 681.5

А. Т. Лелеков, М. Ю. Сахнов, С. А. Галочкин, Е. В. Величко

МОДЕЛИРОВАНИЕ ТЕПЛОВЫХ ОБЪЕКТОВ С РАСПРЕДЕЛЕННЫМИ ПАРАМЕТРАМИ С ИСПОЛЬЗОВАНИЕМ ПРОГРАММНЫХ КОМПЛЕКСОВ COMSOL MULTIPHYSICS–MATLAB–SIMULINK

Рассмотрено построение динамических моделей тепловых объектов с распределенными параметрами. Обсуждаются трудности, связанные с интеграцией модели COMSOL в модель Simulink для решения задач управления.

Ключевые слова: моделирование, автоматическое управление, теплопередача, объект с распределенными параметрами, COMSOL Multiphysics, MATLAB, Simulink.

При проектировании систем автоматического управления (САУ), оптимизации и настройке законов управления, обработки сигналов, моделировании поведения объектов управления нередко возникают трудности, связанные с тем, что объект управления невозможно представить сосредоточенной моделью. Поэтому требуется представить его как объект с распределенными параметрами (ОРП), что подразумевает использование для его описания дифференциальных уравнений (ДУ) в частных производных. В данной статье речь пойдет о решении тепловых задач с учетом их динамики, когда нельзя пренебречь длительностью переходного процесса, так как на вход ОРП поступает меняющееся воздействие и необходимо рассчитать отклик в следующий момент времени.

Обычно в таких случаях пользуются следующими подходами:

– построением модели с сосредоточенными параметрами: выбирается простая модель с похожим поведением и производится подбор таких параметров, которые позволят добиться адекватности;

– построением упрощенной (идеализированной) модели с распределенными параметрами: объект линеаризуется и схематизируется до возможно более простой задачи, параметры также подбираются, задача решается аналитическими способами и полученное решение раскладывается в ряд с ограниченным количеством членов;

– построением численных дискретных моделей с распределенными параметрами, отклики которых ищутся с помощью методов конечных элементов или конечных разностей.

Последний подход, имея ряд существенных недостатков по сравнению с предыдущими (низкая аналитическая ценность, сложность модели, высокие требования к вычислительной мощности и др.), обладает при этом ценными преимуществами: сокращением времени подготовки модели, учетом нелинейностей и сложных форм объектов, сложными начальными (НУ) и граничными условиями (ГУ), а также возможностями, т. е. мультифизичности совместного решения уравнений, описывающих различные по природе, но взаимосвязанные физические явления.

Способы реализации этого подхода варьируются от самостоятельного написания программ на языках программирования высокого уровня до использования специализированных программных комплексов. Для моделирования ОРП авторами были использованы возможности комплекса COMSOL Multiphysics 3.4a, который работал совместно с пакетом MATLAB 2008 и Simulink с заданной в них моделью системы управления распределенным объектом. Эти программные комплексы, взаимодействуя по технологии COM, позволяют проводить совместное имитационное моделирование сложных систем управления объектов с распределенными параметрами.

Конечно, можно обойтись использованием одного пакета MATLAB 2008 и Simulink и моделирование ОРП производить в PDE Toolbox (с дополнительным набором функций для решения ДУ в частных производных). Однако PDE Toolbox пока не допускает использования связанных геометрий и возможностей мультифизичности, которыми обладает комплекс COMSOL Multiphysics.

Ход построения модели таков:

1) задать модель управления объектом в пакете MATLAB–Simulink;

2) задать модель объекта управления в COMSOL Multiphysics;

3) интегрировать модель COMSOL в модель Simulink.

В данной статье будут обсуждаться трудности, связанные с реализацией третьего шага.

Стандартные способы экспорта моделей. Программный комплекс COMSOL Multiphysics имеет в своем составе стандартные средства генерации модели Simulink [1]. Процедура генерации создает в области переменных MATLAB структуру sct, которая представляет собой COM-объект. Она указывается как параметр блока COMSOL Multiphysics Subsystem в модели Simulink. Вид модели настраивается двумя параметрами: формой (General или Linearized) и типом (Dynamic или Static).

Первый параметр фактически определяет, в какой программе будет происходить расчет модели. При нелинейных моделях создается модель General (общая), и при имитационном моделировании будет происходить вызов процедур обсчета (солвера) из COMSOL. Если используется модель без нелинейностей или движения переменных нелинейной модели происходят в малой окрестности рабочей точки, то можно создать линеаризованную модель, и тогда не нужно задействовать специализированные процедуры COMSOL, что значительно ускоряет процесс моделирования. Такая модель представляет собой обычный набор матриц передаточных коэффициентов и к ней можно применить стандартные методы снижения порядка.

Второй параметр определяет, можно ли пренебречь зависимыми от времени членами и производными в уравнениях объекта. В типе Dynamic выходные переменные

зависят как от входов, так и от вектора состояния, который всегда хранится в Simulink. Если инерционность распределенного объекта пренебрежимо мала, то можно ограничиться решением только стационарной задачи в частных производных и вектор состояний не потребуется, поскольку входы модели непосредственно определяют выходы.

В проводимых нами расчетах важное значение имеет учет нелинейностей, а скорость тепловых процессов не позволяет упростить модель до статической, поэтому нужно использовать модель типа General Dynamic.

Отметим, что входы модели задаются через имена определенных в COMSOL переменных, которые могут входить в выражения воздействий в области любой размерности: точке, линии, объеме и т. п. Возможности же определения выходов модели сильно ограничены: можно выбрать значение переменной только в какой-либо одной точке. Сколько выбрано точек, столько и будет выходов, т. е. вектор выходов может быть только одномерным. Это может стать существенным недостатком в случае, если требуется сделать усреднение поля по некоторой области или передать все поле для визуализации или обработки, так как блок, сделанный стандартным образом, таких возможностей не предоставляет.

Увеличение размерности вектора выходов. Расчет модели в Simulink производится посредством s-функции femsfun, которая скрыта в маске блока COMSOL Multiphysics Subsystem. Это s-функция первого уровня (Level 1 S-function), за формирование выходов в которой отвечает подфункция mdlOutputs (в ней по COM-технологии вызываются процедуры COMSOL). Собственно вектор выходов формируется в строке

```
sys = l_computeoutput(sct, jsol, vars);
```

Приведем эту функцию полностью:

```
% ========================================================================
% Compute output variables
% ========================================================================
function sys = l_computeoutput(sct, jsol, vars)
sys = zeros(length(sct.outexprs),1);
piprop = com.femlab.util.Prop;
piprop.setVectorString('const', vars);
piprop.setSolution('u', jsol);
outprop = com.femlab.util.Prop;
for i=1:length(sct.outexprs)
  sct.outpis(i).eval(sct.outexprs(i), piprop, outprop);
  % Temporary constants are cleared
  out = getvmatrixexch(outprop, logical(1));
  if abs(imag(out{1}))>1e-14*abs(real(out{1}))
    error(['Complex-valued output variable: ' sct.outputnames{i}])
  end
  sys(i) = real(out{1});
end
```

Чтобы пожертвовать возможностью иметь несколько отдельных выходов у блока COMSOL, получив взамен возможность вывода всего поля предпоследнюю строку нужно заменить на строку

```
sys = real(out{1});
```

Можно убрать и весь цикл по *i*, так как размер `sct.outexprs` будет равен 1, т. е. количеству независимых выходов.

Генерировать структуру `sct` необходимо также особым способом. Экспорт должен быть произведен не через интерфейс COMSOL, а в командной строке MATLAB. Программы должны быть запущены совместно, для чего вначале производится экспорт структуры fem в MATLAB, а затем делается вызов функции генерации модели Simulink командой

```
sct=femsim_2D(fem, ...
          'input',{'Q'}, ...
          'outnames',{'Temp'}, ...
                    'output',{{'T' xx}},'Matherr','off');
```

где `fem` – структура, содержащая описание задачи; `{'Q'}` – массив с именами констант – входов модели; `{'Temp'}` – массив с именами выходов; `{{'T' xx}}` – массив с выражением для выходной переменной и точек, в которых оно берется. Эта функция отличается от стандартной функции `femsim` строкой

```
outpoints{i} = reshape(out{2}, prod(size(out{2})), 1);
```

замененной на

```
outpoints{i} = out{2};
```

Это позволяет передать на выход неодномерный массив.

Внутри `femsim_2D` производится вызов функции `femsimlowlevel_2D` взамен стандартной `femsimlowlevel`. Текст функции `femsimlowlevel_2D` не отличается от `femsimlowlevel`, кроме одной строки в цикле

```
for i=1:noutputs
   piprop = postproputil(dummyfem, outparams{i});
   piprop.setInt('mcase',mcase);
   xx = outpoints{i}';
                outpis(i) = com.femlab.xmesh.PostInterp(xmesh, piprop, xx, ...
                        size(xx,1),size(xx,2));
end
```

которая заменена на строку

```
for i=1:noutputs
    piprop = postproputil(dummyfem,
outparams{i});
    piprop.setInt('mcase',mcase);
    xx = outpoints{i}; %транспонирование не нужно
                outpis(i) =
com.femlab.xmesh.PostInterp(xmesh, piprop, xx, ...
                    size(xx,1),size(xx,2));
end
```

где параметр xx – матрица координат точек, в которых нужно передать выражение выходной переменной. Для расчета выходов COMSOL использует функцию postinterp, поэтому описание xx можно найти в соответствующем разделе помощи. В колонках матрицы, содержащих координаты *x*, *y*, строка задает номер точки. В двумерном случае, если применяется равномерная прямоугольная сетка, ее можно сгенерировать следующим образом:

```
[X,Y]=meshgrid(0:0.001:0.05,0:0.001:0.05);
n=size(X);
xx=[reshape(X,1,[]);reshape(Y,1,[])];
% Колонки- координаты точек
```

При экспорте структуры fem особое внимание нужно уделить параметрам солвера. Он должен быть типа Transient, и эту задачу обязательно решить в COMSOL заранее, чтобы в дальнейшем исключить труднонаходимые Simulink ошибки.

Построенная таким образом модель в качестве выхода может выдавать поле, рассчитанное в каждый момент времени, однако у нее есть существенный недостаток – модели со связанными (coupled) геометриями здесь не работают.

В Словацком технологическом университете (Братислава, Словакия) был создан пакет подпрограмм для MATLAB–Simulink, позволяющий решать задачи моделирования систем с распределенными параметрами с предоставлением возможностей вывода и визуализации двумерных полей [2]. Разработчики этого пакета используют как собственные технологии моделирования ОРП, так и имеющиеся в составе пакета PDE Toolbox приложения MATLAB. Достоинством предлагаемого пакета является то, что он включает в себя средства анализа и синтеза систем управления ОРП. К недостаткам следует отнести закрытость алгоритмов.

Расширение возможностей модели. Расширить возможности полученной модели за счет использования полной функциональности комплекса COMSOL Multiphysics позволяет s-функция, принцип работы которой основан на классическом методе сшивки. В этом методе конечное состояние объекта на очередном участке переходного процесса служит начальными условиями для следующего участка, благодаря чему он нашел широкое применение для решения тепловых задач, описываемых ДУ первого порядка по времени, которые не требуют в качестве начальных условий значений производных.

Несколько вариантов использования метода сшивки описаны в [3–6].

Автор статей [3– 5] изучает времязависимое решение и использует s-функцию первого уровня, отличающуюся следующими особенностями:

– структура fem, описывающая ОРП, задана как глобальная переменная. Она создается заранее средствами COMSOL;

– величина шага по времени определяется Simulink, тип времени выборки (sample time) – Continuous. Задача пересчитывается каждый раз на новый период времени с одной промежуточной точкой, т. е. временной интервал делится ровно посередине;

– на выход передаются данные – температурные поля в определенных узлах сетки, созданной в COMSOL;

– данные о состоянии объекта хранятся в векторе состояний Simulink.

В [6] изучается статическое решение. Использована s-функция второго уровня, отличающаяся следующими особенностями:

– структура fem создается каждый раз заново скриптом MATLAB;

– величина шага по времени определяется Simulink. Задача стационарная, временной интервал не важен, данные о состоянии объекта не хранятся;

– задача решается с помощью функции формирования выходного вектора. Полученные решения дополнительно обрабатываются, из всего набора данных – полей – выбирается только нужное поле (функция postinterp).

В модели, разработанной авторами данной статьи, применена s-функция второго уровня, передающая на выход как все температурное поле, так и ряд рассчитанных с его использованием переменных. Структура xfem создается заранее в COMSOL и задается как глобальная переменная. В нотации COMSOL структура xfem отличается от fem тем, что первая вводит задачу со связанными геометриями. В этой структуре хранится состояние ОРП, переменные Simulink для этого не используются. Входные воздействия переопределяются на каждом шаге. Величина шага по времени выбирается таким образом, чтобы решение изменилось на величину не более 1 %, что для данного типа ДУ составляет также ~1 % от постоянной времени. Шаг задается таким способом, чтобы Simulink вызывал s-функцию только через указанный промежуток. Это возможно потому, что решения тепловых задач практически всегда устойчивы и имеют значительную постоянную времени. Код функции структурно выглядит так:

1) считать xfem из области глобальных переменных:

```
global xfem
```

2) перезадать константы fem.const – входы модели:

```
Tinlet = block.InputPort(2).Data +
273.15; %Tinlet, переводим в Кельвины
xfem.const{1}=num2str(Tinlet);
```

3) рассчитать НУ текущего шага:

```
init = asseminit(xfem,'init',
xfem.sol,'solnum',length(xfem.sol.tlist));
%В качестве НУ берем последнее решение
```

4) вызвать процедуру расчета решения:

```
xfem.sol=femtime(xfem, ...
    'init',init, ... % параметр 'init'-
недокументированный
```

```
      'solcomp',{'T'}, ...
      'outcomp',{'T'}, ...
        'tlist',11, ... %задает вектор времен, единственный эл-т – конечное время. Можно задать больше промежуточных точек, решение будет точнее
      'tout','tsteps',...
      'initialstep',2.5, ...
      'maxstep',1e3);
```

5) сформировать выходы – температуры и т. п. – через функции `postinterp` и `postint`. На этом шаге удобно извлекать данные из разных геометрий, например `Tfield` извлекается из основной (номер 1), а `Toutlet` – из связанной (номер 3). Положение точек температурного поля задается изначально рассчитанной матрицей `Coords`:

```
  %формируем выходы
  Tfield = postinterp(xfem,'T', Coords.c,'Solnum','end')-273.15; %в градусах Цельсия
  Tfield(find(isnan(Tfield)))=Inf; %NaN недопустимы в s-функции
  Tmean = Tacc_20(Tfield,Coords);
  Toutlet= postint(xfem,'T', ...
      'dl',[2], ...
      'edim',0, ...
      'geomnum',3, ...
     'solnum','end')-273.15; %Toutlet в градусах Цельсия
```

Преимуществами такой модели являются легко задаваемая дополнительная математическая обработка решения, возможность использования связанных геометрий, контроль над ходом решения. Эта модель была успешно опробована и показала устойчивую работу.

Визуализация. Для визуализации рассчитанного температурного поля была создана s-функция первого уровня, не требующая ресурсов COMSOL. Она позволяет визуализировать поле в заранее заданных точках, меняющееся каждый раз после его расчета описанным выше способом. Входные параметры s-функции – матрица координат точек поля и вектор, содержащий значения поля в этих точках. Для решения был использован способ триангуляции, предложенный Б. Н. Делоне в 1934 г. (функция `delaunay`), согласно которому сетка координат точек поля с любым распределением по геометрии объекта разбивается на сетку из треугольников с вершинами в указанных точках. S-функция строит поверхность из объектов patch на этой сетке, рассчитанной заранее. Приведем далее полный текст этой функции:

```
  function [sys, x0, str, ts] = mysfunTRI(t, x, u, flag, Coords, ZLim)
  % Выводит trisurf, на входе значения z в точках [X Y] из Coords
  %Coords – структура .
  %ZLim - жесткие пределы по Z, чтобы не было масштабирования

  % Определение действия в зависимости от значения flag
  switch flag
      case 0 % Инициализация S-функции
            [sys, x0, str, ts] = mdlInitializeSizes(Coords);
     case 2 % Действия, производимые на каждом шаге по времени
         sys=mdlUpdate(t, x, u, Coords, ZLim);
      case 3 % Вычисление значений на выходах S-функции
          sys=mdlOutputs(t,x,u);
      otherwise % выводим ошибку, т.к. действие не определено
            error(['Unhandled flag = ',num2str(flag)]);
  end


  function [sys, x0, str, ts] = mdlInitializeSizes(Coords) % Инициализация S-функции
  sizes = simsizes; % Получение структуры с описанием S-функции
  sizes.NumInputs = length(Coords.n); % задание числа входных портов
  sizes.NumSampleTimes = 1; % число дискретизаций по времени
  sys = simsizes(sizes); % создание вектора с информацией об S-функции

  x0  = [];
  str = [];
  % задание вектора с шагами расчета
  ts  = [-1 0]; %-1 наследование этого параметра от предыдущего блока
  % создание графического окна GUI_Simulink
  try
      close(get_param(gcbh,'Name'));
  end
  % и запись указателя на него в переменную Fig
  Fig=figure('Position',[400 300 400 300],... %    'Color', 'k',...
       'MenuBar', 'none',...
      'Name', get_param(gcbh,'Name'),...
       'NumberTitle', 'off');
  % сохранение указателя на графическое окно в свойстве UserData S-функции
  set_param(gcbh, 'UserData', Fig);

  function sys=mdlUpdate(t, x, u, Coords, ZLim)
  % получаем указатель на графическое окно GUI_Simulink
  %Fig = get_param(gcbh,'UserData');
  % получаем структуру указателей на объекты окна GUI_Simulink
  % Handles = guihandles(Fig);
  % axes(Handles.ax1)
  trisurf(Coords.tri,Coords.c(1,:),Coords.c(2,:),u,'
```

```
EdgeColor','interp','FaceColor','interp')
   axis equal
   view(2)
   colorbar
   set(gca, 'Zlim', ZLim);
   drawnow; % Классическое решение
   sys=[];

   function sys=mdlOutputs(t,x,u)
   % вычисление значений на выходах блока
S-function
   %выходов нет, поэтому присваиваем пу-
стой массив.
   sys=[];
```

Работа выполнена при программной поддержке научно-образовательного центра интегрированных компьютерных технологий информационно-аналитического департамента Сибирского федерального университета.

Библиографические ссылки

1. Storozhenko Y., Gubanski S., Serdyuk Y. COMSOL-Simulink integrated computer model for simulations of high voltage power transformers // Scientific bull. of Norilsk Industrial Inst. Vol. 3. Norilsk, 2008. P. 28–45.

2. Control of Systems Modeled by COMSOL Multiphysics as Distributed Parameter Systems / G. Hulko, C. Belavy, P. Bucek et al. // Proc. of the COMSOL Multiphysics User's Conf. Milan, 2009.

3. Schijndel A. W. M. van. Integrated modeling of dynamic heat, air and moisture processes in buildings and systems using SimuLink and COMSOL // Building Simulation. 2009. Vol. 2, № 2. P. 143–155.

4. Schijndel A. W. M. van. Implementation of FemLab in S-Functions // Proc. of COMSOL Multiphysics User's Conf. Frankfurt, 2005. P. 324–329.

5. Schijndel A. W. M. van. Integrated building physics simulation with FEMLAB/SIMULINK/MATLAB // Proc. of Eighth Intern. IBPSA Conf. Eindhoven, 2003. P. 1177–1184.

6. Jansen R. M. Finite element model of shape and density adaptation in engineered bone : Thesis Presented in Partial Fulfillment of the Requirements for the Degree Master of Science in the Graduate School of The Ohio State University // The Ohio State University. Columbus, Ohio, 2008.

A. T. Lelekov, M. Y. Sahnov, S. A. Galochkin, E. V. Velichko

MODELING OF THERMAL OBJECTS WITH DISTRIBUTED PARAMETERS WITH THE USE OF COMSOL MULTIPHYSICS–MATLAB–SIMULINK SOFTWARE PACKAGES

Construction of dynamical models of thermal objects with distributed parameters heat is considered. The difficulties of integration of COMSOL Multiphysics model in Simulink are discussed.

Keywords: modeling, automatic control, heat transfer, distributed parameters objects, COMSOL multiphysics, MATLAB, Simulink.