

MatLab

Раис Ахметсафин, Римма Ахметсафина, Юрий Курсов

Вопрос Доступен ли интерпретатор MATLAB в системах MMI/SCADA? Как реализовать выполнение команд MATLAB в этих системах?

Ответ

В инженерной, а тем более в исследовательской практике встречаются задачи (например, быстрое преобразование Фурье или адаптивное управление), реализация которых даже средствами Visual Basic for Application (VBA) трудоемка. Специалисты в области управления и обработки сигналов, знакомые с пакетом MATLAB (<http://www.mathworks.com>), наверняка подумают о том, как было бы здорово, если бы в системах MMI/SCADA был доступен интерпретатор MATLAB-скрипт. Так сложилось, что MATLAB-скрипт является своего рода стандартом представления алгоритмов в среде исследователей и разработчиков в области теории управления, обработки сигналов и не только. В состав поставки пакета входят библиотеки (Toolboxes), позволяющие решать большинство инженерных задач, связанных с обработкой сигналов и управлением. Кстати, MATLAB имеет и ресурсы для устройств связи с объектом,

но не такие развитые, как в MMI/SCADA. Кроме того, у этого пакета великолепные возможности графического отображения результатов (рис. 1).

Возможность доступа к интерпретатору MATLAB предусмотрена во всех современных пакетах класса MMI/SCADA. Как же из VB-скрипт выполнять команды MATLAB-скрипт в условиях пакетов MMI/SCADA? Такая возможность предоставляется, если вы проинсталируете у себя MATLAB 5 или MATLAB 4 из комплекта поставки MATLAB 5.x. По интерфейсу DDE (Dynamic Data Exchange) вы имеете командный доступ к интерпретатору. Для DDE в среде Windows должны быть определены три параметра: Server, Topic, Item. Для доступа к параметрам и интерпретатору MATLAB проводится определение: Server — «matlab», Topic — «Engine» (для инженерной, наиболее популярной версии пакета), Item - идентификатор переменной в MATLAB.

Далее приведены операторы и функции VBA, используемые для выполнения команд интерпретатора MATLAB и обмена данными.

Функция **DDEInitiate(application\$, topic\$)** инициализирует DDE-канал и возвращает уникальный код, который используется для доступа к этому приложению.

Оператор **DDEExecute [канал], [команда]** выполняет команду в приложении по открытому каналу.

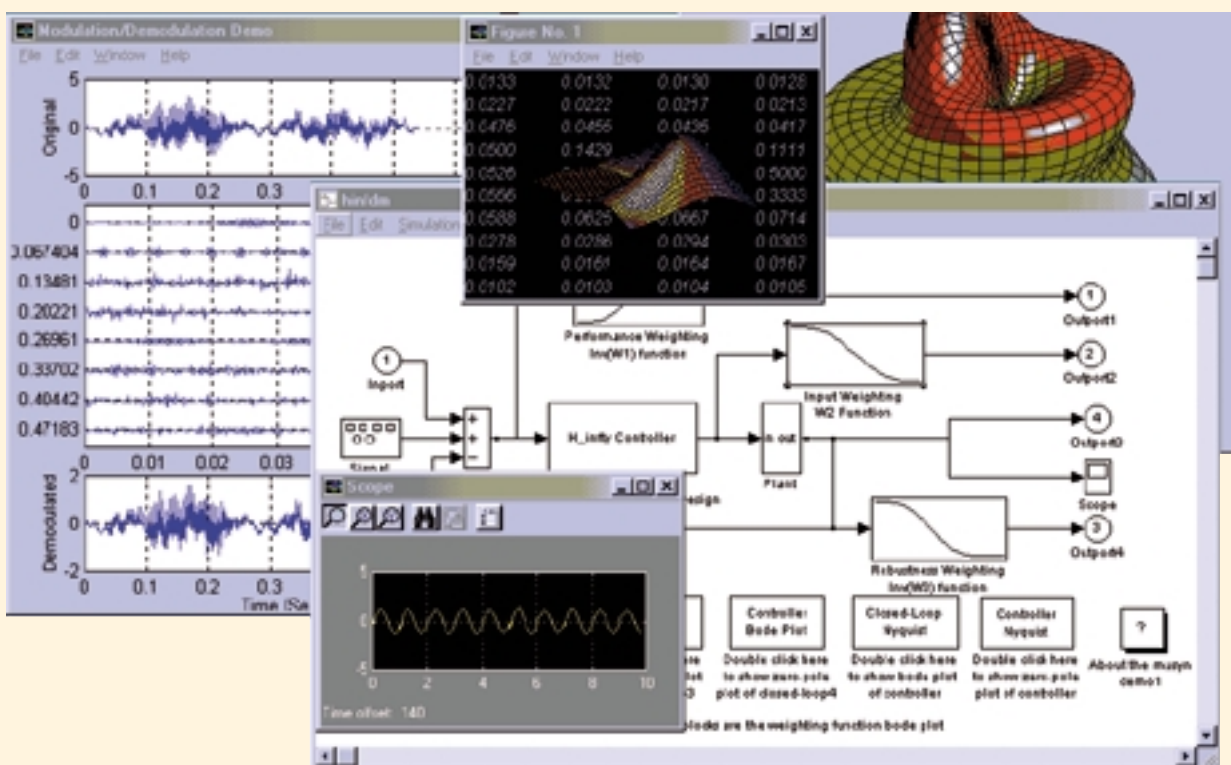


Рис. 1. Графические возможности MATLAB

Пример присвоения переменной «u» в MATLAB значения переменной «a%» в VBA:

```
a% = 124
ch% = DDEInitiate("matlab", "Engine")
cmd$ = "u=" & a%
DDEExecute ch%, cmd$
DDETerminate ch%
```

Пример выполнения строки в интерпретаторе MATLAB:

```
H=A'*B
Y=P'*inv(H)*P

ch% = DDEInitiate("matlab", "Engine")
DDEExecute ch%, _
"H=A'*B" & Chr$(13) & _
"Y=P'*inv(H)*P"
DDETerminate ch%
```

Оператор **DDEPoke channel, DataItem, value** передает значение в переменную с идентификатором DataItem приложения по открытому каналу.

Пример присвоения значения «124» переменной «u»:

```
ch% = DDEInitiate("matlab", "Engine")
DDEPoke ch%, "u", "124"
DDETerminate ch%
```

Функция **DDERequest[s](channel, DataItem\$)** возвращает значение переменной с идентификатором DataItem приложения по открытому каналу.

Пример возвращения значения переменной «u» из MATLAB в переменную «s\$»:

```
ch% = DDEInitiate("matlab", "Engine")
s$ = DDERequest$(ch%, "u")
DDETerminate ch%
```

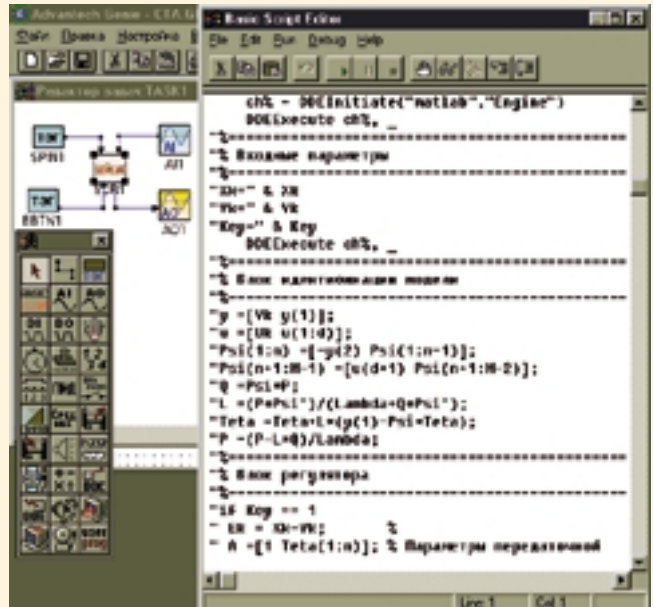


Рис.2. Копия экрана со сценарием BASIC-скрипт SCR1

Оператор **DDETerminate channel** закрывает DDE-канал.

На врезке приведен пример реализации адаптивного управления с идентификатором и регулятором AP(v) [1] в пакете GENIE 3.x. Для адаптивного регулятора входными параметрами являются задающее воздействие, выход объекта (управляемая переменная) и режим (0 — регулятор отключен и выход определяется задающим воздействием; 1 — адаптивный регулятор включен). Выходом регулятора является управляющее воздействие. ●

Литература

1. Изерман Р. Цифровые системы управления / Пер. с англ. — М.: Мир, 1984. — 541 с.

Предварительный сценарий

```

Sub PRE_TASK1()
  ch% = DDEInitiate("matlab","Engine")
  DDEExecute ch%,_
  "%=====
  "% Инициализация параметров идентификации
  "%=====
  "Lambda =0.99;          % Фактор забывания РМНК
  "n = 3;                % Порядок модели
  "d = 1;                % Запозывание модели
  "M = 2*(n+1);          % Кол-во параметров модели"&
  "Alpha = 1.0e11;       % ***Инициализация матриц***
  "P = Alpha*eye(M);     % Обратная матрица ковариаций"&
  "Teta = zeros(M,1);    % Вектор-столбец параметров
  "Psi = zeros(1,M);     % Вектор-строка измерений
  "Psi(M) = 1;          %
  "y = zeros(1,2);       % Вспомогательные
  "u = zeros(1,d+1);     % векторы
  "%=====
  "% Инициализация параметров регулятора
  "%=====
  "MR = M-1+d;          % Размерность уравнения регрессии
  "PsiR = zeros(1,M);   %
  "U_Hi = 100;          % Ограничения выхода
  "U_Lo = 0;            % регулятора
  "Uk = U_Lo;           % Инициализация выхода регистра
  "%=====
  DDETerminate ch%
End Sub
  
```

Сценарий BASIC-script SCR1

```

Sub SCR1()
  Dim Xk As Tag
  Dim Yk As Tag
  Dim Key As Tag
  Set Xk = GetTag("DISP1","SPIN1")
  Set Yk = GetTag("TASK1","AI1")
  Set Key = GetTag("DISP1","BBTN1")

  ch% = DDEInitiate("matlab","Engine")
  DDEExecute ch%,_
  "%=====
  "% Входные параметры
  "%=====
  "Xk=" & Xk
  "Yk=" & Yk
  "Key=" & Key
  DDEExecute ch%,_
  "%=====
  "% Блок идентификации модели
  "%=====
  "y = [Yk y(1)];
  "u = [Uk u(1:d)];
  "Psi(1:n) = [-y(2) Psi(1:n-1)];
  "Psi(n+1:M-1) = [u(d+1) Psi(n+1:M-2)];
  "Q = Psi*P;
  "L = (P*Psi')/(Lambda+Q*Psi');
  "Teta = Teta+L*(y(1)-Psi*Teta);
  "P = (P-L*Q)/Lambda;
  "%=====
  "% Блок регулятора
  "%=====
  "if Key == 1
  " Ek = Xk-Yk;          %
  " A = [1 Teta(1:n)];  % Параметры передаточной
  " B = Teta(n+1:M-1);  % функции объекта
  " q0 = sum(B);        %
  " TetaR = [q0*[zeros(1,d) B] q0*A];
  " PsiR = [[Uk PsiR(1:n+d-1)] [Ek PsiR(n+d+1:MR-1)]];
  " Uk = TetaR*PsiR';   % Выход регулятора
  "else
  " Uk = Xk;
  "if Uk > U_Hi
  " Uk = U_Hi;
  "end
  "if Uk < U_Lo
  " Uk = U_Lo;
  "end
  "%=====
  outputs 0,DDERequest$(ch%,"Uk")
  DDETerminate ch%
End Sub
  
```