

**Московский авиационный институт  
(Национальный исследовательский университет)**

Факультет: «Информационные технологии и прикладная математика»

Дисциплина: Теоретическая механика и основы компьютерного  
моделирования

**Лабораторная работа № 3**

Студент: Дюсекеев А. Е.

Группа: М80-204Б-17

Преподаватель: Владимир  
Иванович

Дата: 02.10.2018

Оценка:

### Отчёт по лабораторной работе 3:

```
syms t
%t - время, f - угол поворота пластины, s=OD, а - угол
альфа
f = 2*t^2
s = sin(3*t)
a = pi/6

om = diff(f,t)
om_ = [0 0 om] %вектор угловой скорости
eps_ = [0 0 diff(om)] %вектор углового ускорения

%координаты в абсолютной системы
x = s*sin(a)*cos(f)
y = s*sin(a)*sin(f)
z=s*cos(a)
r_=[x,y,z]

Vx = diff(x,t)
Vy = diff(y,t)
Vz = diff(z,t)
Wx = diff(Vx,t)
Wy = diff(Vy,t)
Wz = diff(Vz,t)

%Абсолютная скорость и ускорение
V_=diff(r_,t);
% Vab = sqrt(V_(1)^2+V_(2)^2+V_(3)^2)
W_=diff(V_,t);
W_=simplify(W_)

%Теорема сложения скоростей
ir = [cos(f),sin(f),0]
jr = [-sin(f),cos(f),0]
kr = [0 0 1]
```

```
xr = s*sin(a)
yr = 0
zr = s*cos(a)
rr_=[xr, yr, zr]
```

```
% Отн. скорость и ускорение в подвиж. осях
Vr_=Dif(rr_,t,f,s) % Перечисляем все переменные
величины, зависящие от времени НЕЯВНО
```

```
Vr_1=Vr_(1)
```

```
Vr_2=Vr_(2)
```

```
Vr_3=Vr_(3)
```

```
Wr_=diff(Vr_,t); % Перечисляем все переменные
величины, зависящие от времени НЕЯВНО
```

```
% Отн. скорость и ускорение в абс. осях
```

```
Vr_a=ir*Vr_(1)+jr*Vr_(2)+kr*Vr_(3)
```

```
Wr_a=ir*Wr_(1)+jr*Wr_(2)+kr*Wr_(3)
```

```
% Переносные скорость и ускорение в абс. осях
```

```
Ve_=cross(om_,r_)
```

```
Vabc_=Ve_+Vr_a
```

```
dV=V_-Vabc_% Проверка
```

```
We_=cross(eps_,r_)+cross(om_,Ve_)
```

```
Wc_a=2*cross(om_,Vr_a)
```

```
Wabc_=We_+Wr_a+Wc_a
```

```
dW=W_-Wabc_% Проверка
```

```
% t = 0
```

```
% om1 = eval(om)
```

```
% om_1 = eval(om_)
```

```
% eps_1 = eval(eps_)
```

```
% x1 = eval(x)
```

```
% y1 = eval(y)
```

```
% z1 = eval(z)
```

```
%Переходит к matlab вычислениям
```

```

t_ = 0:0.01:4;
l_t = length(t_);
for i = 1 : l_t;
t = t_(i);
x_(i)=eval(x);
    y_(i)=eval(y);
    z_(i)=eval(z);
Wabc__(i) = eval(Wabc_(1));
Vabc__(i) = eval(Vabc_(1));
V__(i) = eval(V_(1));
W__(i) = eval(W_(1));
Vx_(i) = eval(Vx);
    Vy_(i) = eval(Vy);
    Vz_(i) = eval(Vz);
    Wx_(i) = eval(Wx);
    Wy_(i) = eval(Wy);
    Wz_(i) = eval(Wz);
end

```

```

figure
plot(t_,Vabc__)
title ('Vabc(t) - абсолютная скорость, найденная с
помощью теоремы сложения скоростей')
xlabel('x м/с')
legend ('Vabc(t)')
grid on

```

```

figure
plot (t_,Wabc__)
title('Wabc(t) - абсолютное ускорение, найденное с
помощью теоремы сложения ускорений')
xlabel('x м/с^2')
legend('Wabc(t)')
grid on

```

```

figure

```

```

plot (t_,W__)
title('W(t) абсолютное ускорение, найденное просто
дифференцированием')
xlabel('x м/с^2')
legend('W(t)')
grid on

```

```

figure
plot (t_,V__)
title('V(t) - абсолютная скорость, найденная просто
дифференцированием')
xlabel('x м/с')
legend('V(t)')
grid on

```

```

W=[Wx_,Wy_,Wz_]
size(W_)
figure
plot3(x_,y_,z_, 'r')
hold on
quiver3(x_,y_,z_,Wx_,Wy_,Wz_,0.5) %вектор ускорения
траектории

```

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

```

function varargout=Dif(f,varargin)
% СИМВОЛЬНЫЕ ВЫЧИСЛЕНИЯ. Вычисление 1-й и 2-
й производной.
% Два способа обращения к программе :
% вычисление 1-й производной: Df=Dif(f,t,a,b1,...),
% вычисление 1-й и 2-й производных:
[Df,D2f]=Dif(f,t,a,b1,...).
% Здесь varargin - список аргументов
дифференцируемой ф-ции f=f(arg1,arg2,...),
% arg1 - независимая переменная, по которой
выполняется дифф-е (t,x,x1 и т.п.),

```

```

% arg2, arg3,.. - параметры - неявные функции от
независимой переменной,
% напрр., a,b1,...
% Производные от arg2, arg3,.. в результате
дифференцирования обозначаются
% как arg2t, arg3t, arg2tt или arg2x, arg2xx,...и т.п.
% ОКОНЧАНИЕ t используется программой для
обозначения производных
% от параметров по независимой переменной.
% В ВЫЗЫВАЮЩЕЙ программе, в которой после
символьных вычислений
% выполняется переход к матлабовским операциям,
необходимо в syms включить
% не только arg1,arg2,..., но и первые и вторые
производные в виде
% arg2t, arg3t, arg2tt, arg3tt.
% Примеры обращения к функции :
% Ds=Dif(sin(2*t+a),t), Ds=Dif(sin(2*t+a),t,a),
om=Dif(fi,t,fi), [om,eps]=Dif(fi,t,fi)

```

```

% ВЫЧИСЛЕНИЕ 1-Й ПРОИЗВОДНОЙ

```

```

n = nargin; % К-во входных аргументов
namt= inputname(2); % Имя независимой переменной
Dt=diff(f,namt); % Производная по явной независимой
переменной
Df=Dt;
names{1}=namt; % Формирование списка имен
if n>=3
for i=3:n
nami= inputname(i);
names{i-1}=nami; % Имена переменных параметров
Dai=diff(f,nami); % Производная по параметру
% ni_t=[nami,'_',namt]; % % Создание имени
ni_t=[nami,namt]; % % Создание имени (без '_',namt)
names_d{i-2}=ni_t; % Перечень имен

```

```

Dti=Dai*ni_t; % Дифференцирование неявной
зависимости
Df=Df+Dti; % Дифференцирование неявной
зависимости
end; end;
varargout{1}=Df; % 1-я производная
n2 = nargout; % К-во выходных аргументов
if n2==1 return; end;
% ВЫЧИСЛЕНИЕ 2-Й ПРОИЗВОДНОЙ
ft=Df;
Df2=diff(ft,namt); % Дифференцирование явной
зависимости
if n==2, varargout{2}=simplify(Df2); return;end; % 2-я
производная
names_=[names,names_d]; % Переменные величины
Ln=length(names_); % К-во аргументов с неявным
временем
if Ln>=2
for i=2:Ln
nami= names_{i}; % Имя параметра
Dai=diff(ft,nami); % Производная по параметру
% ni_t=[nami,'_',namt]; % Новое имя ... _t
ni_t=[nami,namt]; % Новое имя ... _t
Dti=Dai*ni_t; % Производная по времени
Df2=Df2+Dti; % Дифференцирование неявной
зависимости
end; end;
varargout{2}=simplify(Df2); % 2-я производная
% =====
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

```

$f = 2 \cdot t^2$

$$s = \sin(3*t)$$

$$a = 0.5236$$

$$\omega = 4*t$$

$$\omega\_ = [0, 0, 4*t]$$

$$\epsilon\_ = [0, 0, 4]$$

$$x = (\sin(3*t)*\cos(2*t^2))/2$$

$$y = (\sin(3*t)*\sin(2*t^2))/2$$

$$z = (3^{1/2}*\sin(3*t))/2$$

$$r\_ = [(\sin(3*t)*\cos(2*t^2))/2, (\sin(3*t)*\sin(2*t^2))/2, (3^{1/2}*\sin(3*t))/2]$$

$$W\_ = [- (9*\sin(3*t)*\cos(2*t^2))/2 - 2*\sin(3*t)*\sin(2*t^2) - 8*t^2*\sin(3*t)*\cos(2*t^2) - 12*t*\cos(3*t)*\sin(2*t^2), 2*\sin(3*t)*\cos(2*t^2) - (9*\sin(3*t)*\sin(2*t^2))/2 - 8*t^2*\sin(3*t)*\sin(2*t^2) + 12*t*\cos(3*t)*\cos(2*t^2), -(9*3^{1/2}*\sin(3*t))/2]$$

$$ir = [\cos(2*t^2), \sin(2*t^2), 0]$$

$$jr = [-\sin(2*t^2), \cos(2*t^2), 0]$$

$$kr = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$$

$$xr = \sin(3*t)/2$$

$$yr = 0$$

$$zr = (3^{1/2}*\sin(3*t))/2$$

$$rr\_ = [\sin(3*t)/2, 0, (3^{1/2}*\sin(3*t))/2]$$

$$Vr\_ = [(3*\cos(3*t))/2, 0, (3*3^{1/2}*\cos(3*t))/2]$$

$$Vr\_1 = (3*\cos(3*t))/2$$

$$Vr\_2 = 0$$

$$Vr\_3 = (3*3^{1/2}*\cos(3*t))/2$$

$$Vr\_a = [(3*\cos(3*t)*\cos(2*t^2))/2, (3*\cos(3*t)*\sin(2*t^2))/2, (3*3^{1/2}*\cos(3*t))/2]$$

$$Wr\_a = [-(9*\sin(3*t)*\cos(2*t^2))/2, -(9*\sin(3*t)*\sin(2*t^2))/2, -(9*3^{1/2}*\sin(3*t))/2]$$

$$Ve\_ = [-2*t*\sin(3*t)*\sin(2*t^2), 2*t*\sin(3*t)*\cos(2*t^2), 0]$$

$$Vabc\_ = [(3*\cos(3*t)*\cos(2*t^2))/2 - 2*t*\sin(3*t)*\sin(2*t^2), (3*\cos(3*t)*\sin(2*t^2))/2 + 2*t*\sin(3*t)*\cos(2*t^2), (3*3^{1/2}*\cos(3*t))/2]$$

$$dV = [0, 0, 0]$$



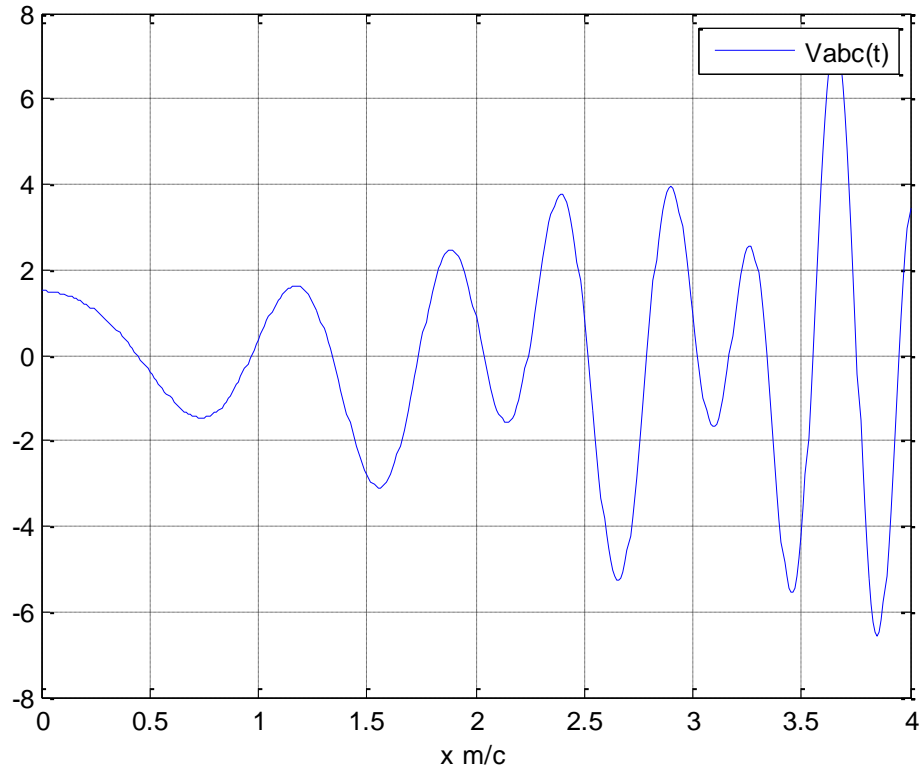
$$W_{e\_} = [-2\sin(3t)\sin(2t^2) - 8t^2\sin(3t)\cos(2t^2), 2\sin(3t)\cos(2t^2) - 8t^2\sin(3t)\sin(2t^2), 0]$$

$$W_{c\_a} = [-12t\cos(3t)\sin(2t^2), 12t\cos(3t)\cos(2t^2), 0]$$

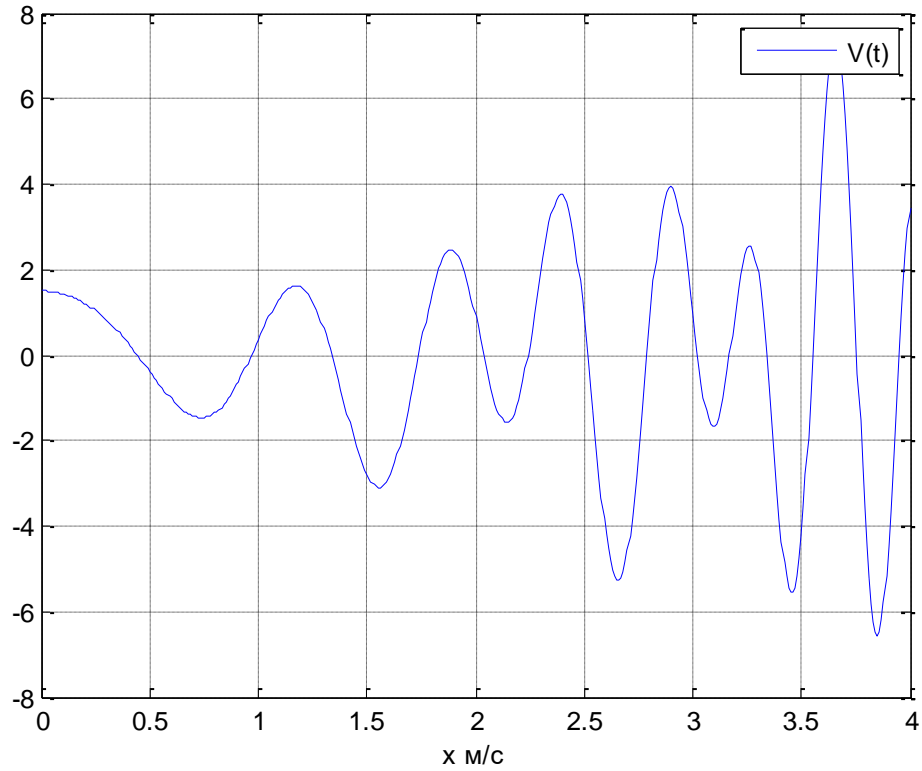
$$W_{abc\_} = [- (9\sin(3t)\cos(2t^2))/2 - 2\sin(3t)\sin(2t^2) - 8t^2\sin(3t)\cos(2t^2) - 12t\cos(3t)\sin(2t^2), 2\sin(3t)\cos(2t^2) - (9\sin(3t)\sin(2t^2))/2 - 8t^2\sin(3t)\sin(2t^2) + 12t\cos(3t)\cos(2t^2), -(9\cdot 3^{1/2}\sin(3t))/2]$$

$$dW = [0, 0, 0]$$

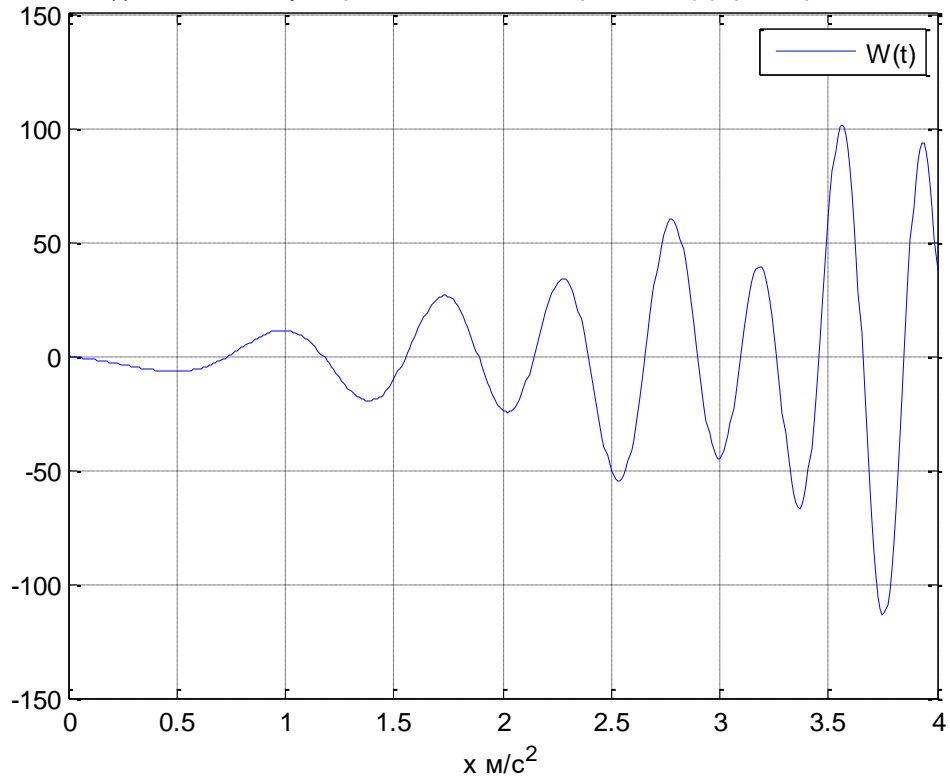
Vabc(t) - абсолютная скорость, найденная с помощью теоремы сложения скоростей



V(t) - абсолютная скорость, найденная просто дифференцированием



W(t) абсолютное ускорение, найденное просто дифференцированием



Wabc(t) - абсолютное ускорение, найденное с помощью теоремы сложения ускорений

