

УДК 303.094.7

Визуализация процесса движения колесной машины в MATLAB/Simulink

Климович Артем Михайлович

klimovicham@student.bmstu.ru

Сидоров Владимир Николаевич

sidorov_vn@bmstu.ru

КФ МГТУ им. Н.Э. Баумана, Калуга, Россия

Статья посвящена визуализации процесса движения колесной машины в среде MATLAB/Simulink. Приведен процесс подготовки данных, а также описание текста программы, содержащей координаты вершин поверхностей, образующих корпус автомобиля, колесо, опорную поверхность, а также переменную, содержащую координаты центра масс автомобиля в каждый момент времени для изображения траектории движения. Визуализация движения позволяет комплексно оценить параметры движения машины и проверить корректность используемых начальных условий.

Ключевые слова: матрица, траектория, координаты, MATLAB/Simulink

Программа, позволяющая визуализировать процесс движения колесной машины по результатам проведенных вычислительных экспериментов, написана на внутреннем языке программирования среды MATLAB.

Визуализация движения позволяет проверить корректность используемых начальных условий, например, определяющих геометрию автомобиля, комплексно оценить параметры движения машины.

На этапе подготовки значения параметров движения импортируются в рабочую область Workspace в виде матрицы с помощью блока «To Workspace». В блоке «To Workspace» хранятся следующие данные:

- координаты центра масс автомобиля в НСК (P_{c2});
- матрица перехода из ПСК в НСК, преобразованная с помощью блока «Reshape» (в настройке блока в качестве параметра Output dimensionality (выходная размерность) выбрать 1-Darray) в матрицу с размерностью 1×9 для хранения;
- вектор координат точек центров колес в НСК (R_{k2});
- вектор координат точек крепления подвесок к кузову в НСК;
- вектор координат точек крепления подвесок к колесу в НСК (формируется с учетом вертикальных перемещений колес z_k);
- углы поворота колес (α_k).

Первоначально в программе задаются переменные, содержащие координаты вершин поверхностей, образующих корпус автомобиля (A_{vagr_0}), колесо (A_{kat_0}), опорную поверхность (x_{gr} , y_{gr} , z_{gr}).

Координаты точек корпуса задаются относительно центра масс автомобиля в ПСК (рис. 1).

Колесо изображается многоугольником, координаты вершин которого задаются относительно центра колеса в МПСК. Приближенность данного многоугольника к окружности можно повысить, уменьшив шаг цикла, заполняющего матрицу A_kat_0 (рис. 2).

```
A_vagr_0{1} = [3; 0.85; -0.4]; %передняя нижняя левая точка кузова
A_vagr_0{2} = [-3.4; 0.85; -0.4]; %задняя нижняя левая точка кузова
A_vagr_0{3} = [-3.4; 0.85; 0.5]; %задняя верхняя левая точка кузова
A_vagr_0{4} = [1.4; 0.85; 0.5]; %задняя левая верхняя точка кабины
A_vagr_0{5} = [3; 0.85; 2]; %передняя верхняя левая точка кузова
A_vagr_0{6} = [3; -0.85; -0.4]; %передняя нижняя правая точка кузова
A_vagr_0{7} = [-3.4; -0.85; -0.4]; %задняя нижняя правая точка кузова
A_vagr_0{8} = [-3.4; -0.85; 0.5]; %задняя верхняя правая точка кузова
A_vagr_0{9} = [1.4; -0.85; 0.5]; %задняя правая верхняя точка кабины
A_vagr_0{10} = [3; -0.85; 2]; %передняя верхняя правая точка кузова
A_vagr_0{11} = [1.4; 0.85; 2]; %задняя верхняя левая точка кабины
A_vagr_0{12} = [1.4; -0.85; 2]; %задняя верхняя правая точка кабины
```

Рис. 1. Программный код, задающий координаты точек корпуса относительно центра масс автомобиля в ПСК

```
i = 1;
for alpha = 0:(pi/16):(2*pi-pi/16)
A_kat_0(i, 1:3) = [rk*cos(alpha) 0 rk*sin(alpha)];
i = i+1;
end
x_gr = [-1000 1000];
y_gr = [-1000 1000];
z_gr = [0 0; 0 0];
Далее необходимо создать графическое окно с помощью следующей функции:
set(gcf, 'DoubleBuffer', 'on');
Имитирующая опорное основание поверхность создается с помощью функции
«surf»:
surf(x_gr, y_gr, z_gr, 'FaceColor', [0.7 0.7 0.7]);
Значение [0.7 0.7 0.7] свойства «FaceColor» определяет цвет поверхности.
```

Рис. 2. Программный код, заполняющий матрицу A_kat_0

Присвоим значения отдельным переменным, соответствующим первому шагу моделирования, поэтому $i = 1$ (рис. 3).

Для отображения в графическом окне непосредственно автомобиля воспользуемся настройками отображения осей с указателем `mult` (рис. 4).

Поверхности, образующие колеса строятся той же функцией, что и кузов. В этом же цикле строятся линии, показывающие деформацию подвески колес (рис. 5).

Для изображения траектории движения автомобиля подготавливается переменная, содержащая координаты его центра масс в каждый момент времени, а также указатель функции, рисующей линию по данным координатам (рис. 6).

Создаются переменные, которые понадобятся в дальнейшем для организации внутреннего цикла: `drawtime = 0; animtime = cputime`.

```

i = 1;
P_vagr_2 = animdata_p(i, 1:3); % вектор координат ЦМ автомобиля
B_vagr = [animdata_p(i, 4:6) animdata_p(i, 7:9) animdata_p(i, 10:12)]; % матрица
перехода из неподвижной в подвижную СК
R_k_vagr_2 = animdata_p(i, 13:30); % матрица координат центров колес в НСК
R_tkp3 = animdata_p(i, 31:48); % матрица координат точек крепления подвесок к
колесам в НСК
R_tkp0 = animdata_p(i, 49:66); % матрица координат точек крепления подвесок к
кузову в НСК
alpha_k = animdata_p(i, 67:72); % матрица углов поворота колес
for j = 1:6
B_k = B_vagr*[cos(alpha_k(j)) -sin(alpha_k(j)) 0;
sin(alpha_k(j)) cos(alpha_k(j)) 0; 0 0 1];
A_kp_x2{j} = B_k(1, 1)*A_kat_0(:, 1)+B_k(1, 2)*A_kat_0(:, 2)+B_k(1, 3)*A_kat_0(:,
3)+R_k_vagr_2(3*(j-1)+1);
A_kp_y2{j} = B_k(2, 1)*A_kat_0(:, 1)+B_k(2, 2)*A_kat_0(:, 2)+B_k(2, 3)*A_kat_0(:,
3)+R_k_vagr_2(3*(j-1)+2);
A_kp_z2{j} = B_k(3, 1)*A_kat_0(:, 1)+B_k(3, 2)*A_kat_0(:, 2)+B_k(3, 3)*A_kat_0(:,
3)+R_k_vagr_2(3*(j-1)+3);
end
% матрица координат точек кузова в НСК

```

Рис. 3. Присвоение значений отдельным переменным

```

set(mult, 'DataAspectRatio', [1 1 1], 'CameraPosition', (P_vagr_2+[-10; -10; 10]),
'CameraTarget', P_vagr_2, 'CameraViewAngle', 50);
Кузов автомобиля строится при помощи функции «patch». Данная функция
создает один или несколько заполненных многоугольников.
vagr = patch('Faces', [1 2 3 4 11 5; 6 7 8 9 12 10;
5 1 6 10 NaN NaN; 1 2 7 6 NaN NaN; 2 3 8 7 NaN NaN;
3 4 9 8 NaN NaN; 4 11 12 9 NaN NaN; 5 10 12 11 NaN NaN],
'Vertices', vert_vagr, 'EdgeColor', [0 0 0],
'FaceVertexCData', [0.7 0.7 0.7; 0.7 0.7 0.7; 0.7 0.7 0.7; 0.7 0.7 0.7; 0.7 0.7 0.7;
0.7 0.7 0.7], 'FaceColor', [0 0.5 0], 'LineWidth', 0.5);

```

Рис. 4. Программный код с настройками отображения осей с указателем mult

```

for j = 1:6
kp{j} = patch(A_kp_x2{j}, A_kp_y2{j}, A_kp_z2{j}, [0.5 0.5 0.5]);
%отрисовка поверхностей, образующих колеса автомобиля
podv{j} = line([R_tkp3(3*(j-1)+1)
R_tkp0(3*(j-1)+1)], [R_tkp3(3*(j-1)+2)
R_tkp0(3*(j-1)+2)], [R_tkp3(3*(j-1)+3) R_tkp0(3*(j-1)+3)], 'Color', [0 0 1], 'LineWidth', 1);
%отрисовка линий, отображающих перемещение колес относительно корпуса
end

```

Рис. 5. Программный код с настройками отображения осей с указателем mult

```

vert_tr = animdata_p(:, 1:3);
tr = patch('Faces', [1], 'Vertices', vert_tr,
'EdgeColor', [1 0 0], 'LineWidth', 0.5);
fac_tr = [];

```

Рис. 6. Программный код для изображения траектории движения автомобиля

Команда `cputime` возвращает общее процессорное время (в секундах), используемое программой с момента ее начала.

Создается цикл `for` с граничным значением счетчика равным количеству строк матрицы `animdata_p`. Так как для первого шага интегрирования значения всех параметров уже были записаны в переменные, то первое значение счетчика цикла — 2. В цикле используются ранее описанные функции построения поверхностей, определяющих корпус и колеса автомобиля (рис. 7).

Для замедления процесса изображения движения автомобиля вводится следующий цикл (рис. 8):

```
for i = 2 : size(animdata_p, 1)
    P_vagp_2 = animdata_p(i, 1:3)';
    B_vagp = [animdata_p(i, 4:6)' animdata_p(i, 7:9)' animdata_p(i, 10:12)'];
    R_k_vagp_2 = animdata_p(i, 13:30)';
    R_tkp3 = animdata_p(i, 31:48)';
    R_tkp0 = animdata_p(i, 49:66)';
    alpha_k = animdata_p(i, 67:72)';
    vert_vagp = [(B_vagp*A_vagp_0{1}+P_vagp_2)';
    (B_vagp*A_vagp_0{2}+P_vagp_2)';
    (B_vagp*A_vagp_0{3}+P_vagp_2)';
    (B_vagp*A_vagp_0{4}+P_vagp_2)';
    (B_vagp*A_vagp_0{5}+P_vagp_2)';
    (B_vagp*A_vagp_0{6}+P_vagp_2)';
    (B_vagp*A_vagp_0{7}+P_vagp_2)';
    (B_vagp*A_vagp_0{8}+P_vagp_2)';
    (B_vagp*A_vagp_0{9}+P_vagp_2)';
    (B_vagp*A_vagp_0{10}+P_vagp_2)';
    (B_vagp*A_vagp_0{11}+P_vagp_2)';
    (B_vagp*A_vagp_0{12}+P_vagp_2)'];
    for j = 1:6
        B_k = B_vagp*[cos(alpha_k(j)) -sin(alpha_k(j)) 0;
        sin(alpha_k(j)) cos(alpha_k(j)) 0; 0 0 1];
        A_kp_x2{j} = B_k(1, 1)*A_kat_0(:, 1)+B_k(1, 2)*A_kat_0(:,
        2)+B_k(1, 3)*A_kat_0(:,
        3)+R_k_vagp_2(3*(j-1)+1); A_kp_y2{j} = B_k(2, 1)*A_kat_0(:, 1)+B_k(2,
        2)*A_kat_0(:, 2)+B_k(2, 3)*A_kat_0(:,
        3)+R_k_vagp_2(3*(j-1)+2); 284 A_kp_z2{j} = B_k(3, 1)*A_kat_0(:, 1)+B_k(3,
        2)*A_kat_0(:, 2)+B_k(3, 3)*A_kat_0(:,
        3)+R_k_vagp_2(3*(j-1)+3);
    end
end
```

Рис. 7. Программный код для описания функций построения поверхностей, определения корпуса и колеса автомобиля

```
drawtime = cputime - drawtime;
tik = cputime; while (cputime - tik) > (1*0.035 - drawtime)
end
drawtime = cputime;
drawnow
end
animtime = cputime - animtime;
```

Рис. 8. Цикл для замедления процесса движения автомобиля

Таким образом, в среде MATLAB/Simulink возможно визуализировать процесс движения колесной машины. Для этого необходимо создать программу, в которой задаются переменные, содержащие координаты вершин поверхностей, образующих корпус автомобиля, колесо, опорную поверхность. Также необходимо подготовить переменную, содержащую координаты центра масс автомобиля в каждый момент времени. Возможность визуализации движения колесной машины позволяет проверить корректность используемых начальных условий, например, определяющих геометрию автомобиля и комплексно оценить параметры движения машины.

Литература

- [1] Афанасьев Б.А., Белоусов Б.Н., Гладов Г.И. *Проектирование полноприводных колесных машин*. Москва, Изд-во МГТУ им. Н.Э. Баумана, 2008.
- [2] Смирнов Г.А. *Теория движения колесных машин*. Москва, Машиностроение, 1990, 352 с.
- [3] Афанасьев Б.А. *Моделирование систем колесных машин*. Москва, Изд-во МГТУ им. Н.Э. Баумана, 1997, 30 с.
- [4] Ларин В.В. *Теория движения полноприводных колесных машин*. Москва, Изд-во МГТУ им. Н.Э. Баумана, 2010, 391 с.

Visualization of the process of movement of a wheeled vehicle in MATLAB/Simulink

Klimovich A.K. klimovicham@student.bmstu.ru

Sidorov V.N. sidorov_vn@bmstu.ru.

BMSTU (KB), Kaluga, Russia

The article is devoted to visualization of the process of movement of a wheeled machine in the Matlab/Simulink environment. The study shows the process of data preparation, as well as a description of the text of the program containing the coordinates of the vertices of the surfaces forming the car body, wheel, support surface, as well as a variable containing the coordinates of the center of mass of the car at each time to image the trajectory of movement. Motion visualization allows you to comprehensively evaluate the parameters of the machine movement and check the correctness of the initial conditions used.

Keywords: matrix, trajectory, coordinates, MATLAB/Simulink