Обработка данных в ЛР УРМУ 1-3

1. Прочитать данные из файла.
2. Проинтегрировать данные по скоростям Vx,Vy,dpsi для получения реальных координат.
3. Определить запаздывание данных о реальной траектории относительно идеальной.

Есть 2 массива данных о координатах и скоростях идеальный и реальный с привязкой по времени.

Находим рассогласование по координатам и скоростям dX,dY,delPsi, dVx,dVy,ddPsi.

Где - идеальная скорость на i шаге;

- реальная скорость из файла данных на i шаге;

- идеальная координата на i шаге;

- реальная координата из файла данных на i шаге; остальные параметры аналогично.

Каждая из описанных переменных это массив размерностью T/dt+1 где T – время моделирования; dt – шаг счета.

Зададим модель ошибок движения по траектории.

*(\*)*

dtzap – неизвестное время запаздывания.

Аналогичные формулы для остальных координат.

Пояснения по нахождению интеграла погрешности по скоростям

- идеальная скорость на i шаге;

- реальная скорость из файла данных на i шаге;

Модель ошибок в виде уравнений (\*) можно записать в матричной форме

- блочная матрица столбец свободных членов;

- блочная матрица матрица коэффициентов;

*Всего n блоков по 3 строчки где n количество точек при движении по траектории.*

*Используем псевдообратные матрицы*

*После определения времени запаздывания и интеграла погрешностей по скоростям необходимо выполнить обратную задачу получить координаты используя следующую формулу*

*Сравниваем восстановленные координаты и идеальные точность должна повысится.*

2 модель погрешностей определение погрешности движения в виде случайной величины с СКО и матожиданием.

Для массивов dX,dY,delPsi определяем СКО и матожидание.

Для массивов dX\_iv,dY\_iv,delPsi\_iv определяем СКО и матожидание.

dX\_iv,dY\_iv,delPsi\_iv – прямые разности между идеальными данными и восстановленными.

И сравнить СКО и матожидание для исходного варианта и варианта с определенной моделью ошибок. (для разности iv СКО и матожидание должны быть меньше чем для простой разности, если это не так, то ищи ошибки).

%из файла полученного от робота вытащим в массив

%и разобьем каждый столбец на переменные

inputDataRob = dlmread('D:\UMRU\lab1\S12b\_19\_brig\_1.txt');

t = inputDataRob(:,1);

omega\_1 = inputDataRob(:,2);

omega\_2= inputDataRob(:,3);

omega\_3= inputDataRob(:,4);

omega\_4= inputDataRob(:,5);

M1= inputDataRob(:,6);

M2= inputDataRob(:,7);

M3= inputDataRob(:,8);

M4= inputDataRob(:,9);

Vx\_o= inputDataRob(:,10);

Vy\_o= inputDataRob(:,11);

dpsi= inputDataRob(:,12);

x0= inputDataRob(:,13);

y0= inputDataRob(:,14);

psi = inputDataRob(:,15);

%откроем файл с данными по идеальным перемещениям

inputDataIdeal = dlmread('D:\UMRU\lab1\perfect\_squareUV1.txt');

Xid = inputDataIdeal(:,1);

Yid = inputDataIdeal(:,2);

psi\_id = inputDataIdeal(:,3);

%откроем файл с данными по идеальным скоростям

inputDataIdSpeed = dlmread('D:\UMRU\lab1\square.txt');

Vxid = inputDataIdSpeed(:,1);

Vyid = inputDataIdSpeed(:,2);

Omid = inputDataIdSpeed(:,3);

%Найдем рассогласование по координатам

dX = Xid-x0;

dY = Yid -y0;

delPsi = psi\_id - psi;

dVx = Vxid - Vx\_o;

dVy = Vyid - Vy\_o;

ddPsi = Omid - dpsi;

%Получим скоростные интегралы для каждой из итераций скоростей

integraDVxdt(1,1) = dVx(1,1);

for i = 2:1:length(dVx)

integraDVxdt(i,1) = dVx(i,1)+integraDVxdt(i-1,1);

end

integraDVxdt = integraDVxdt\*0.1;

integraDVydt(1,1) = dVy(1,1);

for i = 2:1:length(dVx)

integraDVydt(i,1) = dVy(i,1)+integraDVydt(i-1,1);

end

integraDVydt = integraDVydt\*0.1;

integraDdpsi(1,1) = ddPsi(1,1);

for i = 2:1:length(dVx)

integraDdpsi(i,1) = ddPsi(i,1)+integraDdpsi(i-1,1);

end

integraDdpsi = integraDdpsi\*0.1;

%зададим В - блочную матрицу столбцов свободных членов

B = zeros(length(dVx)\*3,1);

k = 1;

for i = 1:3:((length(dVx)\*3))

B(i,1) = dX(k,1)- integraDVxdt(k,1);

B(i+1,1) = dY(k,1)- integraDVydt(k,1);

B(i+2,1) = delPsi(k,1)- integraDdpsi(k,1);

k = k + 1;

end

%зададим А - блочную матрицу коэффициентов

A = zeros(length(dVx)\*3,1);

k = 1;

for i = 1:3:((length(dVx)\*3))

A(i,1) = Vxid(k,1);

A(i+1,1) = Vyid(k,1);

A(i+2,1) = Omid(k,1);

k=k+1;

end

%определим время запаздывания

dtzap = inv(A'\*A)\*A'\*B;

%найдем восстановленные координаты

for i = 1:1:length(dVx)

Xvost(i,1) = x0(i,1)+Vxid(i,1)\*dtzap+integraDVxdt(i,1);

end

for i = 1:1:length(dVx)

Yvost(i,1) = y0(i,1)+Vyid(i,1)\*dtzap+integraDVydt(i,1);

end

hold on;

grid on;

for i = 1:1:length(dVx)

plot(x0,y0,'r');

plot(Xid,Yid,'b');

plot(Xvost,Yvost,'g');

end

xlabel('X,м');

ylabel('Y,м');

title('Движение мекано-платформы youBot');

legend({'Движение платформы снятое датчиками','Идеальное предполагаемое движение плафтормы','Восстановленное движение платформы'},'Location','southwest');