Contents


```
%Matlab file for Glass/Nexus(Pilot) Sim data analysis (from com.csv)
%version 3.0:
    %add trigger_y in the condition of identifying brake event,
    %txbk: post wnd starts from txbk_off/rta_txbk whichever comes later
    %revise checks for rta_txbk
    %revise pre/post formula
    %add codes for "no tx, only txbk" dataset
%version 4.0: revise data_end logic for the last event(if tx_off>=data_end,data_end=double(Count)...etc)
%version 5.0: gr_adj=4, FDadj=5, revise TTC &FD
%version 6.0: change pre to 2 sec and post to 2 sec
%version 7.0: pre: get 2 sec before pace car change lane
%             FD min & avg: FD=0 if y offset-FDadj<0
%             revise lane_pos: use difference to lane centers
%version 8.0: Lane position: use veh_x instead of lane_pos. use sd(LPSD) instead of rms
%             LPSD=sqrt(sumsqr(veh_x-mean(veh_x))/N-1)
%             add LP_EXC, LP_EXC_time
%version "apple": for txbk, use "t to rta" window, same as bk
%Names:
    % t = brake event (t1,t2)
    % tx=text only event, bk=brake only event, txbk=text&brake event
    % gr = release gas
    % bkon = brake onset(touch brake)
    % bkoff = brake offset(release brake)
    % d = gr or bkon whichever comes first
    % m = min speed during a brake event
    % rta = return to acceleration
```

Initiation

```
clear all % clear all objects in Workspace
close all % delete all unhidden figures
clc       % clear Command Window

Output_flatfile_1='Output\Output_Glass_Flat1_Lane_apple.csv';
Output_flatfile_2='Output\Output_Glass_Flat2_Lane_apple.csv';
Output_indexfile='Output\Output_Glass_index_Lane_apple.csv';
Output_logfile='Output\Output_Glass_log_Lane_apple.csv';
row_adj=-1; % adjust header rows (row difference in excel & matlab)
t_speed=4470; % lead car speed before brake event; *originally set at 4480. Change for Ss#67_G
t_speed_decr=15; % minimun lead car speed decrease at brake event
end_speed=4000; % in order to decide end of data before participant vehicle stops
pre_sec = 2; % Pre(baseline) window: 2s before the pre-event lane change
post_sec = 2; % Post window: 2s after RTA or screen_off
% if time between events are not sufficiant for both, "pre" gets the full time and "post" gets the remaining time
row_sec = 60; % every row is 1/60 of a second
FDadj=5; %adjustment for vehicle length(4.97)
Multiplier=1500; % The distance multiplier is used to calculate miles
gr_adj=4; % adj. for release gas due to system glitch(?). use 4 instead of 0 as complete gas pedal release.
trigger_y1=4270;
trigger_y2=6223;
trigger_y3=8261;
trigger_adj=200;
trigger_delay=1900; %1800ms delay+100ms buffer(transmission time etc.)
lane_L=7516.85; %1st event lane center, left lane
```

```
lane_C=7520.3; %beginning & 2nd event lane center, center lane
lane_R=7523.7; %3rd event lane center, right lane
lane_width=3.4;
lane_LL=lane_C-lane_width*1.5; %7515.2
lane_CL=lane_C-lane_width/2; %7518.6
lane_CR=lane_C+lane_width/2; %7522
lane_RR=lane_R+lane_width/2; %7525.4
car_width=1.81; % participant car width

% Construct a questdlg with three options
choice = questdlg('Which file(s) would you like to run?', 'Data Processing Methods', 'Choose one file','Run all files','Run all newly-added file(s)',...
    'Choose one file');

% Handle response
switch choice
```

**Run one file

```
case 'Choose one file'
```

```
% Import data
fid=uigetfile('.csv');
SimData = importdata(fid,',',1);
%SimData = importdata(uigetfile('.csv'),',',1);

% Create new variables in the base workspace from those fields.
for i = 1:size(SimData.colheaders, 2)
    assignin('base', genvarname(SimData.colheaders{i}), SimData.data(:,i));
end

clear i

Count = size(system_time,1);
%creates a variable equal to the amount of points of data collected.
%This is used to initialize variables to reduce computing time.
```

Subject & Condition

```
Ss = str2double(fid(3:4));

if ~isempty(strfind(fid,'glass')), Device = 0;
else if ~isempty(strfind(fid,'nexus')), Device = 1;
        %else if ~isempty(strfind(fid,'test')), Condition = 2;
            else Device=NaN; %end
    end
end

if isnan(Device), device_check=1; msgbox('Check Device.');
else device_check=0;
if Device==0; load('RefData_G.mat'); ref=find(Ss_G==Ss);
else load('RefData_N.mat'); ref=find(Ss_N==Ss); end
trigger=trigger(ref);
txbk_initiate=txbk_initiate(ref)+row_adj; txbk_readthink=txbk_readthink(ref)+row_adj; txbk_reply=txbk_reply(ref)+row_adj; txbk_send=txbk_send(ref)+row_adj; txbk_of
tx_initiate=tx_initiate(ref)+row_adj; tx_readthink=tx_readthink(ref)+row_adj; tx_reply=tx_reply(ref)+row_adj; tx_send=tx_send(ref)+row_adj; tx_off=tx_off(ref)+row_
end
```

Windows

```
t = zeros(Count,1);
for n=3:Count-1;
    if lead_speed(n-1)>=t_speed && lead_speed(n-1)-lead_speed(n)>=t_speed_decr && lead_speed(n-2)<=lead_speed(n-1) && lead_speed(n) >= lead_speed(n+1)...
            &&(system_time(n)-system_time(txbk_readthink)<trigger_delay || abs(lead_y(n)-trigger_y1)<=trigger_adj || ...
            abs(lead_y(n)-trigger_y2)<=trigger_adj || abs(lead_y(n)-trigger_y3)<=trigger_adj);
        t(n,1)=1;
    end
end
t_count=sum(t);

if t_count~=2, msgbox('Check no. of brake events'); t_check=1;
else t_check=0; end

if isnan(Device) || t_check==1;
    msgbox('check problem log');
    Output_log=num2cell([Ss,Device,device_check,t_check,NaN]);
    dlmwrite(strcat(pwd,filesep,Output_logfile),Output_log,'-append');
else
```

```
resize = zeros(Count,1);
for n=1:Count;
    if veh_speed(n) > end_speed;
        resize(n,1)=1;
    end
end
data_end=find(resize,1,'last');
```

```
FD=zeros(Count,1);
for n=1:Count;
   if lead_y(n)-veh_y(n)-FDadj>=0;
       FD(n)=lead_y(n)-veh_y(n)-FDadj;
   end
end

TTC=zeros(Count,1);
for n=1:Count;
   if lead_y(n)-veh_y(n)-FDadj>=0 && veh_speed(n)>lead_speed(n);
       TTC(n)=(((lead_y(n)-veh_y(n)-FDadj)/Multiplier)./(veh_speed(n)/100-lead_speed(n)/100))*360;
   else if veh_speed(n)<=lead_speed(n);
           TTC(n)=NaN;
       else TTC(n)=0;
       end
   end
end

bkoff = zeros(Count,1);
for n=2:Count-1;
    if brake_ped(n) == 0 & brake_ped(n+1) == 0 & brake_ped(n-1) > 0;
        bkoff(n,1)=1;
    end
end

if isempty(find(collision_vehicle, 1));
    Collision=0; else Collision=1;end

ped_check = zeros(Count,1);
for n=1:Count;
    if brake_ped(n)>0 && accel_ped(n)>0;
        ped_check(n,1)=1;
    end
end

excursion=zeros(Count,1);
```

Trigger 1

```
if trigger==1, %tx->txbk->bk
```

```
    t_txbk=find(t,1,'first'); t_bk=find(t,1,'last');
    tx_y=trigger_y1; txbk_y=trigger_y2;
    %tx
    if isnan(tx_off);
        tx_pre_ck=1; tx_pre_rms_steer=NaN; tx_pre_LPSD=NaN; tx_pre_LPEXC=NaN; tx_pre_LPEXC_time=NaN; tx_pre_avg_speed=NaN; tx_pre_avg_FD=NaN; tx_pre_TTC_min=NaN; t
        tx_rms_steer=NaN; tx_LPSD=NaN; tx_LPEXC=NaN; tx_LPEXC_time=NaN; tx_avg_speed=NaN; tx_avg_FD=NaN; tx_TTC_min=NaN; tx_FD_min=NaN; tx_collision=0;
        tx_initiate_rms_steer=NaN; tx_initiate_LPSD=NaN; tx_initiate_LPEXC=NaN; tx_initiate_LPEXC_time=NaN; tx_initiate_avg_speed=NaN; tx_initiate_avg_FD=NaN; tx_i
        tx_readthink_rms_steer=NaN; tx_readthink_LPSD=NaN; tx_readthink_LPEXC=NaN; tx_readthink_LPEXC_time=NaN; tx_readthink_avg_speed=NaN; tx_readthink_avg_FD=NaN
        tx_reply_rms_steer=NaN; tx_reply_LPSD=NaN; tx_reply_LPEXC=NaN; tx_reply_LPEXC_time=NaN; tx_reply_avg_speed=NaN; tx_reply_avg_FD=NaN; tx_reply_TTC_min=NaN; t
        tx_send_rms_steer=NaN; tx_send_LPSD=NaN; tx_send_LPEXC=NaN; tx_send_LPEXC_time=NaN; tx_send_avg_speed=NaN; tx_send_avg_FD=NaN; tx_send_TTC_min=NaN; tx_send
        tx_post_ck=1; tx_post_rms_steer=NaN; tx_post_LPSD=NaN; tx_post_LPEXC=NaN; tx_post_LPEXC_time=NaN; tx_post_avg_speed=NaN; tx_post_avg_FD=NaN; tx_post_TTC_mi

        txbk_pre2=find(lead_x(tx_initiate:txbk_initiate)~=lane_L,1,'first')+tx_initiate-1;
        if txbk_pre2-tx_initiate>pre_sec*row_sec && lead_x(tx_initiate)==lane_L;
            txbk_pre=txbk_pre2-pre_sec*row_sec; txbk_pre_ck=0;
        else txbk_pre=tx_initiate+1; txbk_pre_ck=1;
        end
    else
    %--tx_pre, Lane_C
    tx_pre2=find(lead_x(find(lead_x(1:tx_initiate)==lane_C,1,'first'):tx_initiate)~=lane_C,1,'first')+find(lead_x(1:tx_initiate)==lane_C,1,'first')-1;
    if tx_pre2>pre_sec*row_sec;
        tx_pre=tx_pre2-pre_sec*row_sec; tx_pre_ck=0; else tx_pre=1; tx_pre_ck=1;
    end

    if veh_x(tx_pre)>(lane_CR-car_width/2) || veh_x(tx_pre)<(lane_CL+car_width/2); excursion(tx_pre)=1; end
    for n=tx_pre+1:tx_pre2;
        if (veh_x(n)>(lane_CR-car_width/2) && veh_x(n-1)<=(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2) && veh_x(n-1)>=(lane_CL+car_width/2));
            excursion(n)=1;
        else if (veh_x(n)>(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2)); excursion(n)=2; end
        end
    end

    tx_pre_LPSD=sqrt(sumsqr(veh_x(tx_pre:tx_pre2)-mean(veh_x(tx_pre:tx_pre2)))/(tx_pre2-tx_pre));
    tx_pre_LPEXC=sum(excursion(tx_pre:tx_pre2)==1);
    tx_pre_LPEXC_time=sum(excursion(tx_pre:tx_pre2)~=0)/row_sec; %does not "sum", but "count" instead.
    tx_pre_rms_steer=sqrt(sumsqr(steer_wheel(tx_pre:tx_pre2))/(tx_pre2-tx_pre+1));
    tx_pre_avg_speed=mean(veh_speed(tx_pre:tx_pre2))/100;
    tx_pre_avg_FD=mean(FD(tx_pre:tx_pre2));
    tx_pre_TTC_min=min(TTC(tx_pre:tx_pre2));
    tx_pre_FD_min=min(FD(tx_pre:tx_pre2));
    tx_pre_wnd_length=system_time(tx_pre2)/1000-system_time(tx_pre)/1000;
    %--tx_during, Lane_L
    if veh_x(tx_initiate)>(lane_CL-car_width/2) || veh_x(tx_initiate)<(lane_LL+car_width/2); excursion(tx_initiate)=1; end
    for n=tx_initiate+1:tx_off;
        if (veh_x(n)>(lane_CL-car_width/2) && veh_x(n-1)<=(lane_CL-car_width/2)) || (veh_x(n)<(lane_LL+car_width/2) && veh_x(n-1)>=(lane_LL+car_width/2));
```

```
        excursion(n)=1;
    else if (veh_x(n)>(lane_CL-car_width/2)) || (veh_x(n)<(lane_LL+car_width/2)); excursion(n)=2; end
    end
end

tx_LPSD=sqrt(sumsqr(veh_x(tx_initiate:tx_off)-mean(veh_x(tx_initiate:tx_off)))/(tx_off-tx_initiate));
tx_LPEXC=sum(excursion(tx_initiate:tx_off)==1);
tx_LPEXC_time=sum(excursion(tx_initiate:tx_off)~=0)/row_sec; %does not "sum", but "count" instead.
tx_rms_steer=sqrt(sumsqr(steer_wheel(tx_initiate:tx_off))/(tx_off-tx_initiate+1));
tx_avg_speed=mean(veh_speed(tx_initiate:tx_off))/100;
tx_avg_FD=mean(FD(tx_initiate:tx_off));
tx_TTC_min=min(TTC(tx_initiate:tx_off));
tx_FD_min=min(FD(tx_initiate:tx_off));
if isempty(find(collision_vehicle(tx_initiate:tx_off), 1));
    tx_collision=0; else tx_collision=1;end
%--tx_initiate
tx_initiate_LPSD=sqrt(sumsqr(veh_x(tx_initiate:tx_readthink-1)-mean(veh_x(tx_initiate:tx_readthink-1)))/(tx_readthink-tx_initiate-1));
tx_initiate_LPEXC=sum(excursion(tx_initiate:tx_readthink-1)==1);
tx_initiate_LPEXC_time=sum(excursion(tx_initiate:tx_readthink-1)~=0)/row_sec; %does not "sum", but "count" instead.
tx_initiate_rms_steer=sqrt(sumsqr(steer_wheel(tx_initiate:tx_readthink-1))/(tx_readthink-tx_initiate));
tx_initiate_avg_speed=mean(veh_speed(tx_initiate:tx_readthink-1))/100;
tx_initiate_avg_FD=mean(FD(tx_initiate:tx_readthink-1));
tx_initiate_TTC_min=min(TTC(tx_initiate:tx_readthink-1));
tx_initiate_FD_min=min(FD(tx_initiate:tx_readthink-1));
if isempty(find(collision_vehicle(tx_initiate:tx_readthink-1), 1));
    tx_initiate_collision=0; else tx_initiate_collision=1;end
%--tx_readthink
if veh_x(tx_readthink)>(lane_CL-car_width/2) || veh_x(tx_readthink)<(lane_LL+car_width/2); excursion(tx_readthink)=1; end
tx_readthink_LPSD=sqrt(sumsqr(veh_x(tx_readthink:tx_reply-1)-mean(veh_x(tx_readthink:tx_reply-1)))/(tx_reply-tx_readthink-1));
tx_readthink_LPEXC=sum(excursion(tx_readthink:tx_reply-1)==1);
tx_readthink_LPEXC_time=sum(excursion(tx_readthink:tx_reply-1)~=0)/row_sec; %does not "sum", but "count" instead.
tx_readthink_rms_steer=sqrt(sumsqr(steer_wheel(tx_readthink:tx_reply-1))/(tx_reply-tx_readthink));
tx_readthink_avg_speed=mean(veh_speed(tx_readthink:tx_reply-1))/100;
tx_readthink_avg_FD=mean(FD(tx_readthink:tx_reply-1));
tx_readthink_TTC_min=min(TTC(tx_readthink:tx_reply-1));
tx_readthink_FD_min=min(FD(tx_readthink:tx_reply-1));
if isempty(find(collision_vehicle(tx_readthink:tx_reply-1), 1));
    tx_readthink_collision=0; else tx_readthink_collision=1;end
%--tx_reply
if veh_x(tx_reply)>(lane_CL-car_width/2) || veh_x(tx_reply)<(lane_LL+car_width/2); excursion(tx_reply)=1; end
tx_reply_LPSD=sqrt(sumsqr(veh_x(tx_reply:tx_send)-mean(veh_x(tx_reply:tx_send)))/(tx_send-tx_reply));
tx_reply_LPEXC=sum(excursion(tx_reply:tx_send)==1);
tx_reply_LPEXC_time=sum(excursion(tx_reply:tx_send)~=0)/row_sec; %does not "sum", but "count" instead.
tx_reply_rms_steer=sqrt(sumsqr(steer_wheel(tx_reply:tx_send))/(tx_send-tx_reply+1));
tx_reply_avg_speed=mean(veh_speed(tx_reply:tx_send))/100;
tx_reply_avg_FD=mean(FD(tx_reply:tx_send));
tx_reply_TTC_min=min(TTC(tx_reply:tx_send));
tx_reply_FD_min=min(FD(tx_reply:tx_send));
if isempty(find(collision_vehicle(tx_reply:tx_send), 1));
    tx_reply_collision=0; else tx_reply_collision=1;end
%--tx_send
if veh_x(tx_send+1)>(lane_CL-car_width/2) || veh_x(tx_send+1)<(lane_LL+car_width/2); excursion(tx_send+1)=1; end
tx_send_LPSD=sqrt(sumsqr(veh_x(tx_send+1:tx_off)-mean(veh_x(tx_send+1:tx_off)))/(tx_off-tx_send-1));
tx_send_LPEXC=sum(excursion(tx_send+1:tx_off)==1);
tx_send_LPEXC_time=sum(excursion(tx_send+1:tx_off)~=0)/row_sec; %does not "sum", but "count" instead.
tx_send_rms_steer=sqrt(sumsqr(steer_wheel(tx_send+1:tx_off))/(tx_off-tx_send));
tx_send_avg_speed=mean(veh_speed(tx_send+1:tx_off))/100;
tx_send_avg_FD=mean(FD(tx_send+1:tx_off));
tx_send_TTC_min=min(TTC(tx_send+1:tx_off));
tx_send_FD_min=min(FD(tx_send+1:tx_off));
if isempty(find(collision_vehicle(tx_send+1:tx_off), 1));
    tx_send_collision=0; else tx_send_collision=1;end
%--tx_post, Lane_L
txbk_pre2=find(lead_x(tx_off:txbk_initiate)~=lane_L,1,'first')+tx_off-1;
if txbk_pre2-tx_off>pre_sec*row_sec && lead_x(tx_off)==lane_L;
    txbk_pre=txbk_pre2-pre_sec*row_sec; txbk_pre_ck=0;
    if txbk_pre-tx_off-1>post_sec*row_sec;
    tx_post=tx_off+post_sec*row_sec; tx_post_ck=0; else tx_post=txbk_pre-1; tx_post_ck=1; end

    if veh_x(tx_off+1)>(lane_CL-car_width/2) || veh_x(tx_off+1)<(lane_LL+car_width/2); excursion(tx_off+1)=1; end
    for n=tx_off+1+1:tx_post;
      if (veh_x(n)>(lane_CL-car_width/2) && veh_x(n-1)<=(lane_CL-car_width/2)) || (veh_x(n)<(lane_LL+car_width/2) && veh_x(n-1)>=(lane_LL+car_width/2));
        excursion(n)=1;
        else if (veh_x(n)>(lane_CL-car_width/2)) || (veh_x(n)<(lane_LL+car_width/2)); excursion(n)=2; end
      end
    end

    tx_post_LPSD=sqrt(sumsqr(veh_x(tx_off+1:tx_post)-mean(veh_x(tx_off+1:tx_post)))/(tx_post-tx_off-1));
    tx_post_LPEXC=sum(excursion(tx_off+1:tx_post)==1);
    tx_post_LPEXC_time=sum(excursion(tx_off+1:tx_post)~=0)/row_sec; %does not "sum", but "count" instead.
    tx_post_rms_steer=sqrt(sumsqr(steer_wheel(tx_off+1:tx_post))/(tx_post-tx_off));
    tx_post_avg_speed=mean(veh_speed(tx_off+1:tx_post))/100;
    tx_post_avg_FD=mean(FD(tx_off+1:tx_post));
    tx_post_TTC_min=min(TTC(tx_off+1:tx_post));
    tx_post_FD_min=min(FD(tx_off+1:tx_post));
    tx_post_wnd_length=system_time(tx_post)/1000-system_time(tx_off+1)/1000;

else txbk_pre=tx_off+1; txbk_pre_ck=1; tx_post=NaN; tx_post_ck=1;
    tx_post_rms_steer=NaN;
    tx_post_LPSD=NaN;
    tx_post_LPEXC=NaN;
```

```
    tx_post_LPEXC_time=NaN;
    tx_post_avg_speed=NaN;
    tx_post_avg_FD=NaN;
    tx_post_TTC_min=NaN;
    tx_post_FD_min=NaN;
    tx_post_wnd_length=NaN;
end
end
%txbk
%--txbk_pre, Lane_L
if veh_x(txbk_pre)>(lane_CL-car_width/2) || veh_x(txbk_pre)<(lane_LL+car_width/2); excursion(txbk_pre)=1; end
for n=txbk_pre+1:txbk_pre2;
    if (veh_x(n)>(lane_CL-car_width/2) && veh_x(n-1)<=(lane_CL-car_width/2)) || (veh_x(n)<(lane_LL+car_width/2) && veh_x(n-1)>=(lane_LL+car_width/2));
        excursion(n)=1;
    else if (veh_x(n)>(lane_CL-car_width/2)) || (veh_x(n)<(lane_LL+car_width/2)); excursion(n)=2; end
    end
end

txbk_pre_LPSD=sqrt(sumsqr(veh_x(txbk_pre:txbk_pre2)-mean(veh_x(txbk_pre:txbk_pre2)))/(txbk_pre2-txbk_pre));
txbk_pre_LPEXC=sum(excursion(txbk_pre:txbk_pre2)==1);
txbk_pre_LPEXC_time=sum(excursion(txbk_pre:txbk_pre2)~=0)/row_sec; %does not "sum", but "count" instead.
txbk_pre_rms_steer=sqrt(sumsqr(steer_wheel(txbk_pre:txbk_pre2))/(txbk_pre2-txbk_pre+1));
txbk_pre_avg_speed=mean(veh_speed(txbk_pre:txbk_pre2))/100;
txbk_pre_avg_FD=mean(FD(txbk_pre:txbk_pre2));
txbk_pre_TTC_min=min(TTC(txbk_pre:txbk_pre2));
txbk_pre_FD_min=min(FD(txbk_pre:txbk_pre2));
txbk_pre_wnd_length=system_time(txbk_pre2)/1000-system_time(txbk_pre)/1000;
%--txbk_reaction
n = t_txbk;
bkon_txbk=n;
while n<t_bk-1;
      n=n+1;
      bkon_txbk=bkon_txbk+1;
      if brake_ped(n) ~= 0 & brake_ped((n-3):(n-1)) == 0;
        break
      end
end
n = t_txbk;
gr_txbk=n;
while n<t_bk-3;
      n=n+1;
      gr_txbk=gr_txbk+1;
      if accel_ped((n-3):(n-1)) > gr_adj & accel_ped((n+1):(n+3)) <= gr_adj & accel_ped(n) <= gr_adj & veh_speed(n) > 0;
        break
      end
end
if gr_txbk >= t_bk-3 || accel_ped(t_txbk)<= gr_adj; gr_txbk = NaN; end
if bkon_txbk >= t_bk-1 || brake_ped(t_txbk)~=0; bkon_txbk = NaN; end
if isnan(gr_txbk) && isnan(bkon_txbk); d_txbk=NaN; else d_txbk=min(gr_txbk,bkon_txbk); end
m_txbk=min(veh_speed(max(t_txbk,d_txbk):t_bk-1));
rta_txbk=find(veh_speed(max(t_txbk,d_txbk):t_bk-1)==m_txbk, 1, 'last' )+max(t_txbk,d_txbk)-1;
%min_speed_txbk=find(veh_speed(max(t_txbk,d_txbk):t_bk-1)==m_txbk, 1, 'first' )+max(t_txbk,d_txbk)-1;
if isempty(find(bkoff(max(t_txbk,d_txbk):rta_txbk),1)), bkoff_txbk=NaN; else bkoff_txbk=find(bkoff(max(t_txbk,d_txbk):rta_txbk),1,'last')+max(t_txbk,d_txbk)-1;
%--txbk_during, Lane_C
if veh_x(t_txbk)>(lane_CR-car_width/2) || veh_x(t_txbk)<(lane_CL+car_width/2); excursion(t_txbk)=1; end
for n=t_txbk+1:rta_txbk;
    if (veh_x(n)>(lane_CR-car_width/2) && veh_x(n-1)<=(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2) && veh_x(n-1)>=(lane_CL+car_width/2));
        excursion(n)=1;
    else if (veh_x(n)>(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2)); excursion(n)=2; end
    end
end

txbk_LPSD=sqrt(sumsqr(veh_x(t_txbk:rta_txbk)-mean(veh_x(t_txbk:rta_txbk)))/(rta_txbk-t_txbk));
txbk_LPEXC=sum(excursion(t_txbk:rta_txbk)==1);
txbk_LPEXC_time=sum(excursion(t_txbk:rta_txbk)~=0)/row_sec; %does not "sum", but "count" instead.
txbk_rms_steer=sqrt(sumsqr(steer_wheel(t_txbk:rta_txbk))/(rta_txbk-t_txbk+1));
txbk_avg_speed=mean(veh_speed(t_txbk:rta_txbk))/100;
txbk_avg_FD=mean(FD(t_txbk:rta_txbk));
txbk_TTC_min=min(TTC(t_txbk:rta_txbk));
txbk_FD_min=min(FD(t_txbk:rta_txbk));
if isempty(find(collision_vehicle(t_txbk:rta_txbk), 1));
    txbk_collision=0; else txbk_collision=1;end
%--txbk_initiate
txbk_initiate_LPSD=sqrt(sumsqr(veh_x(txbk_initiate:txbk_readthink-1)-mean(veh_x(txbk_initiate:txbk_readthink-1)))/(txbk_readthink-txbk_initiate-1));
txbk_initiate_LPEXC=sum(excursion(txbk_initiate:txbk_readthink-1)==1);
txbk_initiate_LPEXC_time=sum(excursion(txbk_initiate:txbk_readthink-1)~=0)/row_sec; %does not "sum", but "count" instead.
txbk_initiate_rms_steer=sqrt(sumsqr(steer_wheel(txbk_initiate:txbk_readthink-1))/(txbk_readthink-txbk_initiate));
txbk_initiate_avg_speed=mean(veh_speed(txbk_initiate:txbk_readthink-1))/100;
txbk_initiate_avg_FD=mean(FD(txbk_initiate:txbk_readthink-1));
txbk_initiate_TTC_min=min(TTC(txbk_initiate:txbk_readthink-1));
txbk_initiate_FD_min=min(FD(txbk_initiate:txbk_readthink-1));
if isempty(find(collision_vehicle(txbk_initiate:txbk_readthink-1), 1));
    txbk_initiate_collision=0; else txbk_initiate_collision=1;end
%--txbk_readthink
if veh_x(txbk_readthink)>(lane_CR-car_width/2) || veh_x(txbk_readthink)<(lane_CL+car_width/2); excursion(txbk_readthink)=1; end
txbk_readthink_LPSD=sqrt(sumsqr(veh_x(txbk_readthink:txbk_reply-1)-mean(veh_x(txbk_readthink:txbk_reply-1)))/(txbk_reply-txbk_readthink-1));
txbk_readthink_LPEXC=sum(excursion(txbk_readthink:txbk_reply-1)==1);
txbk_readthink_LPEXC_time=sum(excursion(txbk_readthink:txbk_reply-1)~=0)/row_sec; %does not "sum", but "count" instead.
txbk_readthink_rms_steer=sqrt(sumsqr(steer_wheel(txbk_readthink:txbk_reply-1))/(txbk_reply-txbk_readthink));
txbk_readthink_avg_speed=mean(veh_speed(txbk_readthink:txbk_reply-1))/100;
txbk_readthink_avg_FD=mean(FD(txbk_readthink:txbk_reply-1));
```

```
txbk_readthink_TTC_min=min(TTC(txbk_readthink:txbk_reply-1));
txbk_readthink_FD_min=min(FD(txbk_readthink:txbk_reply-1));
if isempty(find(collision_vehicle(txbk_readthink:txbk_reply-1), 1));
    txbk_readthink_collision=0; else txbk_readthink_collision=1;end
%--txbk_reply
if veh_x(txbk_reply)>(lane_CR-car_width/2) || veh_x(txbk_reply)<(lane_CL+car_width/2); excursion(txbk_reply)=1; end
txbk_reply_LPSD=sqrt(sumsqr(veh_x(txbk_reply:txbk_send)-mean(veh_x(txbk_reply:txbk_send)))/(txbk_send-txbk_reply));
txbk_reply_LPEXC=sum(excursion(txbk_reply:txbk_send)==1);
txbk_reply_LPEXC_time=sum(excursion(txbk_reply:txbk_send)~=0)/row_sec; %does not "sum", but "count" instead.
txbk_reply_rms_steer=sqrt(sumsqr(steer_wheel(txbk_reply:txbk_send))/(txbk_send-txbk_reply+1));
txbk_reply_avg_speed=mean(veh_speed(txbk_reply:txbk_send))/100;
txbk_reply_avg_FD=mean(FD(txbk_reply:txbk_send));
txbk_reply_TTC_min=min(TTC(txbk_reply:txbk_send));
txbk_reply_FD_min=min(FD(txbk_reply:txbk_send));
if isempty(find(collision_vehicle(txbk_reply:txbk_send), 1));
    txbk_reply_collision=0; else txbk_reply_collision=1;end
%--txbk_send
if veh_x(txbk_send+1)>(lane_CR-car_width/2) || veh_x(txbk_send+1)<(lane_CL+car_width/2); excursion(txbk_send+1)=1; end
txbk_send_LPSD=sqrt(sumsqr(veh_x(txbk_send+1:txbk_off)-mean(veh_x(txbk_send+1:txbk_off)))/(txbk_off-txbk_send-1));
txbk_send_LPEXC=sum(excursion(txbk_send+1:txbk_off)==1);
txbk_send_LPEXC_time=sum(excursion(txbk_send+1:txbk_off)~=0)/row_sec; %does not "sum", but "count" instead.
txbk_send_rms_steer=sqrt(sumsqr(steer_wheel(txbk_send+1:txbk_off))/(txbk_off-txbk_send));
txbk_send_avg_speed=mean(veh_speed(txbk_send+1:txbk_off))/100;
txbk_send_avg_FD=mean(FD(txbk_send+1:txbk_off));
txbk_send_TTC_min=min(TTC(txbk_send+1:txbk_off));
txbk_send_FD_min=min(FD(txbk_send+1:txbk_off));
if isempty(find(collision_vehicle(txbk_send+1:txbk_off), 1));
    txbk_send_collision=0; else txbk_send_collision=1;end
%--txbk_post, Lane_C
bk_pre2=find(lead_x(max(rta_txbk,txbk_off):t_bk)~=lane_C,1,'first')+max(rta_txbk,txbk_off)-1;
if bk_pre2-max(rta_txbk,txbk_off)>pre_sec*row_sec && lead_x(max(rta_txbk,txbk_off))==lane_C;
    bk_pre=bk_pre2-pre_sec*row_sec; bk_pre_ck=0;
    if bk_pre-max(rta_txbk,txbk_off)-1>post_sec*row_sec;
    txbk_post=max(rta_txbk,txbk_off)+post_sec*row_sec; txbk_post_ck=0; else txbk_post=bk_pre-1; txbk_post_ck=1; end

    if veh_x(max(rta_txbk,txbk_off)+1)>(lane_CR-car_width/2) || veh_x(max(rta_txbk,txbk_off)+1)<(lane_CL+car_width/2); excursion(max(rta_txbk,txbk_off)+1)=1; e
    for n=max(rta_txbk,txbk_off)+1+1:txbk_post;
        if (veh_x(n)>(lane_CR-car_width/2) && veh_x(n-1)<=(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2) && veh_x(n-1)>=(lane_CL+car_width/2));
            excursion(n)=1;
        else if (veh_x(n)>(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2)); excursion(n)=2; end
        end
    end

    txbk_post_LPSD=sqrt(sumsqr(veh_x(max(rta_txbk,txbk_off)+1:txbk_post)-mean(veh_x(max(rta_txbk,txbk_off)+1:txbk_post)))/(txbk_post-max(rta_txbk,txbk_off)-1))
    txbk_post_LPEXC=sum(excursion(max(rta_txbk,txbk_off)+1:txbk_post)==1);
    txbk_post_LPEXC_time=sum(excursion(max(rta_txbk,txbk_off)+1:txbk_post)~=0)/row_sec; %does not "sum", but "count" instead.
    txbk_post_rms_steer=sqrt(sumsqr(steer_wheel(max(rta_txbk,txbk_off)+1:txbk_post))/(txbk_post-max(rta_txbk,txbk_off)));
    txbk_post_avg_speed=mean(veh_speed(max(rta_txbk,txbk_off)+1:txbk_post))/100;
    txbk_post_avg_FD=mean(FD(max(rta_txbk,txbk_off)+1:txbk_post));
    txbk_post_TTC_min=min(TTC(max(rta_txbk,txbk_off)+1:txbk_post));
    txbk_post_FD_min=min(FD(max(rta_txbk,txbk_off)+1:txbk_post));
    txbk_post_wnd_length=system_time(txbk_post)/1000-system_time(max(rta_txbk,txbk_off)+1)/1000;
else bk_pre=max(rta_txbk,txbk_off)+1; bk_pre_ck=1; txbk_post=NaN; txbk_post_ck=1;
    txbk_post_rms_steer=NaN;
    txbk_post_LPSD=NaN;
    txbk_post_LPEXC=NaN;
    txbk_post_LPEXC_time=NaN;
    txbk_post_avg_speed=NaN;
    txbk_post_avg_FD=NaN;
    txbk_post_TTC_min=NaN;
    txbk_post_FD_min=NaN;
    txbk_post_wnd_length=NaN;
end

%bk
%--bk_pre, Lane_C
if veh_x(bk_pre)>(lane_CR-car_width/2) || veh_x(bk_pre)<(lane_CL+car_width/2); excursion(bk_pre)=1; end
for n=bk_pre+1:bk_pre2;
    if (veh_x(n)>(lane_CR-car_width/2) && veh_x(n-1)<=(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2) && veh_x(n-1)>=(lane_CL+car_width/2));
        excursion(n)=1;
    else if (veh_x(n)>(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2)); excursion(n)=2; end
    end
end

bk_pre_LPSD=sqrt(sumsqr(veh_x(bk_pre:bk_pre2)-mean(veh_x(bk_pre:bk_pre2)))/(bk_pre2-bk_pre));
bk_pre_LPEXC=sum(excursion(bk_pre:bk_pre2)==1);
bk_pre_LPEXC_time=sum(excursion(bk_pre:bk_pre2)~=0)/row_sec; %does not "sum", but "count" instead.
bk_pre_rms_steer=sqrt(sumsqr(steer_wheel(bk_pre:bk_pre2))/(bk_pre2-bk_pre+1));
bk_pre_avg_speed=mean(veh_speed(bk_pre:bk_pre2))/100;
bk_pre_avg_FD=mean(FD(bk_pre:bk_pre2));
bk_pre_TTC_min=min(TTC(bk_pre:bk_pre2));
bk_pre_FD_min=min(FD(bk_pre:bk_pre2));
bk_pre_wnd_length=system_time(bk_pre2)/1000-system_time(bk_pre)/1000;
%--bk_reaction
n = t_bk+2;
t_bk_r=n;
while n<double(Count)-2;
      n=n+1;
      t_bk_r=t_bk_r+1;
      if lead_speed(n-2)>=lead_speed(n-1) && lead_speed(n-1)>=lead_speed(n) && lead_speed(n) < lead_speed(n+1) && lead_speed(n+1) <= lead_speed(n+2)
        break
```

```
      end
end

n = t_bk;
bkon_bk=n;
while n<data_end;
      n=n+1;
      bkon_bk=bkon_bk+1;
      if brake_ped(n) ~= 0 & brake_ped((n-3):(n-1)) == 0;
        break
      end
end
n = t_bk;
gr_bk=n;
while n<data_end-3;
      n=n+1;
      gr_bk=gr_bk+1;
      if accel_ped((n-3):(n-1)) > gr_adj & accel_ped((n+1):(n+3)) <= gr_adj & accel_ped(n) <= gr_adj & veh_speed(n) > 0;
        break
      end
end
if gr_bk >= data_end-3 || accel_ped(t_bk)<= gr_adj; gr_bk = NaN; end
if bkon_bk >= data_end || brake_ped(t_bk)~=0; bkon_bk = NaN; end
if isnan(gr_bk) && isnan(bkon_bk); d_bk=NaN; else d_bk=min(gr_bk,bkon_bk); end

if isempty(find(resize(t_bk_r:Count), 1)); data_end=double(Count);end

m_bk=min(veh_speed(max(t_bk,d_bk):data_end));
rta_bk=find(veh_speed(max(t_bk,d_bk):data_end)==m_bk, 1, 'last' )+max(t_bk,d_bk)-1;
%min_speed_bk=find(veh_speed(max(t_bk,d_bk):data_end)==m_bk, 1, 'first' )+max(t_bk,d_bk)-1;
if isempty(find(bkoff(max(t_bk,d_bk):rta_bk),1)), bkoff_bk=NaN; else bkoff_bk=find(bkoff(max(t_bk,d_bk):rta_bk),1,'last')+max(t_bk,d_bk)-1; end
%--bk_during, Lane_R
if veh_x(t_bk)>(lane_RR-car_width/2) || veh_x(t_bk)<(lane_CR+car_width/2); excursion(t_bk)=1; end
for n=t_bk+1:rta_bk;
    if (veh_x(n)>(lane_RR-car_width/2) && veh_x(n-1)<=(lane_RR-car_width/2)) || (veh_x(n)<(lane_CR+car_width/2) && veh_x(n-1)>=(lane_CR+car_width/2));
        excursion(n)=1;
    else if (veh_x(n)>(lane_RR-car_width/2)) || (veh_x(n)<(lane_CR+car_width/2)); excursion(n)=2; end
    end
end

bk_LPSD=sqrt(sumsqr(veh_x(t_bk:rta_bk)-mean(veh_x(t_bk:rta_bk)))/(rta_bk-t_bk));
bk_LPEXC=sum(excursion(t_bk:rta_bk)==1);
bk_LPEXC_time=sum(excursion(t_bk:rta_bk)~=0)/row_sec; %does not "sum", but "count" instead.
bk_rms_steer=sqrt(sumsqr(steer_wheel(t_bk:rta_bk))/(rta_bk-t_bk+1));
bk_avg_speed=mean(veh_speed(t_bk:rta_bk))/100;
bk_avg_FD=mean(FD(t_bk:rta_bk));
bk_TTC_min=min(TTC(t_bk:rta_bk));
bk_FD_min=min(FD(t_bk:rta_bk));
if isempty(find(collision_vehicle(t_bk:rta_bk), 1));
    bk_collision=0; else bk_collision=1;end
%--bk_post, Lane_R
if data_end==double(Count); bk_post=NaN; bk_post_rms_steer=NaN; bk_post_LPSD=NaN; bk_post_LPEXC=NaN; bk_post_LPEXC_time=NaN; bk_post_avg_speed=NaN;...
        bk_post_avg_FD=NaN; bk_post_TTC_min=NaN; bk_post_FD_min=NaN; bk_post_wnd_length=NaN; bk_post_ck=1; else
if data_end-rta_bk>post_sec*row_sec;
    bk_post=rta_bk+post_sec*row_sec; bk_post_ck=0; else bk_post=data_end; bk_post_ck=1;
end

if veh_x(rta_bk+1)>(lane_RR-car_width/2) || veh_x(rta_bk+1)<(lane_CR+car_width/2); excursion(rta_bk+1)=1; end
for n=rta_bk+1+1:bk_post;
    if (veh_x(n)>(lane_RR-car_width/2) && veh_x(n-1)<=(lane_RR-car_width/2)) || (veh_x(n)<(lane_CR+car_width/2) && veh_x(n-1)>=(lane_CR+car_width/2));
        excursion(n)=1;
    else if (veh_x(n)>(lane_RR-car_width/2)) || (veh_x(n)<(lane_CR+car_width/2)); excursion(n)=2; end
    end
end

bk_post_LPSD=sqrt(sumsqr(veh_x(rta_bk+1:bk_post)-mean(veh_x(rta_bk+1:bk_post)))/(bk_post-rta_bk-1));
bk_post_LPEXC=sum(excursion(rta_bk+1:bk_post)==1);
bk_post_LPEXC_time=sum(excursion(rta_bk+1:bk_post)~=0)/row_sec; %does not "sum", but "count" instead.
bk_post_rms_steer=sqrt(sumsqr(steer_wheel(rta_bk+1:bk_post))/(bk_post-rta_bk));
bk_post_avg_speed=mean(veh_speed(rta_bk+1:bk_post))/100;
bk_post_avg_FD=mean(FD(rta_bk+1:bk_post));
bk_post_TTC_min=min(TTC(rta_bk+1:bk_post));
bk_post_FD_min=min(FD(rta_bk+1:bk_post));
bk_post_wnd_length=system_time(bk_post)/1000-system_time(rta_bk+1)/1000;
end
%--check
if isnan(tx_off); tx_msg_latency=NaN; ck_ped_tx=0;
else
n = tx_pre;
tx_y_row=n;
while n<tx_initiate;
      n=n+1;
      tx_y_row=tx_y_row+1;
      if tx_y-lead_y(n)<trigger_adj;
        break
      end
end
tx_msg_latency=(system_time(tx_initiate)-system_time(tx_y_row))/1000;

if isempty(find(ped_check(tx_pre:max(tx_post,tx_off)), 1)); ck_ped_tx=0; else ck_ped_tx=1; end
end
```

```
    if isempty(find(ped_check(txbk_pre:max(txbk_post,max(rta_txbk,txbk_off))), 1)); ck_ped_txbk=0; else ck_ped_txbk=1; end
    if isempty(find(ped_check(bk_pre:max(bk_post,rta_bk)), 1)); ck_ped_bk=0; else ck_ped_bk=1; end
    if (ck_ped_tx+ck_ped_txbk+ck_ped_bk)>0; ck_ped=1; else ck_ped=0; end
    if rta_txbk<txbk_reply; ck_txbk_rta=0;
        else if rta_txbk<txbk_send; ck_txbk_rta=1;
            else if rta_txbk<=txbk_off; ck_txbk_rta=2; else ck_txbk_rta=3; end
            end
    end
    if (tx_pre_ck+txbk_pre_ck+bk_pre_ck)>0; ck_pre_time=1; else ck_pre_time=1; end
    if (tx_post_ck+txbk_post_ck+bk_post_ck)>0; ck_post_time=1; else ck_post_time=1; end
    n = txbk_pre;
    txbk_y_row=n;
    while n<txbk_initiate;
          n=n+1;
          txbk_y_row=txbk_y_row+1;
          if txbk_y-lead_y(n)<trigger_adj;
            break
          end
    end
    txbk_msg_latency=(system_time(txbk_initiate)-system_time(txbk_y_row))/1000;

    brake_event_delay=(system_time(t_txbk)-system_time(txbk_readthink))/1000;
```

Trigger 2

```
else t_bk=find(t,1,'first'); t_txbk=find(t,1,'last'); %bk->txbk->tx
    txbk_y=trigger_y2; tx_y=trigger_y3;

    %bk
    %--bk_pre, Lane_C

    bk_pre2=find(lead_x(find(lead_x(1:t_bk)==lane_C,1,'first'):t_bk)~=lane_C,1,'first')+find(lead_x(1:t_bk)==lane_C,1,'first')-1;
    if bk_pre2>pre_sec*row_sec;
        bk_pre=bk_pre2-pre_sec*row_sec; bk_pre_ck=0; else bk_pre=1; bk_pre_ck=1;
    end

    if veh_x(bk_pre)>(lane_CR-car_width/2) || veh_x(bk_pre)<(lane_CL+car_width/2); excursion(bk_pre)=1; end
    for n=bk_pre+1:bk_pre2;
        if (veh_x(n)>(lane_CR-car_width/2) && veh_x(n-1)<=(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2) && veh_x(n-1)>=(lane_CL+car_width/2));
            excursion(n)=1;
        else if (veh_x(n)>(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2)); excursion(n)=2; end
        end
    end

    bk_pre_LPSD=sqrt(sumsqr(veh_x(bk_pre:bk_pre2)-mean(veh_x(bk_pre:bk_pre2)))/(bk_pre2-bk_pre));
    bk_pre_LPEXC=sum(excursion(bk_pre:bk_pre2)==1);
    bk_pre_LPEXC_time=sum(excursion(bk_pre:bk_pre2)~=0)/row_sec; %does not "sum", but "count" instead.
    bk_pre_rms_steer=sqrt(sumsqr(steer_wheel(bk_pre:bk_pre2))/(bk_pre2-bk_pre+1));
    bk_pre_avg_speed=mean(veh_speed(bk_pre:bk_pre2))/100;
    bk_pre_avg_FD=mean(FD(bk_pre:bk_pre2));
    bk_pre_TTC_min=min(TTC(bk_pre:bk_pre2));
    bk_pre_FD_min=min(FD(bk_pre:bk_pre2));
    bk_pre_wnd_length=system_time(bk_pre2)/1000-system_time(bk_pre)/1000;
    %--bk_reaction
    n = t_bk;
    bkon_bk=n;
    while n<t_txbk-1;
          n=n+1;
          bkon_bk=bkon_bk+1;
          if brake_ped(n) ~= 0 & brake_ped((n-3):(n-1)) == 0;
            break
          end
    end
    n = t_bk;
    gr_bk=n;
    while n<t_txbk-3;
          n=n+1;
          gr_bk=gr_bk+1;
          if accel_ped((n-3):(n-1)) > gr_adj & accel_ped((n+1):(n+3)) <= gr_adj & accel_ped(n) <= gr_adj & veh_speed(n) > 0;
            break
          end
    end
    if gr_bk >= t_txbk-3 || accel_ped(t_bk)<= gr_adj; gr_bk = NaN; end
    if bkon_bk >= t_txbk-1 || brake_ped(t_bk)~=0; bkon_bk = NaN; end
    if isnan(gr_bk) && isnan(bkon_bk); d_bk=NaN; else d_bk=min(gr_bk,bkon_bk); end

    m_bk=min(veh_speed(max(t_bk,d_bk):t_txbk-1));
    rta_bk=find(veh_speed(max(t_bk,d_bk):t_txbk-1)==m_bk, 1, 'last' )+max(t_bk,d_bk)-1;
    %min_speed_bk=find(veh_speed(max(t_bk,d_bk):t_txbk-1)==m_bk, 1, 'first' )+max(t_bk,d_bk)-1;
    if isempty(find(bkoff(max(t_bk,d_bk):rta_bk),1)), bkoff_bk=NaN; else bkoff_bk=find(bkoff(max(t_bk,d_bk):rta_bk),1,'last')+max(t_bk,d_bk)-1; end
    %--bk_during, Lane_L
    if veh_x(t_bk)>(lane_CL-car_width/2) || veh_x(t_bk)<(lane_LL+car_width/2); excursion(t_bk)=1; end
    for n=t_bk+1:rta_bk;
        if (veh_x(n)>(lane_CL-car_width/2) && veh_x(n-1)<=(lane_CL-car_width/2)) || (veh_x(n)<(lane_LL+car_width/2) && veh_x(n-1)>=(lane_LL+car_width/2));
            excursion(n)=1;
        else if (veh_x(n)>(lane_CL-car_width/2)) || (veh_x(n)<(lane_LL+car_width/2)); excursion(n)=2; end
        end
    end
```

```
bk_LPSD=sqrt(sumsqr(veh_x(t_bk:rta_bk)-mean(veh_x(t_bk:rta_bk)))/(rta_bk-t_bk));
bk_LPEXC=sum(excursion(t_bk:rta_bk)==1);
bk_LPEXC_time=sum(excursion(t_bk:rta_bk)~=0)/row_sec; %does not "sum", but "count" instead.
bk_rms_steer=sqrt(sumsqr(steer_wheel(t_bk:rta_bk))/(rta_bk-t_bk+1));
bk_avg_speed=mean(veh_speed(t_bk:rta_bk))/100;
bk_avg_FD=mean(FD(t_bk:rta_bk));
bk_TTC_min=min(TTC(t_bk:rta_bk));
bk_FD_min=min(FD(t_bk:rta_bk));
if isempty(find(collision_vehicle(t_bk:rta_bk), 1));
    bk_collision=0; else bk_collision=1;end
%--bk_post, Lane_L
txbk_pre2=find(lead_x(rta_bk:txbk_initiate)~=lane_L,1,'first')+rta_bk-1;
if txbk_pre2-rta_bk>pre_sec*row_sec && lead_x(rta_bk)==lane_L;
    txbk_pre=txbk_pre2-pre_sec*row_sec; txbk_pre_ck=0;
    if txbk_pre-rta_bk-1>post_sec*row_sec;
    bk_post=rta_bk+post_sec*row_sec; bk_post_ck=0; else bk_post=txbk_pre-1; bk_post_ck=1; end

    if veh_x(rta_bk+1)>(lane_CL-car_width/2) || veh_x(rta_bk+1)<(lane_LL+car_width/2); excursion(rta_bk+1)=1; end
    for n=rta_bk+1+1:bk_post;
        if (veh_x(n)>(lane_CL-car_width/2) && veh_x(n-1)<=(lane_CL-car_width/2)) || (veh_x(n)<(lane_LL+car_width/2) && veh_x(n-1)>=(lane_LL+car_width/2));
        excursion(n)=1;
        else if (veh_x(n)>(lane_CL-car_width/2)) || (veh_x(n)<(lane_LL+car_width/2)); excursion(n)=2; end
        end
    end

    bk_post_LPSD=sqrt(sumsqr(veh_x(rta_bk+1:bk_post)-mean(veh_x(rta_bk+1:bk_post)))/(bk_post-rta_bk-1));
    bk_post_LPEXC=sum(excursion(rta_bk+1:bk_post)==1);
    bk_post_LPEXC_time=sum(excursion(rta_bk+1:bk_post)~=0)/row_sec; %does not "sum", but "count" instead.
    bk_post_rms_steer=sqrt(sumsqr(steer_wheel(rta_bk+1:bk_post))/(bk_post-rta_bk));
    bk_post_avg_speed=mean(veh_speed(rta_bk+1:bk_post))/100;
    bk_post_avg_FD=mean(FD(rta_bk+1:bk_post));
    bk_post_TTC_min=min(TTC(rta_bk+1:bk_post));
    bk_post_FD_min=min(FD(rta_bk+1:bk_post));
    bk_post_wnd_length=system_time(bk_post)/1000-system_time(rta_bk+1)/1000;
    else txbk_pre=rta_bk+1; txbk_pre_ck=1; bk_post=NaN; bk_post_ck=1;
    bk_post_rms_steer=NaN;
    bk_post_LPSD=NaN;
    bk_post_LPEXC=NaN;
    bk_post_LPEXC_time=NaN;
    bk_post_avg_speed=NaN;
    bk_post_avg_FD=NaN;
    bk_post_TTC_min=NaN;
    bk_post_FD_min=NaN;
    bk_post_wnd_length=NaN;
end

%txbk
%--txbk_pre, Lane_L
if veh_x(txbk_pre)>(lane_CL-car_width/2) || veh_x(txbk_pre)<(lane_LL+car_width/2); excursion(txbk_pre)=1; end
for n=txbk_pre+1:txbk_pre2;
    if (veh_x(n)>(lane_CL-car_width/2) && veh_x(n-1)<=(lane_CL-car_width/2)) || (veh_x(n)<(lane_LL+car_width/2) && veh_x(n-1)>=(lane_LL+car_width/2));
        excursion(n)=1;
    else if (veh_x(n)>(lane_CL-car_width/2)) || (veh_x(n)<(lane_LL+car_width/2)); excursion(n)=2; end
    end
end

txbk_pre_LPSD=sqrt(sumsqr(veh_x(txbk_pre:txbk_pre2)-mean(veh_x(txbk_pre:txbk_pre2)))/(txbk_pre2-txbk_pre));
txbk_pre_LPEXC=sum(excursion(txbk_pre:txbk_pre2)==1);
txbk_pre_LPEXC_time=sum(excursion(txbk_pre:txbk_pre2)~=0)/row_sec; %does not "sum", but "count" instead.
txbk_pre_rms_steer=sqrt(sumsqr(steer_wheel(txbk_pre:txbk_pre2))/(txbk_pre2-txbk_pre+1));
txbk_pre_avg_speed=mean(veh_speed(txbk_pre:txbk_pre2))/100;
txbk_pre_avg_FD=mean(FD(txbk_pre:txbk_pre2));
txbk_pre_TTC_min=min(TTC(txbk_pre:txbk_pre2));
txbk_pre_FD_min=min(FD(txbk_pre:txbk_pre2));
txbk_pre_wnd_length=system_time(txbk_pre2)/1000-system_time(txbk_pre)/1000;
%--txbk_reaction
n = t_txbk;
bkon_txbk=n;
while n<tx_initiate-1;
      n=n+1;
      bkon_txbk=bkon_txbk+1;
      if brake_ped(n) ~= 0 & brake_ped((n-3):(n-1)) == 0;
        break
      end
end
n = t_txbk;
gr_txbk=n;
while n<tx_initiate-3;
      n=n+1;
      gr_txbk=gr_txbk+1;
      if accel_ped((n-3):(n-1)) > gr_adj & accel_ped((n+1):(n+3)) <= gr_adj & accel_ped(n) <= gr_adj & veh_speed(n) > 0;
        break
      end
end
if gr_txbk >= tx_initiate-3 || accel_ped(t_txbk)<= gr_adj; gr_txbk = NaN; end
if bkon_txbk >= tx_initiate-1 || brake_ped(t_txbk)~=0; bkon_txbk = NaN; end
if isnan(gr_txbk) && isnan(bkon_txbk); d_txbk=NaN; else d_txbk=min(gr_txbk,bkon_txbk); end
m_txbk=min(veh_speed(max(t_txbk,d_txbk):tx_initiate-1));
rta_txbk=find(veh_speed(max(t_txbk,d_txbk):tx_initiate-1)==m_txbk, 1, 'last' )+max(t_txbk,d_txbk)-1;
%min_speed_txbk=find(veh_speed(max(t_txbk,d_txbk):tx_initiate-1)==m_txbk, 1, 'first' )+max(t_txbk,d_txbk)-1;
if isempty(find(bkoff(max(t_txbk,d_txbk):rta_txbk),1)), bkoff_txbk=NaN; else bkoff_txbk=find(bkoff(max(t_txbk,d_txbk):rta_txbk),1,'last')+max(t_txbk,d_txbk)-1;
```

```
%--txbk_during, Lane_C
if veh_x(t_txbk)>(lane_CR-car_width/2) || veh_x(t_txbk)<(lane_CL+car_width/2); excursion(t_txbk)=1; end
for n=t_txbk+1:rta_txbk;
    if (veh_x(n)>(lane_CR-car_width/2) && veh_x(n-1)<=(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2) && veh_x(n-1)>=(lane_CL+car_width/2));
        excursion(n)=1;
    else if (veh_x(n)>(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2)); excursion(n)=2; end
    end
end

txbk_LPSD=sqrt(sumsqr(veh_x(t_txbk:rta_txbk)-mean(veh_x(t_txbk:rta_txbk)))/(rta_txbk-t_txbk));
txbk_LPEXC=sum(excursion(t_txbk:rta_txbk)==1);
txbk_LPEXC_time=sum(excursion(t_txbk:rta_txbk)~=0)/row_sec; %does not "sum", but "count" instead.
txbk_rms_steer=sqrt(sumsqr(steer_wheel(t_txbk:rta_txbk))/(rta_txbk-t_txbk+1));
txbk_avg_speed=mean(veh_speed(t_txbk:rta_txbk))/100;
txbk_avg_FD=mean(FD(t_txbk:rta_txbk));
txbk_TTC_min=min(TTC(t_txbk:rta_txbk));
txbk_FD_min=min(FD(t_txbk:rta_txbk));
if isempty(find(collision_vehicle(t_txbk:rta_txbk), 1));
    txbk_collision=0; else txbk_collision=1;end
%--txbk_initiate
txbk_initiate_LPSD=sqrt(sumsqr(veh_x(txbk_initiate:txbk_readthink-1)-mean(veh_x(txbk_initiate:txbk_readthink-1)))/(txbk_readthink-txbk_initiate-1));
txbk_initiate_LPEXC=sum(excursion(txbk_initiate:txbk_readthink-1)==1);
txbk_initiate_LPEXC_time=sum(excursion(txbk_initiate:txbk_readthink-1)~=0)/row_sec; %does not "sum", but "count" instead.
txbk_initiate_rms_steer=sqrt(sumsqr(steer_wheel(txbk_initiate:txbk_readthink-1))/(txbk_readthink-txbk_initiate));
txbk_initiate_avg_speed=mean(veh_speed(txbk_initiate:txbk_readthink-1))/100;
txbk_initiate_avg_FD=mean(FD(txbk_initiate:txbk_readthink-1));
txbk_initiate_TTC_min=min(TTC(txbk_initiate:txbk_readthink-1));
txbk_initiate_FD_min=min(FD(txbk_initiate:txbk_readthink-1));
if isempty(find(collision_vehicle(txbk_initiate:txbk_readthink-1), 1));
    txbk_initiate_collision=0; else txbk_initiate_collision=1;end
%--txbk_readthink
if veh_x(txbk_readthink)>(lane_CR-car_width/2) || veh_x(txbk_readthink)<(lane_CL+car_width/2); excursion(txbk_readthink)=1; end
txbk_readthink_LPSD=sqrt(sumsqr(veh_x(txbk_readthink:txbk_reply-1)-mean(veh_x(txbk_readthink:txbk_reply-1)))/(txbk_reply-txbk_readthink-1));
txbk_readthink_LPEXC=sum(excursion(txbk_readthink:txbk_reply-1)==1);
txbk_readthink_LPEXC_time=sum(excursion(txbk_readthink:txbk_reply-1)~=0)/row_sec; %does not "sum", but "count" instead.
txbk_readthink_rms_steer=sqrt(sumsqr(steer_wheel(txbk_readthink:txbk_reply-1))/(txbk_reply-txbk_readthink));
txbk_readthink_avg_speed=mean(veh_speed(txbk_readthink:txbk_reply-1))/100;
txbk_readthink_avg_FD=mean(FD(txbk_readthink:txbk_reply-1));
txbk_readthink_TTC_min=min(TTC(txbk_readthink:txbk_reply-1));
txbk_readthink_FD_min=min(FD(txbk_readthink:txbk_reply-1));
if isempty(find(collision_vehicle(txbk_readthink:txbk_reply-1), 1));
    txbk_readthink_collision=0; else txbk_readthink_collision=1;end
%--txbk_reply
if veh_x(txbk_reply)>(lane_CR-car_width/2) || veh_x(txbk_reply)<(lane_CL+car_width/2); excursion(txbk_reply)=1; end
txbk_reply_LPSD=sqrt(sumsqr(veh_x(txbk_reply:txbk_send)-mean(veh_x(txbk_reply:txbk_send)))/(txbk_send-txbk_reply));
txbk_reply_LPEXC=sum(excursion(txbk_reply:txbk_send)==1);
txbk_reply_LPEXC_time=sum(excursion(txbk_reply:txbk_send)~=0)/row_sec; %does not "sum", but "count" instead.
txbk_reply_rms_steer=sqrt(sumsqr(steer_wheel(txbk_reply:txbk_send))/(txbk_send-txbk_reply+1));
txbk_reply_avg_speed=mean(veh_speed(txbk_reply:txbk_send))/100;
txbk_reply_avg_FD=mean(FD(txbk_reply:txbk_send));
txbk_reply_TTC_min=min(TTC(txbk_reply:txbk_send));
txbk_reply_FD_min=min(FD(txbk_reply:txbk_send));
if isempty(find(collision_vehicle(txbk_reply:txbk_send), 1));
    txbk_reply_collision=0; else txbk_reply_collision=1;end
%--txbk_send
if veh_x(txbk_send+1)>(lane_CR-car_width/2) || veh_x(txbk_send+1)<(lane_CL+car_width/2); excursion(txbk_send+1)=1; end
txbk_send_LPSD=sqrt(sumsqr(veh_x(txbk_send+1:txbk_off)-mean(veh_x(txbk_send+1:txbk_off)))/(txbk_off-txbk_send-1));
txbk_send_LPEXC=sum(excursion(txbk_send+1:txbk_off)==1);
txbk_send_LPEXC_time=sum(excursion(txbk_send+1:txbk_off)~=0)/row_sec; %does not "sum", but "count" instead.
txbk_send_rms_steer=sqrt(sumsqr(steer_wheel(txbk_send+1:txbk_off))/(txbk_off-txbk_send));
txbk_send_avg_speed=mean(veh_speed(txbk_send+1:txbk_off))/100;
txbk_send_avg_FD=mean(FD(txbk_send+1:txbk_off));
txbk_send_TTC_min=min(TTC(txbk_send+1:txbk_off));
txbk_send_FD_min=min(FD(txbk_send+1:txbk_off));
if isempty(find(collision_vehicle(txbk_send+1:txbk_off), 1));
    txbk_send_collision=0; else txbk_send_collision=1;end
%--txbk_post, Lane_C
if isnan(tx_off);
    tx_pre=NaN; tx_pre2=NaN;
    txbk_post=max(rta_txbk,txbk_off)+post_sec*row_sec; txbk_post_ck=0;

    if veh_x(max(rta_txbk,txbk_off)+1)>(lane_CR-car_width/2) || veh_x(max(rta_txbk,txbk_off)+1)<(lane_CL+car_width/2); excursion(max(rta_txbk,txbk_off)+1)=1; en
    for n=max(rta_txbk,txbk_off)+1+1:txbk_post;
        if (veh_x(n)>(lane_CR-car_width/2) && veh_x(n-1)<=(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2) && veh_x(n-1)>=(lane_CL+car_width/2));
            excursion(n)=1;
        else if (veh_x(n)>(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2)); excursion(n)=2; end
        end
    end

    txbk_post_LPSD=sqrt(sumsqr(veh_x(max(rta_txbk,txbk_off)+1:txbk_post)-mean(veh_x(max(rta_txbk,txbk_off)+1:txbk_post)))/(txbk_post-max(rta_txbk,txbk_off)-1));
    txbk_post_LPEXC=sum(excursion(max(rta_txbk,txbk_off)+1:txbk_post)==1);
    txbk_post_LPEXC_time=sum(excursion(max(rta_txbk,txbk_off)+1:txbk_post)~=0)/row_sec; %does not "sum", but "count" instead.
    txbk_post_rms_steer=sqrt(sumsqr(steer_wheel(max(rta_txbk,txbk_off)+1:txbk_post))/(txbk_post-max(rta_txbk,txbk_off)));
    txbk_post_avg_speed=mean(veh_speed(max(rta_txbk,txbk_off)+1:txbk_post))/100;
    txbk_post_avg_FD=mean(FD(max(rta_txbk,txbk_off)+1:txbk_post));
    txbk_post_TTC_min=min(TTC(max(rta_txbk,txbk_off)+1:txbk_post));
    txbk_post_FD_min=min(FD(max(rta_txbk,txbk_off)+1:txbk_post));
    txbk_post_wnd_length=system_time(txbk_post)/1000-system_time(max(rta_txbk,txbk_off)+1)/1000;
else
tx_pre2=find(lead_x(max(rta_txbk,txbk_off):tx_initiate)~=lane_C,1,'first')+max(rta_txbk,txbk_off)-1;
if tx_pre2-max(rta_txbk,txbk_off)>pre_sec*row_sec && lead_x(max(rta_txbk,txbk_off))==lane_C;
```

```
    tx_pre=tx_pre2-pre_sec*row_sec; tx_pre_ck=0;
    if tx_pre-max(rta_txbk,txbk_off)-1>post_sec*row_sec;
    txbk_post=max(rta_txbk,txbk_off)+post_sec*row_sec; txbk_post_ck=0; else txbk_post=tx_pre-1; txbk_post_ck=1; end

    if veh_x(max(rta_txbk,txbk_off)+1)>(lane_CR-car_width/2) || veh_x(max(rta_txbk,txbk_off)+1)<(lane_CL+car_width/2); excursion(max(rta_txbk,txbk_off)+1)=1; e
    for n=max(rta_txbk,txbk_off)+1+1:txbk_post;
        if (veh_x(n)>(lane_CR-car_width/2) && veh_x(n-1)<=(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2) && veh_x(n-1)>=(lane_CL+car_width/2));
            excursion(n)=1;
        else if (veh_x(n)>(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2)); excursion(n)=2; end
        end
    end

    txbk_post_LPSD=sqrt(sumsqr(veh_x(max(rta_txbk,txbk_off)+1:txbk_post)-mean(veh_x(max(rta_txbk,txbk_off)+1:txbk_post)))/(txbk_post-max(rta_txbk,txbk_off)-1))
    txbk_post_LPEXC=sum(excursion(max(rta_txbk,txbk_off)+1:txbk_post)==1);
    txbk_post_LPEXC_time=sum(excursion(max(rta_txbk,txbk_off)+1:txbk_post)~=0)/row_sec; %does not "sum", but "count" instead.
    txbk_post_rms_steer=sqrt(sumsqr(steer_wheel(max(rta_txbk,txbk_off)+1:txbk_post))/(txbk_post-max(rta_txbk,txbk_off)));
    txbk_post_avg_speed=mean(veh_speed(max(rta_txbk,txbk_off)+1:txbk_post))/100;
    txbk_post_avg_FD=mean(FD(max(rta_txbk,txbk_off)+1:txbk_post));
    txbk_post_TTC_min=min(TTC(max(rta_txbk,txbk_off)+1:txbk_post));
    txbk_post_FD_min=min(FD(max(rta_txbk,txbk_off)+1:txbk_post));
    txbk_post_wnd_length=system_time(txbk_post)/1000-system_time(max(rta_txbk,txbk_off)+1)/1000;
    else tx_pre=max(rta_txbk,txbk_off)+1; tx_pre_ck=1; txbk_post=NaN; txbk_post_ck=1;
    txbk_post_rms_steer=NaN;
    txbk_post_LPSD=NaN;
    txbk_post_LPEXC=NaN;
    txbk_post_LPEXC_time=NaN;
    txbk_post_avg_speed=NaN;
    txbk_post_avg_FD=NaN;
    txbk_post_TTC_min=NaN;
    txbk_post_FD_min=NaN;
    txbk_post_wnd_length=NaN;
end
end

%tx
if isnan(tx_off);
    tx_pre_ck=1; tx_pre_rms_steer=NaN; tx_pre_LPSD=NaN; tx_pre_LPEXC=NaN; tx_pre_LPEXC_time=NaN; tx_pre_avg_speed=NaN; tx_pre_avg_FD=NaN; tx_pre_TTC_min=NaN; t
    tx_rms_steer=NaN; tx_LPSD=NaN; tx_LPEXC=NaN; tx_LPEXC_time=NaN; tx_avg_speed=NaN; tx_avg_FD=NaN; tx_TTC_min=NaN; tx_FD_min=NaN; tx_collision=0;
    tx_initiate_rms_steer=NaN; tx_initiate_LPSD=NaN; tx_initiate_LPEXC=NaN; tx_initiate_LPEXC_time=NaN; tx_initiate_avg_speed=NaN; tx_initiate_avg_FD=NaN; tx_i
    tx_readthink_rms_steer=NaN; tx_readthink_LPSD=NaN; tx_readthink_LPEXC=NaN; tx_readthink_LPEXC_time=NaN; tx_readthink_avg_speed=NaN; tx_readthink_avg_FD=NaN
    tx_reply_rms_steer=NaN; tx_reply_LPSD=NaN; tx_reply_LPEXC=NaN; tx_reply_LPEXC_time=NaN; tx_reply_avg_speed=NaN; tx_reply_avg_FD=NaN; tx_reply_TTC_min=NaN;
    tx_send_rms_steer=NaN; tx_send_LPSD=NaN; tx_send_LPEXC=NaN; tx_send_LPEXC_time=NaN; tx_send_avg_speed=NaN; tx_send_avg_FD=NaN; tx_send_TTC_min=NaN; tx_send_
    tx_post_ck=1; tx_post_rms_steer=NaN; tx_post_LPSD=NaN; tx_post_LPEXC=NaN; tx_post_LPEXC_time=NaN; tx_post_avg_speed=NaN; tx_post_avg_FD=NaN; tx_post_TTC_mi
else
%--tx_pre, Lane_C
if veh_x(tx_pre)>(lane_CR-car_width/2) || veh_x(tx_pre)<(lane_CL+car_width/2); excursion(tx_pre)=1; end
for n=tx_pre+1:tx_pre2;
    if (veh_x(n)>(lane_CR-car_width/2) && veh_x(n-1)<=(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2) && veh_x(n-1)>=(lane_CL+car_width/2));
        excursion(n)=1;
    else if (veh_x(n)>(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2)); excursion(n)=2; end
    end
end

tx_pre_LPSD=sqrt(sumsqr(veh_x(tx_pre:tx_pre2)-mean(veh_x(tx_pre:tx_pre2)))/(tx_pre2-tx_pre));
tx_pre_LPEXC=sum(excursion(tx_pre:tx_pre2)==1);
tx_pre_LPEXC_time=sum(excursion(tx_pre:tx_pre2)~=0)/row_sec; %does not "sum", but "count" instead.
tx_pre_rms_steer=sqrt(sumsqr(steer_wheel(tx_pre:tx_pre2))/(tx_pre2-tx_pre+1));
tx_pre_avg_speed=mean(veh_speed(tx_pre:tx_pre2))/100;
tx_pre_avg_FD=mean(FD(tx_pre:tx_pre2));
tx_pre_TTC_min=min(TTC(tx_pre:tx_pre2));
tx_pre_FD_min=min(FD(tx_pre:tx_pre2));
tx_pre_wnd_length=system_time(tx_pre2)/1000-system_time(tx_pre)/1000;
%--tx_during, Lane_R
if veh_x(tx_initiate)>(lane_RR-car_width/2) || veh_x(tx_initiate)<(lane_CR+car_width/2); excursion(tx_initiate)=1; end
for n=tx_initiate+1:tx_off;
    if (veh_x(n)>(lane_RR-car_width/2) && veh_x(n-1)<=(lane_RR-car_width/2)) || (veh_x(n)<(lane_CR+car_width/2) && veh_x(n-1)>=(lane_CR+car_width/2));
        excursion(n)=1;
    else if (veh_x(n)>(lane_RR-car_width/2)) || (veh_x(n)<(lane_CR+car_width/2)); excursion(n)=2; end
    end
end

tx_LPSD=sqrt(sumsqr(veh_x(tx_initiate:tx_off)-mean(veh_x(tx_initiate:tx_off)))/(tx_off-tx_initiate));
tx_LPEXC=sum(excursion(tx_initiate:tx_off)==1);
tx_LPEXC_time=sum(excursion(tx_initiate:tx_off)~=0)/row_sec; %does not "sum", but "count" instead.
tx_rms_steer=sqrt(sumsqr(steer_wheel(tx_initiate:tx_off))/(tx_off-tx_initiate+1));
tx_avg_speed=mean(veh_speed(tx_initiate:tx_off))/100;
tx_avg_FD=mean(FD(tx_initiate:tx_off));
tx_TTC_min=min(TTC(tx_initiate:tx_off));
tx_FD_min=min(FD(tx_initiate:tx_off));
if isempty(find(collision_vehicle(tx_initiate:tx_off), 1));
    tx_collision=0; else tx_collision=1;end
%--tx_initiate
tx_initiate_LPSD=sqrt(sumsqr(veh_x(tx_initiate:tx_readthink-1)-mean(veh_x(tx_initiate:tx_readthink-1)))/(tx_readthink-tx_initiate-1));
tx_initiate_LPEXC=sum(excursion(tx_initiate:tx_readthink-1)==1);
tx_initiate_LPEXC_time=sum(excursion(tx_initiate:tx_readthink-1)~=0)/row_sec; %does not "sum", but "count" instead.
tx_initiate_rms_steer=sqrt(sumsqr(steer_wheel(tx_initiate:tx_readthink-1))/(tx_readthink-tx_initiate));
tx_initiate_avg_speed=mean(veh_speed(tx_initiate:tx_readthink-1))/100;
tx_initiate_avg_FD=mean(FD(tx_initiate:tx_readthink-1));
tx_initiate_TTC_min=min(TTC(tx_initiate:tx_readthink-1));
tx_initiate_FD_min=min(FD(tx_initiate:tx_readthink-1));
if isempty(find(collision_vehicle(tx_initiate:tx_readthink-1), 1));
```

```
    tx_initiate_collision=0; else tx_initiate_collision=1;end
%--tx_readthink
if veh_x(tx_readthink)>(lane_RR-car_width/2) || veh_x(tx_readthink)<(lane_CR+car_width/2); excursion(tx_readthink)=1; end
tx_readthink_LPSD=sqrt(sumsqr(veh_x(tx_readthink:tx_reply-1)-mean(veh_x(tx_readthink:tx_reply-1)))/(tx_reply-tx_readthink-1));
tx_readthink_LPEXC=sum(excursion(tx_readthink:tx_reply-1)==1);
tx_readthink_LPEXC_time=sum(excursion(tx_readthink:tx_reply-1)~=0)/row_sec; %does not "sum", but "count" instead.
tx_readthink_rms_steer=sqrt(sumsqr(steer_wheel(tx_readthink:tx_reply-1))/(tx_reply-tx_readthink));
tx_readthink_avg_speed=mean(veh_speed(tx_readthink:tx_reply-1))/100;
tx_readthink_avg_FD=mean(FD(tx_readthink:tx_reply-1));
tx_readthink_TTC_min=min(TTC(tx_readthink:tx_reply-1));
tx_readthink_FD_min=min(FD(tx_readthink:tx_reply-1));
if isempty(find(collision_vehicle(tx_readthink:tx_reply-1), 1));
    tx_readthink_collision=0; else tx_readthink_collision=1;end
%--tx_reply
if veh_x(tx_reply)>(lane_RR-car_width/2) || veh_x(tx_reply)<(lane_CR+car_width/2); excursion(tx_reply)=1; end
tx_reply_LPSD=sqrt(sumsqr(veh_x(tx_reply:tx_send)-mean(veh_x(tx_reply:tx_send)))/(tx_send-tx_reply));
tx_reply_LPEXC=sum(excursion(tx_reply:tx_send)==1);
tx_reply_LPEXC_time=sum(excursion(tx_reply:tx_send)~=0)/row_sec; %does not "sum", but "count" instead.
tx_reply_rms_steer=sqrt(sumsqr(steer_wheel(tx_reply:tx_send))/(tx_send-tx_reply+1));
tx_reply_avg_speed=mean(veh_speed(tx_reply:tx_send))/100;
tx_reply_avg_FD=mean(FD(tx_reply:tx_send));
tx_reply_TTC_min=min(TTC(tx_reply:tx_send));
tx_reply_FD_min=min(FD(tx_reply:tx_send));
if isempty(find(collision_vehicle(tx_reply:tx_send), 1));
    tx_reply_collision=0; else tx_reply_collision=1;end
%--tx_send
if veh_x(tx_send+1)>(lane_RR-car_width/2) || veh_x(tx_send+1)<(lane_CR+car_width/2); excursion(tx_send+1)=1; end
tx_send_LPSD=sqrt(sumsqr(veh_x(tx_send+1:tx_off)-mean(veh_x(tx_send+1:tx_off)))/(tx_off-tx_send-1));
tx_send_LPEXC=sum(excursion(tx_send+1:tx_off)==1);
tx_send_LPEXC_time=sum(excursion(tx_send+1:tx_off)~=0)/row_sec; %does not "sum", but "count" instead.
tx_send_rms_steer=sqrt(sumsqr(steer_wheel(tx_send+1:tx_off))/(tx_off-tx_send));
tx_send_avg_speed=mean(veh_speed(tx_send+1:tx_off))/100;
tx_send_avg_FD=mean(FD(tx_send+1:tx_off));
tx_send_TTC_min=min(TTC(tx_send+1:tx_off));
tx_send_FD_min=min(FD(tx_send+1:tx_off));
if isempty(find(collision_vehicle(tx_send+1:tx_off), 1));
    tx_send_collision=0; else tx_send_collision=1;end
%--tx_post, Lane_R
if tx_off>=data_end; data_end=double(Count); end
if data_end-tx_off>post_sec*row_sec;
    tx_post=tx_off+post_sec*row_sec; tx_post_ck=0; else tx_post=data_end; tx_post_ck=1;
end

if veh_x(tx_off+1)>(lane_RR-car_width/2) || veh_x(tx_off+1)<(lane_CR+car_width/2); excursion(tx_off+1)=1; end
for n=tx_off+1+1:tx_post;
  if (veh_x(n)>(lane_RR-car_width/2) && veh_x(n-1)<=(lane_RR-car_width/2)) || (veh_x(n)<(lane_CR+car_width/2) && veh_x(n-1)>=(lane_CR+car_width/2));
       excursion(n)=1;
    else if (veh_x(n)>(lane_RR-car_width/2)) || (veh_x(n)<(lane_CR+car_width/2)); excursion(n)=2; end
  end
end

tx_post_LPSD=sqrt(sumsqr(veh_x(tx_off+1:tx_post)-mean(veh_x(tx_off+1:tx_post)))/(tx_post-tx_off-1));
tx_post_LPEXC=sum(excursion(tx_off+1:tx_post)==1);
tx_post_LPEXC_time=sum(excursion(tx_off+1:tx_post)~=0)/row_sec; %does not "sum", but "count" instead.
tx_post_rms_steer=sqrt(sumsqr(steer_wheel(tx_off+1:tx_post))/(tx_post-tx_off));
tx_post_avg_speed=mean(veh_speed(tx_off+1:tx_post))/100;
tx_post_avg_FD=mean(FD(tx_off+1:tx_post));
tx_post_TTC_min=min(TTC(tx_off+1:tx_post));
tx_post_FD_min=min(FD(tx_off+1:tx_post));
tx_post_wnd_length=system_time(tx_post)/1000-system_time(tx_off+1)/1000;
end
%--check
if isnan(tx_off); tx_msg_latency=NaN; ck_ped_tx=0;
else
n = tx_pre;
tx_y_row=n;
while n<tx_initiate;
      n=n+1;
      tx_y_row=tx_y_row+1;
      if tx_y-lead_y(n)<trigger_adj;
        break
      end
end
tx_msg_latency=(system_time(tx_initiate)-system_time(tx_y_row))/1000;

if isempty(find(ped_check(tx_pre:max(tx_post,tx_off)), 1)); ck_ped_tx=0; else ck_ped_tx=1; end
end

if isempty(find(ped_check(txbk_pre:max(txbk_post,max(rta_txbk,txbk_off))), 1)); ck_ped_txbk=0; else ck_ped_txbk=1; end
if isempty(find(ped_check(bk_pre:max(bk_post,rta_bk)), 1)); ck_ped_bk=0; else ck_ped_bk=1; end
if (ck_ped_tx+ck_ped_txbk+ck_ped_bk)>0; ck_ped=1; else ck_ped=0; end
if rta_txbk<txbk_reply; ck_txbk_rta=0;
    else if rta_txbk<txbk_send; ck_txbk_rta=1;
        else if rta_txbk<=txbk_off; ck_txbk_rta=2; else ck_txbk_rta=3; end
        end
end
if (tx_pre_ck+txbk_pre_ck+bk_pre_ck)>0; ck_pre_time=1; else ck_pre_time=1; end
if (tx_post_ck+txbk_post_ck+bk_post_ck)>0; ck_post_time=1; else ck_post_time=1; end
n = txbk_pre;
txbk_y_row=n;
while n<txbk_initiate;
```

```
            n=n+1;
            txbk_y_row=txbk_y_row+1;
            if txbk_y-lead_y(n)<trigger_adj;
              break
            end
        end
        txbk_msg_latency=(system_time(txbk_initiate)-system_time(txbk_y_row))/1000;

        brake_event_delay=(system_time(t_txbk)-system_time(txbk_readthink))/1000;
    end

    if isnan(bkon_txbk); txbk_bk_onset=NaN; txbk_bk_offset=NaN; txbk_rta=NaN;
    else txbk_bk_onset=system_time(bkon_txbk)/1000-system_time(t_txbk)/1000;
        if isnan(bkoff_txbk); txbk_bk_offset=NaN; else txbk_bk_offset=system_time(bkoff_txbk)/1000-system_time(bkon_txbk)/1000; end
        txbk_rta=system_time(rta_txbk)/1000-system_time(bkon_txbk)/1000;
    end
    if isnan(d_txbk); txbk_hy_res=NaN; else txbk_hy_res=system_time(d_txbk)/1000-system_time(t_txbk)/1000; end

    if isnan(bkon_bk); bk_bk_onset=NaN; bk_bk_offset=NaN; bk_rta=NaN;
    else bk_bk_onset=system_time(bkon_bk)/1000-system_time(t_bk)/1000;
        if isnan(bkoff_bk); bk_bk_offset=NaN; else bk_bk_offset=system_time(bkoff_bk)/1000-system_time(bkon_bk)/1000; end
        if rta_bk==data_end; bk_rta=NaN; else bk_rta=system_time(rta_bk)/1000-system_time(bkon_bk)/1000; end
    end
    if isnan(d_bk); bk_hy_res=NaN; else bk_hy_res=system_time(d_bk)/1000-system_time(t_bk)/1000; end

    if (Collision==0 && (tx_collision+bk_collision+txbk_collision)==0) || (Collision==1 && (tx_collision+bk_collision+txbk_collision)>0);
        c_check=0; else c_check=1;
    end

    if ~isnan(txbk_pre_LPEXC) && txbk_pre_LPEXC==0 && txbk_pre_LPEXC_time~=0; LPEXC_check=1; else LPEXC_check=0; end
    if ~isnan(txbk_LPEXC) && txbk_LPEXC==0 && txbk_LPEXC_time~=0; LPEXC_check=LPEXC_check+1; end
    if ~isnan(txbk_initiate_LPEXC) && txbk_initiate_LPEXC==0 && txbk_initiate_LPEXC_time~=0; LPEXC_check=LPEXC_check+1; end
    if ~isnan(txbk_readthink_LPEXC) && txbk_readthink_LPEXC==0 && txbk_readthink_LPEXC_time~=0; LPEXC_check=LPEXC_check+1; end
    if ~isnan(txbk_reply_LPEXC) && txbk_reply_LPEXC==0 && txbk_reply_LPEXC_time~=0; LPEXC_check=LPEXC_check+1; end
    if ~isnan(txbk_send_LPEXC) && txbk_send_LPEXC==0 && txbk_send_LPEXC_time~=0; LPEXC_check=LPEXC_check+1; end
    if ~isnan(txbk_post_LPEXC) && txbk_post_LPEXC==0 && txbk_post_LPEXC_time~=0; LPEXC_check=LPEXC_check+1; end

    if ~isnan(tx_pre_LPEXC) && tx_pre_LPEXC==0 && tx_pre_LPEXC_time~=0; LPEXC_check=LPEXC_check+1; end
    if ~isnan(tx_LPEXC) && tx_LPEXC==0 && tx_LPEXC_time~=0; LPEXC_check=LPEXC_check+1; end
    if ~isnan(tx_initiate_LPEXC) && tx_initiate_LPEXC==0 && tx_initiate_LPEXC_time~=0; LPEXC_check=LPEXC_check+1; end
    if ~isnan(tx_readthink_LPEXC) && tx_readthink_LPEXC==0 && tx_readthink_LPEXC_time~=0; LPEXC_check=LPEXC_check+1; end
    if ~isnan(tx_reply_LPEXC) && tx_reply_LPEXC==0 && tx_reply_LPEXC_time~=0; LPEXC_check=LPEXC_check+1; end
    if ~isnan(tx_send_LPEXC) && tx_send_LPEXC==0 && tx_send_LPEXC_time~=0; LPEXC_check=LPEXC_check+1; end
    if ~isnan(tx_post_LPEXC) && tx_post_LPEXC==0 && tx_post_LPEXC_time~=0; LPEXC_check=LPEXC_check+1; end

    if ~isnan(bk_pre_LPEXC) && bk_pre_LPEXC==0 && bk_pre_LPEXC_time~=0; LPEXC_check=LPEXC_check+1; end
    if ~isnan(bk_LPEXC) && bk_LPEXC==0 && bk_LPEXC_time~=0; LPEXC_check=LPEXC_check+1; end
    if ~isnan(bk_post_LPEXC) && bk_post_LPEXC==0 && bk_post_LPEXC_time~=0; LPEXC_check=LPEXC_check+1; end
```

Create Database and export

```
Output_1=num2cell([Ss,Device,trigger,Collision,NaN,ck_ped,ck_txbk_rta,ck_pre_time,ck_post_time,txbk_msg_latency,tx_msg_latency,brake_event_delay,...
    txbk_pre_rms_steer,txbk_pre_LPSD,txbk_pre_LPEXC,txbk_pre_LPEXC_time,txbk_pre_avg_speed,txbk_pre_avg_FD,txbk_pre_TTC_min,txbk_pre_FD_min,txbk_pre_wnd_length,...
    txbk_rms_steer,txbk_LPSD,txbk_LPEXC,txbk_LPEXC_time,txbk_avg_speed,txbk_avg_FD,txbk_TTC_min,txbk_FD_min,txbk_collision,txbk_bk_onset,txbk_bk_offset,txbk_hy_res,
    NaN,NaN,NaN,NaN,NaN,NaN,NaN,txbk_post_rms_steer,txbk_post_LPSD, txbk_post_LPEXC, txbk_post_LPEXC_time,txbk_post_avg_speed,txbk_post_avg_FD,txbk_post_TTC_min,txb
    tx_pre_rms_steer,tx_pre_LPSD,tx_pre_LPEXC, tx_pre_LPEXC_time,tx_pre_avg_speed,tx_pre_avg_FD,tx_pre_TTC_min,tx_pre_FD_min,tx_pre_wnd_length,...
    tx_rms_steer,tx_LPSD,tx_LPEXC,tx_LPEXC_time,tx_avg_speed,tx_avg_FD,tx_TTC_min,tx_FD_min,tx_collision,NaN,NaN,NaN,NaN,NaN,NaN,NaN,...
    tx_post_rms_steer,tx_post_LPSD,tx_post_LPEXC,tx_post_LPEXC_time,tx_post_avg_speed,tx_post_avg_FD,tx_post_TTC_min,tx_post_FD_min,tx_post_wnd_length,...
    bk_pre_rms_steer,bk_pre_LPSD,bk_pre_LPEXC,bk_pre_LPEXC_time,bk_pre_avg_speed,bk_pre_avg_FD,bk_pre_TTC_min,bk_pre_FD_min,bk_pre_wnd_length,...
    bk_rms_steer,bk_LPSD,bk_LPEXC,bk_LPEXC_time,bk_avg_speed,bk_avg_FD,bk_TTC_min,bk_FD_min,bk_collision,bk_bk_onset,bk_bk_offset,bk_hy_res,bk_rta,...
    bk_post_rms_steer,bk_post_LPSD,bk_post_LPEXC,bk_post_LPEXC_time,bk_post_avg_speed,bk_post_avg_FD,bk_post_TTC_min,bk_post_FD_min,bk_post_wnd_length]);
dlmwrite(strcat(pwd,filesep,Output_flatfile_1),Output_1,'-append');

Output_2=num2cell([Ss,Device,trigger,Collision,NaN,ck_ped,ck_txbk_rta,ck_pre_time,ck_post_time,txbk_msg_latency,tx_msg_latency,brake_event_delay,...
    txbk_pre_rms_steer,txbk_pre_LPSD,txbk_pre_LPEXC,txbk_pre_LPEXC_time,txbk_pre_avg_speed,txbk_pre_avg_FD,txbk_pre_TTC_min,txbk_pre_FD_min,txbk_pre_wnd_length,...
    txbk_initiate_rms_steer,txbk_initiate_LPSD,txbk_initiate_LPEXC,txbk_initiate_LPEXC_time,txbk_initiate_avg_speed,txbk_initiate_avg_FD,txbk_initiate_TTC_min,txbk_
    txbk_readthink_rms_steer,txbk_readthink_LPSD,txbk_readthink_LPEXC,txbk_readthink_LPEXC_time,txbk_readthink_avg_speed,txbk_readthink_avg_FD,txbk_readthink_TTC_m
    txbk_bk_onset,txbk_bk_offset,txbk_hy_res,txbk_rta,NaN,...
    txbk_reply_rms_steer,txbk_reply_LPSD,txbk_reply_LPEXC,txbk_reply_LPEXC_time,txbk_reply_avg_speed,txbk_reply_avg_FD,txbk_reply_TTC_min,txbk_reply_FD_min,txbk_re
    txbk_send_rms_steer,txbk_send_LPSD,txbk_send_LPEXC,txbk_send_LPEXC_time,txbk_send_avg_speed,txbk_send_avg_FD,txbk_send_TTC_min,txbk_send_FD_min,txbk_send_colli
    txbk_post_rms_steer,txbk_post_LPSD,txbk_post_LPEXC,txbk_post_LPEXC_time,txbk_post_avg_speed,txbk_post_avg_FD,txbk_post_TTC_min,txbk_post_FD_min,txbk_post_wnd_l
    tx_pre_rms_steer,tx_pre_LPSD,tx_pre_LPEXC,tx_pre_LPEXC_time,tx_pre_avg_speed,tx_pre_avg_FD,tx_pre_TTC_min,tx_pre_FD_min,tx_pre_wnd_length,...
    tx_initiate_rms_steer,tx_initiate_LPSD,tx_initiate_LPEXC,tx_initiate_LPEXC_time,tx_initiate_avg_speed,tx_initiate_avg_FD,tx_initiate_TTC_min,tx_initiate_FD_min
    tx_readthink_rms_steer,tx_readthink_LPSD,tx_readthink_LPEXC,tx_readthink_LPEXC_time,tx_readthink_avg_speed,tx_readthink_avg_FD,tx_readthink_TTC_min,tx_readthink
    tx_reply_rms_steer,tx_reply_LPSD,tx_reply_LPEXC,tx_reply_LPEXC_time,tx_reply_avg_speed,tx_reply_avg_FD,tx_reply_TTC_min,tx_reply_FD_min,tx_reply_collision,NaN,
    tx_send_rms_steer,tx_send_LPSD,tx_send_LPEXC,tx_send_LPEXC_time,tx_send_avg_speed,tx_send_avg_FD,tx_send_TTC_min,tx_send_FD_min,tx_send_collision,NaN,...
    tx_post_rms_steer,tx_post_LPSD,tx_post_LPEXC,tx_post_LPEXC_time,tx_post_avg_speed,tx_post_avg_FD,tx_post_TTC_min,tx_post_FD_min,tx_post_wnd_length,...
    bk_pre_rms_steer,bk_pre_LPSD,bk_pre_LPEXC,bk_pre_LPEXC_time,bk_pre_avg_speed,bk_pre_avg_FD,bk_pre_TTC_min,bk_pre_FD_min,bk_pre_wnd_length,...
    bk_rms_steer,bk_LPSD,bk_LPEXC,bk_LPEXC_time,bk_avg_speed,bk_avg_FD,bk_TTC_min,bk_FD_min,bk_collision,bk_bk_onset,bk_bk_offset,bk_hy_res,bk_rta,...
    bk_post_rms_steer,bk_post_LPSD,bk_post_LPEXC,bk_post_LPEXC_time,bk_post_avg_speed,bk_post_avg_FD,bk_post_TTC_min,bk_post_FD_min,bk_post_wnd_length]);
dlmwrite(strcat(pwd,filesep,Output_flatfile_2),Output_2,'-append');

Output_index=num2cell([Ss,Device,trigger,t_count,data_end,double(Count),txbk_pre,txbk_initiate,t_txbk,gr_txbk,bkon_txbk,bkoff_txbk,rta_txbk,txbk_off,txbk_post,...
    bk_pre,t_bk,gr_bk,bkon_bk,bkoff_bk,rta_bk,bk_post,tx_pre,tx_initiate,tx_off,tx_post]);
dlmwrite(strcat(pwd,filesep,Output_indexfile),Output_index,'-append');
```

Problem log

```
if (c_check+LPEXC_check) >0;
    Output_log=num2cell([Ss,Device,device_check,t_check,c_check,LPEXC_check]);
    dlmwrite(strcat(pwd,filesep,Output_logfile),Output_log,'-append');
end
```

```
end
msgbox(strcat('Ss#  ', num2str(Ss),'  Device=  ', num2str(Device), '  Done.  Collision = ', num2str(Collision)))
```

**Run all files

```
case 'Run all files'
```

```
fPath = uigetdir('.', pwd);
if fPath==0, error('no folder selected'), end
fNames = dir( fullfile(fPath,'*.csv') );
fNames = strcat(fPath, filesep, {fNames.name});

for f=1:length(fNames)
```

```
   SimData = importdata(fNames{f},',',1);

% Create new variables in the base workspace from those fields.
for i = 1:size(SimData.colheaders, 2)
    assignin('base', genvarname(SimData.colheaders{i}), SimData.data(:,i));
end
clear i

Count = size(system_time,1);
%creates a variable equal to the amount of points of data collected.
%This is used to initialize variables to reduce computing time.
```

Subject & Condition

```
fid=strrep(fNames{f},strcat(fPath, filesep),'');
Ss = str2double(fid(3:4));

if ~isempty(strfind(fid,'glass')), Device = 0;
else if ~isempty(strfind(fid,'nexus')), Device = 1;
        %else if ~isempty(strfind(fid,'test')), Condition = 2;
            else Device=NaN; %end
    end
end

if isnan(Device), device_check=1; msgbox('Check Device.');
else device_check=0;
if Device==0; load('RefData_G.mat'); ref=find(Ss_G==Ss);
else load('RefData_N.mat'); ref=find(Ss_N==Ss); end
trigger=trigger(ref);
txbk_initiate=txbk_initiate(ref)+row_adj; txbk_readthink=txbk_readthink(ref)+row_adj; txbk_reply=txbk_reply(ref)+row_adj; txbk_send=txbk_send(ref)+row_adj; txbk_of
tx_initiate=tx_initiate(ref)+row_adj; tx_readthink=tx_readthink(ref)+row_adj; tx_reply=tx_reply(ref)+row_adj; tx_send=tx_send(ref)+row_adj; tx_off=tx_off(ref)+row_
end
```

Windows

```
t = zeros(Count,1);
for n=3:Count-1;
    if lead_speed(n-1)>=t_speed && lead_speed(n-1)-lead_speed(n)>=t_speed_decr && lead_speed(n-2)<=lead_speed(n-1) && lead_speed(n) >= lead_speed(n+1)...
            &&(system_time(n)-system_time(txbk_readthink)<trigger_delay || abs(lead_y(n)-trigger_y1)<=trigger_adj || ...
            abs(lead_y(n)-trigger_y2)<=trigger_adj || abs(lead_y(n)-trigger_y3)<=trigger_adj);
        t(n,1)=1;
    end
end
t_count=sum(t);

if t_count~=2, msgbox('Check no. of brake events'); t_check=1;
else t_check=0; end

if isnan(Device) || t_check==1;
    msgbox('check problem log');
    Output_log=num2cell([Ss,Device,device_check,t_check,NaN]);
    dlmwrite(strcat(pwd,filesep,Output_logfile),Output_log,'-append');
else
```

```
resize = zeros(Count,1);
for n=1:Count;
    if veh_speed(n) > end_speed;
        resize(n,1)=1;
    end
end
data_end=find(resize,1,'last');
```

```
FD=zeros(Count,1);
for n=1:Count;
   if lead_y(n)-veh_y(n)-FDadj>=0;
       FD(n)=lead_y(n)-veh_y(n)-FDadj;
   end
end

TTC=zeros(Count,1);
for n=1:Count;
   if lead_y(n)-veh_y(n)-FDadj>=0 && veh_speed(n)>lead_speed(n);
       TTC(n)=(((lead_y(n)-veh_y(n)-FDadj)/Multiplier)./(veh_speed(n)/100-lead_speed(n)/100))*360;
   else if veh_speed(n)<=lead_speed(n);
           TTC(n)=NaN;
       else TTC(n)=0;
       end
   end
end

bkoff = zeros(Count,1);
for n=2:Count-1;
    if brake_ped(n) == 0 & brake_ped(n+1) == 0 & brake_ped(n-1) > 0;
        bkoff(n,1)=1;
    end
end

if isempty(find(collision_vehicle, 1));
    Collision=0; else Collision=1;end

ped_check = zeros(Count,1);
for n=1:Count;
    if brake_ped(n)>0 && accel_ped(n)>0;
        ped_check(n,1)=1;
    end
end

excursion=zeros(Count,1);
```

Trigger 1

```
if trigger==1, %tx->txbk->bk
```

```
    t_txbk=find(t,1,'first'); t_bk=find(t,1,'last');
    tx_y=trigger_y1; txbk_y=trigger_y2;
    %tx
    if isnan(tx_off);
        tx_pre_ck=1; tx_pre_rms_steer=NaN; tx_pre_LPSD=NaN; tx_pre_LPEXC=NaN; tx_pre_LPEXC_time=NaN; tx_pre_avg_speed=NaN; tx_pre_avg_FD=NaN; tx_pre_TTC_min=NaN; t
        tx_rms_steer=NaN; tx_LPSD=NaN; tx_LPEXC=NaN; tx_LPEXC_time=NaN; tx_avg_speed=NaN; tx_avg_FD=NaN; tx_TTC_min=NaN; tx_FD_min=NaN; tx_collision=0;
        tx_initiate_rms_steer=NaN; tx_initiate_LPSD=NaN; tx_initiate_LPEXC=NaN; tx_initiate_LPEXC_time=NaN; tx_initiate_avg_speed=NaN; tx_initiate_avg_FD=NaN; tx_i
        tx_readthink_rms_steer=NaN; tx_readthink_LPSD=NaN; tx_readthink_LPEXC=NaN; tx_readthink_LPEXC_time=NaN; tx_readthink_avg_speed=NaN; tx_readthink_avg_FD=NaN
        tx_reply_rms_steer=NaN; tx_reply_LPSD=NaN; tx_reply_LPEXC=NaN; tx_reply_LPEXC_time=NaN; tx_reply_avg_speed=NaN; tx_reply_avg_FD=NaN; tx_reply_TTC_min=NaN;
        tx_send_rms_steer=NaN; tx_send_LPSD=NaN; tx_send_LPEXC=NaN; tx_send_LPEXC_time=NaN; tx_send_avg_speed=NaN; tx_send_avg_FD=NaN; tx_send_TTC_min=NaN; tx_send_
        tx_post_ck=1; tx_post_rms_steer=NaN; tx_post_LPSD=NaN; tx_post_LPEXC=NaN; tx_post_LPEXC_time=NaN; tx_post_avg_speed=NaN; tx_post_avg_FD=NaN; tx_post_TTC_mi

        txbk_pre2=find(lead_x(tx_initiate:txbk_initiate)~=lane_L,1,'first')+tx_initiate-1;
        if txbk_pre2-tx_initiate>pre_sec*row_sec && lead_x(tx_initiate)==lane_L;
            txbk_pre=txbk_pre2-pre_sec*row_sec; txbk_pre_ck=0;
        else txbk_pre=tx_initiate+1; txbk_pre_ck=1;
        end
    else
    %--tx_pre, Lane_C
    tx_pre2=find(lead_x(find(lead_x(1:tx_initiate)==lane_C,1,'first'):tx_initiate)~=lane_C,1,'first')+find(lead_x(1:tx_initiate)==lane_C,1,'first')-1;
    if tx_pre2>pre_sec*row_sec;
        tx_pre=tx_pre2-pre_sec*row_sec; tx_pre_ck=0; else tx_pre=1; tx_pre_ck=1;
    end

    if veh_x(tx_pre)>(lane_CR-car_width/2) || veh_x(tx_pre)<(lane_CL+car_width/2); excursion(tx_pre)=1; end
    for n=tx_pre+1:tx_pre2;
        if (veh_x(n)>(lane_CR-car_width/2) && veh_x(n-1)<=(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2) && veh_x(n-1)>=(lane_CL+car_width/2));
            excursion(n)=1;
        else if (veh_x(n)>(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2)); excursion(n)=2; end
        end
    end

    tx_pre_LPSD=sqrt(sumsqr(veh_x(tx_pre:tx_pre2)-mean(veh_x(tx_pre:tx_pre2)))/(tx_pre2-tx_pre));
    tx_pre_LPEXC=sum(excursion(tx_pre:tx_pre2)==1);
    tx_pre_LPEXC_time=sum(excursion(tx_pre:tx_pre2)~=0)/row_sec; %does not "sum", but "count" instead.
    tx_pre_rms_steer=sqrt(sumsqr(steer_wheel(tx_pre:tx_pre2))/(tx_pre2-tx_pre+1));
    tx_pre_avg_speed=mean(veh_speed(tx_pre:tx_pre2))/100;
    tx_pre_avg_FD=mean(FD(tx_pre:tx_pre2));
    tx_pre_TTC_min=min(TTC(tx_pre:tx_pre2));
    tx_pre_FD_min=min(FD(tx_pre:tx_pre2));
    tx_pre_wnd_length=system_time(tx_pre2)/1000-system_time(tx_pre)/1000;
    %--tx_during, Lane_L
    if veh_x(tx_initiate)>(lane_CL-car_width/2) || veh_x(tx_initiate)<(lane_LL+car_width/2); excursion(tx_initiate)=1; end
    for n=tx_initiate+1:tx_off;
        if (veh_x(n)>(lane_CL-car_width/2) && veh_x(n-1)<=(lane_CL-car_width/2)) || (veh_x(n)<(lane_LL+car_width/2) && veh_x(n-1)>=(lane_LL+car_width/2));
            excursion(n)=1;
```

```
    else if (veh_x(n)>(lane_CL-car_width/2)) || (veh_x(n)<(lane_LL+car_width/2)); excursion(n)=2; end
    end
end

tx_LPSD=sqrt(sumsqr(veh_x(tx_initiate:tx_off)-mean(veh_x(tx_initiate:tx_off)))/(tx_off-tx_initiate));
tx_LPEXC=sum(excursion(tx_initiate:tx_off)==1);
tx_LPEXC_time=sum(excursion(tx_initiate:tx_off)~=0)/row_sec; %does not "sum", but "count" instead.
tx_rms_steer=sqrt(sumsqr(steer_wheel(tx_initiate:tx_off))/(tx_off-tx_initiate+1));
tx_avg_speed=mean(veh_speed(tx_initiate:tx_off))/100;
tx_avg_FD=mean(FD(tx_initiate:tx_off));
tx_TTC_min=min(TTC(tx_initiate:tx_off));
tx_FD_min=min(FD(tx_initiate:tx_off));
if isempty(find(collision_vehicle(tx_initiate:tx_off), 1));
    tx_collision=0; else tx_collision=1;end
%--tx_initiate
tx_initiate_LPSD=sqrt(sumsqr(veh_x(tx_initiate:tx_readthink-1)-mean(veh_x(tx_initiate:tx_readthink-1)))/(tx_readthink-tx_initiate-1));
tx_initiate_LPEXC=sum(excursion(tx_initiate:tx_readthink-1)==1);
tx_initiate_LPEXC_time=sum(excursion(tx_initiate:tx_readthink-1)~=0)/row_sec; %does not "sum", but "count" instead.
tx_initiate_rms_steer=sqrt(sumsqr(steer_wheel(tx_initiate:tx_readthink-1))/(tx_readthink-tx_initiate));
tx_initiate_avg_speed=mean(veh_speed(tx_initiate:tx_readthink-1))/100;
tx_initiate_avg_FD=mean(FD(tx_initiate:tx_readthink-1));
tx_initiate_TTC_min=min(TTC(tx_initiate:tx_readthink-1));
tx_initiate_FD_min=min(FD(tx_initiate:tx_readthink-1));
if isempty(find(collision_vehicle(tx_initiate:tx_readthink-1), 1));
    tx_initiate_collision=0; else tx_initiate_collision=1;end
%--tx_readthink
if veh_x(tx_readthink)>(lane_CL-car_width/2) || veh_x(tx_readthink)<(lane_LL+car_width/2); excursion(tx_readthink)=1; end
tx_readthink_LPSD=sqrt(sumsqr(veh_x(tx_readthink:tx_reply-1)-mean(veh_x(tx_readthink:tx_reply-1)))/(tx_reply-tx_readthink-1));
tx_readthink_LPEXC=sum(excursion(tx_readthink:tx_reply-1)==1);
tx_readthink_LPEXC_time=sum(excursion(tx_readthink:tx_reply-1)~=0)/row_sec; %does not "sum", but "count" instead.
tx_readthink_rms_steer=sqrt(sumsqr(steer_wheel(tx_readthink:tx_reply-1))/(tx_reply-tx_readthink));
tx_readthink_avg_speed=mean(veh_speed(tx_readthink:tx_reply-1))/100;
tx_readthink_avg_FD=mean(FD(tx_readthink:tx_reply-1));
tx_readthink_TTC_min=min(TTC(tx_readthink:tx_reply-1));
tx_readthink_FD_min=min(FD(tx_readthink:tx_reply-1));
if isempty(find(collision_vehicle(tx_readthink:tx_reply-1), 1));
    tx_readthink_collision=0; else tx_readthink_collision=1;end
%--tx_reply
if veh_x(tx_reply)>(lane_CL-car_width/2) || veh_x(tx_reply)<(lane_LL+car_width/2); excursion(tx_reply)=1; end
tx_reply_LPSD=sqrt(sumsqr(veh_x(tx_reply:tx_send)-mean(veh_x(tx_reply:tx_send)))/(tx_send-tx_reply));
tx_reply_LPEXC=sum(excursion(tx_reply:tx_send)==1);
tx_reply_LPEXC_time=sum(excursion(tx_reply:tx_send)~=0)/row_sec; %does not "sum", but "count" instead.
tx_reply_rms_steer=sqrt(sumsqr(steer_wheel(tx_reply:tx_send))/(tx_send-tx_reply+1));
tx_reply_avg_speed=mean(veh_speed(tx_reply:tx_send))/100;
tx_reply_avg_FD=mean(FD(tx_reply:tx_send));
tx_reply_TTC_min=min(TTC(tx_reply:tx_send));
tx_reply_FD_min=min(FD(tx_reply:tx_send));
if isempty(find(collision_vehicle(tx_reply:tx_send), 1));
    tx_reply_collision=0; else tx_reply_collision=1;end
%--tx_send
if veh_x(tx_send+1)>(lane_CL-car_width/2) || veh_x(tx_send+1)<(lane_LL+car_width/2); excursion(tx_send+1)=1; end
tx_send_LPSD=sqrt(sumsqr(veh_x(tx_send+1:tx_off)-mean(veh_x(tx_send+1:tx_off)))/(tx_off-tx_send-1));
tx_send_LPEXC=sum(excursion(tx_send+1:tx_off)==1);
tx_send_LPEXC_time=sum(excursion(tx_send+1:tx_off)~=0)/row_sec; %does not "sum", but "count" instead.
tx_send_rms_steer=sqrt(sumsqr(steer_wheel(tx_send+1:tx_off))/(tx_off-tx_send));
tx_send_avg_speed=mean(veh_speed(tx_send+1:tx_off))/100;
tx_send_avg_FD=mean(FD(tx_send+1:tx_off));
tx_send_TTC_min=min(TTC(tx_send+1:tx_off));
tx_send_FD_min=min(FD(tx_send+1:tx_off));
if isempty(find(collision_vehicle(tx_send+1:tx_off), 1));
    tx_send_collision=0; else tx_send_collision=1;end
%--tx_post, Lane_L
txbk_pre2=find(lead_x(tx_off:txbk_initiate)~=lane_L,1,'first')+tx_off-1;
if txbk_pre2-tx_off>pre_sec*row_sec && lead_x(tx_off)==lane_L;
    txbk_pre=txbk_pre2-pre_sec*row_sec; txbk_pre_ck=0;
    if txbk_pre-tx_off-1>post_sec*row_sec;
    tx_post=tx_off+post_sec*row_sec; tx_post_ck=0; else tx_post=txbk_pre-1; tx_post_ck=1; end

    if veh_x(tx_off+1)>(lane_CL-car_width/2) || veh_x(tx_off+1)<(lane_LL+car_width/2); excursion(tx_off+1)=1; end
    for n=tx_off+1+1:tx_post;
      if (veh_x(n)>(lane_CL-car_width/2) && veh_x(n-1)<=(lane_CL-car_width/2)) || (veh_x(n)<(lane_LL+car_width/2) && veh_x(n-1)>=(lane_LL+car_width/2));
        excursion(n)=1;
        else if (veh_x(n)>(lane_CL-car_width/2)) || (veh_x(n)<(lane_LL+car_width/2)); excursion(n)=2; end
      end
    end

    tx_post_LPSD=sqrt(sumsqr(veh_x(tx_off+1:tx_post)-mean(veh_x(tx_off+1:tx_post)))/(tx_post-tx_off-1));
    tx_post_LPEXC=sum(excursion(tx_off+1:tx_post)==1);
    tx_post_LPEXC_time=sum(excursion(tx_off+1:tx_post)~=0)/row_sec; %does not "sum", but "count" instead.
    tx_post_rms_steer=sqrt(sumsqr(steer_wheel(tx_off+1:tx_post))/(tx_post-tx_off));
    tx_post_avg_speed=mean(veh_speed(tx_off+1:tx_post))/100;
    tx_post_avg_FD=mean(FD(tx_off+1:tx_post));
    tx_post_TTC_min=min(TTC(tx_off+1:tx_post));
    tx_post_FD_min=min(FD(tx_off+1:tx_post));
    tx_post_wnd_length=system_time(tx_post)/1000-system_time(tx_off+1)/1000;

else txbk_pre=tx_off+1; txbk_pre_ck=1; tx_post=NaN; tx_post_ck=1;
    tx_post_rms_steer=NaN;
    tx_post_LPSD=NaN;
    tx_post_LPEXC=NaN;
    tx_post_LPEXC_time=NaN;
```

```
    tx_post_avg_speed=NaN;
    tx_post_avg_FD=NaN;
    tx_post_TTC_min=NaN;
    tx_post_FD_min=NaN;
    tx_post_wnd_length=NaN;
end
end
%txbk
%--txbk_pre, Lane_L
if veh_x(txbk_pre)>(lane_CL-car_width/2) || veh_x(txbk_pre)<(lane_LL+car_width/2); excursion(txbk_pre)=1; end
for n=txbk_pre+1:txbk_pre2;
    if (veh_x(n)>(lane_CL-car_width/2) && veh_x(n-1)<=(lane_CL-car_width/2)) || (veh_x(n)<(lane_LL+car_width/2) && veh_x(n-1)>=(lane_LL+car_width/2));
        excursion(n)=1;
    else if (veh_x(n)>(lane_CL-car_width/2)) || (veh_x(n)<(lane_LL+car_width/2)); excursion(n)=2; end
    end
end

txbk_pre_LPSD=sqrt(sumsqr(veh_x(txbk_pre:txbk_pre2)-mean(veh_x(txbk_pre:txbk_pre2)))/(txbk_pre2-txbk_pre));
txbk_pre_LPEXC=sum(excursion(txbk_pre:txbk_pre2)==1);
txbk_pre_LPEXC_time=sum(excursion(txbk_pre:txbk_pre2)~=0)/row_sec; %does not "sum", but "count" instead.
txbk_pre_rms_steer=sqrt(sumsqr(steer_wheel(txbk_pre:txbk_pre2))/(txbk_pre2-txbk_pre+1));
txbk_pre_avg_speed=mean(veh_speed(txbk_pre:txbk_pre2))/100;
txbk_pre_avg_FD=mean(FD(txbk_pre:txbk_pre2));
txbk_pre_TTC_min=min(TTC(txbk_pre:txbk_pre2));
txbk_pre_FD_min=min(FD(txbk_pre:txbk_pre2));
txbk_pre_wnd_length=system_time(txbk_pre2)/1000-system_time(txbk_pre)/1000;
%--txbk_reaction
n = t_txbk;
bkon_txbk=n;
while n<t_bk-1;
      n=n+1;
      bkon_txbk=bkon_txbk+1;
      if brake_ped(n) ~= 0 & brake_ped((n-3):(n-1)) == 0;
        break
      end
end
n = t_txbk;
gr_txbk=n;
while n<t_bk-3;
      n=n+1;
      gr_txbk=gr_txbk+1;
      if accel_ped((n-3):(n-1)) > gr_adj & accel_ped((n+1):(n+3)) <= gr_adj & accel_ped(n) <= gr_adj & veh_speed(n) > 0;
        break
      end
end
if gr_txbk >= t_bk-3 || accel_ped(t_txbk)<= gr_adj; gr_txbk = NaN; end
if bkon_txbk >= t_bk-1 || brake_ped(t_txbk)~=0; bkon_txbk = NaN; end
if isnan(gr_txbk) && isnan(bkon_txbk); d_txbk=NaN; else d_txbk=min(gr_txbk,bkon_txbk); end
m_txbk=min(veh_speed(max(t_txbk,d_txbk):t_bk-1));
rta_txbk=find(veh_speed(max(t_txbk,d_txbk):t_bk-1)==m_txbk, 1, 'last' )+max(t_txbk,d_txbk)-1;
%min_speed_txbk=find(veh_speed(max(t_txbk,d_txbk):t_bk-1)==m_txbk, 1, 'first' )+max(t_txbk,d_txbk)-1;
if isempty(find(bkoff(max(t_txbk,d_txbk):rta_txbk),1)), bkoff_txbk=NaN; else bkoff_txbk=find(bkoff(max(t_txbk,d_txbk):rta_txbk),1,'last')+max(t_txbk,d_txbk)-1;
%--txbk_during, Lane_C
if veh_x(t_txbk)>(lane_CR-car_width/2) || veh_x(t_txbk)<(lane_CL+car_width/2); excursion(t_txbk)=1; end
for n=t_txbk+1:rta_txbk;
    if (veh_x(n)>(lane_CR-car_width/2) && veh_x(n-1)<=(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2) && veh_x(n-1)>=(lane_CL+car_width/2));
        excursion(n)=1;
    else if (veh_x(n)>(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2)); excursion(n)=2; end
    end
end

txbk_LPSD=sqrt(sumsqr(veh_x(t_txbk:rta_txbk)-mean(veh_x(t_txbk:rta_txbk)))/(rta_txbk-t_txbk));
txbk_LPEXC=sum(excursion(t_txbk:rta_txbk)==1);
txbk_LPEXC_time=sum(excursion(t_txbk:rta_txbk)~=0)/row_sec; %does not "sum", but "count" instead.
txbk_rms_steer=sqrt(sumsqr(steer_wheel(t_txbk:rta_txbk))/(rta_txbk-t_txbk+1));
txbk_avg_speed=mean(veh_speed(t_txbk:rta_txbk))/100;
txbk_avg_FD=mean(FD(t_txbk:rta_txbk));
txbk_TTC_min=min(TTC(t_txbk:rta_txbk));
txbk_FD_min=min(FD(t_txbk:rta_txbk));
if isempty(find(collision_vehicle(t_txbk:rta_txbk), 1));
    txbk_collision=0; else txbk_collision=1;end
%--txbk_initiate
txbk_initiate_LPSD=sqrt(sumsqr(veh_x(txbk_initiate:txbk_readthink-1)-mean(veh_x(txbk_initiate:txbk_readthink-1)))/(txbk_readthink-txbk_initiate-1));
txbk_initiate_LPEXC=sum(excursion(txbk_initiate:txbk_readthink-1)==1);
txbk_initiate_LPEXC_time=sum(excursion(txbk_initiate:txbk_readthink-1)~=0)/row_sec; %does not "sum", but "count" instead.
txbk_initiate_rms_steer=sqrt(sumsqr(steer_wheel(txbk_initiate:txbk_readthink-1))/(txbk_readthink-txbk_initiate));
txbk_initiate_avg_speed=mean(veh_speed(txbk_initiate:txbk_readthink-1))/100;
txbk_initiate_avg_FD=mean(FD(txbk_initiate:txbk_readthink-1));
txbk_initiate_TTC_min=min(TTC(txbk_initiate:txbk_readthink-1));
txbk_initiate_FD_min=min(FD(txbk_initiate:txbk_readthink-1));
if isempty(find(collision_vehicle(txbk_initiate:txbk_readthink-1), 1));
    txbk_initiate_collision=0; else txbk_initiate_collision=1;end
%--txbk_readthink
if veh_x(txbk_readthink)>(lane_CR-car_width/2) || veh_x(txbk_readthink)<(lane_CL+car_width/2); excursion(txbk_readthink)=1; end
txbk_readthink_LPSD=sqrt(sumsqr(veh_x(txbk_readthink:txbk_reply-1)-mean(veh_x(txbk_readthink:txbk_reply-1)))/(txbk_reply-txbk_readthink-1));
txbk_readthink_LPEXC=sum(excursion(txbk_readthink:txbk_reply-1)==1);
txbk_readthink_LPEXC_time=sum(excursion(txbk_readthink:txbk_reply-1)~=0)/row_sec; %does not "sum", but "count" instead.
txbk_readthink_rms_steer=sqrt(sumsqr(steer_wheel(txbk_readthink:txbk_reply-1))/(txbk_reply-txbk_readthink));
txbk_readthink_avg_speed=mean(veh_speed(txbk_readthink:txbk_reply-1))/100;
txbk_readthink_avg_FD=mean(FD(txbk_readthink:txbk_reply-1));
txbk_readthink_TTC_min=min(TTC(txbk_readthink:txbk_reply-1));
```

```
txbk_readthink_FD_min=min(FD(txbk_readthink:txbk_reply-1));
if isempty(find(collision_vehicle(txbk_readthink:txbk_reply-1), 1));
    txbk_readthink_collision=0; else txbk_readthink_collision=1;end
%--txbk_reply
if veh_x(txbk_reply)>(lane_CR-car_width/2) || veh_x(txbk_reply)<(lane_CL+car_width/2); excursion(txbk_reply)=1; end
txbk_reply_LPSD=sqrt(sumsqr(veh_x(txbk_reply:txbk_send)-mean(veh_x(txbk_reply:txbk_send)))/(txbk_send-txbk_reply));
txbk_reply_LPEXC=sum(excursion(txbk_reply:txbk_send)==1);
txbk_reply_LPEXC_time=sum(excursion(txbk_reply:txbk_send)~=0)/row_sec; %does not "sum", but "count" instead.
txbk_reply_rms_steer=sqrt(sumsqr(steer_wheel(txbk_reply:txbk_send))/(txbk_send-txbk_reply+1));
txbk_reply_avg_speed=mean(veh_speed(txbk_reply:txbk_send))/100;
txbk_reply_avg_FD=mean(FD(txbk_reply:txbk_send));
txbk_reply_TTC_min=min(TTC(txbk_reply:txbk_send));
txbk_reply_FD_min=min(FD(txbk_reply:txbk_send));
if isempty(find(collision_vehicle(txbk_reply:txbk_send), 1));
    txbk_reply_collision=0; else txbk_reply_collision=1;end
%--txbk_send
if veh_x(txbk_send+1)>(lane_CR-car_width/2) || veh_x(txbk_send+1)<(lane_CL+car_width/2); excursion(txbk_send+1)=1; end
txbk_send_LPSD=sqrt(sumsqr(veh_x(txbk_send+1:txbk_off)-mean(veh_x(txbk_send+1:txbk_off)))/(txbk_off-txbk_send-1));
txbk_send_LPEXC=sum(excursion(txbk_send+1:txbk_off)==1);
txbk_send_LPEXC_time=sum(excursion(txbk_send+1:txbk_off)~=0)/row_sec; %does not "sum", but "count" instead.
txbk_send_rms_steer=sqrt(sumsqr(steer_wheel(txbk_send+1:txbk_off))/(txbk_off-txbk_send));
txbk_send_avg_speed=mean(veh_speed(txbk_send+1:txbk_off))/100;
txbk_send_avg_FD=mean(FD(txbk_send+1:txbk_off));
txbk_send_TTC_min=min(TTC(txbk_send+1:txbk_off));
txbk_send_FD_min=min(FD(txbk_send+1:txbk_off));
if isempty(find(collision_vehicle(txbk_send+1:txbk_off), 1));
    txbk_send_collision=0; else txbk_send_collision=1;end
%--txbk_post, Lane_C
bk_pre2=find(lead_x(max(rta_txbk,txbk_off):t_bk)~=lane_C,1,'first')+max(rta_txbk,txbk_off)-1;
if bk_pre2-max(rta_txbk,txbk_off)>pre_sec*row_sec && lead_x(max(rta_txbk,txbk_off))==lane_C;
    bk_pre=bk_pre2-pre_sec*row_sec; bk_pre_ck=0;
    if bk_pre-max(rta_txbk,txbk_off)-1>post_sec*row_sec;
    txbk_post=max(rta_txbk,txbk_off)+post_sec*row_sec; txbk_post_ck=0; else txbk_post=bk_pre-1; txbk_post_ck=1; end

    if veh_x(max(rta_txbk,txbk_off)+1)>(lane_CR-car_width/2) || veh_x(max(rta_txbk,txbk_off)+1)<(lane_CL+car_width/2); excursion(max(rta_txbk,txbk_off)+1)=1; e
    for n=max(rta_txbk,txbk_off)+1+1:txbk_post;
        if (veh_x(n)>(lane_CR-car_width/2) && veh_x(n-1)<=(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2) && veh_x(n-1)>=(lane_CL+car_width/2));
            excursion(n)=1;
        else if (veh_x(n)>(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2)); excursion(n)=2; end
        end
    end

    txbk_post_LPSD=sqrt(sumsqr(veh_x(max(rta_txbk,txbk_off)+1:txbk_post)-mean(veh_x(max(rta_txbk,txbk_off)+1:txbk_post)))/(txbk_post-max(rta_txbk,txbk_off)-1))
    txbk_post_LPEXC=sum(excursion(max(rta_txbk,txbk_off)+1:txbk_post)==1);
    txbk_post_LPEXC_time=sum(excursion(max(rta_txbk,txbk_off)+1:txbk_post)~=0)/row_sec; %does not "sum", but "count" instead.
    txbk_post_rms_steer=sqrt(sumsqr(steer_wheel(max(rta_txbk,txbk_off)+1:txbk_post))/(txbk_post-max(rta_txbk,txbk_off)));
    txbk_post_avg_speed=mean(veh_speed(max(rta_txbk,txbk_off)+1:txbk_post))/100;
    txbk_post_avg_FD=mean(FD(max(rta_txbk,txbk_off)+1:txbk_post));
    txbk_post_TTC_min=min(TTC(max(rta_txbk,txbk_off)+1:txbk_post));
    txbk_post_FD_min=min(FD(max(rta_txbk,txbk_off)+1:txbk_post));
    txbk_post_wnd_length=system_time(txbk_post)/1000-system_time(max(rta_txbk,txbk_off)+1)/1000;
else bk_pre=max(rta_txbk,txbk_off)+1; bk_pre_ck=1; txbk_post=NaN; txbk_post_ck=1;
    txbk_post_rms_steer=NaN;
    txbk_post_LPSD=NaN;
    txbk_post_LPEXC=NaN;
    txbk_post_LPEXC_time=NaN;
    txbk_post_avg_speed=NaN;
    txbk_post_avg_FD=NaN;
    txbk_post_TTC_min=NaN;
    txbk_post_FD_min=NaN;
    txbk_post_wnd_length=NaN;
end

%bk
%--bk_pre, Lane_C
if veh_x(bk_pre)>(lane_CR-car_width/2) || veh_x(bk_pre)<(lane_CL+car_width/2); excursion(bk_pre)=1; end
for n=bk_pre+1:bk_pre2;
    if (veh_x(n)>(lane_CR-car_width/2) && veh_x(n-1)<=(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2) && veh_x(n-1)>=(lane_CL+car_width/2));
        excursion(n)=1;
    else if (veh_x(n)>(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2)); excursion(n)=2; end
    end
end

bk_pre_LPSD=sqrt(sumsqr(veh_x(bk_pre:bk_pre2)-mean(veh_x(bk_pre:bk_pre2)))/(bk_pre2-bk_pre));
bk_pre_LPEXC=sum(excursion(bk_pre:bk_pre2)==1);
bk_pre_LPEXC_time=sum(excursion(bk_pre:bk_pre2)~=0)/row_sec; %does not "sum", but "count" instead.
bk_pre_rms_steer=sqrt(sumsqr(steer_wheel(bk_pre:bk_pre2))/(bk_pre2-bk_pre+1));
bk_pre_avg_speed=mean(veh_speed(bk_pre:bk_pre2))/100;
bk_pre_avg_FD=mean(FD(bk_pre:bk_pre2));
bk_pre_TTC_min=min(TTC(bk_pre:bk_pre2));
bk_pre_FD_min=min(FD(bk_pre:bk_pre2));
bk_pre_wnd_length=system_time(bk_pre2)/1000-system_time(bk_pre)/1000;
%--bk_reaction
n = t_bk+2;
t_bk_r=n;
while n<double(Count)-2;
      n=n+1;
      t_bk_r=t_bk_r+1;
      if lead_speed(n-2)>=lead_speed(n-1) && lead_speed(n-1)>=lead_speed(n) && lead_speed(n) < lead_speed(n+1) && lead_speed(n+1) <= lead_speed(n+2)
        break
      end
```

```
end

n = t_bk;
bkon_bk=n;
while n<data_end;
      n=n+1;
      bkon_bk=bkon_bk+1;
      if brake_ped(n) ~= 0 & brake_ped((n-3):(n-1)) == 0;
        break
      end
end
n = t_bk;
gr_bk=n;
while n<data_end-3;
      n=n+1;
      gr_bk=gr_bk+1;
      if accel_ped((n-3):(n-1)) > gr_adj & accel_ped((n+1):(n+3)) <= gr_adj & accel_ped(n) <= gr_adj & veh_speed(n) > 0;
        break
      end
end
if gr_bk >= data_end-3 || accel_ped(t_bk)<= gr_adj; gr_bk = NaN; end
if bkon_bk >= data_end || brake_ped(t_bk)~=0; bkon_bk = NaN; end
if isnan(gr_bk) && isnan(bkon_bk); d_bk=NaN; else d_bk=min(gr_bk,bkon_bk); end

if isempty(find(resize(t_bk_r:Count), 1)); data_end=double(Count);end

m_bk=min(veh_speed(max(t_bk,d_bk):data_end));
rta_bk=find(veh_speed(max(t_bk,d_bk):data_end)==m_bk, 1, 'last' )+max(t_bk,d_bk)-1;
%min_speed_bk=find(veh_speed(max(t_bk,d_bk):data_end)==m_bk, 1, 'first' )+max(t_bk,d_bk)-1;
if isempty(find(bkoff(max(t_bk,d_bk):rta_bk),1)), bkoff_bk=NaN; else bkoff_bk=find(bkoff(max(t_bk,d_bk):rta_bk),1,'last')+max(t_bk,d_bk)-1; end
%--bk_during, Lane_R
if veh_x(t_bk)>(lane_RR-car_width/2) || veh_x(t_bk)<(lane_CR+car_width/2); excursion(t_bk)=1; end
for n=t_bk+1:rta_bk;
    if (veh_x(n)>(lane_RR-car_width/2) && veh_x(n-1)<=(lane_RR-car_width/2)) || (veh_x(n)<(lane_CR+car_width/2) && veh_x(n-1)>=(lane_CR+car_width/2));
        excursion(n)=1;
    else if (veh_x(n)>(lane_RR-car_width/2)) || (veh_x(n)<(lane_CR+car_width/2)); excursion(n)=2; end
    end
end

bk_LPSD=sqrt(sumsqr(veh_x(t_bk:rta_bk)-mean(veh_x(t_bk:rta_bk)))/(rta_bk-t_bk));
bk_LPEXC=sum(excursion(t_bk:rta_bk)==1);
bk_LPEXC_time=sum(excursion(t_bk:rta_bk)~=0)/row_sec; %does not "sum", but "count" instead.
bk_rms_steer=sqrt(sumsqr(steer_wheel(t_bk:rta_bk))/(rta_bk-t_bk+1));
bk_avg_speed=mean(veh_speed(t_bk:rta_bk))/100;
bk_avg_FD=mean(FD(t_bk:rta_bk));
bk_TTC_min=min(TTC(t_bk:rta_bk));
bk_FD_min=min(FD(t_bk:rta_bk));
if isempty(find(collision_vehicle(t_bk:rta_bk), 1));
    bk_collision=0; else bk_collision=1;end
%--bk_post, Lane_R
if data_end==double(Count); bk_post=NaN; bk_post_rms_steer=NaN; bk_post_LPSD=NaN; bk_post_LPEXC=NaN; bk_post_LPEXC_time=NaN; bk_post_avg_speed=NaN;...
        bk_post_avg_FD=NaN; bk_post_TTC_min=NaN; bk_post_FD_min=NaN; bk_post_wnd_length=NaN; bk_post_ck=1; else
if data_end-rta_bk>post_sec*row_sec;
    bk_post=rta_bk+post_sec*row_sec; bk_post_ck=0; else bk_post=data_end; bk_post_ck=1;
end

if veh_x(rta_bk+1)>(lane_RR-car_width/2) || veh_x(rta_bk+1)<(lane_CR+car_width/2); excursion(rta_bk+1)=1; end
for n=rta_bk+1+1:bk_post;
    if (veh_x(n)>(lane_RR-car_width/2) && veh_x(n-1)<=(lane_RR-car_width/2)) || (veh_x(n)<(lane_CR+car_width/2) && veh_x(n-1)>=(lane_CR+car_width/2));
        excursion(n)=1;
    else if (veh_x(n)>(lane_RR-car_width/2)) || (veh_x(n)<(lane_CR+car_width/2)); excursion(n)=2; end
    end
end

bk_post_LPSD=sqrt(sumsqr(veh_x(rta_bk+1:bk_post)-mean(veh_x(rta_bk+1:bk_post)))/(bk_post-rta_bk-1));
bk_post_LPEXC=sum(excursion(rta_bk+1:bk_post)==1);
bk_post_LPEXC_time=sum(excursion(rta_bk+1:bk_post)~=0)/row_sec; %does not "sum", but "count" instead.
bk_post_rms_steer=sqrt(sumsqr(steer_wheel(rta_bk+1:bk_post))/(bk_post-rta_bk));
bk_post_avg_speed=mean(veh_speed(rta_bk+1:bk_post))/100;
bk_post_avg_FD=mean(FD(rta_bk+1:bk_post));
bk_post_TTC_min=min(TTC(rta_bk+1:bk_post));
bk_post_FD_min=min(FD(rta_bk+1:bk_post));
bk_post_wnd_length=system_time(bk_post)/1000-system_time(rta_bk+1)/1000;
end
%--check
if isnan(tx_off); tx_msg_latency=NaN; ck_ped_tx=0;
else
n = tx_pre;
tx_y_row=n;
while n<tx_initiate;
      n=n+1;
      tx_y_row=tx_y_row+1;
      if tx_y-lead_y(n)<trigger_adj;
        break
      end
end
tx_msg_latency=(system_time(tx_initiate)-system_time(tx_y_row))/1000;

if isempty(find(ped_check(tx_pre:max(tx_post,tx_off)), 1)); ck_ped_tx=0; else ck_ped_tx=1; end
end
```

```
    if isempty(find(ped_check(txbk_pre:max(txbk_post,max(rta_txbk,txbk_off))), 1)); ck_ped_txbk=0; else ck_ped_txbk=1; end
    if isempty(find(ped_check(bk_pre:max(bk_post,rta_bk)), 1)); ck_ped_bk=0; else ck_ped_bk=1; end
    if (ck_ped_tx+ck_ped_txbk+ck_ped_bk)>0; ck_ped=1; else ck_ped=0; end
    if rta_txbk<txbk_reply; ck_txbk_rta=0;
        else if rta_txbk<txbk_send; ck_txbk_rta=1;
            else if rta_txbk<=txbk_off; ck_txbk_rta=2; else ck_txbk_rta=3; end
            end
    end
    if (tx_pre_ck+txbk_pre_ck+bk_pre_ck)>0; ck_pre_time=1; else ck_pre_time=1; end
    if (tx_post_ck+txbk_post_ck+bk_post_ck)>0; ck_post_time=1; else ck_post_time=1; end
    n = txbk_pre;
    txbk_y_row=n;
    while n<txbk_initiate;
          n=n+1;
          txbk_y_row=txbk_y_row+1;
          if txbk_y-lead_y(n)<trigger_adj;
            break
          end
    end
    txbk_msg_latency=(system_time(txbk_initiate)-system_time(txbk_y_row))/1000;

    brake_event_delay=(system_time(t_txbk)-system_time(txbk_readthink))/1000;
```

Trigger 2

```
else t_bk=find(t,1,'first'); t_txbk=find(t,1,'last'); %bk->txbk->tx
    txbk_y=trigger_y2; tx_y=trigger_y3;

    %bk
    %--bk_pre, Lane_C

    bk_pre2=find(lead_x(find(lead_x(1:t_bk)==lane_C,1,'first'):t_bk)~=lane_C,1,'first')+find(lead_x(1:t_bk)==lane_C,1,'first')-1;
    if bk_pre2>pre_sec*row_sec;
        bk_pre=bk_pre2-pre_sec*row_sec; bk_pre_ck=0; else bk_pre=1; bk_pre_ck=1;
    end

    if veh_x(bk_pre)>(lane_CR-car_width/2) || veh_x(bk_pre)<(lane_CL+car_width/2); excursion(bk_pre)=1; end
    for n=bk_pre+1:bk_pre2;
        if (veh_x(n)>(lane_CR-car_width/2) && veh_x(n-1)<=(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2) && veh_x(n-1)>=(lane_CL+car_width/2));
            excursion(n)=1;
        else if (veh_x(n)>(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2)); excursion(n)=2; end
        end
    end

    bk_pre_LPSD=sqrt(sumsqr(veh_x(bk_pre:bk_pre2)-mean(veh_x(bk_pre:bk_pre2)))/(bk_pre2-bk_pre));
    bk_pre_LPEXC=sum(excursion(bk_pre:bk_pre2)==1);
    bk_pre_LPEXC_time=sum(excursion(bk_pre:bk_pre2)~=0)/row_sec; %does not "sum", but "count" instead.
    bk_pre_rms_steer=sqrt(sumsqr(steer_wheel(bk_pre:bk_pre2))/(bk_pre2-bk_pre+1));
    bk_pre_avg_speed=mean(veh_speed(bk_pre:bk_pre2))/100;
    bk_pre_avg_FD=mean(FD(bk_pre:bk_pre2));
    bk_pre_TTC_min=min(TTC(bk_pre:bk_pre2));
    bk_pre_FD_min=min(FD(bk_pre:bk_pre2));
    bk_pre_wnd_length=system_time(bk_pre2)/1000-system_time(bk_pre)/1000;
    %--bk_reaction
    n = t_bk;
    bkon_bk=n;
    while n<t_txbk-1;
          n=n+1;
          bkon_bk=bkon_bk+1;
          if brake_ped(n) ~= 0 & brake_ped((n-3):(n-1)) == 0;
            break
          end
    end
    n = t_bk;
    gr_bk=n;
    while n<t_txbk-3;
          n=n+1;
          gr_bk=gr_bk+1;
          if accel_ped((n-3):(n-1)) > gr_adj & accel_ped((n+1):(n+3)) <= gr_adj & accel_ped(n) <= gr_adj & veh_speed(n) > 0;
            break
          end
    end
    if gr_bk >= t_txbk-3 || accel_ped(t_bk)<= gr_adj; gr_bk = NaN; end
    if bkon_bk >= t_txbk-1 || brake_ped(t_bk)~=0; bkon_bk = NaN; end
    if isnan(gr_bk) && isnan(bkon_bk); d_bk=NaN; else d_bk=min(gr_bk,bkon_bk); end

    m_bk=min(veh_speed(max(t_bk,d_bk):t_txbk-1));
    rta_bk=find(veh_speed(max(t_bk,d_bk):t_txbk-1)==m_bk, 1, 'last' )+max(t_bk,d_bk)-1;
    %min_speed_bk=find(veh_speed(max(t_bk,d_bk):t_txbk-1)==m_bk, 1, 'first' )+max(t_bk,d_bk)-1;
    if isempty(find(bkoff(max(t_bk,d_bk):rta_bk),1)), bkoff_bk=NaN; else bkoff_bk=find(bkoff(max(t_bk,d_bk):rta_bk),1,'last')+max(t_bk,d_bk)-1; end
    %--bk_during, Lane_L
    if veh_x(t_bk)>(lane_CL-car_width/2) || veh_x(t_bk)<(lane_LL+car_width/2); excursion(t_bk)=1; end
    for n=t_bk+1:rta_bk;
        if (veh_x(n)>(lane_CL-car_width/2) && veh_x(n-1)<=(lane_CL-car_width/2)) || (veh_x(n)<(lane_LL+car_width/2) && veh_x(n-1)>=(lane_LL+car_width/2));
            excursion(n)=1;
        else if (veh_x(n)>(lane_CL-car_width/2)) || (veh_x(n)<(lane_LL+car_width/2)); excursion(n)=2; end
        end
    end

    bk_LPSD=sqrt(sumsqr(veh_x(t_bk:rta_bk)-mean(veh_x(t_bk:rta_bk)))/(rta_bk-t_bk));
```

```
bk_LPEXC=sum(excursion(t_bk:rta_bk)==1);
bk_LPEXC_time=sum(excursion(t_bk:rta_bk)~=0)/row_sec; %does not "sum", but "count" instead.
bk_rms_steer=sqrt(sumsqr(steer_wheel(t_bk:rta_bk))/(rta_bk-t_bk+1));
bk_avg_speed=mean(veh_speed(t_bk:rta_bk))/100;
bk_avg_FD=mean(FD(t_bk:rta_bk));
bk_TTC_min=min(TTC(t_bk:rta_bk));
bk_FD_min=min(FD(t_bk:rta_bk));
if isempty(find(collision_vehicle(t_bk:rta_bk), 1));
    bk_collision=0; else bk_collision=1;end
%--bk_post, Lane_L
txbk_pre2=find(lead_x(rta_bk:txbk_initiate)~=lane_L,1,'first')+rta_bk-1;
if txbk_pre2-rta_bk>pre_sec*row_sec && lead_x(rta_bk)==lane_L;
    txbk_pre=txbk_pre2-pre_sec*row_sec; txbk_pre_ck=0;
    if txbk_pre-rta_bk-1>post_sec*row_sec;
    bk_post=rta_bk+post_sec*row_sec; bk_post_ck=0; else bk_post=txbk_pre-1; bk_post_ck=1; end

    if veh_x(rta_bk+1)>(lane_CL-car_width/2) || veh_x(rta_bk+1)<(lane_LL+car_width/2); excursion(rta_bk+1)=1; end
    for n=rta_bk+1+1:bk_post;
        if (veh_x(n)>(lane_CL-car_width/2) && veh_x(n-1)<=(lane_CL-car_width/2)) || (veh_x(n)<(lane_LL+car_width/2) && veh_x(n-1)>=(lane_LL+car_width/2));
        excursion(n)=1;
        else if (veh_x(n)>(lane_CL-car_width/2)) || (veh_x(n)<(lane_LL+car_width/2)); excursion(n)=2; end
        end
    end

    bk_post_LPSD=sqrt(sumsqr(veh_x(rta_bk+1:bk_post)-mean(veh_x(rta_bk+1:bk_post)))/(bk_post-rta_bk-1));
    bk_post_LPEXC=sum(excursion(rta_bk+1:bk_post)==1);
    bk_post_LPEXC_time=sum(excursion(rta_bk+1:bk_post)~=0)/row_sec; %does not "sum", but "count" instead.
    bk_post_rms_steer=sqrt(sumsqr(steer_wheel(rta_bk+1:bk_post))/(bk_post-rta_bk));
    bk_post_avg_speed=mean(veh_speed(rta_bk+1:bk_post))/100;
    bk_post_avg_FD=mean(FD(rta_bk+1:bk_post));
    bk_post_TTC_min=min(TTC(rta_bk+1:bk_post));
    bk_post_FD_min=min(FD(rta_bk+1:bk_post));
    bk_post_wnd_length=system_time(bk_post)/1000-system_time(rta_bk+1)/1000;
    else txbk_pre=rta_bk+1; txbk_pre_ck=1; bk_post=NaN; bk_post_ck=1;
    bk_post_rms_steer=NaN;
    bk_post_LPSD=NaN;
    bk_post_LPEXC=NaN;
    bk_post_LPEXC_time=NaN;
    bk_post_avg_speed=NaN;
    bk_post_avg_FD=NaN;
    bk_post_TTC_min=NaN;
    bk_post_FD_min=NaN;
    bk_post_wnd_length=NaN;
end

%txbk
%--txbk_pre, Lane_L
if veh_x(txbk_pre)>(lane_CL-car_width/2) || veh_x(txbk_pre)<(lane_LL+car_width/2); excursion(txbk_pre)=1; end
for n=txbk_pre+1:txbk_pre2;
    if (veh_x(n)>(lane_CL-car_width/2) && veh_x(n-1)<=(lane_CL-car_width/2)) || (veh_x(n)<(lane_LL+car_width/2) && veh_x(n-1)>=(lane_LL+car_width/2));
        excursion(n)=1;
    else if (veh_x(n)>(lane_CL-car_width/2)) || (veh_x(n)<(lane_LL+car_width/2)); excursion(n)=2; end
    end
end

txbk_pre_LPSD=sqrt(sumsqr(veh_x(txbk_pre:txbk_pre2)-mean(veh_x(txbk_pre:txbk_pre2)))/(txbk_pre2-txbk_pre));
txbk_pre_LPEXC=sum(excursion(txbk_pre:txbk_pre2)==1);
txbk_pre_LPEXC_time=sum(excursion(txbk_pre:txbk_pre2)~=0)/row_sec; %does not "sum", but "count" instead.
txbk_pre_rms_steer=sqrt(sumsqr(steer_wheel(txbk_pre:txbk_pre2))/(txbk_pre2-txbk_pre+1));
txbk_pre_avg_speed=mean(veh_speed(txbk_pre:txbk_pre2))/100;
txbk_pre_avg_FD=mean(FD(txbk_pre:txbk_pre2));
txbk_pre_TTC_min=min(TTC(txbk_pre:txbk_pre2));
txbk_pre_FD_min=min(FD(txbk_pre:txbk_pre2));
txbk_pre_wnd_length=system_time(txbk_pre2)/1000-system_time(txbk_pre)/1000;
%--txbk_reaction
n = t_txbk;
bkon_txbk=n;
while n<tx_initiate-1;
      n=n+1;
      bkon_txbk=bkon_txbk+1;
      if brake_ped(n) ~= 0 & brake_ped((n-3):(n-1)) == 0;
        break
      end
end
n = t_txbk;
gr_txbk=n;
while n<tx_initiate-3;
      n=n+1;
      gr_txbk=gr_txbk+1;
      if accel_ped((n-3):(n-1)) > gr_adj & accel_ped((n+1):(n+3)) <= gr_adj & accel_ped(n) <= gr_adj & veh_speed(n) > 0;
        break
      end
end
if gr_txbk >= tx_initiate-3 || accel_ped(t_txbk)<= gr_adj; gr_txbk = NaN; end
if bkon_txbk >= tx_initiate-1 || brake_ped(t_txbk)~=0; bkon_txbk = NaN; end
if isnan(gr_txbk) && isnan(bkon_txbk); d_txbk=NaN; else d_txbk=min(gr_txbk,bkon_txbk); end
m_txbk=min(veh_speed(max(t_txbk,d_txbk):tx_initiate-1));
rta_txbk=find(veh_speed(max(t_txbk,d_txbk):tx_initiate-1)==m_txbk, 1, 'last' )+max(t_txbk,d_txbk)-1;
%min_speed_txbk=find(veh_speed(max(t_txbk,d_txbk):tx_initiate-1)==m_txbk, 1, 'first' )+max(t_txbk,d_txbk)-1;
if isempty(find(bkoff(max(t_txbk,d_txbk):rta_txbk),1)), bkoff_txbk=NaN; else bkoff_txbk=find(bkoff(max(t_txbk,d_txbk):rta_txbk),1,'last')+max(t_txbk,d_txbk)-1;
%--txbk_during, Lane_C
```

```
if veh_x(t_txbk)>(lane_CR-car_width/2) || veh_x(t_txbk)<(lane_CL+car_width/2); excursion(t_txbk)=1; end
for n=t_txbk+1:rta_txbk;
    if (veh_x(n)>(lane_CR-car_width/2) && veh_x(n-1)<=(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2) && veh_x(n-1)>=(lane_CL+car_width/2));
        excursion(n)=1;
    else if (veh_x(n)>(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2)); excursion(n)=2; end
    end
end

txbk_LPSD=sqrt(sumsqr(veh_x(t_txbk:rta_txbk)-mean(veh_x(t_txbk:rta_txbk)))/(rta_txbk-t_txbk));
txbk_LPEXC=sum(excursion(t_txbk:rta_txbk)==1);
txbk_LPEXC_time=sum(excursion(t_txbk:rta_txbk)~=0)/row_sec; %does not "sum", but "count" instead.
txbk_rms_steer=sqrt(sumsqr(steer_wheel(t_txbk:rta_txbk))/(rta_txbk-t_txbk+1));
txbk_avg_speed=mean(veh_speed(t_txbk:rta_txbk))/100;
txbk_avg_FD=mean(FD(t_txbk:rta_txbk));
txbk_TTC_min=min(TTC(t_txbk:rta_txbk));
txbk_FD_min=min(FD(t_txbk:rta_txbk));
if isempty(find(collision_vehicle(t_txbk:rta_txbk), 1));
    txbk_collision=0; else txbk_collision=1;end
%--txbk_initiate
txbk_initiate_LPSD=sqrt(sumsqr(veh_x(txbk_initiate:txbk_readthink-1)-mean(veh_x(txbk_initiate:txbk_readthink-1)))/(txbk_readthink-txbk_initiate-1));
txbk_initiate_LPEXC=sum(excursion(txbk_initiate:txbk_readthink-1)==1);
txbk_initiate_LPEXC_time=sum(excursion(txbk_initiate:txbk_readthink-1)~=0)/row_sec; %does not "sum", but "count" instead.
txbk_initiate_rms_steer=sqrt(sumsqr(steer_wheel(txbk_initiate:txbk_readthink-1))/(txbk_readthink-txbk_initiate));
txbk_initiate_avg_speed=mean(veh_speed(txbk_initiate:txbk_readthink-1))/100;
txbk_initiate_avg_FD=mean(FD(txbk_initiate:txbk_readthink-1));
txbk_initiate_TTC_min=min(TTC(txbk_initiate:txbk_readthink-1));
txbk_initiate_FD_min=min(FD(txbk_initiate:txbk_readthink-1));
if isempty(find(collision_vehicle(txbk_initiate:txbk_readthink-1), 1));
    txbk_initiate_collision=0; else txbk_initiate_collision=1;end
%--txbk_readthink
if veh_x(txbk_readthink)>(lane_CR-car_width/2) || veh_x(txbk_readthink)<(lane_CL+car_width/2); excursion(txbk_readthink)=1; end
txbk_readthink_LPSD=sqrt(sumsqr(veh_x(txbk_readthink:txbk_reply-1)-mean(veh_x(txbk_readthink:txbk_reply-1)))/(txbk_reply-txbk_readthink-1));
txbk_readthink_LPEXC=sum(excursion(txbk_readthink:txbk_reply-1)==1);
txbk_readthink_LPEXC_time=sum(excursion(txbk_readthink:txbk_reply-1)~=0)/row_sec; %does not "sum", but "count" instead.
txbk_readthink_rms_steer=sqrt(sumsqr(steer_wheel(txbk_readthink:txbk_reply-1))/(txbk_reply-txbk_readthink));
txbk_readthink_avg_speed=mean(veh_speed(txbk_readthink:txbk_reply-1))/100;
txbk_readthink_avg_FD=mean(FD(txbk_readthink:txbk_reply-1));
txbk_readthink_TTC_min=min(TTC(txbk_readthink:txbk_reply-1));
txbk_readthink_FD_min=min(FD(txbk_readthink:txbk_reply-1));
if isempty(find(collision_vehicle(txbk_readthink:txbk_reply-1), 1));
    txbk_readthink_collision=0; else txbk_readthink_collision=1;end
%--txbk_reply
if veh_x(txbk_reply)>(lane_CR-car_width/2) || veh_x(txbk_reply)<(lane_CL+car_width/2); excursion(txbk_reply)=1; end
txbk_reply_LPSD=sqrt(sumsqr(veh_x(txbk_reply:txbk_send)-mean(veh_x(txbk_reply:txbk_send)))/(txbk_send-txbk_reply));
txbk_reply_LPEXC=sum(excursion(txbk_reply:txbk_send)==1);
txbk_reply_LPEXC_time=sum(excursion(txbk_reply:txbk_send)~=0)/row_sec; %does not "sum", but "count" instead.
txbk_reply_rms_steer=sqrt(sumsqr(steer_wheel(txbk_reply:txbk_send))/(txbk_send-txbk_reply+1));
txbk_reply_avg_speed=mean(veh_speed(txbk_reply:txbk_send))/100;
txbk_reply_avg_FD=mean(FD(txbk_reply:txbk_send));
txbk_reply_TTC_min=min(TTC(txbk_reply:txbk_send));
txbk_reply_FD_min=min(FD(txbk_reply:txbk_send));
if isempty(find(collision_vehicle(txbk_reply:txbk_send), 1));
    txbk_reply_collision=0; else txbk_reply_collision=1;end
%--txbk_send
if veh_x(txbk_send+1)>(lane_CR-car_width/2) || veh_x(txbk_send+1)<(lane_CL+car_width/2); excursion(txbk_send+1)=1; end
txbk_send_LPSD=sqrt(sumsqr(veh_x(txbk_send+1:txbk_off)-mean(veh_x(txbk_send+1:txbk_off)))/(txbk_off-txbk_send-1));
txbk_send_LPEXC=sum(excursion(txbk_send+1:txbk_off)==1);
txbk_send_LPEXC_time=sum(excursion(txbk_send+1:txbk_off)~=0)/row_sec; %does not "sum", but "count" instead.
txbk_send_rms_steer=sqrt(sumsqr(steer_wheel(txbk_send+1:txbk_off))/(txbk_off-txbk_send));
txbk_send_avg_speed=mean(veh_speed(txbk_send+1:txbk_off))/100;
txbk_send_avg_FD=mean(FD(txbk_send+1:txbk_off));
txbk_send_TTC_min=min(TTC(txbk_send+1:txbk_off));
txbk_send_FD_min=min(FD(txbk_send+1:txbk_off));
if isempty(find(collision_vehicle(txbk_send+1:txbk_off), 1));
    txbk_send_collision=0; else txbk_send_collision=1;end
%--txbk_post, Lane_C
if isnan(tx_off);
    tx_pre=NaN; tx_pre2=NaN;
    txbk_post=max(rta_txbk,txbk_off)+post_sec*row_sec; txbk_post_ck=0;

    if veh_x(max(rta_txbk,txbk_off)+1)>(lane_CR-car_width/2) || veh_x(max(rta_txbk,txbk_off)+1)<(lane_CL+car_width/2); excursion(max(rta_txbk,txbk_off)+1)=1; e
    for n=max(rta_txbk,txbk_off)+1+1:txbk_post;
        if (veh_x(n)>(lane_CR-car_width/2) && veh_x(n-1)<=(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2) && veh_x(n-1)>=(lane_CL+car_width/2));
            excursion(n)=1;
        else if (veh_x(n)>(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2)); excursion(n)=2; end
        end
    end

    txbk_post_LPSD=sqrt(sumsqr(veh_x(max(rta_txbk,txbk_off)+1:txbk_post)-mean(veh_x(max(rta_txbk,txbk_off)+1:txbk_post)))/(txbk_post-max(rta_txbk,txbk_off)-1))
    txbk_post_LPEXC=sum(excursion(max(rta_txbk,txbk_off)+1:txbk_post)==1);
    txbk_post_LPEXC_time=sum(excursion(max(rta_txbk,txbk_off)+1:txbk_post)~=0)/row_sec; %does not "sum", but "count" instead.
    txbk_post_rms_steer=sqrt(sumsqr(steer_wheel(max(rta_txbk,txbk_off)+1:txbk_post))/(txbk_post-max(rta_txbk,txbk_off)));
    txbk_post_avg_speed=mean(veh_speed(max(rta_txbk,txbk_off)+1:txbk_post))/100;
    txbk_post_avg_FD=mean(FD(max(rta_txbk,txbk_off)+1:txbk_post));
    txbk_post_TTC_min=min(TTC(max(rta_txbk,txbk_off)+1:txbk_post));
    txbk_post_FD_min=min(FD(max(rta_txbk,txbk_off)+1:txbk_post));
    txbk_post_wnd_length=system_time(txbk_post)/1000-system_time(max(rta_txbk,txbk_off)+1)/1000;
else
tx_pre2=find(lead_x(max(rta_txbk,txbk_off):tx_initiate)~=lane_C,1,'first')+max(rta_txbk,txbk_off)-1;
if tx_pre2-max(rta_txbk,txbk_off)>pre_sec*row_sec && lead_x(max(rta_txbk,txbk_off))==lane_C;
    tx_pre=tx_pre2-pre_sec*row_sec; tx_pre_ck=0;
```

```
    if tx_pre-max(rta_txbk,txbk_off)-1>post_sec*row_sec;
    txbk_post=max(rta_txbk,txbk_off)+post_sec*row_sec; txbk_post_ck=0; else txbk_post=tx_pre-1; txbk_post_ck=1; end

    if veh_x(max(rta_txbk,txbk_off)+1)>(lane_CR-car_width/2) || veh_x(max(rta_txbk,txbk_off)+1)<(lane_CL+car_width/2); excursion(max(rta_txbk,txbk_off)+1)=1; e
    for n=max(rta_txbk,txbk_off)+1+1:txbk_post;
        if (veh_x(n)>(lane_CR-car_width/2) && veh_x(n-1)<=(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2) && veh_x(n-1)>=(lane_CL+car_width/2));
            excursion(n)=1;
        else if (veh_x(n)>(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2)); excursion(n)=2; end
        end
    end

    txbk_post_LPSD=sqrt(sumsqr(veh_x(max(rta_txbk,txbk_off)+1:txbk_post)-mean(veh_x(max(rta_txbk,txbk_off)+1:txbk_post)))/(txbk_post-max(rta_txbk,txbk_off)-1))
    txbk_post_LPEXC=sum(excursion(max(rta_txbk,txbk_off)+1:txbk_post)==1);
    txbk_post_LPEXC_time=sum(excursion(max(rta_txbk,txbk_off)+1:txbk_post)~=0)/row_sec; %does not "sum", but "count" instead.
    txbk_post_rms_steer=sqrt(sumsqr(steer_wheel(max(rta_txbk,txbk_off)+1:txbk_post))/(txbk_post-max(rta_txbk,txbk_off)));
    txbk_post_avg_speed=mean(veh_speed(max(rta_txbk,txbk_off)+1:txbk_post))/100;
    txbk_post_avg_FD=mean(FD(max(rta_txbk,txbk_off)+1:txbk_post));
    txbk_post_TTC_min=min(TTC(max(rta_txbk,txbk_off)+1:txbk_post));
    txbk_post_FD_min=min(FD(max(rta_txbk,txbk_off)+1:txbk_post));
    txbk_post_wnd_length=system_time(txbk_post)/1000-system_time(max(rta_txbk,txbk_off)+1)/1000;
    else tx_pre=max(rta_txbk,txbk_off)+1; tx_pre_ck=1; txbk_post=NaN; txbk_post_ck=1;
    txbk_post_rms_steer=NaN;
    txbk_post_LPSD=NaN;
    txbk_post_LPEXC=NaN;
    txbk_post_LPEXC_time=NaN;
    txbk_post_avg_speed=NaN;
    txbk_post_avg_FD=NaN;
    txbk_post_TTC_min=NaN;
    txbk_post_FD_min=NaN;
    txbk_post_wnd_length=NaN;
end
end

%tx
if isnan(tx_off);
    tx_pre_ck=1; tx_pre_rms_steer=NaN; tx_pre_LPSD=NaN; tx_pre_LPEXC=NaN; tx_pre_LPEXC_time=NaN; tx_pre_avg_speed=NaN; tx_pre_avg_FD=NaN; tx_pre_TTC_min=NaN; t
    tx_rms_steer=NaN; tx_LPSD=NaN; tx_LPEXC=NaN; tx_LPEXC_time=NaN; tx_avg_speed=NaN; tx_avg_FD=NaN; tx_TTC_min=NaN; tx_FD_min=NaN; tx_collision=0;
    tx_initiate_rms_steer=NaN; tx_initiate_LPSD=NaN; tx_initiate_LPEXC=NaN; tx_initiate_LPEXC_time=NaN; tx_initiate_avg_speed=NaN; tx_initiate_avg_FD=NaN; tx_i
    tx_readthink_rms_steer=NaN; tx_readthink_LPSD=NaN; tx_readthink_LPEXC=NaN; tx_readthink_LPEXC_time=NaN; tx_readthink_avg_speed=NaN; tx_readthink_avg_FD=NaN
    tx_reply_rms_steer=NaN; tx_reply_LPSD=NaN; tx_reply_LPEXC=NaN; tx_reply_LPEXC_time=NaN; tx_reply_avg_speed=NaN; tx_reply_avg_FD=NaN; tx_reply_TTC_min=NaN;
    tx_send_rms_steer=NaN; tx_send_LPSD=NaN; tx_send_LPEXC=NaN; tx_send_LPEXC_time=NaN; tx_send_avg_speed=NaN; tx_send_avg_FD=NaN; tx_send_TTC_min=NaN; tx_send
    tx_post_ck=1; tx_post_rms_steer=NaN; tx_post_LPSD=NaN; tx_post_LPEXC=NaN; tx_post_LPEXC_time=NaN; tx_post_avg_speed=NaN; tx_post_avg_FD=NaN; tx_post_TTC_mi
else
%--tx_pre, Lane_C
if veh_x(tx_pre)>(lane_CR-car_width/2) || veh_x(tx_pre)<(lane_CL+car_width/2); excursion(tx_pre)=1; end
for n=tx_pre+1:tx_pre2;
    if (veh_x(n)>(lane_CR-car_width/2) && veh_x(n-1)<=(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2) && veh_x(n-1)>=(lane_CL+car_width/2));
        excursion(n)=1;
    else if (veh_x(n)>(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2)); excursion(n)=2; end
    end
end

tx_pre_LPSD=sqrt(sumsqr(veh_x(tx_pre:tx_pre2)-mean(veh_x(tx_pre:tx_pre2)))/(tx_pre2-tx_pre));
tx_pre_LPEXC=sum(excursion(tx_pre:tx_pre2)==1);
tx_pre_LPEXC_time=sum(excursion(tx_pre:tx_pre2)~=0)/row_sec; %does not "sum", but "count" instead.
tx_pre_rms_steer=sqrt(sumsqr(steer_wheel(tx_pre:tx_pre2))/(tx_pre2-tx_pre+1));
tx_pre_avg_speed=mean(veh_speed(tx_pre:tx_pre2))/100;
tx_pre_avg_FD=mean(FD(tx_pre:tx_pre2));
tx_pre_TTC_min=min(TTC(tx_pre:tx_pre2));
tx_pre_FD_min=min(FD(tx_pre:tx_pre2));
tx_pre_wnd_length=system_time(tx_pre2)/1000-system_time(tx_pre)/1000;
%--tx_during, Lane_R
if veh_x(tx_initiate)>(lane_RR-car_width/2) || veh_x(tx_initiate)<(lane_CR+car_width/2); excursion(tx_initiate)=1; end
for n=tx_initiate+1:tx_off;
    if (veh_x(n)>(lane_RR-car_width/2) && veh_x(n-1)<=(lane_RR-car_width/2)) || (veh_x(n)<(lane_CR+car_width/2) && veh_x(n-1)>=(lane_CR+car_width/2));
        excursion(n)=1;
    else if (veh_x(n)>(lane_RR-car_width/2)) || (veh_x(n)<(lane_CR+car_width/2)); excursion(n)=2; end
    end
end

tx_LPSD=sqrt(sumsqr(veh_x(tx_initiate:tx_off)-mean(veh_x(tx_initiate:tx_off)))/(tx_off-tx_initiate));
tx_LPEXC=sum(excursion(tx_initiate:tx_off)==1);
tx_LPEXC_time=sum(excursion(tx_initiate:tx_off)~=0)/row_sec; %does not "sum", but "count" instead.
tx_rms_steer=sqrt(sumsqr(steer_wheel(tx_initiate:tx_off))/(tx_off-tx_initiate+1));
tx_avg_speed=mean(veh_speed(tx_initiate:tx_off))/100;
tx_avg_FD=mean(FD(tx_initiate:tx_off));
tx_TTC_min=min(TTC(tx_initiate:tx_off));
tx_FD_min=min(FD(tx_initiate:tx_off));
if isempty(find(collision_vehicle(tx_initiate:tx_off), 1));
    tx_collision=0; else tx_collision=1;end
%--tx_initiate
tx_initiate_LPSD=sqrt(sumsqr(veh_x(tx_initiate:tx_readthink-1)-mean(veh_x(tx_initiate:tx_readthink-1)))/(tx_readthink-tx_initiate-1));
tx_initiate_LPEXC=sum(excursion(tx_initiate:tx_readthink-1)==1);
tx_initiate_LPEXC_time=sum(excursion(tx_initiate:tx_readthink-1)~=0)/row_sec; %does not "sum", but "count" instead.
tx_initiate_rms_steer=sqrt(sumsqr(steer_wheel(tx_initiate:tx_readthink-1))/(tx_readthink-tx_initiate));
tx_initiate_avg_speed=mean(veh_speed(tx_initiate:tx_readthink-1))/100;
tx_initiate_avg_FD=mean(FD(tx_initiate:tx_readthink-1));
tx_initiate_TTC_min=min(TTC(tx_initiate:tx_readthink-1));
tx_initiate_FD_min=min(FD(tx_initiate:tx_readthink-1));
if isempty(find(collision_vehicle(tx_initiate:tx_readthink-1), 1));
    tx_initiate_collision=0; else tx_initiate_collision=1;end
```

```
%--tx_readthink
if veh_x(tx_readthink)>(lane_RR-car_width/2) || veh_x(tx_readthink)<(lane_CR+car_width/2); excursion(tx_readthink)=1; end
tx_readthink_LPSD=sqrt(sumsqr(veh_x(tx_readthink:tx_reply-1)-mean(veh_x(tx_readthink:tx_reply-1)))/(tx_reply-tx_readthink-1));
tx_readthink_LPEXC=sum(excursion(tx_readthink:tx_reply-1)==1);
tx_readthink_LPEXC_time=sum(excursion(tx_readthink:tx_reply-1)~=0)/row_sec; %does not "sum", but "count" instead.
tx_readthink_rms_steer=sqrt(sumsqr(steer_wheel(tx_readthink:tx_reply-1))/(tx_reply-tx_readthink));
tx_readthink_avg_speed=mean(veh_speed(tx_readthink:tx_reply-1))/100;
tx_readthink_avg_FD=mean(FD(tx_readthink:tx_reply-1));
tx_readthink_TTC_min=min(TTC(tx_readthink:tx_reply-1));
tx_readthink_FD_min=min(FD(tx_readthink:tx_reply-1));
if isempty(find(collision_vehicle(tx_readthink:tx_reply-1), 1));
    tx_readthink_collision=0; else tx_readthink_collision=1;end
%--tx_reply
if veh_x(tx_reply)>(lane_RR-car_width/2) || veh_x(tx_reply)<(lane_CR+car_width/2); excursion(tx_reply)=1; end
tx_reply_LPSD=sqrt(sumsqr(veh_x(tx_reply:tx_send)-mean(veh_x(tx_reply:tx_send)))/(tx_send-tx_reply));
tx_reply_LPEXC=sum(excursion(tx_reply:tx_send)==1);
tx_reply_LPEXC_time=sum(excursion(tx_reply:tx_send)~=0)/row_sec; %does not "sum", but "count" instead.
tx_reply_rms_steer=sqrt(sumsqr(steer_wheel(tx_reply:tx_send))/(tx_send-tx_reply+1));
tx_reply_avg_speed=mean(veh_speed(tx_reply:tx_send))/100;
tx_reply_avg_FD=mean(FD(tx_reply:tx_send));
tx_reply_TTC_min=min(TTC(tx_reply:tx_send));
tx_reply_FD_min=min(FD(tx_reply:tx_send));
if isempty(find(collision_vehicle(tx_reply:tx_send), 1));
    tx_reply_collision=0; else tx_reply_collision=1;end
%--tx_send
if veh_x(tx_send+1)>(lane_RR-car_width/2) || veh_x(tx_send+1)<(lane_CR+car_width/2); excursion(tx_send+1)=1; end
tx_send_LPSD=sqrt(sumsqr(veh_x(tx_send+1:tx_off)-mean(veh_x(tx_send+1:tx_off)))/(tx_off-tx_send-1));
tx_send_LPEXC=sum(excursion(tx_send+1:tx_off)==1);
tx_send_LPEXC_time=sum(excursion(tx_send+1:tx_off)~=0)/row_sec; %does not "sum", but "count" instead.
tx_send_rms_steer=sqrt(sumsqr(steer_wheel(tx_send+1:tx_off))/(tx_off-tx_send));
tx_send_avg_speed=mean(veh_speed(tx_send+1:tx_off))/100;
tx_send_avg_FD=mean(FD(tx_send+1:tx_off));
tx_send_TTC_min=min(TTC(tx_send+1:tx_off));
tx_send_FD_min=min(FD(tx_send+1:tx_off));
if isempty(find(collision_vehicle(tx_send+1:tx_off), 1));
    tx_send_collision=0; else tx_send_collision=1;end
%--tx_post, Lane_R
if tx_off>=data_end; data_end=double(Count); end
if data_end-tx_off>post_sec*row_sec;
    tx_post=tx_off+post_sec*row_sec; tx_post_ck=0; else tx_post=data_end; tx_post_ck=1;
end

if veh_x(tx_off+1)>(lane_RR-car_width/2) || veh_x(tx_off+1)<(lane_CR+car_width/2); excursion(tx_off+1)=1; end
for n=tx_off+1+1:tx_post;
  if (veh_x(n)>(lane_RR-car_width/2) && veh_x(n-1)<=(lane_RR-car_width/2)) || (veh_x(n)<(lane_CR+car_width/2) && veh_x(n-1)>=(lane_CR+car_width/2));
        excursion(n)=1;
    else if (veh_x(n)>(lane_RR-car_width/2)) || (veh_x(n)<(lane_CR+car_width/2)); excursion(n)=2; end
  end
end

tx_post_LPSD=sqrt(sumsqr(veh_x(tx_off+1:tx_post)-mean(veh_x(tx_off+1:tx_post)))/(tx_post-tx_off-1));
tx_post_LPEXC=sum(excursion(tx_off+1:tx_post)==1);
tx_post_LPEXC_time=sum(excursion(tx_off+1:tx_post)~=0)/row_sec; %does not "sum", but "count" instead.
tx_post_rms_steer=sqrt(sumsqr(steer_wheel(tx_off+1:tx_post))/(tx_post-tx_off));
tx_post_avg_speed=mean(veh_speed(tx_off+1:tx_post))/100;
tx_post_avg_FD=mean(FD(tx_off+1:tx_post));
tx_post_TTC_min=min(TTC(tx_off+1:tx_post));
tx_post_FD_min=min(FD(tx_off+1:tx_post));
tx_post_wnd_length=system_time(tx_post)/1000-system_time(tx_off+1)/1000;
end
%--check
if isnan(tx_off); tx_msg_latency=NaN; ck_ped_tx=0;
else
n = tx_pre;
tx_y_row=n;
while n<tx_initiate;
      n=n+1;
      tx_y_row=tx_y_row+1;
      if tx_y-lead_y(n)<trigger_adj;
        break
      end
end
tx_msg_latency=(system_time(tx_initiate)-system_time(tx_y_row))/1000;

if isempty(find(ped_check(tx_pre:max(tx_post,tx_off)), 1)); ck_ped_tx=0; else ck_ped_tx=1; end
end

if isempty(find(ped_check(txbk_pre:max(txbk_post,max(rta_txbk,txbk_off))), 1)); ck_ped_txbk=0; else ck_ped_txbk=1; end
if isempty(find(ped_check(bk_pre:max(bk_post,rta_bk)), 1)); ck_ped_bk=0; else ck_ped_bk=1; end
if (ck_ped_tx+ck_ped_txbk+ck_ped_bk)>0; ck_ped=1; else ck_ped=0; end
if rta_txbk<txbk_reply; ck_txbk_rta=0;
    else if rta_txbk<txbk_send; ck_txbk_rta=1;
        else if rta_txbk<=txbk_off; ck_txbk_rta=2; else ck_txbk_rta=3; end
        end
end
if (tx_pre_ck+txbk_pre_ck+bk_pre_ck)>0; ck_pre_time=1; else ck_pre_time=1; end
if (tx_post_ck+txbk_post_ck+bk_post_ck)>0; ck_post_time=1; else ck_post_time=1; end
n = txbk_pre;
txbk_y_row=n;
while n<txbk_initiate;
      n=n+1;
```

```
            txbk_y_row=txbk_y_row+1;
            if txbk_y-lead_y(n)<trigger_adj;
              break
            end
    end
    txbk_msg_latency=(system_time(txbk_initiate)-system_time(txbk_y_row))/1000;

    brake_event_delay=(system_time(t_txbk)-system_time(txbk_readthink))/1000;
end

if isnan(bkon_txbk); txbk_bk_onset=NaN; txbk_bk_offset=NaN; txbk_rta=NaN;
else txbk_bk_onset=system_time(bkon_txbk)/1000-system_time(t_txbk)/1000;
    if isnan(bkoff_txbk); txbk_bk_offset=NaN; else txbk_bk_offset=system_time(bkoff_txbk)/1000-system_time(bkon_txbk)/1000; end
    txbk_rta=system_time(rta_txbk)/1000-system_time(bkon_txbk)/1000;
end
if isnan(d_txbk); txbk_hy_res=NaN; else txbk_hy_res=system_time(d_txbk)/1000-system_time(t_txbk)/1000; end

if isnan(bkon_bk); bk_bk_onset=NaN; bk_bk_offset=NaN; bk_rta=NaN;
else bk_bk_onset=system_time(bkon_bk)/1000-system_time(t_bk)/1000;
    if isnan(bkoff_bk); bk_bk_offset=NaN; else bk_bk_offset=system_time(bkoff_bk)/1000-system_time(bkon_bk)/1000; end
    if rta_bk==data_end; bk_rta=NaN; else bk_rta=system_time(rta_bk)/1000-system_time(bkon_bk)/1000; end
end
if isnan(d_bk); bk_hy_res=NaN; else bk_hy_res=system_time(d_bk)/1000-system_time(t_bk)/1000; end

if (Collision==0 && (tx_collision+bk_collision+txbk_collision)==0) || (Collision==1 && (tx_collision+bk_collision+txbk_collision)>0);
    c_check=0; else c_check=1;
end

if ~isnan(txbk_pre_LPEXC) && txbk_pre_LPEXC==0 && txbk_pre_LPEXC_time~=0; LPEXC_check=1; else LPEXC_check=0; end
if ~isnan(txbk_LPEXC) && txbk_LPEXC==0 && txbk_LPEXC_time~=0; LPEXC_check=LPEXC_check+1; end
if ~isnan(txbk_initiate_LPEXC) && txbk_initiate_LPEXC==0 && txbk_initiate_LPEXC_time~=0; LPEXC_check=LPEXC_check+1; end
if ~isnan(txbk_readthink_LPEXC) && txbk_readthink_LPEXC==0 && txbk_readthink_LPEXC_time~=0; LPEXC_check=LPEXC_check+1; end
if ~isnan(txbk_reply_LPEXC) && txbk_reply_LPEXC==0 && txbk_reply_LPEXC_time~=0; LPEXC_check=LPEXC_check+1; end
if ~isnan(txbk_send_LPEXC) && txbk_send_LPEXC==0 && txbk_send_LPEXC_time~=0; LPEXC_check=LPEXC_check+1; end
if ~isnan(txbk_post_LPEXC) && txbk_post_LPEXC==0 && txbk_post_LPEXC_time~=0; LPEXC_check=LPEXC_check+1; end

if ~isnan(tx_pre_LPEXC) && tx_pre_LPEXC==0 && tx_pre_LPEXC_time~=0; LPEXC_check=LPEXC_check+1; end
if ~isnan(tx_LPEXC) && tx_LPEXC==0 && tx_LPEXC_time~=0; LPEXC_check=LPEXC_check+1; end
if ~isnan(tx_initiate_LPEXC) && tx_initiate_LPEXC==0 && tx_initiate_LPEXC_time~=0; LPEXC_check=LPEXC_check+1; end
if ~isnan(tx_readthink_LPEXC) && tx_readthink_LPEXC==0 && tx_readthink_LPEXC_time~=0; LPEXC_check=LPEXC_check+1; end
if ~isnan(tx_reply_LPEXC) && tx_reply_LPEXC==0 && tx_reply_LPEXC_time~=0; LPEXC_check=LPEXC_check+1; end
if ~isnan(tx_send_LPEXC) && tx_send_LPEXC==0 && tx_send_LPEXC_time~=0; LPEXC_check=LPEXC_check+1; end
if ~isnan(tx_post_LPEXC) && tx_post_LPEXC==0 && tx_post_LPEXC_time~=0; LPEXC_check=LPEXC_check+1; end

if ~isnan(bk_pre_LPEXC) && bk_pre_LPEXC==0 && bk_pre_LPEXC_time~=0; LPEXC_check=LPEXC_check+1; end
if ~isnan(bk_LPEXC) && bk_LPEXC==0 && bk_LPEXC_time~=0; LPEXC_check=LPEXC_check+1; end
if ~isnan(bk_post_LPEXC) && bk_post_LPEXC==0 && bk_post_LPEXC_time~=0; LPEXC_check=LPEXC_check+1; end
```

Create Database and export

```
Output_1=num2cell([Ss,Device,trigger,Collision,NaN,ck_ped,ck_txbk_rta,ck_pre_time,ck_post_time,txbk_msg_latency,tx_msg_latency,brake_event_delay,...
    txbk_pre_rms_steer,txbk_pre_LPSD,txbk_pre_LPEXC,txbk_pre_LPEXC_time,txbk_pre_avg_speed,txbk_pre_avg_FD,txbk_pre_TTC_min,txbk_pre_FD_min,txbk_pre_wnd_length,...
    txbk_rms_steer,txbk_LPSD,txbk_LPEXC,txbk_LPEXC_time,txbk_avg_speed,txbk_avg_FD,txbk_TTC_min,txbk_FD_min,txbk_collision,txbk_bk_onset,txbk_bk_offset,txbk_hy_res,
    NaN,NaN,NaN,NaN,NaN,NaN,NaN,txbk_post_rms_steer,txbk_post_LPSD, txbk_post_LPEXC, txbk_post_LPEXC_time,txbk_post_avg_speed,txbk_post_avg_FD,txbk_post_TTC_min,txl
    tx_pre_rms_steer,tx_pre_LPSD,tx_pre_LPEXC, tx_pre_LPEXC_time,tx_pre_avg_speed,tx_pre_avg_FD,tx_pre_TTC_min,tx_pre_FD_min,tx_pre_wnd_length,...
    tx_rms_steer,tx_LPSD,tx_LPEXC,tx_LPEXC_time,tx_avg_speed,tx_avg_FD,tx_TTC_min,tx_FD_min,tx_collision,NaN,NaN,NaN,NaN,NaN,NaN,NaN,...
    tx_post_rms_steer,tx_post_LPSD,tx_post_LPEXC,tx_post_LPEXC_time,tx_post_avg_speed,tx_post_avg_FD,tx_post_TTC_min,tx_post_FD_min,tx_post_wnd_length,...
    bk_pre_rms_steer,bk_pre_LPSD,bk_pre_LPEXC,bk_pre_LPEXC_time,bk_pre_avg_speed,bk_pre_avg_FD,bk_pre_TTC_min,bk_pre_FD_min,bk_pre_wnd_length,...
    bk_rms_steer,bk_LPSD,bk_LPEXC,bk_LPEXC_time,bk_avg_speed,bk_avg_FD,bk_TTC_min,bk_FD_min,bk_collision,bk_bk_onset,bk_bk_offset,bk_hy_res,bk_rta,...
    bk_post_rms_steer,bk_post_LPSD,bk_post_LPEXC,bk_post_LPEXC_time,bk_post_avg_speed,bk_post_avg_FD,bk_post_TTC_min,bk_post_FD_min,bk_post_wnd_length,f]);
dlmwrite(strcat(pwd,filesep,Output_flatfile_1),Output_1,'-append');

Output_2=num2cell([Ss,Device,trigger,Collision,NaN,ck_ped,ck_txbk_rta,ck_pre_time,ck_post_time,txbk_msg_latency,tx_msg_latency,brake_event_delay,...
    txbk_pre_rms_steer,txbk_pre_LPSD,txbk_pre_LPEXC,txbk_pre_LPEXC_time,txbk_pre_avg_speed,txbk_pre_avg_FD,txbk_pre_TTC_min,txbk_pre_FD_min,txbk_pre_wnd_length,...
    txbk_initiate_rms_steer,txbk_initiate_LPSD,txbk_initiate_LPEXC,txbk_initiate_LPEXC_time,txbk_initiate_avg_speed,txbk_initiate_avg_FD,txbk_initiate_TTC_min,txbk_
    txbk_readthink_rms_steer,txbk_readthink_LPSD,txbk_readthink_LPEXC,txbk_readthink_LPEXC_time,txbk_readthink_avg_speed,txbk_readthink_avg_FD,txbk_readthink_TTC_m
    txbk_bk_onset,txbk_bk_offset,txbk_hy_res,txbk_rta,NaN,...
    txbk_reply_rms_steer,txbk_reply_LPSD,txbk_reply_LPEXC,txbk_reply_LPEXC_time,txbk_reply_avg_speed,txbk_reply_avg_FD,txbk_reply_TTC_min,txbk_reply_FD_min,txbk_re
    txbk_send_rms_steer,txbk_send_LPSD,txbk_send_LPEXC,txbk_send_LPEXC_time,txbk_send_avg_speed,txbk_send_avg_FD,txbk_send_TTC_min,txbk_send_FD_min,txbk_send_colli
    txbk_post_rms_steer,txbk_post_LPSD,txbk_post_LPEXC,txbk_post_LPEXC_time,txbk_post_avg_speed,txbk_post_avg_FD,txbk_post_TTC_min,txbk_post_FD_min,txbk_post_wnd_l
    tx_pre_rms_steer,tx_pre_LPSD,tx_pre_LPEXC,tx_pre_LPEXC_time,tx_pre_avg_speed,tx_pre_avg_FD,tx_pre_TTC_min,tx_pre_FD_min,tx_pre_wnd_length,...
    tx_initiate_rms_steer,tx_initiate_LPSD,tx_initiate_LPEXC,tx_initiate_LPEXC_time,tx_initiate_avg_speed,tx_initiate_avg_FD,tx_initiate_TTC_min,tx_initiate_FD_min
    tx_readthink_rms_steer,tx_readthink_LPSD,tx_readthink_LPEXC,tx_readthink_LPEXC_time,tx_readthink_avg_speed,tx_readthink_avg_FD,tx_readthink_TTC_min,tx_readthinl
    tx_reply_rms_steer,tx_reply_LPSD,tx_reply_LPEXC,tx_reply_LPEXC_time,tx_reply_avg_speed,tx_reply_avg_FD,tx_reply_TTC_min,tx_reply_FD_min,tx_reply_collision,NaN,
    tx_send_rms_steer,tx_send_LPSD,tx_send_LPEXC,tx_send_LPEXC_time,tx_send_avg_speed,tx_send_avg_FD,tx_send_TTC_min,tx_send_FD_min,tx_send_collision,NaN,...
    tx_post_rms_steer,tx_post_LPSD,tx_post_LPEXC,tx_post_LPEXC_time,tx_post_avg_speed,tx_post_avg_FD,tx_post_TTC_min,tx_post_FD_min,tx_post_wnd_length,...
    bk_pre_rms_steer,bk_pre_LPSD,bk_pre_LPEXC,bk_pre_LPEXC_time,bk_pre_avg_speed,bk_pre_avg_FD,bk_pre_TTC_min,bk_pre_FD_min,bk_pre_wnd_length,...
    bk_rms_steer,bk_LPSD,bk_LPEXC,bk_LPEXC_time,bk_avg_speed,bk_avg_FD,bk_TTC_min,bk_FD_min,bk_collision,bk_bk_onset,bk_bk_offset,bk_hy_res,bk_rta,...
    bk_post_rms_steer,bk_post_LPSD,bk_post_LPEXC,bk_post_LPEXC_time,bk_post_avg_speed,bk_post_avg_FD,bk_post_TTC_min,bk_post_FD_min,bk_post_wnd_length,f]);
dlmwrite(strcat(pwd,filesep,Output_flatfile_2),Output_2,'-append');

Output_index=num2cell([Ss,Device,trigger,t_count,data_end,double(Count),txbk_pre,txbk_initiate,t_txbk,gr_txbk,bkon_txbk,bkoff_txbk,rta_txbk,txbk_off,txbk_post,...
    bk_pre,t_bk,gr_bk,bkon_bk,bkoff_bk,rta_bk,bk_post,tx_pre,tx_initiate,tx_off,tx_post,f]);
dlmwrite(strcat(pwd,filesep,Output_indexfile),Output_index,'-append');
```

Problem log

```
if (c_check+LPEXC_check) >0;
    Output_log=num2cell([Ss,Device,device_check,t_check,c_check,LPEXC_check,f]);
    dlmwrite(strcat(pwd,filesep,Output_logfile),Output_log,'-append');
end
```

```
end
```

```
end

fprintf('Number of files processed: %d\n', f);

save('fNames.mat','fNames');
msgbox(strcat('Number of files processed:  ', num2str(f)));
```

**Run all new-added file(s)

```
case 'Run all newly-added file(s)'
```

```
load('fNames.mat');
fPath = uigetdir('.', pwd);
if fPath==0, error('no folder selected'), end
fNames_re = dir( fullfile(fPath,'*.csv') );
fNames_re = strcat(fPath, filesep, {fNames_re.name});
fNames_check=~ismember(fNames_re,fNames);
fNames_new=fNames_re(fNames_check);
f_diff=numel(fNames_re)-numel(fNames);

for f=1:length(fNames_new)
```

```
    SimData = importdata(fNames_new{f},',',1);
% Create new variables in the base workspace from those fields.
for i = 1:size(SimData.colheaders, 2)
    assignin('base', genvarname(SimData.colheaders{i}), SimData.data(:,i));
end

clear i

Count = size(system_time,1);
%creates a variable equal to the amount of points of data collected.
%This is used to initialize variables to reduce computing time.
```

Subject & Condition

```
fid=strrep(fNames_new{f},strcat(fPath, filesep),'');
Ss = str2double(fid(3:4));

if ~isempty(strfind(fid,'glass')), Device = 0;
else if ~isempty(strfind(fid,'nexus')), Device = 1;
        %else if ~isempty(strfind(fid,'test')), Condition = 2;
            else Device=NaN; %end
    end
end

if isnan(Device), device_check=1; msgbox('Check Device.');
else device_check=0;
if Device==0; load('RefData_G.mat'); ref=find(Ss_G==Ss);
else load('RefData_N.mat'); ref=find(Ss_N==Ss); end
trigger=trigger(ref);
txbk_initiate=txbk_initiate(ref)+row_adj; txbk_readthink=txbk_readthink(ref)+row_adj; txbk_reply=txbk_reply(ref)+row_adj; txbk_send=txbk_send(ref)+row_adj; txbk_of
tx_initiate=tx_initiate(ref)+row_adj; tx_readthink=tx_readthink(ref)+row_adj; tx_reply=tx_reply(ref)+row_adj; tx_send=tx_send(ref)+row_adj; tx_off=tx_off(ref)+row_
end
```

Windows

```
t = zeros(Count,1);
for n=3:Count-1;
    if lead_speed(n-1)>=t_speed && lead_speed(n-1)-lead_speed(n)>=t_speed_decr && lead_speed(n-2)<=lead_speed(n-1) && lead_speed(n) >= lead_speed(n+1)...
            &&(system_time(n)-system_time(txbk_readthink)<trigger_delay || abs(lead_y(n)-trigger_y1)<=trigger_adj || ...
            abs(lead_y(n)-trigger_y2)<=trigger_adj || abs(lead_y(n)-trigger_y3)<=trigger_adj);
        t(n,1)=1;
    end
end
t_count=sum(t);

if t_count~=2, msgbox('Check no. of brake events'); t_check=1;
else t_check=0; end

if isnan(Device) || t_check==1;
    msgbox('check problem log');
    Output_log=num2cell([Ss,Device,device_check,t_check,NaN]);
```

```
    dlmwrite(strcat(pwd,filesep,Output_logfile),Output_log,'-append');
else
```

```
resize = zeros(Count,1);
for n=1:Count;
    if veh_speed(n) > end_speed;
        resize(n,1)=1;
    end
end
data_end=find(resize,1,'last');


FD=zeros(Count,1);
for n=1:Count;
   if lead_y(n)-veh_y(n)-FDadj>=0;
       FD(n)=lead_y(n)-veh_y(n)-FDadj;
   end
end

TTC=zeros(Count,1);
for n=1:Count;
   if lead_y(n)-veh_y(n)-FDadj>=0 && veh_speed(n)>lead_speed(n);
       TTC(n)=(((lead_y(n)-veh_y(n)-FDadj)/Multiplier)./(veh_speed(n)/100-lead_speed(n)/100))*360;
   else if veh_speed(n)<=lead_speed(n);
           TTC(n)=NaN;
       else TTC(n)=0;
       end
   end
end

bkoff = zeros(Count,1);
for n=2:Count-1;
    if brake_ped(n) == 0 && brake_ped(n+1) == 0 && brake_ped(n-1) > 0;
        bkoff(n,1)=1;
    end
end

if isempty(find(collision_vehicle, 1));
    Collision=0; else Collision=1;end

ped_check = zeros(Count,1);
for n=1:Count;
    if brake_ped(n)>0 && accel_ped(n)>0;
        ped_check(n,1)=1;
    end
end

excursion=zeros(Count,1);
```

Trigger 1

```
if trigger==1, %tx->txbk->bk
```

```
    t_txbk=find(t,1,'first'); t_bk=find(t,1,'last');
    tx_y=trigger_y1; txbk_y=trigger_y2;
    %tx
    if isnan(tx_off);
        tx_pre_ck=1; tx_pre_rms_steer=NaN; tx_pre_LPSD=NaN; tx_pre_LPEXC=NaN; tx_pre_LPEXC_time=NaN; tx_pre_avg_speed=NaN; tx_pre_avg_FD=NaN; tx_pre_TTC_min=NaN; t
        tx_rms_steer=NaN; tx_LPSD=NaN; tx_LPEXC=NaN; tx_LPEXC_time=NaN; tx_avg_speed=NaN; tx_avg_FD=NaN; tx_TTC_min=NaN; tx_FD_min=NaN; tx_collision=0;
        tx_initiate_rms_steer=NaN; tx_initiate_LPSD=NaN; tx_initiate_LPEXC=NaN; tx_initiate_LPEXC_time=NaN; tx_initiate_avg_speed=NaN; tx_initiate_avg_FD=NaN; tx_i
        tx_readthink_rms_steer=NaN; tx_readthink_LPSD=NaN; tx_readthink_LPEXC=NaN; tx_readthink_LPEXC_time=NaN; tx_readthink_avg_speed=NaN; tx_readthink_avg_FD=NaN
        tx_reply_rms_steer=NaN; tx_reply_LPSD=NaN; tx_reply_LPEXC=NaN; tx_reply_LPEXC_time=NaN; tx_reply_avg_speed=NaN; tx_reply_avg_FD=NaN; tx_reply_TTC_min=NaN;
        tx_send_rms_steer=NaN; tx_send_LPSD=NaN; tx_send_LPEXC=NaN; tx_send_LPEXC_time=NaN; tx_send_avg_speed=NaN; tx_send_avg_FD=NaN; tx_send_TTC_min=NaN; tx_send
        tx_post_ck=1; tx_post_rms_steer=NaN; tx_post_LPSD=NaN; tx_post_LPEXC=NaN; tx_post_LPEXC_time=NaN; tx_post_avg_speed=NaN; tx_post_avg_FD=NaN; tx_post_TTC_mi

        txbk_pre2=find(lead_x(tx_initiate:txbk_initiate)~=lane_L,1,'first')+tx_initiate-1;
        if txbk_pre2-tx_initiate>pre_sec*row_sec && lead_x(tx_initiate)==lane_L;
            txbk_pre=txbk_pre2-pre_sec*row_sec; txbk_pre_ck=0;
        else txbk_pre=tx_initiate+1; txbk_pre_ck=1;
        end
    else
    %--tx_pre, Lane_C
    tx_pre2=find(lead_x(find(lead_x(1:tx_initiate)==lane_C,1,'first'):tx_initiate)~=lane_C,1,'first')+find(lead_x(1:tx_initiate)==lane_C,1,'first')-1;
    if tx_pre2>pre_sec*row_sec;
        tx_pre=tx_pre2-pre_sec*row_sec; tx_pre_ck=0; else tx_pre=1; tx_pre_ck=1;
    end

    if veh_x(tx_pre)>(lane_CR-car_width/2) || veh_x(tx_pre)<(lane_CL+car_width/2); excursion(tx_pre)=1; end
    for n=tx_pre+1:tx_pre2;
        if (veh_x(n)>(lane_CR-car_width/2) && veh_x(n-1)<=(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2) && veh_x(n-1)>=(lane_CL+car_width/2));
            excursion(n)=1;
        else if (veh_x(n)>(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2)); excursion(n)=2; end
        end
    end

    tx_pre_LPSD=sqrt(sumsqr(veh_x(tx_pre:tx_pre2)-mean(veh_x(tx_pre:tx_pre2)))/(tx_pre2-tx_pre));
    tx_pre_LPEXC=sum(excursion(tx_pre:tx_pre2)==1);
```

```
tx_pre_LPEXC_time=sum(excursion(tx_pre:tx_pre2)~=0)/row_sec; %does not "sum", but "count" instead.
tx_pre_rms_steer=sqrt(sumsqr(steer_wheel(tx_pre:tx_pre2))/(tx_pre2-tx_pre+1));
tx_pre_avg_speed=mean(veh_speed(tx_pre:tx_pre2))/100;
tx_pre_avg_FD=mean(FD(tx_pre:tx_pre2));
tx_pre_TTC_min=min(TTC(tx_pre:tx_pre2));
tx_pre_FD_min=min(FD(tx_pre:tx_pre2));
tx_pre_wnd_length=system_time(tx_pre2)/1000-system_time(tx_pre)/1000;
%--tx_during, Lane_L
if veh_x(tx_initiate)>(lane_CL-car_width/2) || veh_x(tx_initiate)<(lane_LL+car_width/2); excursion(tx_initiate)=1; end
for n=tx_initiate+1:tx_off;
    if (veh_x(n)>(lane_CL-car_width/2) && veh_x(n-1)<=(lane_CL-car_width/2)) || (veh_x(n)<(lane_LL+car_width/2) && veh_x(n-1)>=(lane_LL+car_width/2));
        excursion(n)=1;
    else if (veh_x(n)>(lane_CL-car_width/2)) || (veh_x(n)<(lane_LL+car_width/2)); excursion(n)=2; end
    end
end

tx_LPSD=sqrt(sumsqr(veh_x(tx_initiate:tx_off)-mean(veh_x(tx_initiate:tx_off)))/(tx_off-tx_initiate));
tx_LPEXC=sum(excursion(tx_initiate:tx_off)==1);
tx_LPEXC_time=sum(excursion(tx_initiate:tx_off)~=0)/row_sec; %does not "sum", but "count" instead.
tx_rms_steer=sqrt(sumsqr(steer_wheel(tx_initiate:tx_off))/(tx_off-tx_initiate+1));
tx_avg_speed=mean(veh_speed(tx_initiate:tx_off))/100;
tx_avg_FD=mean(FD(tx_initiate:tx_off));
tx_TTC_min=min(TTC(tx_initiate:tx_off));
tx_FD_min=min(FD(tx_initiate:tx_off));
if isempty(find(collision_vehicle(tx_initiate:tx_off), 1));
    tx_collision=0; else tx_collision=1;end
%--tx_initiate
tx_initiate_LPSD=sqrt(sumsqr(veh_x(tx_initiate:tx_readthink-1)-mean(veh_x(tx_initiate:tx_readthink-1)))/(tx_readthink-tx_initiate-1));
tx_initiate_LPEXC=sum(excursion(tx_initiate:tx_readthink-1)==1);
tx_initiate_LPEXC_time=sum(excursion(tx_initiate:tx_readthink-1)~=0)/row_sec; %does not "sum", but "count" instead.
tx_initiate_rms_steer=sqrt(sumsqr(steer_wheel(tx_initiate:tx_readthink-1))/(tx_readthink-tx_initiate));
tx_initiate_avg_speed=mean(veh_speed(tx_initiate:tx_readthink-1))/100;
tx_initiate_avg_FD=mean(FD(tx_initiate:tx_readthink-1));
tx_initiate_TTC_min=min(TTC(tx_initiate:tx_readthink-1));
tx_initiate_FD_min=min(FD(tx_initiate:tx_readthink-1));
if isempty(find(collision_vehicle(tx_initiate:tx_readthink-1), 1));
    tx_initiate_collision=0; else tx_initiate_collision=1;end
%--tx_readthink
if veh_x(tx_readthink)>(lane_CL-car_width/2) || veh_x(tx_readthink)<(lane_LL+car_width/2); excursion(tx_readthink)=1; end
tx_readthink_LPSD=sqrt(sumsqr(veh_x(tx_readthink:tx_reply-1)-mean(veh_x(tx_readthink:tx_reply-1)))/(tx_reply-tx_readthink-1));
tx_readthink_LPEXC=sum(excursion(tx_readthink:tx_reply-1)==1);
tx_readthink_LPEXC_time=sum(excursion(tx_readthink:tx_reply-1)~=0)/row_sec; %does not "sum", but "count" instead.
tx_readthink_rms_steer=sqrt(sumsqr(steer_wheel(tx_readthink:tx_reply-1))/(tx_reply-tx_readthink));
tx_readthink_avg_speed=mean(veh_speed(tx_readthink:tx_reply-1))/100;
tx_readthink_avg_FD=mean(FD(tx_readthink:tx_reply-1));
tx_readthink_TTC_min=min(TTC(tx_readthink:tx_reply-1));
tx_readthink_FD_min=min(FD(tx_readthink:tx_reply-1));
if isempty(find(collision_vehicle(tx_readthink:tx_reply-1), 1));
    tx_readthink_collision=0; else tx_readthink_collision=1;end
%--tx_reply
if veh_x(tx_reply)>(lane_CL-car_width/2) || veh_x(tx_reply)<(lane_LL+car_width/2); excursion(tx_reply)=1; end
tx_reply_LPSD=sqrt(sumsqr(veh_x(tx_reply:tx_send)-mean(veh_x(tx_reply:tx_send)))/(tx_send-tx_reply));
tx_reply_LPEXC=sum(excursion(tx_reply:tx_send)==1);
tx_reply_LPEXC_time=sum(excursion(tx_reply:tx_send)~=0)/row_sec; %does not "sum", but "count" instead.
tx_reply_rms_steer=sqrt(sumsqr(steer_wheel(tx_reply:tx_send))/(tx_send-tx_reply+1));
tx_reply_avg_speed=mean(veh_speed(tx_reply:tx_send))/100;
tx_reply_avg_FD=mean(FD(tx_reply:tx_send));
tx_reply_TTC_min=min(TTC(tx_reply:tx_send));
tx_reply_FD_min=min(FD(tx_reply:tx_send));
if isempty(find(collision_vehicle(tx_reply:tx_send), 1));
    tx_reply_collision=0; else tx_reply_collision=1;end
%--tx_send
if veh_x(tx_send+1)>(lane_CL-car_width/2) || veh_x(tx_send+1)<(lane_LL+car_width/2); excursion(tx_send+1)=1; end
tx_send_LPSD=sqrt(sumsqr(veh_x(tx_send+1:tx_off)-mean(veh_x(tx_send+1:tx_off)))/(tx_off-tx_send-1));
tx_send_LPEXC=sum(excursion(tx_send+1:tx_off)==1);
tx_send_LPEXC_time=sum(excursion(tx_send+1:tx_off)~=0)/row_sec; %does not "sum", but "count" instead.
tx_send_rms_steer=sqrt(sumsqr(steer_wheel(tx_send+1:tx_off))/(tx_off-tx_send));
tx_send_avg_speed=mean(veh_speed(tx_send+1:tx_off))/100;
tx_send_avg_FD=mean(FD(tx_send+1:tx_off));
tx_send_TTC_min=min(TTC(tx_send+1:tx_off));
tx_send_FD_min=min(FD(tx_send+1:tx_off));
if isempty(find(collision_vehicle(tx_send+1:tx_off), 1));
    tx_send_collision=0; else tx_send_collision=1;end
%--tx_post, Lane_L
txbk_pre2=find(lead_x(tx_off:txbk_initiate)~=lane_L,1,'first')+tx_off-1;
if txbk_pre2-tx_off>pre_sec*row_sec && lead_x(tx_off)==lane_L;
    txbk_pre=txbk_pre2-pre_sec*row_sec; txbk_pre_ck=0;
    if txbk_pre-tx_off-1>post_sec*row_sec;
    tx_post=tx_off+post_sec*row_sec; tx_post_ck=0; else tx_post=txbk_pre-1; tx_post_ck=1; end

    if veh_x(tx_off+1)>(lane_CL-car_width/2) || veh_x(tx_off+1)<(lane_LL+car_width/2); excursion(tx_off+1)=1; end
    for n=tx_off+1+1:tx_post;
      if (veh_x(n)>(lane_CL-car_width/2) && veh_x(n-1)<=(lane_CL-car_width/2)) || (veh_x(n)<(lane_LL+car_width/2) && veh_x(n-1)>=(lane_LL+car_width/2));
        excursion(n)=1;
        else if (veh_x(n)>(lane_CL-car_width/2)) || (veh_x(n)<(lane_LL+car_width/2)); excursion(n)=2; end
      end
    end

    tx_post_LPSD=sqrt(sumsqr(veh_x(tx_off+1:tx_post)-mean(veh_x(tx_off+1:tx_post)))/(tx_post-tx_off-1));
    tx_post_LPEXC=sum(excursion(tx_off+1:tx_post)==1);
    tx_post_LPEXC_time=sum(excursion(tx_off+1:tx_post)~=0)/row_sec; %does not "sum", but "count" instead.
```

```
    tx_post_rms_steer=sqrt(sumsqr(steer_wheel(tx_off+1:tx_post))/(tx_post-tx_off));
    tx_post_avg_speed=mean(veh_speed(tx_off+1:tx_post))/100;
    tx_post_avg_FD=mean(FD(tx_off+1:tx_post));
    tx_post_TTC_min=min(TTC(tx_off+1:tx_post));
    tx_post_FD_min=min(FD(tx_off+1:tx_post));
    tx_post_wnd_length=system_time(tx_post)/1000-system_time(tx_off+1)/1000;

else txbk_pre=tx_off+1; txbk_pre_ck=1; tx_post=NaN; tx_post_ck=1;
    tx_post_rms_steer=NaN;
    tx_post_LPSD=NaN;
    tx_post_LPEXC=NaN;
    tx_post_LPEXC_time=NaN;
    tx_post_avg_speed=NaN;
    tx_post_avg_FD=NaN;
    tx_post_TTC_min=NaN;
    tx_post_FD_min=NaN;
    tx_post_wnd_length=NaN;
end
end
%txbk
%--txbk_pre, Lane_L
if veh_x(txbk_pre)>(lane_CL-car_width/2) || veh_x(txbk_pre)<(lane_LL+car_width/2); excursion(txbk_pre)=1; end
for n=txbk_pre+1:txbk_pre2;
    if (veh_x(n)>(lane_CL-car_width/2) && veh_x(n-1)<=(lane_CL-car_width/2)) || (veh_x(n)<(lane_LL+car_width/2) && veh_x(n-1)>=(lane_LL+car_width/2));
        excursion(n)=1;
    else if (veh_x(n)>(lane_CL-car_width/2)) || (veh_x(n)<(lane_LL+car_width/2)); excursion(n)=2; end
    end
end

txbk_pre_LPSD=sqrt(sumsqr(veh_x(txbk_pre:txbk_pre2)-mean(veh_x(txbk_pre:txbk_pre2)))/(txbk_pre2-txbk_pre));
txbk_pre_LPEXC=sum(excursion(txbk_pre:txbk_pre2)==1);
txbk_pre_LPEXC_time=sum(excursion(txbk_pre:txbk_pre2)~=0)/row_sec; %does not "sum", but "count" instead.
txbk_pre_rms_steer=sqrt(sumsqr(steer_wheel(txbk_pre:txbk_pre2))/(txbk_pre2-txbk_pre+1));
txbk_pre_avg_speed=mean(veh_speed(txbk_pre:txbk_pre2))/100;
txbk_pre_avg_FD=mean(FD(txbk_pre:txbk_pre2));
txbk_pre_TTC_min=min(TTC(txbk_pre:txbk_pre2));
txbk_pre_FD_min=min(FD(txbk_pre:txbk_pre2));
txbk_pre_wnd_length=system_time(txbk_pre2)/1000-system_time(txbk_pre)/1000;
%--txbk_reaction
n = t_txbk;
bkon_txbk=n;
while n<t_bk-1;
      n=n+1;
      bkon_txbk=bkon_txbk+1;
      if brake_ped(n) ~= 0 & brake_ped((n-3):(n-1)) == 0;
        break
      end
end
n = t_txbk;
gr_txbk=n;
while n<t_bk-3;
      n=n+1;
      gr_txbk=gr_txbk+1;
      if accel_ped((n-3):(n-1)) > gr_adj & accel_ped((n+1):(n+3)) <= gr_adj & accel_ped(n) <= gr_adj & veh_speed(n) > 0;
        break
      end
end
if gr_txbk >= t_bk-3 || accel_ped(t_txbk)<= gr_adj; gr_txbk = NaN; end
if bkon_txbk >= t_bk-1 || brake_ped(t_txbk)~=0; bkon_txbk = NaN; end
if isnan(gr_txbk) && isnan(bkon_txbk); d_txbk=NaN; else d_txbk=min(gr_txbk,bkon_txbk); end
m_txbk=min(veh_speed(max(t_txbk,d_txbk):t_bk-1));
rta_txbk=find(veh_speed(max(t_txbk,d_txbk):t_bk-1)==m_txbk, 1, 'last' )+max(t_txbk,d_txbk)-1;
%min_speed_txbk=find(veh_speed(max(t_txbk,d_txbk):t_bk-1)==m_txbk, 1, 'first' )+max(t_txbk,d_txbk)-1;
if isempty(find(bkoff(max(t_txbk,d_txbk):rta_txbk),1)), bkoff_txbk=NaN; else bkoff_txbk=find(bkoff(max(t_txbk,d_txbk):rta_txbk),1,'last')+max(t_txbk,d_txbk)-1;
%--txbk_during, Lane_C
if veh_x(t_txbk)>(lane_CR-car_width/2) || veh_x(t_txbk)<(lane_CL+car_width/2); excursion(t_txbk)=1; end
for n=t_txbk+1:rta_txbk;
    if (veh_x(n)>(lane_CR-car_width/2) && veh_x(n-1)<=(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2) && veh_x(n-1)>=(lane_CL+car_width/2));
        excursion(n)=1;
    else if (veh_x(n)>(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2)); excursion(n)=2; end
    end
end

txbk_LPSD=sqrt(sumsqr(veh_x(t_txbk:rta_txbk)-mean(veh_x(t_txbk:rta_txbk)))/(rta_txbk-t_txbk));
txbk_LPEXC=sum(excursion(t_txbk:rta_txbk)==1);
txbk_LPEXC_time=sum(excursion(t_txbk:rta_txbk)~=0)/row_sec; %does not "sum", but "count" instead.
txbk_rms_steer=sqrt(sumsqr(steer_wheel(t_txbk:rta_txbk))/(rta_txbk-t_txbk+1));
txbk_avg_speed=mean(veh_speed(t_txbk:rta_txbk))/100;
txbk_avg_FD=mean(FD(t_txbk:rta_txbk));
txbk_TTC_min=min(TTC(t_txbk:rta_txbk));
txbk_FD_min=min(FD(t_txbk:rta_txbk));
if isempty(find(collision_vehicle(t_txbk:rta_txbk), 1));
    txbk_collision=0; else txbk_collision=1;end
%--txbk_initiate
txbk_initiate_LPSD=sqrt(sumsqr(veh_x(txbk_initiate:txbk_readthink-1)-mean(veh_x(txbk_initiate:txbk_readthink-1)))/(txbk_readthink-txbk_initiate-1));
txbk_initiate_LPEXC=sum(excursion(txbk_initiate:txbk_readthink-1)==1);
txbk_initiate_LPEXC_time=sum(excursion(txbk_initiate:txbk_readthink-1)~=0)/row_sec; %does not "sum", but "count" instead.
txbk_initiate_rms_steer=sqrt(sumsqr(steer_wheel(txbk_initiate:txbk_readthink-1))/(txbk_readthink-txbk_initiate));
txbk_initiate_avg_speed=mean(veh_speed(txbk_initiate:txbk_readthink-1))/100;
txbk_initiate_avg_FD=mean(FD(txbk_initiate:txbk_readthink-1));
txbk_initiate_TTC_min=min(TTC(txbk_initiate:txbk_readthink-1));
```

```
txbk_initiate_FD_min=min(FD(txbk_initiate:txbk_readthink-1));
if isempty(find(collision_vehicle(txbk_initiate:txbk_readthink-1), 1));
    txbk_initiate_collision=0; else txbk_initiate_collision=1;end
%--txbk_readthink
if veh_x(txbk_readthink)>(lane_CR-car_width/2) || veh_x(txbk_readthink)<(lane_CL+car_width/2); excursion(txbk_readthink)=1; end
txbk_readthink_LPSD=sqrt(sumsqr(veh_x(txbk_readthink:txbk_reply-1)-mean(veh_x(txbk_readthink:txbk_reply-1)))/(txbk_reply-txbk_readthink-1));
txbk_readthink_LPEXC=sum(excursion(txbk_readthink:txbk_reply-1)==1);
txbk_readthink_LPEXC_time=sum(excursion(txbk_readthink:txbk_reply-1)~=0)/row_sec; %does not "sum", but "count" instead.
txbk_readthink_rms_steer=sqrt(sumsqr(steer_wheel(txbk_readthink:txbk_reply-1))/(txbk_reply-txbk_readthink));
txbk_readthink_avg_speed=mean(veh_speed(txbk_readthink:txbk_reply-1))/100;
txbk_readthink_avg_FD=mean(FD(txbk_readthink:txbk_reply-1));
txbk_readthink_TTC_min=min(TTC(txbk_readthink:txbk_reply-1));
txbk_readthink_FD_min=min(FD(txbk_readthink:txbk_reply-1));
if isempty(find(collision_vehicle(txbk_readthink:txbk_reply-1), 1));
    txbk_readthink_collision=0; else txbk_readthink_collision=1;end
%--txbk_reply
if veh_x(txbk_reply)>(lane_CR-car_width/2) || veh_x(txbk_reply)<(lane_CL+car_width/2); excursion(txbk_reply)=1; end
txbk_reply_LPSD=sqrt(sumsqr(veh_x(txbk_reply:txbk_send)-mean(veh_x(txbk_reply:txbk_send)))/(txbk_send-txbk_reply));
txbk_reply_LPEXC=sum(excursion(txbk_reply:txbk_send)==1);
txbk_reply_LPEXC_time=sum(excursion(txbk_reply:txbk_send)~=0)/row_sec; %does not "sum", but "count" instead.
txbk_reply_rms_steer=sqrt(sumsqr(steer_wheel(txbk_reply:txbk_send))/(txbk_send-txbk_reply+1));
txbk_reply_avg_speed=mean(veh_speed(txbk_reply:txbk_send))/100;
txbk_reply_avg_FD=mean(FD(txbk_reply:txbk_send));
txbk_reply_TTC_min=min(TTC(txbk_reply:txbk_send));
txbk_reply_FD_min=min(FD(txbk_reply:txbk_send));
if isempty(find(collision_vehicle(txbk_reply:txbk_send), 1));
    txbk_reply_collision=0; else txbk_reply_collision=1;end
%--txbk_send
if veh_x(txbk_send+1)>(lane_CR-car_width/2) || veh_x(txbk_send+1)<(lane_CL+car_width/2); excursion(txbk_send+1)=1; end
txbk_send_LPSD=sqrt(sumsqr(veh_x(txbk_send+1:txbk_off)-mean(veh_x(txbk_send+1:txbk_off)))/(txbk_off-txbk_send-1));
txbk_send_LPEXC=sum(excursion(txbk_send+1:txbk_off)==1);
txbk_send_LPEXC_time=sum(excursion(txbk_send+1:txbk_off)~=0)/row_sec; %does not "sum", but "count" instead.
txbk_send_rms_steer=sqrt(sumsqr(steer_wheel(txbk_send+1:txbk_off))/(txbk_off-txbk_send));
txbk_send_avg_speed=mean(veh_speed(txbk_send+1:txbk_off))/100;
txbk_send_avg_FD=mean(FD(txbk_send+1:txbk_off));
txbk_send_TTC_min=min(TTC(txbk_send+1:txbk_off));
txbk_send_FD_min=min(FD(txbk_send+1:txbk_off));
if isempty(find(collision_vehicle(txbk_send+1:txbk_off), 1));
    txbk_send_collision=0; else txbk_send_collision=1;end
%--txbk_post, Lane_C
bk_pre2=find(lead_x(max(rta_txbk,txbk_off):t_bk)~=lane_C,1,'first')+max(rta_txbk,txbk_off)-1;
if bk_pre2-max(rta_txbk,txbk_off)>pre_sec*row_sec && lead_x(max(rta_txbk,txbk_off))==lane_C;
    bk_pre=bk_pre2-pre_sec*row_sec; bk_pre_ck=0;
    if bk_pre-max(rta_txbk,txbk_off)-1>post_sec*row_sec;
    txbk_post=max(rta_txbk,txbk_off)+post_sec*row_sec; txbk_post_ck=0; else txbk_post=bk_pre-1; txbk_post_ck=1; end

    if veh_x(max(rta_txbk,txbk_off)+1)>(lane_CR-car_width/2) || veh_x(max(rta_txbk,txbk_off)+1)<(lane_CL+car_width/2); excursion(max(rta_txbk,txbk_off)+1)=1; e
    for n=max(rta_txbk,txbk_off)+1+1:txbk_post;
        if (veh_x(n)>(lane_CR-car_width/2) && veh_x(n-1)<=(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2) && veh_x(n-1)>=(lane_CL+car_width/2));
            excursion(n)=1;
        else if (veh_x(n)>(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2)); excursion(n)=2; end
        end
    end

    txbk_post_LPSD=sqrt(sumsqr(veh_x(max(rta_txbk,txbk_off)+1:txbk_post)-mean(veh_x(max(rta_txbk,txbk_off)+1:txbk_post)))/(txbk_post-max(rta_txbk,txbk_off)-1))
    txbk_post_LPEXC=sum(excursion(max(rta_txbk,txbk_off)+1:txbk_post)==1);
    txbk_post_LPEXC_time=sum(excursion(max(rta_txbk,txbk_off)+1:txbk_post)~=0)/row_sec; %does not "sum", but "count" instead.
    txbk_post_rms_steer=sqrt(sumsqr(steer_wheel(max(rta_txbk,txbk_off)+1:txbk_post))/(txbk_post-max(rta_txbk,txbk_off)));
    txbk_post_avg_speed=mean(veh_speed(max(rta_txbk,txbk_off)+1:txbk_post))/100;
    txbk_post_avg_FD=mean(FD(max(rta_txbk,txbk_off)+1:txbk_post));
    txbk_post_TTC_min=min(TTC(max(rta_txbk,txbk_off)+1:txbk_post));
    txbk_post_FD_min=min(FD(max(rta_txbk,txbk_off)+1:txbk_post));
    txbk_post_wnd_length=system_time(txbk_post)/1000-system_time(max(rta_txbk,txbk_off)+1)/1000;
else bk_pre=max(rta_txbk,txbk_off)+1; bk_pre_ck=1; txbk_post=NaN; txbk_post_ck=1;
    txbk_post_rms_steer=NaN;
    txbk_post_LPSD=NaN;
    txbk_post_LPEXC=NaN;
    txbk_post_LPEXC_time=NaN;
    txbk_post_avg_speed=NaN;
    txbk_post_avg_FD=NaN;
    txbk_post_TTC_min=NaN;
    txbk_post_FD_min=NaN;
    txbk_post_wnd_length=NaN;
end

%bk
%--bk_pre, Lane_C
if veh_x(bk_pre)>(lane_CR-car_width/2) || veh_x(bk_pre)<(lane_CL+car_width/2); excursion(bk_pre)=1; end
for n=bk_pre+1:bk_pre2;
    if (veh_x(n)>(lane_CR-car_width/2) && veh_x(n-1)<=(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2) && veh_x(n-1)>=(lane_CL+car_width/2));
        excursion(n)=1;
    else if (veh_x(n)>(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2)); excursion(n)=2; end
    end
end

bk_pre_LPSD=sqrt(sumsqr(veh_x(bk_pre:bk_pre2)-mean(veh_x(bk_pre:bk_pre2)))/(bk_pre2-bk_pre));
bk_pre_LPEXC=sum(excursion(bk_pre:bk_pre2)==1);
bk_pre_LPEXC_time=sum(excursion(bk_pre:bk_pre2)~=0)/row_sec; %does not "sum", but "count" instead.
bk_pre_rms_steer=sqrt(sumsqr(steer_wheel(bk_pre:bk_pre2))/(bk_pre2-bk_pre+1));
bk_pre_avg_speed=mean(veh_speed(bk_pre:bk_pre2))/100;
bk_pre_avg_FD=mean(FD(bk_pre:bk_pre2));
```

```
bk_pre_TTC_min=min(TTC(bk_pre:bk_pre2));
bk_pre_FD_min=min(FD(bk_pre:bk_pre2));
bk_pre_wnd_length=system_time(bk_pre2)/1000-system_time(bk_pre)/1000;
%--bk_reaction
n = t_bk+2;
t_bk_r=n;
while n<double(Count)-2;
      n=n+1;
      t_bk_r=t_bk_r+1;
      if lead_speed(n-2)>=lead_speed(n-1) && lead_speed(n-1)>=lead_speed(n) && lead_speed(n) < lead_speed(n+1) && lead_speed(n+1) <= lead_speed(n+2)
        break
      end
end

n = t_bk;
bkon_bk=n;
while n<data_end;
      n=n+1;
      bkon_bk=bkon_bk+1;
      if brake_ped(n) ~= 0 & brake_ped((n-3):(n-1)) == 0;
        break
      end
end
n = t_bk;
gr_bk=n;
while n<data_end-3;
      n=n+1;
      gr_bk=gr_bk+1;
      if accel_ped((n-3):(n-1)) > gr_adj & accel_ped((n+1):(n+3)) <= gr_adj & accel_ped(n) <= gr_adj & veh_speed(n) > 0;
        break
      end
end
if gr_bk >= data_end-3 || accel_ped(t_bk)<= gr_adj; gr_bk = NaN; end
if bkon_bk >= data_end || brake_ped(t_bk)~=0; bkon_bk = NaN; end
if isnan(gr_bk) && isnan(bkon_bk); d_bk=NaN; else d_bk=min(gr_bk,bkon_bk); end

if isempty(find(resize(t_bk_r:Count), 1)); data_end=double(Count);end

m_bk=min(veh_speed(max(t_bk,d_bk):data_end));
rta_bk=find(veh_speed(max(t_bk,d_bk):data_end)==m_bk, 1, 'last' )+max(t_bk,d_bk)-1;
%min_speed_bk=find(veh_speed(max(t_bk,d_bk):data_end)==m_bk, 1, 'first' )+max(t_bk,d_bk)-1;
if isempty(find(bkoff(max(t_bk,d_bk):rta_bk),1)), bkoff_bk=NaN; else bkoff_bk=find(bkoff(max(t_bk,d_bk):rta_bk),1,'last')+max(t_bk,d_bk)-1; end
%--bk_during, Lane_R
if veh_x(t_bk)>(lane_RR-car_width/2) || veh_x(t_bk)<(lane_CR+car_width/2); excursion(t_bk)=1; end
for n=t_bk+1:rta_bk;
    if (veh_x(n)>(lane_RR-car_width/2) && veh_x(n-1)<=(lane_RR-car_width/2)) || (veh_x(n)<(lane_CR+car_width/2) && veh_x(n-1)>=(lane_CR+car_width/2));
        excursion(n)=1;
    else if (veh_x(n)>(lane_RR-car_width/2)) || (veh_x(n)<(lane_CR+car_width/2)); excursion(n)=2; end
    end
end

bk_LPSD=sqrt(sumsqr(veh_x(t_bk:rta_bk)-mean(veh_x(t_bk:rta_bk)))/(rta_bk-t_bk));
bk_LPEXC=sum(excursion(t_bk:rta_bk)==1);
bk_LPEXC_time=sum(excursion(t_bk:rta_bk)~=0)/row_sec; %does not "sum", but "count" instead.
bk_rms_steer=sqrt(sumsqr(steer_wheel(t_bk:rta_bk))/(rta_bk-t_bk+1));
bk_avg_speed=mean(veh_speed(t_bk:rta_bk))/100;
bk_avg_FD=mean(FD(t_bk:rta_bk));
bk_TTC_min=min(TTC(t_bk:rta_bk));
bk_FD_min=min(FD(t_bk:rta_bk));
if isempty(find(collision_vehicle(t_bk:rta_bk), 1));
    bk_collision=0; else bk_collision=1;end
%--bk_post, Lane_R
if data_end==double(Count); bk_post=NaN; bk_post_rms_steer=NaN; bk_post_LPSD=NaN; bk_post_LPEXC=NaN; bk_post_LPEXC_time=NaN; bk_post_avg_speed=NaN;...
        bk_post_avg_FD=NaN; bk_post_TTC_min=NaN; bk_post_FD_min=NaN; bk_post_wnd_length=NaN; bk_post_ck=1; else
if data_end-rta_bk>post_sec*row_sec;
    bk_post=rta_bk+post_sec*row_sec; bk_post_ck=0; else bk_post=data_end; bk_post_ck=1;
end

if veh_x(rta_bk+1)>(lane_RR-car_width/2) || veh_x(rta_bk+1)<(lane_CR+car_width/2); excursion(rta_bk+1)=1; end
for n=rta_bk+1+1:bk_post;
    if (veh_x(n)>(lane_RR-car_width/2) && veh_x(n-1)<=(lane_RR-car_width/2)) || (veh_x(n)<(lane_CR+car_width/2) && veh_x(n-1)>=(lane_CR+car_width/2));
        excursion(n)=1;
    else if (veh_x(n)>(lane_RR-car_width/2)) || (veh_x(n)<(lane_CR+car_width/2)); excursion(n)=2; end
    end
end

bk_post_LPSD=sqrt(sumsqr(veh_x(rta_bk+1:bk_post)-mean(veh_x(rta_bk+1:bk_post)))/(bk_post-rta_bk-1));
bk_post_LPEXC=sum(excursion(rta_bk+1:bk_post)==1);
bk_post_LPEXC_time=sum(excursion(rta_bk+1:bk_post)~=0)/row_sec; %does not "sum", but "count" instead.
bk_post_rms_steer=sqrt(sumsqr(steer_wheel(rta_bk+1:bk_post))/(bk_post-rta_bk));
bk_post_avg_speed=mean(veh_speed(rta_bk+1:bk_post))/100;
bk_post_avg_FD=mean(FD(rta_bk+1:bk_post));
bk_post_TTC_min=min(TTC(rta_bk+1:bk_post));
bk_post_FD_min=min(FD(rta_bk+1:bk_post));
bk_post_wnd_length=system_time(bk_post)/1000-system_time(rta_bk+1)/1000;
end
%--check
if isnan(tx_off); tx_msg_latency=NaN; ck_ped_tx=0;
else
n = tx_pre;
tx_y_row=n;
```

```
    while n<tx_initiate;
          n=n+1;
          tx_y_row=tx_y_row+1;
          if tx_y-lead_y(n)<trigger_adj;
            break
          end
    end
    tx_msg_latency=(system_time(tx_initiate)-system_time(tx_y_row))/1000;

    if isempty(find(ped_check(tx_pre:max(tx_post,tx_off)), 1)); ck_ped_tx=0; else ck_ped_tx=1; end
    end

    if isempty(find(ped_check(txbk_pre:max(txbk_post,max(rta_txbk,txbk_off))), 1)); ck_ped_txbk=0; else ck_ped_txbk=1; end
    if isempty(find(ped_check(bk_pre:max(bk_post,rta_bk)), 1)); ck_ped_bk=0; else ck_ped_bk=1; end
    if (ck_ped_tx+ck_ped_txbk+ck_ped_bk)>0; ck_ped=1; else ck_ped=0; end
    if rta_txbk<txbk_reply; ck_txbk_rta=0;
        else if rta_txbk<txbk_send; ck_txbk_rta=1;
            else if rta_txbk<=txbk_off; ck_txbk_rta=2; else ck_txbk_rta=3; end
            end
    end
    if (tx_pre_ck+txbk_pre_ck+bk_pre_ck)>0; ck_pre_time=1; else ck_pre_time=1; end
    if (tx_post_ck+txbk_post_ck+bk_post_ck)>0; ck_post_time=1; else ck_post_time=1; end
    n = txbk_pre;
    txbk_y_row=n;
    while n<txbk_initiate;
          n=n+1;
          txbk_y_row=txbk_y_row+1;
          if txbk_y-lead_y(n)<trigger_adj;
            break
          end
    end
    txbk_msg_latency=(system_time(txbk_initiate)-system_time(txbk_y_row))/1000;

    brake_event_delay=(system_time(t_txbk)-system_time(txbk_readthink))/1000;
```

Trigger 2

```
else t_bk=find(t,1,'first'); t_txbk=find(t,1,'last'); %bk->txbk->tx
    txbk_y=trigger_y2; tx_y=trigger_y3;

    %bk
    %--bk_pre, Lane_C

    bk_pre2=find(lead_x(find(lead_x(1:t_bk)==lane_C,1,'first'):t_bk)~=lane_C,1,'first')+find(lead_x(1:t_bk)==lane_C,1,'first')-1;
    if bk_pre2>pre_sec*row_sec;
        bk_pre=bk_pre2-pre_sec*row_sec; bk_pre_ck=0; else bk_pre=1; bk_pre_ck=1;
    end

    if veh_x(bk_pre)>(lane_CR-car_width/2) || veh_x(bk_pre)<(lane_CL+car_width/2); excursion(bk_pre)=1; end
    for n=bk_pre+1:bk_pre2;
        if (veh_x(n)>(lane_CR-car_width/2) && veh_x(n-1)<=(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2) && veh_x(n-1)>=(lane_CL+car_width/2));
            excursion(n)=1;
        else if (veh_x(n)>(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2)); excursion(n)=2; end
        end
    end

    bk_pre_LPSD=sqrt(sumsqr(veh_x(bk_pre:bk_pre2)-mean(veh_x(bk_pre:bk_pre2)))/(bk_pre2-bk_pre));
    bk_pre_LPEXC=sum(excursion(bk_pre:bk_pre2)==1);
    bk_pre_LPEXC_time=sum(excursion(bk_pre:bk_pre2)~=0)/row_sec; %does not "sum", but "count" instead.
    bk_pre_rms_steer=sqrt(sumsqr(steer_wheel(bk_pre:bk_pre2))/(bk_pre2-bk_pre+1));
    bk_pre_avg_speed=mean(veh_speed(bk_pre:bk_pre2))/100;
    bk_pre_avg_FD=mean(FD(bk_pre:bk_pre2));
    bk_pre_TTC_min=min(TTC(bk_pre:bk_pre2));
    bk_pre_FD_min=min(FD(bk_pre:bk_pre2));
    bk_pre_wnd_length=system_time(bk_pre2)/1000-system_time(bk_pre)/1000;
    %--bk_reaction
    n = t_bk;
    bkon_bk=n;
    while n<t_txbk-1;
          n=n+1;
          bkon_bk=bkon_bk+1;
          if brake_ped(n) ~= 0 & brake_ped((n-3):(n-1)) == 0;
            break
          end
    end
    n = t_bk;
    gr_bk=n;
    while n<t_txbk-3;
          n=n+1;
          gr_bk=gr_bk+1;
          if accel_ped((n-3):(n-1)) > gr_adj & accel_ped((n+1):(n+3)) <= gr_adj & accel_ped(n) <= gr_adj & veh_speed(n) > 0;
            break
          end
    end
    if gr_bk >= t_txbk-3 || accel_ped(t_bk)<= gr_adj; gr_bk = NaN; end
    if bkon_bk >= t_txbk-1 || brake_ped(t_bk)~=0; bkon_bk = NaN; end
    if isnan(gr_bk) && isnan(bkon_bk); d_bk=NaN; else d_bk=min(gr_bk,bkon_bk); end

    m_bk=min(veh_speed(max(t_bk,d_bk):t_txbk-1));
    rta_bk=find(veh_speed(max(t_bk,d_bk):t_txbk-1)==m_bk, 1, 'last' )+max(t_bk,d_bk)-1;
```

```
%min_speed_bk=find(veh_speed(max(t_bk,d_bk):t_txbk-1)==m_bk, 1, 'first' )+max(t_bk,d_bk)-1;
if isempty(find(bkoff(max(t_bk,d_bk):rta_bk),1)), bkoff_bk=NaN; else bkoff_bk=find(bkoff(max(t_bk,d_bk):rta_bk),1,'last')+max(t_bk,d_bk)-1; end
%--bk_during, Lane_L
if veh_x(t_bk)>(lane_CL-car_width/2) || veh_x(t_bk)<(lane_LL+car_width/2); excursion(t_bk)=1; end
for n=t_bk+1:rta_bk;
    if (veh_x(n)>(lane_CL-car_width/2) && veh_x(n-1)<=(lane_CL-car_width/2)) || (veh_x(n)<(lane_LL+car_width/2) && veh_x(n-1)>=(lane_LL+car_width/2));
        excursion(n)=1;
    else if (veh_x(n)>(lane_CL-car_width/2)) || (veh_x(n)<(lane_LL+car_width/2)); excursion(n)=2; end
    end
end

bk_LPSD=sqrt(sumsqr(veh_x(t_bk:rta_bk)-mean(veh_x(t_bk:rta_bk)))/(rta_bk-t_bk));
bk_LPEXC=sum(excursion(t_bk:rta_bk)==1);
bk_LPEXC_time=sum(excursion(t_bk:rta_bk)~=0)/row_sec; %does not "sum", but "count" instead.
bk_rms_steer=sqrt(sumsqr(steer_wheel(t_bk:rta_bk))/(rta_bk-t_bk+1));
bk_avg_speed=mean(veh_speed(t_bk:rta_bk))/100;
bk_avg_FD=mean(FD(t_bk:rta_bk));
bk_TTC_min=min(TTC(t_bk:rta_bk));
bk_FD_min=min(FD(t_bk:rta_bk));
if isempty(find(collision_vehicle(t_bk:rta_bk), 1));
    bk_collision=0; else bk_collision=1;end
%--bk_post, Lane_L
txbk_pre2=find(lead_x(rta_bk:txbk_initiate)~=lane_L,1,'first')+rta_bk-1;
if txbk_pre2-rta_bk>pre_sec*row_sec && lead_x(rta_bk)==lane_L;
    txbk_pre=txbk_pre2-pre_sec*row_sec; txbk_pre_ck=0;
    if txbk_pre-rta_bk-1>post_sec*row_sec;
    bk_post=rta_bk+post_sec*row_sec; bk_post_ck=0; else bk_post=txbk_pre-1; bk_post_ck=1; end

    if veh_x(rta_bk+1)>(lane_CL-car_width/2) || veh_x(rta_bk+1)<(lane_LL+car_width/2); excursion(rta_bk+1)=1; end
    for n=rta_bk+1+1:bk_post;
        if (veh_x(n)>(lane_CL-car_width/2) && veh_x(n-1)<=(lane_CL-car_width/2)) || (veh_x(n)<(lane_LL+car_width/2) && veh_x(n-1)>=(lane_LL+car_width/2));
        excursion(n)=1;
        else if (veh_x(n)>(lane_CL-car_width/2)) || (veh_x(n)<(lane_LL+car_width/2)); excursion(n)=2; end
        end
    end

    bk_post_LPSD=sqrt(sumsqr(veh_x(rta_bk+1:bk_post)-mean(veh_x(rta_bk+1:bk_post)))/(bk_post-rta_bk-1));
    bk_post_LPEXC=sum(excursion(rta_bk+1:bk_post)==1);
    bk_post_LPEXC_time=sum(excursion(rta_bk+1:bk_post)~=0)/row_sec; %does not "sum", but "count" instead.
    bk_post_rms_steer=sqrt(sumsqr(steer_wheel(rta_bk+1:bk_post))/(bk_post-rta_bk));
    bk_post_avg_speed=mean(veh_speed(rta_bk+1:bk_post))/100;
    bk_post_avg_FD=mean(FD(rta_bk+1:bk_post));
    bk_post_TTC_min=min(TTC(rta_bk+1:bk_post));
    bk_post_FD_min=min(FD(rta_bk+1:bk_post));
    bk_post_wnd_length=system_time(bk_post)/1000-system_time(rta_bk+1)/1000;
    else txbk_pre=rta_bk+1; txbk_pre_ck=1; bk_post=NaN; bk_post_ck=1;
    bk_post_rms_steer=NaN;
    bk_post_LPSD=NaN;
    bk_post_LPEXC=NaN;
    bk_post_LPEXC_time=NaN;
    bk_post_avg_speed=NaN;
    bk_post_avg_FD=NaN;
    bk_post_TTC_min=NaN;
    bk_post_FD_min=NaN;
    bk_post_wnd_length=NaN;
end

%txbk
%--txbk_pre, Lane_L
if veh_x(txbk_pre)>(lane_CL-car_width/2) || veh_x(txbk_pre)<(lane_LL+car_width/2); excursion(txbk_pre)=1; end
for n=txbk_pre+1:txbk_pre2;
    if (veh_x(n)>(lane_CL-car_width/2) && veh_x(n-1)<=(lane_CL-car_width/2)) || (veh_x(n)<(lane_LL+car_width/2) && veh_x(n-1)>=(lane_LL+car_width/2));
        excursion(n)=1;
    else if (veh_x(n)>(lane_CL-car_width/2)) || (veh_x(n)<(lane_LL+car_width/2)); excursion(n)=2; end
    end
end

txbk_pre_LPSD=sqrt(sumsqr(veh_x(txbk_pre:txbk_pre2)-mean(veh_x(txbk_pre:txbk_pre2)))/(txbk_pre2-txbk_pre));
txbk_pre_LPEXC=sum(excursion(txbk_pre:txbk_pre2)==1);
txbk_pre_LPEXC_time=sum(excursion(txbk_pre:txbk_pre2)~=0)/row_sec; %does not "sum", but "count" instead.
txbk_pre_rms_steer=sqrt(sumsqr(steer_wheel(txbk_pre:txbk_pre2))/(txbk_pre2-txbk_pre+1));
txbk_pre_avg_speed=mean(veh_speed(txbk_pre:txbk_pre2))/100;
txbk_pre_avg_FD=mean(FD(txbk_pre:txbk_pre2));
txbk_pre_TTC_min=min(TTC(txbk_pre:txbk_pre2));
txbk_pre_FD_min=min(FD(txbk_pre:txbk_pre2));
txbk_pre_wnd_length=system_time(txbk_pre2)/1000-system_time(txbk_pre)/1000;
%--txbk_reaction
n = t_txbk;
bkon_txbk=n;
while n<tx_initiate-1;
      n=n+1;
      bkon_txbk=bkon_txbk+1;
      if brake_ped(n) ~= 0 & brake_ped((n-3):(n-1)) == 0;
        break
      end
end
n = t_txbk;
gr_txbk=n;
while n<tx_initiate-3;
      n=n+1;
      gr_txbk=gr_txbk+1;
```

```
      if accel_ped((n-3):(n-1)) > gr_adj & accel_ped((n+1):(n+3)) <= gr_adj & accel_ped(n) <= gr_adj & veh_speed(n) > 0;
        break
      end
end
if gr_txbk >= tx_initiate-3 || accel_ped(t_txbk)<= gr_adj; gr_txbk = NaN; end
if bkon_txbk >= tx_initiate-1 || brake_ped(t_txbk)~=0; bkon_txbk = NaN; end
if isnan(gr_txbk) && isnan(bkon_txbk); d_txbk=NaN; else d_txbk=min(gr_txbk,bkon_txbk); end
m_txbk=min(veh_speed(max(t_txbk,d_txbk):tx_initiate-1));
rta_txbk=find(veh_speed(max(t_txbk,d_txbk):tx_initiate-1)==m_txbk, 1, 'last' )+max(t_txbk,d_txbk)-1;
%min_speed_txbk=find(veh_speed(max(t_txbk,d_txbk):tx_initiate-1)==m_txbk, 1, 'first' )+max(t_txbk,d_txbk)-1;
if isempty(find(bkoff(max(t_txbk,d_txbk):rta_txbk),1)), bkoff_txbk=NaN; else bkoff_txbk=find(bkoff(max(t_txbk,d_txbk):rta_txbk),1,'last')+max(t_txbk,d_txbk)-1;
%--txbk_during, Lane_C
if veh_x(t_txbk)>(lane_CR-car_width/2) || veh_x(t_txbk)<(lane_CL+car_width/2); excursion(t_txbk)=1; end
for n=t_txbk+1:rta_txbk;
    if (veh_x(n)>(lane_CR-car_width/2) && veh_x(n-1)<=(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2) && veh_x(n-1)>=(lane_CL+car_width/2));
        excursion(n)=1;
    else if (veh_x(n)>(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2)); excursion(n)=2; end
    end
end

txbk_LPSD=sqrt(sumsqr(veh_x(t_txbk:rta_txbk)-mean(veh_x(t_txbk:rta_txbk)))/(rta_txbk-t_txbk));
txbk_LPEXC=sum(excursion(t_txbk:rta_txbk)==1);
txbk_LPEXC_time=sum(excursion(t_txbk:rta_txbk)~=0)/row_sec; %does not "sum", but "count" instead.
txbk_rms_steer=sqrt(sumsqr(steer_wheel(t_txbk:rta_txbk))/(rta_txbk-t_txbk+1));
txbk_avg_speed=mean(veh_speed(t_txbk:rta_txbk))/100;
txbk_avg_FD=mean(FD(t_txbk:rta_txbk));
txbk_TTC_min=min(TTC(t_txbk:rta_txbk));
txbk_FD_min=min(FD(t_txbk:rta_txbk));
if isempty(find(collision_vehicle(t_txbk:rta_txbk), 1));
    txbk_collision=0; else txbk_collision=1;end
%--txbk_initiate
txbk_initiate_LPSD=sqrt(sumsqr(veh_x(txbk_initiate:txbk_readthink-1)-mean(veh_x(txbk_initiate:txbk_readthink-1)))/(txbk_readthink-txbk_initiate-1));
txbk_initiate_LPEXC=sum(excursion(txbk_initiate:txbk_readthink-1)==1);
txbk_initiate_LPEXC_time=sum(excursion(txbk_initiate:txbk_readthink-1)~=0)/row_sec; %does not "sum", but "count" instead.
txbk_initiate_rms_steer=sqrt(sumsqr(steer_wheel(txbk_initiate:txbk_readthink-1))/(txbk_readthink-txbk_initiate));
txbk_initiate_avg_speed=mean(veh_speed(txbk_initiate:txbk_readthink-1))/100;
txbk_initiate_avg_FD=mean(FD(txbk_initiate:txbk_readthink-1));
txbk_initiate_TTC_min=min(TTC(txbk_initiate:txbk_readthink-1));
txbk_initiate_FD_min=min(FD(txbk_initiate:txbk_readthink-1));
if isempty(find(collision_vehicle(txbk_initiate:txbk_readthink-1), 1));
    txbk_initiate_collision=0; else txbk_initiate_collision=1;end
%--txbk_readthink
if veh_x(txbk_readthink)>(lane_CR-car_width/2) || veh_x(txbk_readthink)<(lane_CL+car_width/2); excursion(txbk_readthink)=1; end
txbk_readthink_LPSD=sqrt(sumsqr(veh_x(txbk_readthink:txbk_reply-1)-mean(veh_x(txbk_readthink:txbk_reply-1)))/(txbk_reply-txbk_readthink-1));
txbk_readthink_LPEXC=sum(excursion(txbk_readthink:txbk_reply-1)==1);
txbk_readthink_LPEXC_time=sum(excursion(txbk_readthink:txbk_reply-1)~=0)/row_sec; %does not "sum", but "count" instead.
txbk_readthink_rms_steer=sqrt(sumsqr(steer_wheel(txbk_readthink:txbk_reply-1))/(txbk_reply-txbk_readthink));
txbk_readthink_avg_speed=mean(veh_speed(txbk_readthink:txbk_reply-1))/100;
txbk_readthink_avg_FD=mean(FD(txbk_readthink:txbk_reply-1));
txbk_readthink_TTC_min=min(TTC(txbk_readthink:txbk_reply-1));
txbk_readthink_FD_min=min(FD(txbk_readthink:txbk_reply-1));
if isempty(find(collision_vehicle(txbk_readthink:txbk_reply-1), 1));
    txbk_readthink_collision=0; else txbk_readthink_collision=1;end
%--txbk_reply
if veh_x(txbk_reply)>(lane_CR-car_width/2) || veh_x(txbk_reply)<(lane_CL+car_width/2); excursion(txbk_reply)=1; end
txbk_reply_LPSD=sqrt(sumsqr(veh_x(txbk_reply:txbk_send)-mean(veh_x(txbk_reply:txbk_send)))/(txbk_send-txbk_reply));
txbk_reply_LPEXC=sum(excursion(txbk_reply:txbk_send)==1);
txbk_reply_LPEXC_time=sum(excursion(txbk_reply:txbk_send)~=0)/row_sec; %does not "sum", but "count" instead.
txbk_reply_rms_steer=sqrt(sumsqr(steer_wheel(txbk_reply:txbk_send))/(txbk_send-txbk_reply+1));
txbk_reply_avg_speed=mean(veh_speed(txbk_reply:txbk_send))/100;
txbk_reply_avg_FD=mean(FD(txbk_reply:txbk_send));
txbk_reply_TTC_min=min(TTC(txbk_reply:txbk_send));
txbk_reply_FD_min=min(FD(txbk_reply:txbk_send));
if isempty(find(collision_vehicle(txbk_reply:txbk_send), 1));
    txbk_reply_collision=0; else txbk_reply_collision=1;end
%--txbk_send
if veh_x(txbk_send+1)>(lane_CR-car_width/2) || veh_x(txbk_send+1)<(lane_CL+car_width/2); excursion(txbk_send+1)=1; end
txbk_send_LPSD=sqrt(sumsqr(veh_x(txbk_send+1:txbk_off)-mean(veh_x(txbk_send+1:txbk_off)))/(txbk_off-txbk_send-1));
txbk_send_LPEXC=sum(excursion(txbk_send+1:txbk_off)==1);
txbk_send_LPEXC_time=sum(excursion(txbk_send+1:txbk_off)~=0)/row_sec; %does not "sum", but "count" instead.
txbk_send_rms_steer=sqrt(sumsqr(steer_wheel(txbk_send+1:txbk_off))/(txbk_off-txbk_send));
txbk_send_avg_speed=mean(veh_speed(txbk_send+1:txbk_off))/100;
txbk_send_avg_FD=mean(FD(txbk_send+1:txbk_off));
txbk_send_TTC_min=min(TTC(txbk_send+1:txbk_off));
txbk_send_FD_min=min(FD(txbk_send+1:txbk_off));
if isempty(find(collision_vehicle(txbk_send+1:txbk_off), 1));
    txbk_send_collision=0; else txbk_send_collision=1;end
%--txbk_post, Lane_C
if isnan(tx_off);
    tx_pre=NaN; tx_pre2=NaN;
    txbk_post=max(rta_txbk,txbk_off)+post_sec*row_sec; txbk_post_ck=0;

    if veh_x(max(rta_txbk,txbk_off)+1)>(lane_CR-car_width/2) || veh_x(max(rta_txbk,txbk_off)+1)<(lane_CL+car_width/2); excursion(max(rta_txbk,txbk_off)+1)=1; e
    for n=max(rta_txbk,txbk_off)+1+1:txbk_post;
        if (veh_x(n)>(lane_CR-car_width/2) && veh_x(n-1)<=(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2) && veh_x(n-1)>=(lane_CL+car_width/2));
            excursion(n)=1;
        else if (veh_x(n)>(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2)); excursion(n)=2; end
        end
    end

    txbk_post_LPSD=sqrt(sumsqr(veh_x(max(rta_txbk,txbk_off)+1:txbk_post)-mean(veh_x(max(rta_txbk,txbk_off)+1:txbk_post)))/(txbk_post-max(rta_txbk,txbk_off)-1))
```

```
    txbk_post_LPEXC=sum(excursion(max(rta_txbk,txbk_off)+1:txbk_post)==1);
    txbk_post_LPEXC_time=sum(excursion(max(rta_txbk,txbk_off)+1:txbk_post)~=0)/row_sec; %does not "sum", but "count" instead.
    txbk_post_rms_steer=sqrt(sumsqr(steer_wheel(max(rta_txbk,txbk_off)+1:txbk_post))/(txbk_post-max(rta_txbk,txbk_off)));
    txbk_post_avg_speed=mean(veh_speed(max(rta_txbk,txbk_off)+1:txbk_post))/100;
    txbk_post_avg_FD=mean(FD(max(rta_txbk,txbk_off)+1:txbk_post));
    txbk_post_TTC_min=min(TTC(max(rta_txbk,txbk_off)+1:txbk_post));
    txbk_post_FD_min=min(FD(max(rta_txbk,txbk_off)+1:txbk_post));
    txbk_post_wnd_length=system_time(txbk_post)/1000-system_time(max(rta_txbk,txbk_off)+1)/1000;
else
tx_pre2=find(lead_x(max(rta_txbk,txbk_off):tx_initiate)~=lane_C,1,'first')+max(rta_txbk,txbk_off)-1;
if tx_pre2-max(rta_txbk,txbk_off)>pre_sec*row_sec && lead_x(max(rta_txbk,txbk_off))==lane_C;
    tx_pre=tx_pre2-pre_sec*row_sec; tx_pre_ck=0;
    if tx_pre-max(rta_txbk,txbk_off)-1>post_sec*row_sec;
    txbk_post=max(rta_txbk,txbk_off)+post_sec*row_sec; txbk_post_ck=0; else txbk_post=tx_pre-1; txbk_post_ck=1; end

    if veh_x(max(rta_txbk,txbk_off)+1)>(lane_CR-car_width/2) || veh_x(max(rta_txbk,txbk_off)+1)<(lane_CL+car_width/2); excursion(max(rta_txbk,txbk_off)+1)=1; e
    for n=max(rta_txbk,txbk_off)+1+1:txbk_post;
        if (veh_x(n)>(lane_CR-car_width/2) && veh_x(n-1)<=(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2) && veh_x(n-1)>=(lane_CL+car_width/2));
            excursion(n)=1;
        else if (veh_x(n)>(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2)); excursion(n)=2; end
        end
    end

    txbk_post_LPSD=sqrt(sumsqr(veh_x(max(rta_txbk,txbk_off)+1:txbk_post)-mean(veh_x(max(rta_txbk,txbk_off)+1:txbk_post)))/(txbk_post-max(rta_txbk,txbk_off)-1))
    txbk_post_LPEXC=sum(excursion(max(rta_txbk,txbk_off)+1:txbk_post)==1);
    txbk_post_LPEXC_time=sum(excursion(max(rta_txbk,txbk_off)+1:txbk_post)~=0)/row_sec; %does not "sum", but "count" instead.
    txbk_post_rms_steer=sqrt(sumsqr(steer_wheel(max(rta_txbk,txbk_off)+1:txbk_post))/(txbk_post-max(rta_txbk,txbk_off)));
    txbk_post_avg_speed=mean(veh_speed(max(rta_txbk,txbk_off)+1:txbk_post))/100;
    txbk_post_avg_FD=mean(FD(max(rta_txbk,txbk_off)+1:txbk_post));
    txbk_post_TTC_min=min(TTC(max(rta_txbk,txbk_off)+1:txbk_post));
    txbk_post_FD_min=min(FD(max(rta_txbk,txbk_off)+1:txbk_post));
    txbk_post_wnd_length=system_time(txbk_post)/1000-system_time(max(rta_txbk,txbk_off)+1)/1000;
    else tx_pre=max(rta_txbk,txbk_off)+1; tx_pre_ck=1; txbk_post=NaN; txbk_post_ck=1;
    txbk_post_rms_steer=NaN;
    txbk_post_LPSD=NaN;
    txbk_post_LPEXC=NaN;
    txbk_post_LPEXC_time=NaN;
    txbk_post_avg_speed=NaN;
    txbk_post_avg_FD=NaN;
    txbk_post_TTC_min=NaN;
    txbk_post_FD_min=NaN;
    txbk_post_wnd_length=NaN;
end
end

%tx
if isnan(tx_off);
    tx_pre_ck=1; tx_pre_rms_steer=NaN; tx_pre_LPSD=NaN; tx_pre_LPEXC=NaN; tx_pre_LPEXC_time=NaN; tx_pre_avg_speed=NaN; tx_pre_avg_FD=NaN; tx_pre_TTC_min=NaN; t
    tx_rms_steer=NaN; tx_LPSD=NaN; tx_LPEXC=NaN; tx_LPEXC_time=NaN; tx_avg_speed=NaN; tx_avg_FD=NaN; tx_TTC_min=NaN; tx_FD_min=NaN; tx_collision=0;
    tx_initiate_rms_steer=NaN; tx_initiate_LPSD=NaN; tx_initiate_LPEXC=NaN; tx_initiate_LPEXC_time=NaN; tx_initiate_avg_speed=NaN; tx_initiate_avg_FD=NaN; tx_i
    tx_readthink_rms_steer=NaN; tx_readthink_LPSD=NaN; tx_readthink_LPEXC=NaN; tx_readthink_LPEXC_time=NaN; tx_readthink_avg_speed=NaN; tx_readthink_avg_FD=NaN
    tx_reply_rms_steer=NaN; tx_reply_LPSD=NaN; tx_reply_LPEXC=NaN; tx_reply_LPEXC_time=NaN; tx_reply_avg_speed=NaN; tx_reply_avg_FD=NaN; tx_reply_TTC_min=NaN;
    tx_send_rms_steer=NaN; tx_send_LPSD=NaN; tx_send_LPEXC=NaN; tx_send_LPEXC_time=NaN; tx_send_avg_speed=NaN; tx_send_avg_FD=NaN; tx_send_TTC_min=NaN; tx_send_
    tx_post_ck=1; tx_post_rms_steer=NaN; tx_post_LPSD=NaN; tx_post_LPEXC=NaN; tx_post_LPEXC_time=NaN; tx_post_avg_speed=NaN; tx_post_avg_FD=NaN; tx_post_TTC_mi
else
%--tx_pre, Lane_C
if veh_x(tx_pre)>(lane_CR-car_width/2) || veh_x(tx_pre)<(lane_CL+car_width/2); excursion(tx_pre)=1; end
for n=tx_pre+1:tx_pre2;
    if (veh_x(n)>(lane_CR-car_width/2) && veh_x(n-1)<=(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2) && veh_x(n-1)>=(lane_CL+car_width/2));
        excursion(n)=1;
    else if (veh_x(n)>(lane_CR-car_width/2)) || (veh_x(n)<(lane_CL+car_width/2)); excursion(n)=2; end
    end
end

tx_pre_LPSD=sqrt(sumsqr(veh_x(tx_pre:tx_pre2)-mean(veh_x(tx_pre:tx_pre2)))/(tx_pre2-tx_pre));
tx_pre_LPEXC=sum(excursion(tx_pre:tx_pre2)==1);
tx_pre_LPEXC_time=sum(excursion(tx_pre:tx_pre2)~=0)/row_sec; %does not "sum", but "count" instead.
tx_pre_rms_steer=sqrt(sumsqr(steer_wheel(tx_pre:tx_pre2))/(tx_pre2-tx_pre+1));
tx_pre_avg_speed=mean(veh_speed(tx_pre:tx_pre2))/100;
tx_pre_avg_FD=mean(FD(tx_pre:tx_pre2));
tx_pre_TTC_min=min(TTC(tx_pre:tx_pre2));
tx_pre_FD_min=min(FD(tx_pre:tx_pre2));
tx_pre_wnd_length=system_time(tx_pre2)/1000-system_time(tx_pre)/1000;
%--tx_during, Lane_R
if veh_x(tx_initiate)>(lane_RR-car_width/2) || veh_x(tx_initiate)<(lane_CR+car_width/2); excursion(tx_initiate)=1; end
for n=tx_initiate+1:tx_off;
    if (veh_x(n)>(lane_RR-car_width/2) && veh_x(n-1)<=(lane_RR-car_width/2)) || (veh_x(n)<(lane_CR+car_width/2) && veh_x(n-1)>=(lane_CR+car_width/2));
        excursion(n)=1;
    else if (veh_x(n)>(lane_RR-car_width/2)) || (veh_x(n)<(lane_CR+car_width/2)); excursion(n)=2; end
    end
end

tx_LPSD=sqrt(sumsqr(veh_x(tx_initiate:tx_off)-mean(veh_x(tx_initiate:tx_off)))/(tx_off-tx_initiate));
tx_LPEXC=sum(excursion(tx_initiate:tx_off)==1);
tx_LPEXC_time=sum(excursion(tx_initiate:tx_off)~=0)/row_sec; %does not "sum", but "count" instead.
tx_rms_steer=sqrt(sumsqr(steer_wheel(tx_initiate:tx_off))/(tx_off-tx_initiate+1));
tx_avg_speed=mean(veh_speed(tx_initiate:tx_off))/100;
tx_avg_FD=mean(FD(tx_initiate:tx_off));
tx_TTC_min=min(TTC(tx_initiate:tx_off));
tx_FD_min=min(FD(tx_initiate:tx_off));
if isempty(find(collision_vehicle(tx_initiate:tx_off), 1));
```

```
    tx_collision=0; else tx_collision=1;end
%--tx_initiate
tx_initiate_LPSD=sqrt(sumsqr(veh_x(tx_initiate:tx_readthink-1)-mean(veh_x(tx_initiate:tx_readthink-1)))/(tx_readthink-tx_initiate-1));
tx_initiate_LPEXC=sum(excursion(tx_initiate:tx_readthink-1)==1);
tx_initiate_LPEXC_time=sum(excursion(tx_initiate:tx_readthink-1)~=0)/row_sec; %does not "sum", but "count" instead.
tx_initiate_rms_steer=sqrt(sumsqr(steer_wheel(tx_initiate:tx_readthink-1))/(tx_readthink-tx_initiate));
tx_initiate_avg_speed=mean(veh_speed(tx_initiate:tx_readthink-1))/100;
tx_initiate_avg_FD=mean(FD(tx_initiate:tx_readthink-1));
tx_initiate_TTC_min=min(TTC(tx_initiate:tx_readthink-1));
tx_initiate_FD_min=min(FD(tx_initiate:tx_readthink-1));
if isempty(find(collision_vehicle(tx_initiate:tx_readthink-1), 1));
    tx_initiate_collision=0; else tx_initiate_collision=1;end
%--tx_readthink
if veh_x(tx_readthink)>(lane_RR-car_width/2) || veh_x(tx_readthink)<(lane_CR+car_width/2); excursion(tx_readthink)=1; end
tx_readthink_LPSD=sqrt(sumsqr(veh_x(tx_readthink:tx_reply-1)-mean(veh_x(tx_readthink:tx_reply-1)))/(tx_reply-tx_readthink-1));
tx_readthink_LPEXC=sum(excursion(tx_readthink:tx_reply-1)==1);
tx_readthink_LPEXC_time=sum(excursion(tx_readthink:tx_reply-1)~=0)/row_sec; %does not "sum", but "count" instead.
tx_readthink_rms_steer=sqrt(sumsqr(steer_wheel(tx_readthink:tx_reply-1))/(tx_reply-tx_readthink));
tx_readthink_avg_speed=mean(veh_speed(tx_readthink:tx_reply-1))/100;
tx_readthink_avg_FD=mean(FD(tx_readthink:tx_reply-1));
tx_readthink_TTC_min=min(TTC(tx_readthink:tx_reply-1));
tx_readthink_FD_min=min(FD(tx_readthink:tx_reply-1));
if isempty(find(collision_vehicle(tx_readthink:tx_reply-1), 1));
    tx_readthink_collision=0; else tx_readthink_collision=1;end
%--tx_reply
if veh_x(tx_reply)>(lane_RR-car_width/2) || veh_x(tx_reply)<(lane_CR+car_width/2); excursion(tx_reply)=1; end
tx_reply_LPSD=sqrt(sumsqr(veh_x(tx_reply:tx_send)-mean(veh_x(tx_reply:tx_send))))/(tx_send-tx_reply));
tx_reply_LPEXC=sum(excursion(tx_reply:tx_send)==1);
tx_reply_LPEXC_time=sum(excursion(tx_reply:tx_send)~=0)/row_sec; %does not "sum", but "count" instead.
tx_reply_rms_steer=sqrt(sumsqr(steer_wheel(tx_reply:tx_send))/(tx_send-tx_reply+1));
tx_reply_avg_speed=mean(veh_speed(tx_reply:tx_send))/100;
tx_reply_avg_FD=mean(FD(tx_reply:tx_send));
tx_reply_TTC_min=min(TTC(tx_reply:tx_send));
tx_reply_FD_min=min(FD(tx_reply:tx_send));
if isempty(find(collision_vehicle(tx_reply:tx_send), 1));
    tx_reply_collision=0; else tx_reply_collision=1;end
%--tx_send
if veh_x(tx_send+1)>(lane_RR-car_width/2) || veh_x(tx_send+1)<(lane_CR+car_width/2); excursion(tx_send+1)=1; end
tx_send_LPSD=sqrt(sumsqr(veh_x(tx_send+1:tx_off)-mean(veh_x(tx_send+1:tx_off)))/(tx_off-tx_send-1));
tx_send_LPEXC=sum(excursion(tx_send+1:tx_off)==1);
tx_send_LPEXC_time=sum(excursion(tx_send+1:tx_off)~=0)/row_sec; %does not "sum", but "count" instead.
tx_send_rms_steer=sqrt(sumsqr(steer_wheel(tx_send+1:tx_off))/(tx_off-tx_send));
tx_send_avg_speed=mean(veh_speed(tx_send+1:tx_off))/100;
tx_send_avg_FD=mean(FD(tx_send+1:tx_off));
tx_send_TTC_min=min(TTC(tx_send+1:tx_off));
tx_send_FD_min=min(FD(tx_send+1:tx_off));
if isempty(find(collision_vehicle(tx_send+1:tx_off), 1));
    tx_send_collision=0; else tx_send_collision=1;end
%--tx_post, Lane_R
if tx_off>=data_end; data_end=double(Count); end
if data_end-tx_off>post_sec*row_sec;
    tx_post=tx_off+post_sec*row_sec; tx_post_ck=0; else tx_post=data_end; tx_post_ck=1;
end

if veh_x(tx_off+1)>(lane_RR-car_width/2) || veh_x(tx_off+1)<(lane_CR+car_width/2); excursion(tx_off+1)=1; end
for n=tx_off+1+1:tx_post;
  if (veh_x(n)>(lane_RR-car_width/2) && veh_x(n-1)<=(lane_RR-car_width/2)) || (veh_x(n)<(lane_CR+car_width/2) && veh_x(n-1)>=(lane_CR+car_width/2));
        excursion(n)=1;
    else if (veh_x(n)>(lane_RR-car_width/2)) || (veh_x(n)<(lane_CR+car_width/2)); excursion(n)=2; end
  end
end

tx_post_LPSD=sqrt(sumsqr(veh_x(tx_off+1:tx_post)-mean(veh_x(tx_off+1:tx_post)))/(tx_post-tx_off-1));
tx_post_LPEXC=sum(excursion(tx_off+1:tx_post)==1);
tx_post_LPEXC_time=sum(excursion(tx_off+1:tx_post)~=0)/row_sec; %does not "sum", but "count" instead.
tx_post_rms_steer=sqrt(sumsqr(steer_wheel(tx_off+1:tx_post))/(tx_post-tx_off));
tx_post_avg_speed=mean(veh_speed(tx_off+1:tx_post))/100;
tx_post_avg_FD=mean(FD(tx_off+1:tx_post));
tx_post_TTC_min=min(TTC(tx_off+1:tx_post));
tx_post_FD_min=min(FD(tx_off+1:tx_post));
tx_post_wnd_length=system_time(tx_post)/1000-system_time(tx_off+1)/1000;
end
%--check
if isnan(tx_off); tx_msg_latency=NaN; ck_ped_tx=0;
else
n = tx_pre;
tx_y_row=n;
while n<tx_initiate;
      n=n+1;
      tx_y_row=tx_y_row+1;
      if tx_y-lead_y(n)<trigger_adj;
        break
      end
end
tx_msg_latency=(system_time(tx_initiate)-system_time(tx_y_row))/1000;

if isempty(find(ped_check(tx_pre:max(tx_post,tx_off)), 1)); ck_ped_tx=0; else ck_ped_tx=1; end
end

if isempty(find(ped_check(txbk_pre:max(txbk_post,max(rta_txbk,txbk_off))), 1)); ck_ped_txbk=0; else ck_ped_txbk=1; end
if isempty(find(ped_check(bk_pre:max(bk_post,rta_bk)), 1)); ck_ped_bk=0; else ck_ped_bk=1; end
```

```
    if (ck_ped_tx+ck_ped_txbk+ck_ped_bk)>0; ck_ped=1; else ck_ped=0; end
    if rta_txbk<txbk_reply; ck_txbk_rta=0;
        else if rta_txbk<txbk_send; ck_txbk_rta=1;
            else if rta_txbk<=txbk_off; ck_txbk_rta=2; else ck_txbk_rta=3; end
            end
    end
    if (tx_pre_ck+txbk_pre_ck+bk_pre_ck)>0; ck_pre_time=1; else ck_pre_time=1; end
    if (tx_post_ck+txbk_post_ck+bk_post_ck)>0; ck_post_time=1; else ck_post_time=1; end
    n = txbk_pre;
    txbk_y_row=n;
    while n<txbk_initiate;
          n=n+1;
          txbk_y_row=txbk_y_row+1;
          if txbk_y-lead_y(n)<trigger_adj;
            break
          end
    end
    txbk_msg_latency=(system_time(txbk_initiate)-system_time(txbk_y_row))/1000;

    brake_event_delay=(system_time(t_txbk)-system_time(txbk_readthink))/1000;
end

if isnan(bkon_txbk); txbk_bk_onset=NaN; txbk_bk_offset=NaN; txbk_rta=NaN;
else txbk_bk_onset=system_time(bkon_txbk)/1000-system_time(t_txbk)/1000;
    if isnan(bkoff_txbk); txbk_bk_offset=NaN; else txbk_bk_offset=system_time(bkoff_txbk)/1000-system_time(bkon_txbk)/1000; end
    txbk_rta=system_time(rta_txbk)/1000-system_time(bkon_txbk)/1000;
end
if isnan(d_txbk); txbk_hy_res=NaN; else txbk_hy_res=system_time(d_txbk)/1000-system_time(t_txbk)/1000; end

if isnan(bkon_bk); bk_bk_onset=NaN; bk_bk_offset=NaN; bk_rta=NaN;
else bk_bk_onset=system_time(bkon_bk)/1000-system_time(t_bk)/1000;
    if isnan(bkoff_bk); bk_bk_offset=NaN; else bk_bk_offset=system_time(bkoff_bk)/1000-system_time(bkon_bk)/1000; end
    if rta_bk==data_end; bk_rta=NaN; else bk_rta=system_time(rta_bk)/1000-system_time(bkon_bk)/1000; end
end
if isnan(d_bk); bk_hy_res=NaN; else bk_hy_res=system_time(d_bk)/1000-system_time(t_bk)/1000; end

if (Collision==0 && (tx_collision+bk_collision+txbk_collision)==0) || (Collision==1 && (tx_collision+bk_collision+txbk_collision)>0);
    c_check=0; else c_check=1;
end

if ~isnan(txbk_pre_LPEXC) && txbk_pre_LPEXC==0 && txbk_pre_LPEXC_time~=0; LPEXC_check=1; else LPEXC_check=0; end
if ~isnan(txbk_LPEXC) && txbk_LPEXC==0 && txbk_LPEXC_time~=0; LPEXC_check=LPEXC_check+1; end
if ~isnan(txbk_initiate_LPEXC) && txbk_initiate_LPEXC==0 && txbk_initiate_LPEXC_time~=0; LPEXC_check=LPEXC_check+1; end
if ~isnan(txbk_readthink_LPEXC) && txbk_readthink_LPEXC==0 && txbk_readthink_LPEXC_time~=0; LPEXC_check=LPEXC_check+1; end
if ~isnan(txbk_reply_LPEXC) && txbk_reply_LPEXC==0 && txbk_reply_LPEXC_time~=0; LPEXC_check=LPEXC_check+1; end
if ~isnan(txbk_send_LPEXC) && txbk_send_LPEXC==0 && txbk_send_LPEXC_time~=0; LPEXC_check=LPEXC_check+1; end
if ~isnan(txbk_post_LPEXC) && txbk_post_LPEXC==0 && txbk_post_LPEXC_time~=0; LPEXC_check=LPEXC_check+1; end

if ~isnan(tx_pre_LPEXC) && tx_pre_LPEXC==0 && tx_pre_LPEXC_time~=0; LPEXC_check=LPEXC_check+1; end
if ~isnan(tx_LPEXC) && tx_LPEXC==0 && tx_LPEXC_time~=0; LPEXC_check=LPEXC_check+1; end
if ~isnan(tx_initiate_LPEXC) && tx_initiate_LPEXC==0 && tx_initiate_LPEXC_time~=0; LPEXC_check=LPEXC_check+1; end
if ~isnan(tx_readthink_LPEXC) && tx_readthink_LPEXC==0 && tx_readthink_LPEXC_time~=0; LPEXC_check=LPEXC_check+1; end
if ~isnan(tx_reply_LPEXC) && tx_reply_LPEXC==0 && tx_reply_LPEXC_time~=0; LPEXC_check=LPEXC_check+1; end
if ~isnan(tx_send_LPEXC) && tx_send_LPEXC==0 && tx_send_LPEXC_time~=0; LPEXC_check=LPEXC_check+1; end
if ~isnan(tx_post_LPEXC) && tx_post_LPEXC==0 && tx_post_LPEXC_time~=0; LPEXC_check=LPEXC_check+1; end

if ~isnan(bk_pre_LPEXC) && bk_pre_LPEXC==0 && bk_pre_LPEXC_time~=0; LPEXC_check=LPEXC_check+1; end
if ~isnan(bk_LPEXC) && bk_LPEXC==0 && bk_LPEXC_time~=0; LPEXC_check=LPEXC_check+1; end
if ~isnan(bk_post_LPEXC) && bk_post_LPEXC==0 && bk_post_LPEXC_time~=0; LPEXC_check=LPEXC_check+1; end
```

Create Database and export

```
Output_1=num2cell([Ss,Device,trigger,Collision,NaN,ck_ped,ck_tx_rta,ck_pre_time,ck_post_time,txbk_msg_latency,tx_msg_latency,brake_event_delay,...
    txbk_pre_rms_steer,txbk_pre_LPSD,txbk_pre_LPEXC,txbk_pre_LPEXC_time,txbk_pre_avg_speed,txbk_pre_avg_FD,txbk_pre_TTC_min,txbk_pre_FD_min,txbk_pre_wnd_length,...
    txbk_rms_steer,txbk_LPSD,txbk_LPEXC,txbk_LPEXC_time,txbk_avg_speed,txbk_avg_FD,txbk_TTC_min,txbk_FD_min,txbk_collision,txbk_bk_onset,txbk_bk_offset,txbk_hy_res,
    NaN,NaN,NaN,NaN,NaN,NaN,NaN,txbk_post_rms_steer,txbk_post_LPSD, txbk_post_LPEXC, txbk_post_LPEXC_time,txbk_post_avg_speed,txbk_post_avg_FD,txbk_post_TTC_min,txb
    tx_pre_rms_steer,tx_pre_LPSD,tx_pre_LPEXC, tx_pre_LPEXC_time,tx_pre_avg_speed,tx_pre_avg_FD,tx_pre_TTC_min,tx_pre_FD_min,tx_pre_wnd_length,...
    tx_rms_steer,tx_LPSD,tx_LPEXC,tx_LPEXC_time,tx_avg_speed,tx_avg_FD,tx_TTC_min,tx_FD_min,tx_collision,NaN,NaN,NaN,NaN,NaN,NaN,...
    tx_post_rms_steer,tx_post_LPSD,tx_post_LPEXC,tx_post_LPEXC_time,tx_post_avg_speed,tx_post_avg_FD,tx_post_TTC_min,tx_post_FD_min,tx_post_wnd_length,...
    bk_pre_rms_steer,bk_pre_LPSD,bk_pre_LPEXC,bk_pre_LPEXC_time,bk_pre_avg_speed,bk_pre_avg_FD,bk_pre_TTC_min,bk_pre_FD_min,bk_pre_wnd_length,...
    bk_rms_steer,bk_LPSD,bk_LPEXC,bk_LPEXC_time,bk_avg_speed,bk_avg_FD,bk_TTC_min,bk_FD_min,bk_collision,bk_bk_onset,bk_bk_offset,bk_hy_res,bk_rta,...
    bk_post_rms_steer,bk_post_LPSD,bk_post_LPEXC,bk_post_LPEXC_time,bk_post_avg_speed,bk_post_avg_FD,bk_post_TTC_min,bk_post_FD_min,bk_post_wnd_length,f+numel(fNam
dlmwrite(strcat(pwd,filesep,Output_flatfile_1),Output_1,'-append');

Output_2=num2cell([Ss,Device,trigger,Collision,NaN,ck_ped,ck_txbk_rta,ck_pre_time,ck_post_time,txbk_msg_latency,tx_msg_latency,brake_event_delay,...
    txbk_pre_rms_steer,txbk_pre_LPSD,txbk_pre_LPEXC,txbk_pre_LPEXC_time,txbk_pre_avg_speed,txbk_pre_avg_FD,txbk_pre_TTC_min,txbk_pre_FD_min,txbk_pre_wnd_length,...
    txbk_initiate_rms_steer,txbk_initiate_LPSD,txbk_initiate_LPEXC,txbk_initiate_LPEXC_time,txbk_initiate_avg_speed,txbk_initiate_avg_FD,txbk_initiate_TTC_min,txbk_
    txbk_readthink_rms_steer,txbk_readthink_LPSD,txbk_readthink_LPEXC,txbk_readthink_LPEXC_time,txbk_readthink_avg_speed,txbk_readthink_avg_FD,txbk_readthink_TTC_m
    txbk_bk_onset,txbk_bk_offset,txbk_hy_res,txbk_rta,NaN,...
    txbk_reply_rms_steer,txbk_reply_LPSD,txbk_reply_LPEXC,txbk_reply_LPEXC_time,txbk_reply_avg_speed,txbk_reply_avg_FD,txbk_reply_TTC_min,txbk_reply_FD_min,txbk_re
    txbk_send_rms_steer,txbk_send_LPSD,txbk_send_LPEXC,txbk_send_LPEXC_time,txbk_send_avg_speed,txbk_send_avg_FD,txbk_send_TTC_min,txbk_send_FD_min,txbk_send_colli
    txbk_post_rms_steer,txbk_post_LPSD,txbk_post_LPEXC,txbk_post_LPEXC_time,txbk_post_avg_speed,txbk_post_avg_FD,txbk_post_TTC_min,txbk_post_FD_min,txbk_post_wnd_l
    tx_pre_rms_steer,tx_pre_LPSD,tx_pre_LPEXC,tx_pre_LPEXC_time,tx_pre_avg_speed,tx_pre_avg_FD,tx_pre_TTC_min,tx_pre_FD_min,tx_pre_wnd_length,...
    tx_initiate_rms_steer,tx_initiate_LPSD,tx_initiate_LPEXC,tx_initiate_LPEXC_time,tx_initiate_avg_speed,tx_initiate_avg_FD,tx_initiate_TTC_min,tx_initiate_FD_min
    tx_readthink_rms_steer,tx_readthink_LPSD,tx_readthink_LPEXC,tx_readthink_LPEXC_time,tx_readthink_avg_speed,tx_readthink_avg_FD,tx_readthink_TTC_min,tx_readthink
    tx_reply_rms_steer,tx_reply_LPSD,tx_reply_LPEXC,tx_reply_LPEXC_time,tx_reply_avg_speed,tx_reply_avg_FD,tx_reply_TTC_min,tx_reply_FD_min,tx_reply_collision,NaN,
    tx_send_rms_steer,tx_send_LPSD,tx_send_LPEXC,tx_send_LPEXC_time,tx_send_avg_speed,tx_send_avg_FD,tx_send_TTC_min,tx_send_FD_min,tx_send_collision,NaN,...
    tx_post_rms_steer,tx_post_LPSD,tx_post_LPEXC,tx_post_LPEXC_time,tx_post_avg_speed,tx_post_avg_FD,tx_post_TTC_min,tx_post_FD_min,tx_post_wnd_length,...
    bk_pre_rms_steer,bk_pre_LPSD,bk_pre_LPEXC,bk_pre_LPEXC_time,bk_pre_avg_speed,bk_pre_avg_FD,bk_pre_TTC_min,bk_pre_FD_min,bk_pre_wnd_length,...
    bk_rms_steer,bk_LPSD,bk_LPEXC,bk_LPEXC_time,bk_avg_speed,bk_avg_FD,bk_TTC_min,bk_FD_min,bk_collision,bk_bk_onset,bk_bk_offset,bk_hy_res,bk_rta,...
```

```
    bk_post_rms_steer,bk_post_LPSD,bk_post_LPEXC,bk_post_LPEXC_time,bk_post_avg_speed,bk_post_avg_FD,bk_post_TTC_min,bk_post_FD_min,bk_post_wnd_length,f+numel(fNam
dlmwrite(strcat(pwd,filesep,Output_flatfile_2),Output_2,'-append');

Output_index=num2cell([Ss,Device,trigger,t_count,data_end,double(Count),txbk_pre,txbk_initiate,t_txbk,gr_txbk,bkon_txbk,bkoff_txbk,rta_txbk,txbk_off,txbk_post,...
    bk_pre,t_bk,gr_bk,bkon_bk,bkoff_bk,rta_bk,bk_post,tx_pre,tx_initiate,tx_off,tx_post,f+numel(fNames)]);
dlmwrite(strcat(pwd,filesep,Output_indexfile),Output_index,'-append');
```

Problem log

```
if (c_check+LPEXC_check) >0;
    Output_log=num2cell([Ss,Device,device_check,t_check,c_check,LPEXC_check,f+numel(fNames)]);
    dlmwrite(strcat(pwd,filesep,Output_logfile),Output_log,'-append');
end
```

```
end
```

```
end

fprintf('Number of files processed: %d\n', f);
fprintf('Number of files difference: %d\n', f_diff);

fNames=fNames_re;
save('fNames.mat','fNames');

msgbox(strcat('Number of files processed:  ', num2str(f), '. Number of files difference: ', num2str(f_diff)));

end

%clear n br resize ped_check Output_flatfile_1 Output_flatfile_2 Output_index Output_logfile row_adj t_speed t_speed_decr end_speed pre_sec post_sec FDadj Multipli
```

Published with MATLAB® R2013a