```
function [duration,curvature,d2x,d2y,can_lane_changing,x,y,acceleration,distance,amax]↙
= getDurationandCurvatureandAcceleration(velocity_vehicle_lead,↙
initial_lane_changing_velocity_ego,t,coor_trans_t,turn_signal,velocity_right_lane_max,↙
velocity_left_lane_max)

% Calculate the desired curvature,longitudinal velocity and duration of lane changing↙
maneuver.

% Inputs:
% velocity_lead_vehicle: speed of the lead vehicle B. Unit: m/s
% safety_distance: safety distance

% initial_lane_changing_velocity_ego: the initial speed of lane changing maneuver.

% coor_trans_t: the start time of lane changing maneuver,which is used for coordinate↙
transformation.

% turn_signal: 0->lane keeping,1,2->straight lane changing,3,4->curve lane,changing,

% velocity_right_lane_max: maximal speed allowed on the right lane.
% velocity_left_lane_max: maximal speed allowed on the left lane.


% Outputs:
% curvature: the desired curvature of lane changing.
% d2x: the desired longitudinal acceleration of lane changing.
% duration: the duration of lane changing or u-turn maneuver.
% can_lane_changing: 1->lane changing allowed;0->lane chan ging not allowed.

can_lane_changing=1;
%
% % the value will be set to one until a good condition for lane change is found.
 lateral_acceleration_max = 4; % maximal lateral acceleration
  lateral_acceleration_min = -4;% minimal lateral acceleration
  longitudinal_acceleration_max = 2; % maximal longitudinal acceleration
%longitudinal_acceleration_min = -2; % minimal longitudinal acceleration

% % % maximum speed allowed on the left/right lane. Unit:m/s.

% Lane change
if turn_signal == 1 || turn_signal == 2

    % turn_signal = 1:switch left
    % turn_signal = 2:switch right

    velocity_lane_max =  velocity_left_lane_max;

if turn_signal == 2
      velocity_lane_max =  velocity_right_lane_max;
end

 % straight lane change

 % the target speed velocity_ego_target should be greater than the overtaken
    % vehicle speed by at least 20km/h, while respecting the maximum speed
```

```
    % allowed on the left lane, which yields:

  % velocity_ego_target = min(velocity_vehicle_lead+20km/h,velocity_left_lane_max)

  % velocity_leftlane_max is the maximum speed allowed on the left lane.

  % the vehicle A can not overtake the vehicle B unless its speed is greater than or↙
equal to the overtaken vehicle speed,this yields:

 % velocity_ego_target = max(min(velocity_vehicle_lead+20km/h,velocity_left_lane_max),↙
velocity_ego_initial)
 % https://tel.archives-ouvertes.fr/tel-01727720/file/These_UTC_Alia_Chebly.pdf,page↙
40,formula (2.32);

velocity_ego_target=max(min(velocity_vehicle_lead + 25*1000/3600, velocity_lane_max),↙
initial_lane_changing_velocity_ego);

 distance_between_ego_B_initial = 20;

  %the distance between the ego vehicle and the overtaken vehicle at the beginning of↙
the lane changing maneuver

 safety_distance = 3; % safety_distance: safety distance between the projection
%     % of the vehicle A on its lane and the position of the vehicle B on that
    % lane, when A reaches iTS lane;or the distance between vehicle ego and B when ego↙
vehicle finished the returning

    lane_width=3.6;%lane_width

    % The derive of duration is at page 40,41,part:
    % Respecting safety requirements,formula (2.36)
    % https://tel.archives-ouvertes.fr/tel-01727720/file/These_UTC_Alia_Chebly.pdf

%duration= 2;
 duration = 2*(distance_between_ego_B_initial - safety_distance )/(velocity_ego_target↙
+ initial_lane_changing_velocity_ego - 2*velocity_vehicle_lead);

 % https://tel.archives-ouvertes.fr/tel-01727720/file/These_UTC_Alia_Chebly.pdf
    % The derive of duration_lane_changing_min is at page 39,40,part:
    % Respecting the vehicle dynamic constraints,formula (2.25), (2.30)

 duration_lane_changing_min_1 = max(sqrt(5.77*lane_width/lateral_acceleration_max),↙
sqrt(-5.77*lane_width/lateral_acceleration_min));
 duration_lane_changing_min_2 = (velocity_ego_target -↙
initial_lane_changing_velocity_ego)/(2/3*longitudinal_acceleration_max);
% %
%     % page 41, formula(2.39)
%     % In order to find a proper duration(the solution space is not empty),the
%     % minimal duration must less than the maximal duration.If the solution
%     % space is empty,we decide not to change lane.
%
    duration_lane_changing_max = duration;
%
     if max(duration_lane_changing_min_1,duration_lane_changing_min_2) <↙
duration_lane_changing_max
```

```
        can_lane_changing = 1;
     else
        can_lane_changing = 0;
      end

    t = t - coor_trans_t;

  distance = velocity_ego_target+ initial_lane_changing_velocity_ego*duration/2;
  % coordinate transformation
  % the equation for longitudinal position:


% https://tel.archives-ouvertes.fr/tel-01727720/file/These_UTC_Alia_Chebly.pdf
% page 37,formula (2.15);page 38,formula (2.18)
  %4th and 5th order equation

%     a0 = 0;
%     a1 = initial_lane_changing_velocity_ego;
%     a2 = 0;
%     a3 = (velocity_ego_target - initial_lane_changing_velocity_ego)/duration^2;
%     a4 = (-a3/(2*duration));

    %5th order equation (v1.0)
% a0 = 0;
% a1 = initial_lane_changing_velocity_ego;
% a2 = 0;
% a3 = 10*(velocity_ego_target - initial_lane_changing_velocity_ego)/duration^3;
% a4 = -15*(velocity_ego_target -  initial_lane_changing_velocity_ego)/duration^4;
% a5 = 6*(velocity_ego_target - initial_lane_changing_velocity_ego)/duration^5;

%  %5th order equation (v1.1)
% a0 = 0;
% a1 = initial_lane_changing_velocity_ego;
% a2 = 0;
% a3 = 5*(velocity_ego_target - initial_lane_changing_velocity_ego)/duration^2;
% a4 = -(15/2)*(velocity_ego_target -  initial_lane_changing_velocity_ego)/duration^3;
% a5 = 3*(velocity_ego_target - initial_lane_changing_velocity_ego)/duration^4;

% % %6th order equation
a0 = 0;
a1 = initial_lane_changing_velocity_ego;
a2 = 0;
a6= 0.0001;
a3 = (-1/duration^2)*(initial_lane_changing_velocity_ego - velocity_ego_target +↙
a6*duration^5);
a4 = (1/duration^3)*(0.5*initial_lane_changing_velocity_ego-0.5*velocity_ego_target +↙
3* a6*duration^5);
a5 = -(3*a6*duration);

%4th order
% x = a0 + a1*t + a2*t^2 + a3*t^3 + a4*t^4 ;
%  dx = a1 + 3*a3*t^2 + 4*a4*t^3 ;  % longitudinal velocity
%  d2x= 6*a3*t + 12*a4*t^2 ; % longitudinal acceleration
% % %
 %5th order
```

```
% x = a0 + a1*t + a2*t^2 + a3*t^3 + a4*t^4 + a5*t^5;
%  dx = a1 + 3*a3*t^2 + 4*a4*t^3 + 5*a4*t^4;  % longitudinal velocity
%  d2x= 6*a3*t + 12*a4*t^2 + 20*a5*t^3; % longitudinal acceleration
% %
% %6th order
x = a0 + a1*t + a2*t^2 + a3*t^3 + a4*t^4 + a5*t^5+ a6*t^6;
dx = a1 + 3*a3*t^2 + 4*a4*t^3 + 5*a4*t^4 +6*a6*t^5;  % longitudinal velocity
d2x= 6*a3*t + 12*a4*t^2 + 20*a5*t^3 +30*a6*t^4; % longitudinal acceleration

% the equation for lateral position:


% https://tel.archives-ouvertes.fr/tel-01727720/file/These_UTC_Alia_Chebly.pdf
% page 37,formula (2.16);page 38,formula (2.20)

%     b0 = 0;
%     b1 = 0;
%     b2 = 0;
%     b3 = ((-1)^(turn_signal + 1))*10*lane_width/duration^3;
%     b4 = ((-1)^(turn_signal + 1))*(-15)*lane_width/duration^4;
%     b5 = ((-1)^(turn_signal + 1))*6*lane_width/duration^5;
%
% y = b0 + b1*t + b2*t^2 + b3*t^3 + b4*t^4 + b5*t^5;
%
%   Desired curvature;
%     https://en.wikipedia.org/wiki/Curvature
%     part:In terms of a general parametrization
%
%     Desired x position

%     x = a1*t+a3*t^3+a4*t^4+a5*t^5;
%     Desired y position

%     y = b3*t^3+b4*t^4+b5*t^5;
%
%     dy = 3*b3*t^2 + 4*b4*t^3 + 5*b5*t^4; % lateral velocity
%
%     d2y = 6*b3*t + 12*b4*t^2 + 20*b5*t^3; % lateral acceleration


%
%     %7th degree for lateral displacement
    b0 = 0;
    b1 = 0;
    b2 = 0;
    b3 = 0;

%     tn= velocity_ego_target*duration;

    b4 = ((-1)^(turn_signal + 1))*(35)*(lane_width)/duration^4;
    b5 = ((-1)^(turn_signal + 1))*(-84)*(lane_width)/duration^5;
    b6 = ((-1)^(turn_signal + 1))*(70)*(lane_width)/duration^6;
    b7 = ((-1)^(turn_signal + 1))*(-20)*(lane_width)/duration^7;

%tn= velocity_ego_target*duration;
```

```
%     b4 = 35*lane_width/tn^4;
%     b5 = (-84)*lane_width/tn^5;
%     b6 = (70)*lane_width/tn^6;
%     b7 = (-20)*lane_width/tn^7;

    y = b0 + b1*t + b2*t^2 + b3*t^3 + b4*t^4 + b5*t^5+ b6*t^6+ b7*t^7;
%
    dy = 4*b4*t^3 + 5*b5*t^4+ 6*b6*t^5+ 7*b7*t^6; % lateral velocity

    d2y = 12*b4*t^2 + 20*b5*t^3 +30*b6*t^4 +42*b7*t^5; % lateral acceleration

    %d3y = 24*b4*t + 60*b5*t^2 +120*b6*t^3 +210*b7*t^4;

    curvature = (dx*d2y - d2x*dy)/(dx^2 + dy^2)^(3/2);

    acceleration = sqrt(d2x^2+ d2y^2);
    amax= (sqrt(lane_width^2+(0.5*initial_lane_changing_velocity_ego - 0.5↙
*velocity_ego_target)^2)*(10*sqrt(3)))/(3*duration^2);
% elseif turn_signal == 2
%
%  d2x = 1.4 ;
%
%  duration = 2 ;
%
%  curvature = 0;

else
     x=0;
     y=0;
     d2x = 0;
     d2y = 0;
     %d3y=0;
    curvature=0;
    duration=0;
   acceleration= 0;
   distance= 0;
   amax=0;
end
```