Lane changing operations of autonomous vehicles

TIANCHI YU, Institut Polytechnique de Paris, France

This is a mini-project that models and simulates of lane change maneuver of autonomous vehicles. Here we construct a bicycle model for each vehicle and solve the dynamical system based on state-space equations. Using LQR linear controller, we construct a closed loop model of system and simulate the whole process by MATLAB/Simulink. The results can verify our design, to a certain extent.

CCS Concepts: • **Cyber-Physical System**; • **Modelling Physical** → *Lane change*; • **Model-based Simulation**; • **Simulation** → ODE dynamical system;

Additional Key Words and Phrases: simulink,autonomous

ACM Reference Format:
Tianchi YU. 2021. Lane changing operations of autonomous vehicles. In . France, 19 pages.

1 INTRODUCTION

The lane change strategy can be divided according to the existence of road infrastructure or reference trajectory. Here, we provide a model that could display the maneuver of lane change and express the security based on the simulation. To specify, our model include four autonomous vehicles, where three cars are driving in the right lane and one is in the left lane. The purpose is that the vehicle in the left lane want to move to the right while avoiding collisions. Suppose that each vehicle is equipped with sensors (with reasonable errors) and can communicate with its neighboring cars (send necessary information). This maneuver can be considered as an automated process.[7]

The vehicle dynamics are represented by a dynamic bicycle model, and each vehicle is composed of a linear controller (which is LQR controller actually) that regulates its own lateral and longitudinal behavior. In order to ensure safe handling and meet traffic regulations, we use a cooperative driving control scheme that determines the actions of each vehicle.

2 MODEL - SYSTEM DESCRIPTION

In this section, we present the details of the scenario and describe the whole system. At first, let's consider the real scenario on the road. There are four vehicles on the road, three of them are on the right lane with same speed and another one are on the left lane. Now we want to implement one maneuver of automated merging maneuver, i.e. how to insert a vehicle from on-ramp in the middle between two pre-selected vehicles of a platoon in the main lane. To be specifically, the vehicle on the lane has to merge to the right one, because of high layer like road infrastructure or the emergency, e.g. obstacle avoidance.

Suppose that all vehicles are equipped with sensors used to measure the orientation, position and velocity. Besides, all vehicles have the capacity to communicate with their neighboring vehicles, the important information are longitudinal position and speed.

Manuscript submitted to ACM

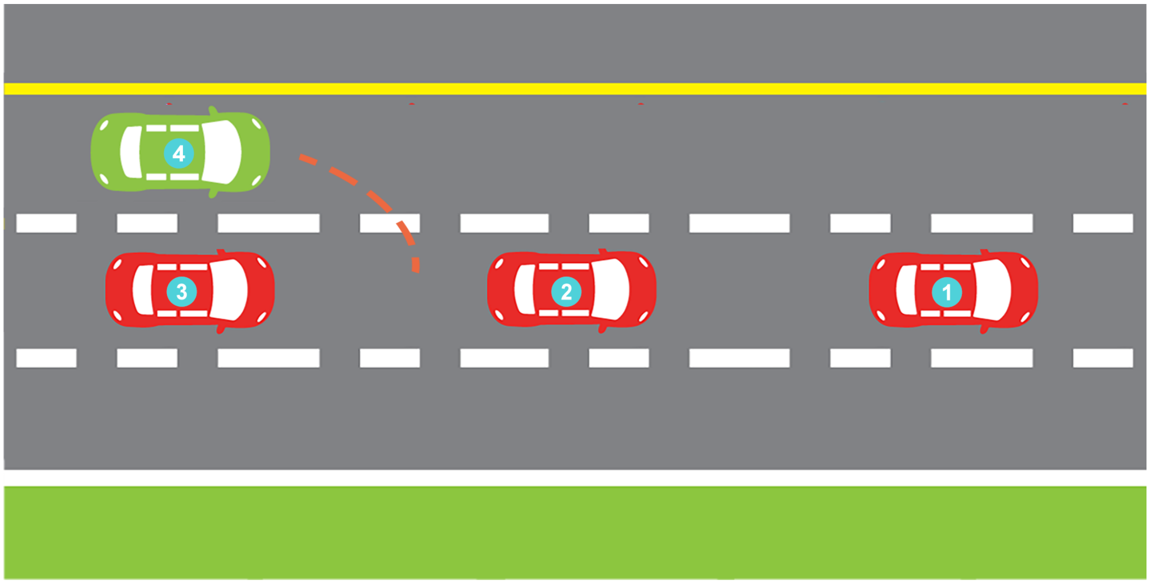


Fig. 1. Lane Change Scenario

Notice that, we suppose the width of each lane is 5m, and we take the y coordinate of middle line of the right lane as 0.

2.1 System Specifications

Before design the model, we should define the specifications of our system. For each vehicle, we care about its safety margins with surrounding vehicles, the respect for traffic rules and the physical constraints,etc.[5] More precisely, they could be interpreted as:

1. The distance of two neighboring vehicles of the platoon should always maintain larger than a given threshold,

2. The vehicles of the platoon should maintain a constant time gap(t_{gap}) (a.k.a time-to-collision[10]) between each other;

3. The manoeuvre should only be initiated if the time gap is greater than a given value(t_{gap_m}),

4. Once the manoeuvre is finished, the vehicles should form a platoon and the velocity of all vehicles should reach $\|v_{des} \pm \epsilon\|$, where ϵ is a user-defined metric,

5. The practical velocity bounds of vehicles exist, e.g. $v_{min} \leq v \leq v_{max}$,

6. The control inputs are bounded.

2.2 Vehicle Dynamics

For vehicle dynamics, there are a large variety of models. As in the literature of autonomous vehicles, dynamic and kinematic bicycle models are commonly used[4]. In this case, instead of a kinematic model, a dynamic model for lateral vehicle motion must be developed. So we consider a dynamic bicycle model with a linear tire model. The model is assumed linear to avoid computational complexity. The physical interpretation of the bicycle model is showed as the Fig2.

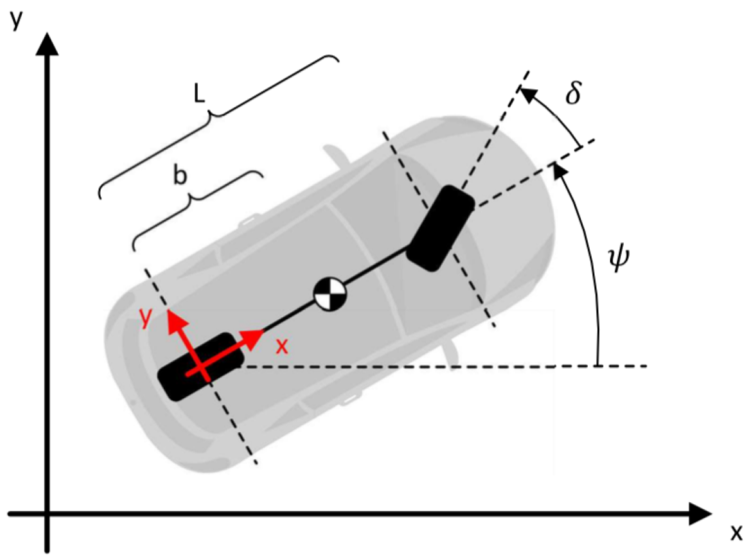


Fig. 2. Bicycle Model[6]

The state vector contains (we take the position of the rear axle of vehicles as the reference of longitudinal and lateral position)[9]:

- the longitudinal position of the rear axle $p_{x,r}$
- the lateral position of the rear axle $p_{y,r}$
- the yaw angle ψ
- the longitudinal velocity v_x
- the lateral velocity at the center of the rear axle v_y
- the yaw rate ω

The inputs of command are the longitudinal acceleration a_x and the steering angle δ. The state vector is measured and we model additive measurement noise in all state dimensions, which are $e^m_{x,r}, e^m_{y,r}, e^{'}_{\psi}, e^m_{v_x}, e^m_{v_y}, e^m_{\omega}$.

Then define the disturbances as three normalized forces, with the error force $e^d_{f_x}$ acting in longitudinal direction, $e^d_{f_y,f}$ acting in lateral direction at the front axle and $e^d_{f_y,r}$ acting in lateral direction at the rear axle. Finally, the state vector could be expressed as(in vector formal)

Table 1. Parameters for bicycle model

Description	Symbol	Value
wheelbase(m)	L	2.7
gravitational constant (m/s^2)	g	9.81
friction coefficient	μ	0.8
distance from front wheels to center of gravity	a	$(1-\frac{b}{L})\cdot L$
ratio of mass to rotational inertia (m^2/s^2)	J/m	1.57
relative position of center of gravity	b/L	0.57
relative front tire stiffness	c_f	-10.8
relative rear tire stiffness	c_r	-17.8

$$x = [p_{x,r}, p_{y,r}, \psi, v_x, v_y, \omega] \quad \text{state wrt. rear axle} \tag{1}$$

$$u = [a_x, \delta]^T \quad \text{input} \tag{2}$$

$$y = [p_{x,r}, p_{y,r}, \psi, v_x, v_y, \omega]^T \quad \text{measurement/output} \tag{3}$$

$$v = [e^m_{x,r}, e^m_{y,r}, e_{\psi}, e^m_{v_x}, e^m_{v_y}, e^m_{\omega}]^T \quad \text{measurement error} \tag{4}$$

$$w = [e^d_{f_x}, e^d_{f_{y,f}}, e^d_{f_{y,r}}]^T \quad \text{disturbance} \tag{5}$$

Starting from first-principles, as shown in [9], the differential equations of the dynamic bicycle model are defined:

$$f_B(x,u) = \begin{cases} \dot{x}_1 &= x_4 cos(x_3) - x_5 sin(x_3) \\ \dot{x}_2 &= x_4 sin(x_3) + x_5 cos(x_3) \\ \dot{x}_3 &= x_6 \\ \dot{x}_4 &= u_1 + x_5 x_6 + w_1 \\ \dot{x}_5 &= f_{y,f}(x,u,w) + f_{y,r}(x,w) - x_4 x_6 \\ \dot{x}_6 &= a\frac{m}{J}(f_{y,f}(x,u,w)) - b\frac{m}{J}(f_{y,r}(x,w)) \end{cases} \tag{6}$$

with the normalized front and rear lateral forces $f_{y,f}(x,u), f_{y,r}(x)$ given as

$$f_{y,f}(x,u,w) = c_f \mu g \frac{b}{a+b}\left(\frac{x_5 + (a+b)\cdot x_6}{x_4} - u_2\right) + w_2$$

$$f_{y,r}(x,u) = c_r \mu g \frac{a}{a+b}\frac{x_5}{x_4} + w_3$$

(Notice that c_f and c_r are negative.)

In the equations above, x_i stands for the i-th element of the state vector.

The model parameters that we use are taken from [8] and are provided in Table 1. The model measurement errors and disturbance are provided in Table 2 and Table 3.

The maximum disturbances and maximum measurement errors are slightly changed based on the article[8],

Table 2. Maximum values of measurement errors

$v_1 : e^m_{x,r}[m]$	$v_2 : e^m_{y,r}[m]$	$v_3 : e^m_{\psi}[degree]$	$v_4 : e^m_{v_x}[m/s]$	$v_5 : e^m_{v_y}[m/s]$	$v_6 : v[degree/s]$
0.04	0.04	0.1	0.05	0.05	0.1

Table 3. Maximum values of disturbances

$w_1 : e^d_{f_x}[m/s^2]$	$w_2 : e^d_{f_{y,f}}[m/s^2]$	$w_3 : e^d_{f_{y,r}}[m/s^2]$
0.1	0.057	0.043

2.3 Linearization

The nonlinear model is linearized around a set of operating points using standard point-wise linearization.[2] For linearization purposes in our case, because of the stable equilibrium and the destined velocity of vehicles(we set as 70 km/h), we consider a set point $x_{op} = [0; 0; 0; 70/3.6; 0; 0]$ and do Taylor expansion around the point.

For example, $cos(x_3) = 1+O(x_3^2)$, and in the equation of $f_{y,f}(x,u,w)$, the $c_f \mu g \frac{b}{a+b}\frac{1}{x_4}$ could be expand as $c_f \mu g \frac{b}{(a+b)(70/3.6)^2} x_4 + O(x_4^2)$. Then after linearization, $x_1 = x_4 - x_{op,5} \cdot x_3 - x_{op,3} x_5$, where $x_{op,5} = 0$, and $x_{op,3} = 0$.

In total, the equations are represented as(ignore the values of 0 in the following function):

$$f_{BL}(x,u) = \begin{cases} \dot{x}_1 & = x_4 \\ \dot{x}_2 & = x_{op,4} \cdot x_3 + x_5 \\ \dot{x}_3 & = x_6 \\ \dot{x}_4 & = u_1 + w_1 \\ \dot{x}_5 & = c_f \mu g \frac{b}{(a+b)(70/3.6)^2} x_{op,4} x_5 + c_f \mu g \frac{b}{(70/3.6)^2} x_{op,4} x_6 \\ & \quad -c_f \mu g \frac{b}{a+b} u_2 + w_2 + c_r \mu g \frac{a}{(a+b)(70/3.6)^2} x_{op,4} x_5 + w_3 - x_{op,4} x_6 \\ \dot{x}_6 & = a\frac{m}{J}(c_f \mu g \frac{b}{(a+b)(70/3.6)^2} x_{op,4} x_5 + c_f \mu g \frac{b}{(70/3.6)^2} x_{op,4} x_6 \\ & \quad -c_f \mu g \frac{b}{a+b} u_2 + w_2) - b\frac{m}{J}(c_r \mu g \frac{a}{(a+b)(70/3.6)^2} x_{op,4} x_5 + w_3) \end{cases} \tag{7}$$

Then, we obtain the state-space representation:

$$\text{where } A =\approx \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 19.4444 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -5.5739 & -26.1530 \\ 0 & 0 & 0 & 0 & 1.1909 & -4.9609 \end{bmatrix}, B \approx \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 48.3123 \\ 0 & 35.7265 \end{bmatrix},$$

$$B_d \approx \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0.7395 & -0.9803 \end{bmatrix}, C \approx \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

2.4 Linear Controller

The objective of the linear controller of each vehicle is to regulate its position and velocity in accordance with the behavior of the other vehicles. At the same time, the lane change maneuver should be safety. Once the maneuver is completed, the vehicle platoon should maintain the predefined vehicle speed v_{des}.

In particular, we utilize a linear controller

$$u = -K \cdot y = -K \cdot (x + v) \tag{8}$$

We opt for an LQR (Linear Quadratic Regulation) controller since it is a well established design technique that provides practical feedback gains. LQR is an optimal multivariable feedback control approach which minimizes the deviation of the state trajectories of the closed-loop system while requiring minimum controller effort. The behavior of an LQR controller is determined by two parameters: state and control weighting matrices. These two matrices are design parameters and influence the success of the LQR controller synthesis.[3] **Now we need to determine the weighting matrices of the cost function.**

The choice of the design matrices Q and R is normally a problem of trial and error. There are not much literature in this sense. However, we can use **the Bryson's rule** as a first choice.[1] The Bryson's Rule: According to this rule, Q and R are diagonal matrices whose diagonal elements are respectively expressed as the reciprocals of the squares of the maximum acceptable values of the state variable (X) and the input control variable (u).

We can find the rule in the book Digital Control of Dynamic Systems by Franklin, Powell and Workman (page 400). The rule was proposed in Bryson an Ho (Applied Optimal Control, 1975).

$$J = \int_{0^{\infty}} [x^T Q x + u^T R u] dt \tag{9}$$

Where Q is the state weighting matrix with real symmetry and positive semi-definite in nature. R, is the control weighting matrix of real symmetry but positive definite in nature.[3]

$$\mathbf{Q_{ii}} = \frac{1}{\text{maximum acceptable value of } x_i^2} \tag{10}$$

where $i \in (1, 2, 3..., l)$
And the diagonal elements R_{jj} of matrix R, also, can be written as:

$$\mathbf{R_{jj}} = \frac{1}{\text{maximum acceptable value of } u_j^2} \tag{11}$$

where $j \in (1, 2, 3..., k)$

Applying Bryson's rule to the state-space equation for attitude, as in Equation, the following initial Q and R values were obtained:

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/180 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5/180 \end{bmatrix}, R = \begin{bmatrix} 1 & 0 \\ 0 & 180/\pi \end{bmatrix}$$

Then we numerically solve the continuous Algebraic Ricatti Equation with MATLAB and obtain our state-feedback matrix:

$$K = \begin{bmatrix} 1 & 0 & 0 & 2.6458 & 0 & 0 \\ 0 & 0.1321 & 2.3308 & 0 & -0.0075 & 0.4835 \end{bmatrix}$$

The eigenvalues of $A - BK$ can be calculated as: [-0.4569 + 0.0000i, -2.1889 + 0.0000i, -12.5037 + 7.5751i, -12.5037 - 7.5751i, -1.2191 + 1.2644i, -1.2191 - 1.2644i], so the system is stable.

3 SUPERVISORY CONTROL

For the controller of each vehicle, we need to define the references x_{ref}. Actually, the references for each vehicle are defined by the states of the neighboring vehicles. As such, we can conclude the expression of the supervisory controllers as:

$$u = -K(y - x_{ref}) = -K(x - x_{ref} + v) \tag{12}$$

where $x = [p_{ref,x}, p_{ref,y}, \psi_{ref}, v_{ref,x}, v_{ref,y}, \omega_{ref}]$
The control inputs could be bounded. In particular, the acceleration of vehicle is bounded as

$$-3 \leq u_1 \leq 2,$$

and the steering angle is bounded as

$$-\pi/4 \leq u_2 \leq \pi/4.$$

Then we need to define the references for different vehicles. Note that, we numerate the vehicles and give an order like: leader vehicle in the right lane(No.1), middle vehicle in the right lane(No.2), rear vehicle in the right lane(No.3), merging vehicle(No.4).

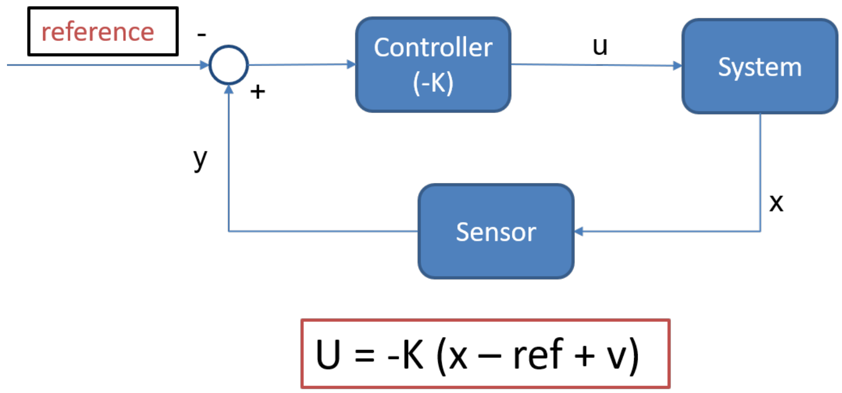


Fig. 3. Control Design(changed from [3])

3.1 Leader Vehicle

Scenario: the leader vehicle should maintain the platoon velocity or accelerate depending on the vehicle behind. (In the equations, the number inside the bracket stands for the ID of the vehicles)

$$\begin{aligned} p^{(1)}_{ref,x} &= max(p^{(1)}_{x,r}, p^{(2)}_{x,r} + t_{gap} \cdot v_x), p^{(1)}_{ref,y} = 0, \\ \psi^{(1)}_{ref} &= 0, v^{(1)}_{ref,x} = max(v_{des}, v^{(2)}_r), \\ v^{(1)}_{ref,y} &= 0, \omega^{(1)}_{ref} = 0 \end{aligned}$$

3.2 Rear Vehicle

Scenario: For the vehicle at the tail of the platoon, it should maintain the smaller speed between the merging vehicle and the vehicle in front of it to avoid any crashes.

$$\begin{aligned} p^{(3)}_{ref,x} &= min(p^{(4)}_{x,r} - t_{gap} \cdot v^{(4)}_x, p^{(2)}_{x,r} - t_{gap} \cdot v^{(2)}_x), p^{(3)}_{ref,y} = 0, \\ \psi^{(3)}_{ref} &= 0, v^{(3)}_{ref,x} = min(v^{(2)}_r, v^{(4)}_r), \\ v^{(3)}_{ref,y} &= 0, \omega^{(3)}_{ref} = 0 \end{aligned}$$

3.3 Middle Vehicle

Scenario: For the vehicle in the middle, to make sure that the merging vehicle could merge into the second and third vehicle, it should accelerate if there is not enough space for the lane change and respect the speed of the first leader vehicle.

$$
\begin{aligned}
&p_{ref,x}^{(2)} = max(\frac{p_{x,r}^{(2)} - t_{gap} \cdot v_x^{(2)} + max(p_{x,r}^{(3)} + t_{gap} \cdot v_x^{(3)}, p_{x,r}^{(4)} + t_{gap} \cdot v_x^{(4)})}{2}, p_{x,r}^{(1)} - t_{gap} \cdot v_x^{(1)}),\\
&p_{ref,y}^{(2)} = 0, \psi_{ref}^{(2)} = 0, v_{ref,x}^{(2)} = v_x^{(1)}, v_{ref,y}^{(2)} = 0, \omega_{ref}^{(2)} = 0
\end{aligned}
$$

3.4 Merging Vehicle

To finish the lane change, we assume that there are two phrases for merging vehicle. The first phrase is kind of preparation step, which is before the manoeuvre. The vehicle needs to check if and when it is feasible to do the lane change. In essence, the vehicle needs to regulate its velocity with respect to the platoon velocity (while guaranteeing that there is enough space margin).

The merging manoeuvre starts in the second phrase, This practically means that its lateral position should change. Because we set the width of the lane is 5m, and we set the middle line of right lane as y axle, so we have:

Phase 1:

$$
\begin{aligned}
&p_{ref,x}^{(4)} = min(\frac{p_{x,r}^{(2)} - t_{gap} \cdot v_x^{(2)} + p_{x,r}^{(3)} + t_{gap} \cdot v_x^{(3)}}{2}, p_{x,r}^{(2)} - t_{gap} \cdot v_x^{(2)}),\\
&p_{ref,y}^{(2)} = 5, \psi_{ref}^{(4)} = 0, v_{ref,x}^{(4)} = v_x^{(1)}, v_{ref,y}^{(4)} = 0, \omega_{ref}^{(4)} = 0
\end{aligned}
$$

Phase 2:

$$
\begin{aligned}
&p_{ref,x}^{(4)} = min(\frac{p_{x,r}^{(2)} - t_{gap} \cdot v_x^{(2)} + p_{x,r}^{(3)} + t_{gap} \cdot v_x^{(3)}}{2}, p_{x,r}^{(2)} - t_{gap} \cdot v_x^{(2)}),\\
&p_{ref,y}^{(2)} = 0, \psi_{ref}^{(4)} = 0, v_{ref,x}^{(4)} = v_x^{(1)}, v_{ref,y}^{(4)} = 0, \omega_{ref}^{(4)} = 0
\end{aligned}
$$

Besides, the transition need to meet one condition, that should be valid to initiate a safe lane change. The condition is defined as:

$$
\phi := p_{ref,x}^{(4)} < p_{ref,x}^{(2)} - t_{gap\_m} \cdot v_x 2 \text{ and } p_{ref,x}^{(4)} > p_{ref,x}^{(3)} + t_{gap\_m} \cdot v_x(3) \tag{13}
$$

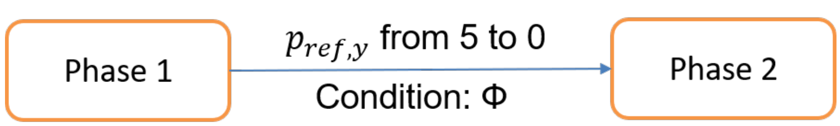


Fig. 4. Phases Transition for merging vehicle

Table 4. Parameters for control design

Description	Symbol	Value
minimum velocity (km/h)	v_{min}	0
maximum velocity (km/h)	v_{max}	150
minimum acceleration (m/s^2)	a_{min}	-3
maximum acceleration (m/s^2)	a_{max}	2
minimum steering angle (rad)	δ_{min}	$-\pi/4$
maximum steering angle (rad)	δ_{max}	$\pi/4$
destined velocity (km/h)	v_{des}	70
time gap(s)	t_{gap}	1.5
minimum time gap(s)	t_{gap_m}	1
constant initial gap(m)	gap	70/3.6 * 1.5

4 SIMULATION

4.1 Parameters and Initial state

Before simulation, we need to settle some other basic parameters for our model in Table 4. We simulate the whold model by MATLAB/Simulink. At first we construct the control model for each vehicle. Then, connect all vehicles following the dependencies between each other. We construct the global system and realize the communication among the vehicles, using the basic information(longitudinal position and velocity, etc).

The initial state for four vehile:

Leader vehicle(No.1): $x^{(1)} = [2 * gap, 0, 0, v_{des}, 0, 0]$,

Middle vehicle(No.2): $x^{(2)} = [gap, 0, 0, v_{des}, 0, 0]$,

Rear vehicle(No.3): $x^{(3)} = [0, 0, 0, v_{des}, 0, 0]$,

Merging vehicle(No.4): $x^{(4)} = [2 * gap, 5, 0, v_{des}/2, 0, 0]$

4.2 Simulation Model

The global model and the curves displays could be represented like Fig 5. The references come from the output of other vehicles, with the errors of sensors. The signal processing and dynamical control are realised separately by models of each vehicle.

Now, we go deeper for the model of each vehicle.

Example of leader Vehicle

The Fig.6 show the model of leader vehicle. We can see that, the model is separated to two parts. The first part obtain the reference and the second part realize the linear control for the dynamical system.

Actually the second part is LQR controller, for each vehicle, there is no big difference among them.

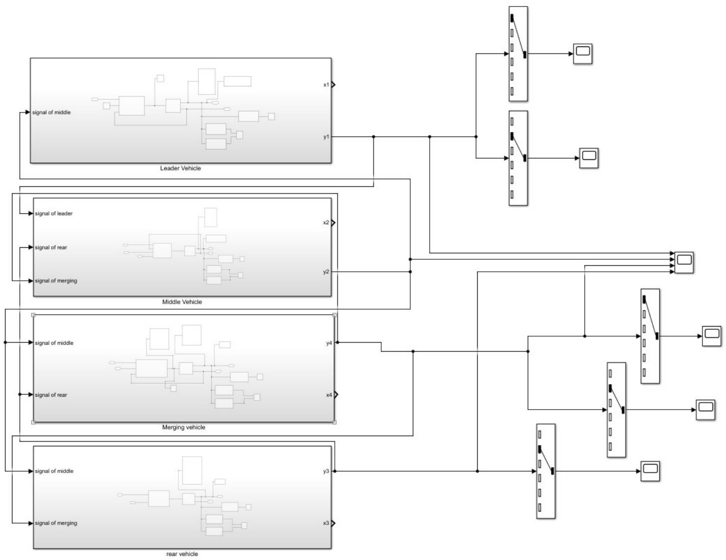


Fig. 5. Global model for whole system

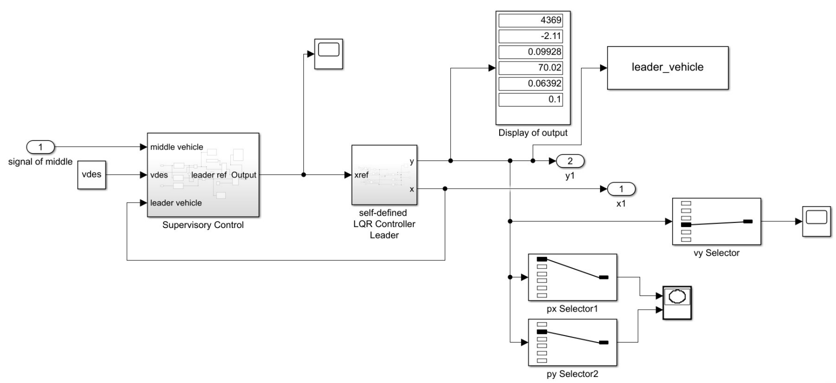


Fig. 6. Leader vehicle Model

Specialize for Merging Vehicle

For the merging vehicle, we need to consider it as a special case. As we have divided the lane change maneuver into two phase, we need to set one logic switching based on the condition ϕ we have defined. The structures are displayed in appendix at the end of the project.

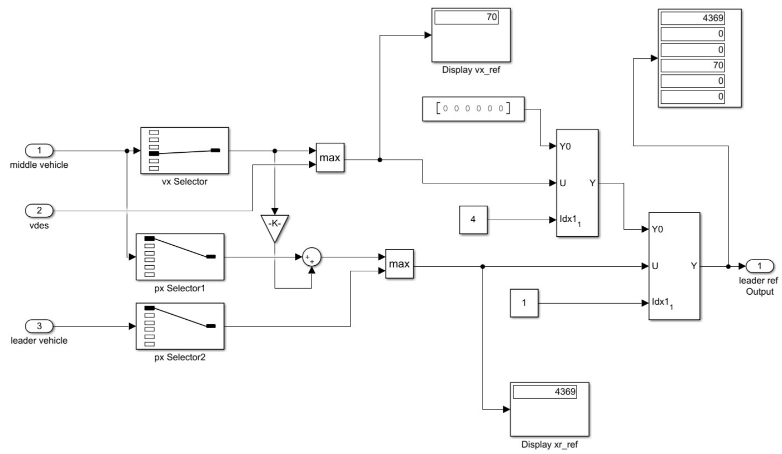


Fig. 7. Supervisory Control Part of Leader Vehicle

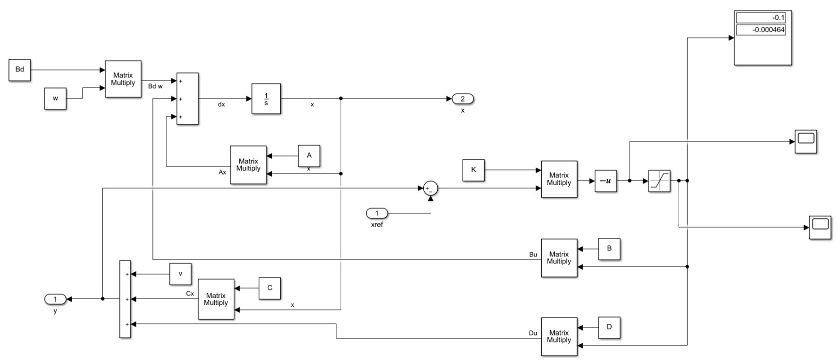
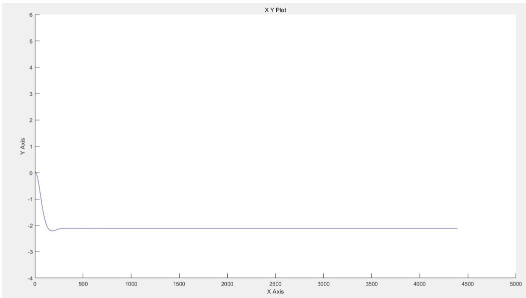


Fig. 8. LQR Controller of Leader Vehicle

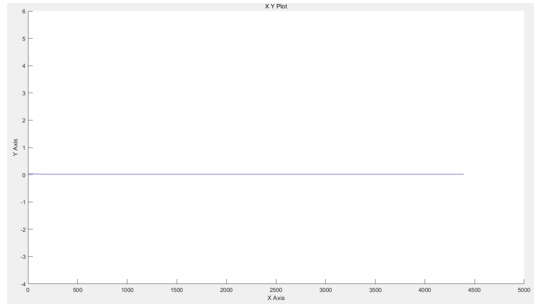
4.3 Simulation Results

We display the trajectory of three vehicles in the right lane(The trajectories are similar for them). Here we can see that, at first the initial y position is 0. Later, the position change to around -2m and keep stable at that coordinate. The reason for this kind of movement is that the errors of measurements has a bad influence in the velocity in y direction and the yaw angle, especially $v_3 : e^m_\psi$ and $v_6 : e^m_\omega$. If we set these two errors as 0, there will be no mistake/offset in y direction.

Similarly, we display the trajectory of left lane vehicle. We can see the similar performance. With the errors of measurement, there would be one offset both for the phase 1 and phase 2. The controller makes a misunderstanding for the v_y and yaw angle. However, if we don't consider these errors, the trajectory of merging vehicle seems have a good representation. At first, it prepares for the merging and keep moving(with the velocity changing), when the condition is satisfied, it starts to merge and stay in the platoon of vehicles "forever". Then we show the v_y changing and the satisfaction of the transition condition to explain its performance.

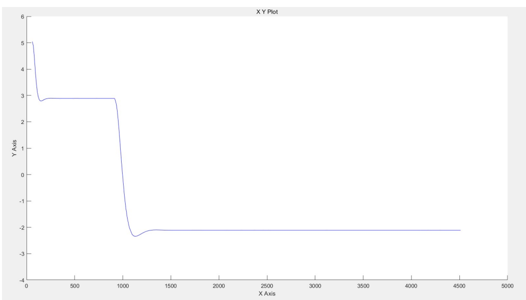


(a) pic1. with error

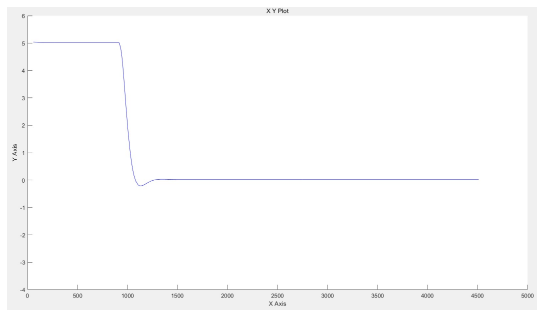


(b) pic2. e_{ψ}^{m} and v_6 : e_{ω}^{m} are 0

Fig. 9. simulation of right vehicles

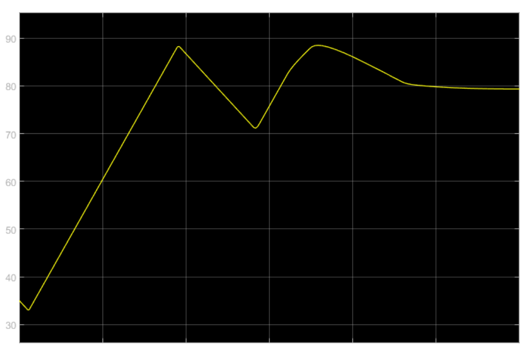


(a) pic1. with error

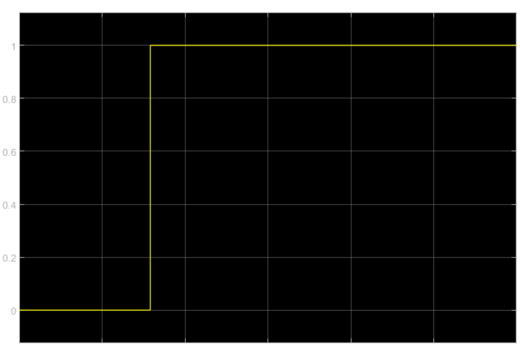


(b) pic2. e_{ψ}^{m} and v_6 : e_{ω}^{m} are 0

Fig. 10. simulation of merging vehicle



(a) pic1. velocity in x direction

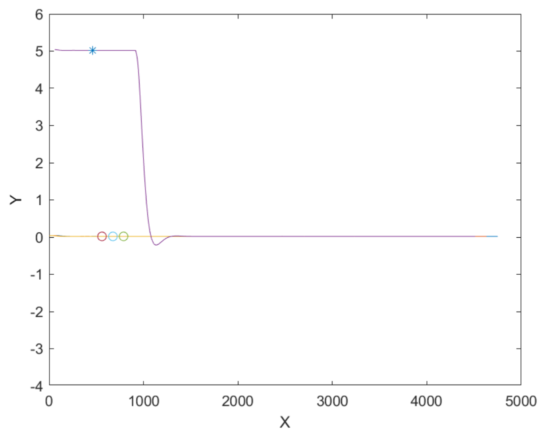


(b) pic2. the satisfaction of transition condition

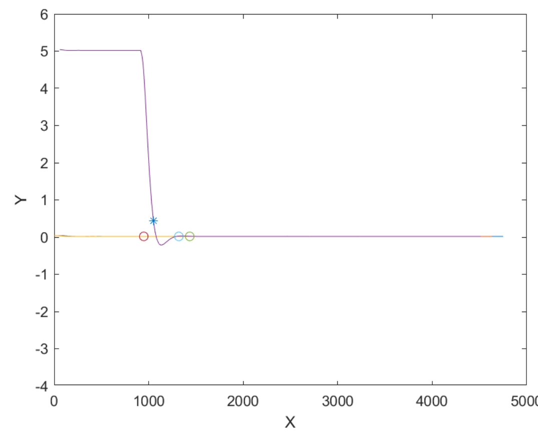
Fig. 11. simulation of right vehicles

We can easily catch the state of acceleration and the moment when merging vehicle decides to merge into the platoon by the pictures above.

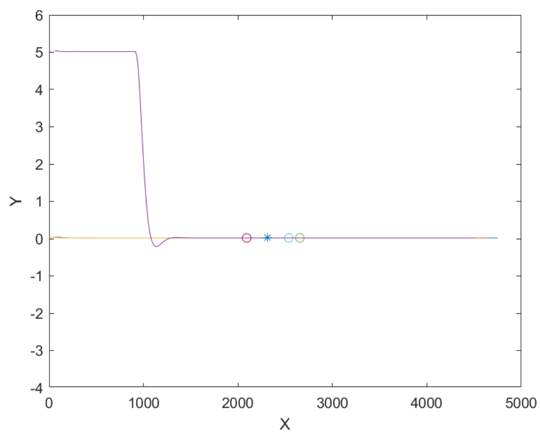
Further, to make a clear implementation of the lane change maneuver and ignore the unnecessary errors(e_{ψ}^{m} and e_{ω}^{m}), we get a complete/dynamic lane change process for our case. Here, circles stand for the position of the vehicles in the right lane, and star stands for the merging vehicle.



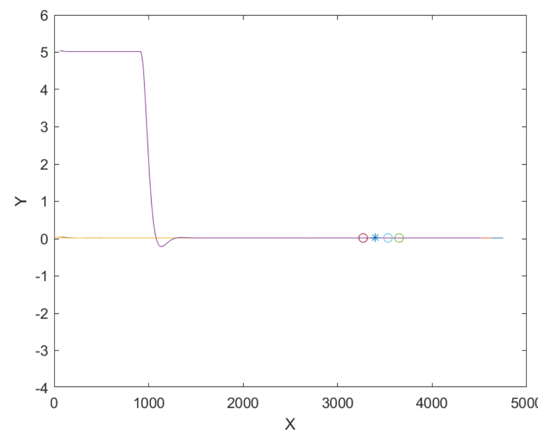
(a) pic1. Phase 1: Preparation



(b) pic2. Phase 2: Merging



(c) pic3. After merging



(d) pic4. Stable results

Fig. 12. Lane Change Process

5 CONCLUSION

We construct the dynamical model of vehicles by classical bicycle model and study one case of real scenario of lane change using a global system based on the vehicle model. To realize the performance of autonomous control, we design a LQR controller and solve the state-space equations, calculate the feedback gain which is used in the simulation. Also, we design a supervisory control to meet the safety requirements during the merging process, using the references and dependencies of the velocity and position information. Finally, we simulate the lane change maneuver by Simulink and verify the correctness of our system.

REFERENCES

[1] E. Okyere; A. bousbaine; G. T. Poyi; A.K. Joseph; J.M. Andrade. [n.d.]. LQR controller design for quad-rotor helicopters. *The Journal of Engineering* ([n. d.]).

[2] Marcelo Forets Daniel Heß, Bastian Sch¨urmann and Goran Frehse. 2017. *UnCoVerCPS Deliverable D3.2: Report on Precomputation of Reachable Sets and Advances in Reachability Analysis.* https://cps-vo.org/node/39010

[3] Kaveh Hassani and Won-Sook Lee. 2016. Multi-objective design of state feedback controllers using reinforced quantum-behaved particle swarm optimization. In *Applied Soft Computing*. 41:66–76.

[4] Georg Schildbach Jason Kong, Mark Pfeiffer and Francesco Borrelli. 2015. Kinematic and dynamic vehicle models for autonomous driving control design. In *IV Symposium.*

[5] Erik Coelingh Julia Nilsson, Mattias Br¨annstr¨om and Jonas Fredriksson. 2017. Lane change maneuvers for automated vehicles. *ITS* (2017).

[6] Michael Kaiser. 2017. *Untersuchung zur online abgesicherten Fahrer-Fahrzeug Interaktion in Kollisionsvermeidungs-Szenarien.* Ph.D. Dissertation. Technische Universit¨at Braunschweig.

[7] Xiao-Yun Lu and J Karl Hedrick. 2003. Longitudinal control algorithm for automated vehicle merging. *Internat. J. Control* (2003).

[8] Ray Lattarulo Jens Oehlerking JoshuéPérez Marianne Hartung, Daniel Heß and Alexander Rausch. 2016. *UnCoVerCPS Deliverable D5.2: Report on Conformance Testing of Application Models.* https://cps-vo.org/node/39012

[9] Rajesh Rajamani. 2011. Vehicle dynamics and control. *Springer Science Business Media* (2011).

[10] Richard van der Horst and Jeroen Hogema. 1993. *Time-to-collision and collision avoidance systems.*

A APPENDIX

A.1 Model pictures of merging vehicle

More details could be found in the MATLAB and Simulink files.

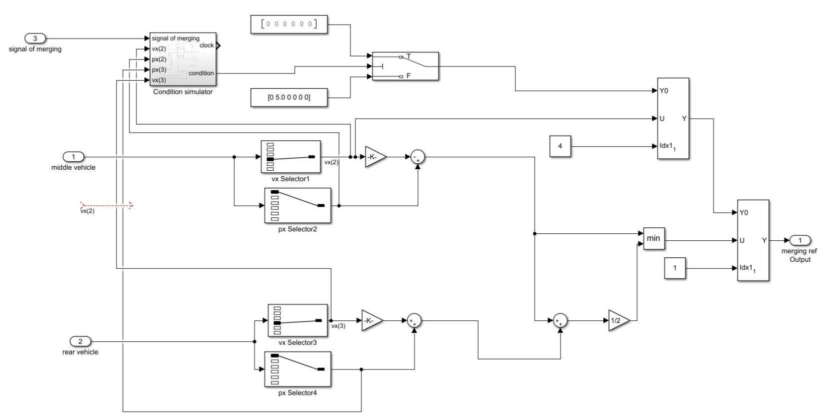


Fig. 13. Supervisory Control Design and Transition between phases

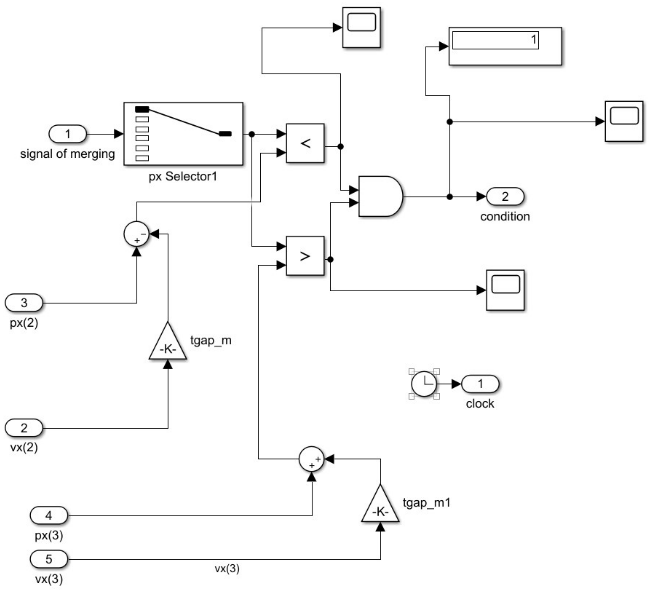


Fig. 14. Condition Construction

A.2 Calculation part of Matlab code

The more details could be found in the Matlab code file.

```
L = 2.70;
g = 9.81;
mu = 0.80;
relative_cent = 0.57 ;% b/L
b = relative_cent * L;
a = L - b; %(1.00 - relative_cent)*L;
cf = -10.80%-9.7; %;
cr = -17.80 %-25.2; %;
ratio_mass_rot = 1.57; % J/m

x_op = [0.0,0.0,0.0,70.0/3.6,0.0,0.0];

% linalisation

% x1_d
para1 = [0.0,0.0,0.0,cos(x_op(3)),0.0,0.0];
para1_u = [0.0,0.0];
para1_w = [0.0,0.0,0.0];
```

```
% x2_d
% para2 = [0.0,0.0,x_op(2),sin(x_op(3)),cos(x_op(3)),0.0]; % standard
para2 = [0.0,0.0,x_op(4),0.0,cos(x_op(3)),0.0];
para2_u = [0.0,0.0];
para2_w = [0.0,0.0,0.0];
% x3_d
para3 = [0.0,0.0,0.0,0.0,0.0,1.0];
para3_u = [0.0,0.0];
para3_w = [0.0,0.0,0.0];

% x4_d
para4_u = [1.00,0.0];
para4_w = [1.00,0.0,0.0];
para4 = [0.0,0.0,0.0,0.0,0.0,0.0];

%x5_d

% For normalized front lateral force f_yf(x,u)
para5 = [0.0,0.0,0.0,0.0,
cf*mu*g*b/L/(70.0*70.0/3.6/3.6)*x_op(4)+cr*mu*g*a/L/(70.0*70.0/3.6/3.6)*x_op(4),
cf*mu*g*b/(70.0*70.0/3.6/3.6)*x_op(4)-x_op(4)]
para5_u = [0.00,-cf*mu*g*b/L];
para5_w = [0.00,1.00,1.00];

% For normalized rear lateral force f_fr(x)
%-cf*mu*g*a/L


%x6_d

amj = a/ratio_mass_rot;
bmj = b/ratio_mass_rot;
para6 = [0.0,0.0,0.0,0.0,
amj*(cf*mu*g*b/L/(70.0*70.0/3.6/3.6)*x_op(4)) - bmj*(cr*mu*g*a/L/(70.0*70.0/3.6/3.6)*x_op(4)),
amj*(cf*mu*g*b/(70.0*70.0/3.6/3.6)*x_op(4))]
para = b/relative_cent * 1.00;

para6_u = [0.0,-cf*mu*g*b/L*a/ratio_mass_rot]; %35.7265];
para6_w = [0.0,a/ratio_mass_rot,-b/ratio_mass_rot];
```

```
A = [para1;para2;para3;para4;para5;para6]
B = [para1_u;para2_u;para3_u;para4_u;para5_u;para6_u]
Bd = [para1_w;para2_w;para3_w;para4_w;para5_w;para6_w]
Iv = [1.0,1.0,1.0,1.0,1.0,1.0];
C = diag(Iv)
D = [0.0,0.0;
     0.0,0.0;
     0.0,0.0;
     0.0,0.0;
     0.0,0.0;
     0.0,0.0];
Q = diag([1.0,1.0,1.0/180,5.0,5.0,5.0/180])
R = diag([1.0,180.0/pi])
K = lqr(A,B,Q,R)
E = eye(6);

e = eig(A - B*K)


%% parameters of model
v = [0.04,0.04,0,0.05,0.05,0];
% If we want to add some error in angle
% v = [0.04,0.04,0.1,0.05,0.05,0.1];
w = [0.1,0.057,0.043]

%% constrains of control design
v_min = 0;
v_max = 150;
a_min = -3;
a_max = 2;
delta_min = -pi/4.0;
delta_max = pi/4.0;
tgap = 1.5;
tgap_m = 1;
vdes = 70.0;
gap = 70.0/3.6 * 1.5;

%% initial conditions of each vehicle

x1_init = [2.0*gap,0.0,0.0,vdes,0.0,0.0];
x2_init = [gap,0.0,0.0,vdes,0.0,0.0];
```

```
x3_init = [0.0,0.0,0.0,vdes,0.0,0.0];
x4_init = [2.0*gap,5.0,0.0,vdes/2.0,0.0,0.0];


%% simulation part

options=simset('SrcWorkspace','current');
sim('Mymodel.slx',[],options);
```