

An Automatic Steering and Tracking System for Autonomous Vehicles based on EPS

Xiangyu Chen*, Lingzhu Xiang*, Huijing Zhao*, Hongbin Zha*

*School of Information Science/Peking university, Beijing, P.R.China

Abstract—The steering control system serves as the foundation of all autonomous vehicle activities, directly supporting trajectory planning and tracking. Prior research assumes ideal characteristics of the control system and leaves the burden of realizing such assumptions to physical implementation. However, expense of high-performance actuators and mechanical difficulty of hardware installation render this a dilemma between financial infeasibility and loss of theoretical coherence. To overcome such predicament, we propose a tightly integrated predictive steering and tracking framework capable of achieving comparable performance while uses only factory-shipped equipment built in the vehicle with cheap sensors and controllers. Careful dynamical modeling of each component and extensive on-road experiments prove the practicality and efficacy of our approach.

I. INTRODUCTION

The study of steering system is very important in the autonomous vehicle research, for the steering system directly contributes to the behavior of the autonomous vehicle. In recent years, methodology that widely applied to the steering system is rather mature. Such as in DARPA Grand Challenge in 2004 and 2005, the steering system of *Sandstorm*(the autonomous car of CMU) is made of a high torque motor, and a well designed PID control law. Other participate teams also apply similar methods to their steering systems[1]. When CMU won the champion of DARPAR Urban Challenge in 2007, their autonomous car *BOSS* still adopt the classic steering system and the pure pursuit tracking algorithm[2]. However, if we put it for pragmatic use now, the entire steering system is much too expensive. A compromise is made between the performance and the cost when researchers begin to design a steering system. Meanwhile, looking on whole, the steering part is a constituent part of the autonomous vehicle framework which is also becoming more and more concise and economical.

A. General Frameworks of Autonomous Vehicle

It is generally acknowledged that the frameworks of autonomous vehicle consist of three parts: Perception, Planning and Control. Perception part mainly deal with the data that the sensors(Camera, Laser, Radar, GPS/IMU) collected. Semantic information extraction, transient location and posture estimation is very urgent. Only by acquiring these data, the steering system could function well.

Planning part made decisions during the entire task by taking every data that possesses semantic information into consideration. Basically there are two kind of planning, global planning and local planing. The global planning is responsible for finding a solution to the final destination of the task while



Fig. 1. Steering Motors

the local planning deal with each subtasks. Both of them have been studied intensively in recent decades and several relatively mature planning methodologies have been established[3] [4] [5].

The last part is more fundamental. The control law and actuator have considered vehicle kinematic characters, driving sequences, traffic behaviors and the comfortableness of the passengers when driving at a certain speed. The robustness and accuracy are the core indicators of the control system.

B. Related Steering and Tracking Method

Most research institutes apply BLDCM(Brushless DC Motor Fig. 1) to their autonomous vehicles[1] [2] [6], because the BLDCM can provide both enough torque and turning speed while steering, and the BLDCM is relatively cheaper than step motor. Both of these two motors belong to the additional accessories, which means manual installing is necessary, and higher accuracy the motor is, the higher cost the system would be. To make a compromise between the accuracy and cost, researches have to design the corresponding tracking methods which varies from each research institutes in recent study.

The algorithm that widely used in path tracking issue is the Pure Pursuit algorithm [7] which is proved effective in practical use. Since Pure Pursuit is a simple combination of vehicle dynamics and road geometric[8], the result severely depend on the look-ahead distance. So some of the researchers try to analyze how to choose a better look-ahead distance so that the robustness of the pure pursuit could be guaranteed. At the

¹Acknowledgement (This paper is partially supported by the NSFC Grants(91120004, 91120010, 90920304))

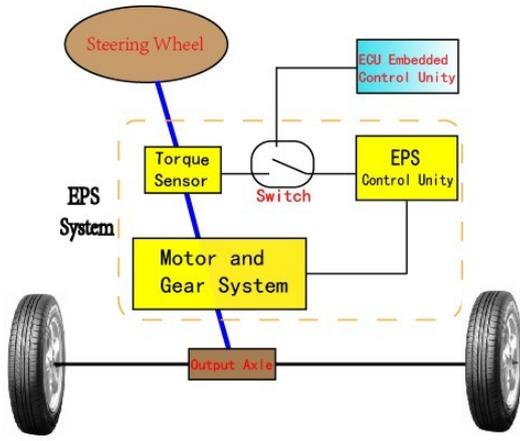


Fig. 2. Steering Motors

meantime, some improved pure pursuit algorithm is proposed by other researchers[9].

C. Outline of the Research

Under the tendency that the steering system is required to be cheaper and more convenient to settle down, this paper provide a simple steering structure both in hardware and software, and give a solution to the corresponding tracking algorithm. The advantage of our system is the low expense and it is easy to add, which means our method does not require additional motor and connection elements between the turning structure and the motor. Only by connecting our ECU(Electronic Control Unity) to the EPS module which the car already equipped and apply the control and tracking algorithm to the system, this method can give a robust, real-time and switchable solution to the steering system.

The EPS system is introduced in Section 2. By illustrating the characters of the EPS system, the automatic steering system is described in Section 3 and the corresponding tracking method is given in Section 4. Experimental validations and future work are presented in Section 5 and 6 respectively.

II. THE EPS SYSTEM

As a matter of fact, the mechanism between the steering wheel and the tire is complicate. The EPS system is an important part of this mechanism. It helps turning the front tyres by detecting the manual input torque from the steering wheel, which made it much easier for manual steering.

Despite of some mechanical connection structure, the EPS system is made of a brushless DC motor and a torque sensor. In this method, we set up a switch between the torque sensor and the motor control unity, which allow us to control the brushless DC motor directly. When it is necessary for manual steering, switching to manual mode simply requires pushing a button or giving an order to the controller. As Fig. 2 show, the torque sensor is taken as the input and the motor actuator is regarded as a black box because the accurate structure of the EPS system

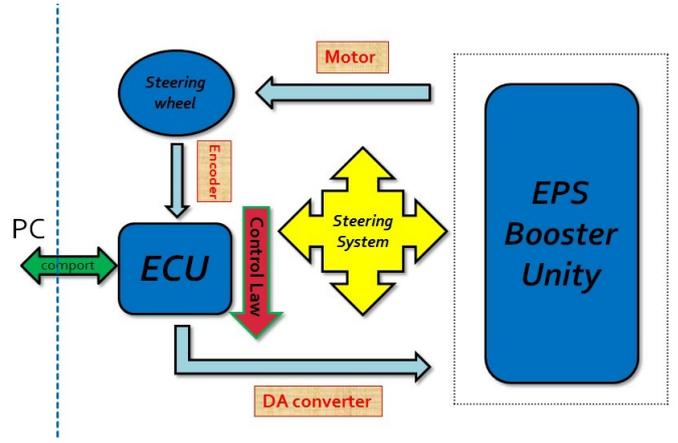


Fig. 3. Steering System

is unknown and secured. Building a model to describe how the EPS system works and estimated the key parameters is also very necessary. By simulating the output signal of torque sensor, the controller give similar signal to the actuator through a simple PID control law programmed in the embedded system. The experiment suggest our system is robust and effective.

Table. I shows the Characteristics of the vehicle EPS system. By cutting down the connection between the Torque output and the EPS's ECU, we directly provide the $0 \sim 5V$ analog voltage to the EPS' ECU unity(validity range $1 \sim 4V$), which is a way to give indications to the DC motor indirectly and finally turning the front tyres of the vehicle.

TABLE I
THE EPS CHARACTERISTICS[10]

Sensor and actuator	Input and Output				
	Input	Num	Output	Num	Other
Torque Sensor	Steering Torque	$\leq 6N.m$	Analog Voltage	$1 \sim 4V$	Unknown
Motor Cluth	Logic Signal	0 or 1	Direct Current	30A	Unknown
DC Motor	Direct Current	$\leq 30A$	Turning Torque	$22N.m$	Unknown
ECU Unity	Analog Voltage	$1 \sim 4V$	Logic Indication	Pulse Signal	Unknown

III. STEERING CONTROL SYSTEM

In our Steering control system, we install an encoder on the steering wheel to record the state of the steering wheel and made our system a close-loop control system(Fig. 3). Then an improved PID control law is implemented in our ECU. Through a digital to analog converter, the output signal can be directly imported to the EPS system.

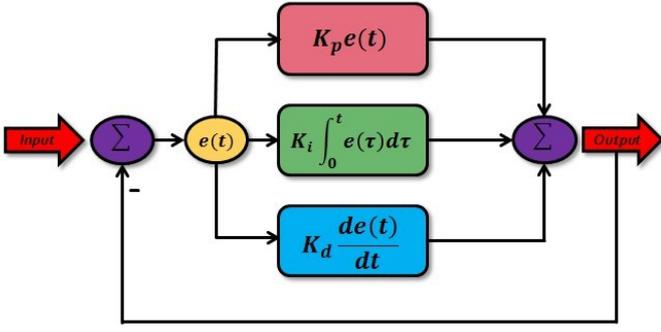


Fig. 4. Steering System

A. PID Control Law

The PID control law has been widely used in the process feed-back control ever since it is first proposed in the early 20 century, because it is simple and highly effective for linear and time-invariant dynamic response system[11]. Fig. 4 shows the framework of PID algorithm. Firstly, we choose to control the steering angle whose mathematic expression is just as same as the direct PID equation:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (1)$$

where $e(t)$ is the D-value of the current state and aim state. By adjust parameters K_p, K_i, K_d , we can achieve an acceptable result. However, taking the D-value of the steering wheel location state may bring some unsatisfiable consequence in our system such as the lag of the stable time, steering speed can not be modified so the exact trajectory could not be estimated and the K_p parameter make the turning speed so fast that the comfortableness can not be guaranteed. So we change the control variable and trying to maintain the steering speed at a constant value. The speed control equations is given by:

$$\Delta V(t) = K_p e_v(t) + K_i \int_0^t e_v(\tau) d\tau + K_d \frac{de_v(t)}{dt} \quad (2)$$

where $\Delta V(t)$ is the increment of the output voltage and $\frac{de_v(t)}{dt}$ is the acceleration a of the steering wheel. This algorithm made it possible for steering speed controlling, and the result is more flexible and stable.

B. State Estimation

As equation. 2 shows that if we want to calculate the increment of the output voltage, a nice estimation of the current speed, location and acceleration is required. Direct approach like measuring the angle value in a fixed sampling frequency was limited to the encoder resolution. Besides, sampling at a constant frequency will bring quantization error too. So, directly acquiring the current speed and acceleration from the differential operation is inadvisable(Fig. 5).

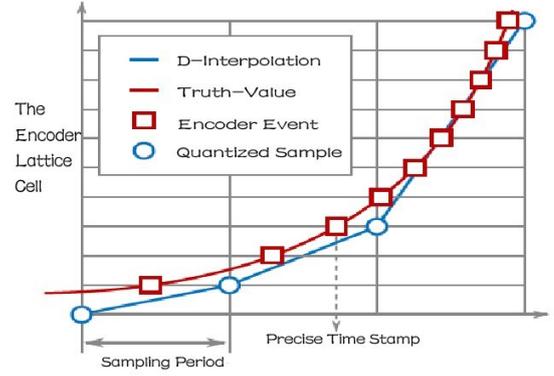


Fig. 5. Encoder State Estimation

There are several approach to improve the estimation accuracy of the encoder, [12] proposed a polynomial fit method to estimate the speed and the acceleration of the encoder, [13] gave another method based on adaptive noise variance kalman filter, [14] suggested an adaptive window hybrid estimation method. While consider the limited computation source of our ECU, we choose the polynomial fit method[12]. When we get the accurate time sequence and the correspond encoder state, we have:

$$P = (p_1 p_2 p_3)^T \quad (3)$$

P is the parameters matrix, p_1, p_2, p_3 are the steering wheel location, steering speed and the steering acceleration respectively. And this matrix is what we need to estimate.

$$X = (x_1 x_2 \dots x_n)^T \quad (4)$$

$$H = \begin{bmatrix} t_1^2 & t_1 & 1 \\ t_2^2 & t_2 & 1 \\ \vdots & \vdots & \vdots \\ t_n^2 & t_n & 1 \end{bmatrix} \quad (5)$$

Matrix X is the encoder value correspond to the matrix H, for example, x_1 is the measurement of encoder value at time t_1 . When $n > 2$, the linear equation is given by:

$$HP = X \quad (6)$$

Simple Least Square method can be used to estimate:

$$P = (H^T H)^{-1} H^T X \quad (7)$$

We choose $n = 5$, which is a compromise to the estimation accuracy and the ECU computational capability. Then we have estimated the location, steering speed and the acceleration at a higher accuracy. So the control system is much more robust and accurate for real-time implementation.

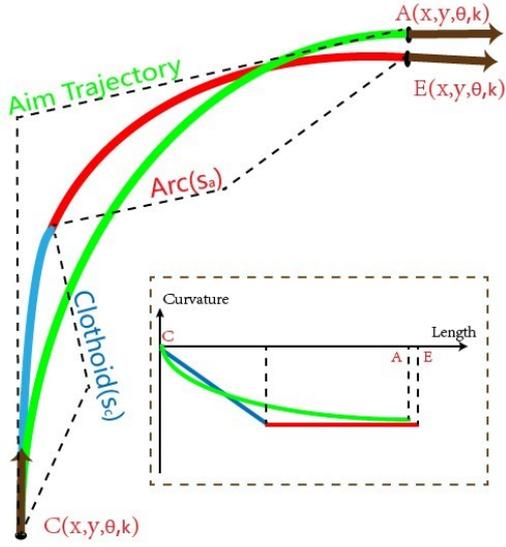


Fig. 6. Tracking Problem

IV. ROBUST TRACKING SYSTEM

This tracking system is well tailored for the steering control system described above. Since the steering wheel could be either holding or uniform rotating, the trajectory of the vehicle is either arc or first-order clothoid. Similar approach is proposed in [15] [16], which request two clothoid and one arc in each section of trajectory and the curvature at the start point and end point of each trajectory section is set to 0 so that the optimize algorithm could be implemented.

In the tracking processes, firstly the planning part generated an aim trajectory, and then send the trajectory to the control system. For a given trajectory Q , each point in the trajectory is a state space:

$$q = \begin{bmatrix} x \\ y \\ \theta \\ k \end{bmatrix} \quad (8)$$

where x, y are the corresponding coordinates and θ, k are the orientation and the curvature of each point. For two points p_1 and p_2 , the D-state $q(p_2 - p_1)$ can be written as a function of path length s :

$$\Delta q = \begin{bmatrix} x(s) \\ y(s) \\ \theta(s) \\ k(s) \end{bmatrix} \quad (9)$$

So we have the current state q_c , and the aim point state q_a . Then an assumption is made that the tracking process can be divided into two courses just as manual driving. Firstly a steering course was made and then holding the steering wheel to achieve the second course so that the trajectory is made of a clothoid and an arc, with the length of s_c and s_a respectively. The end point

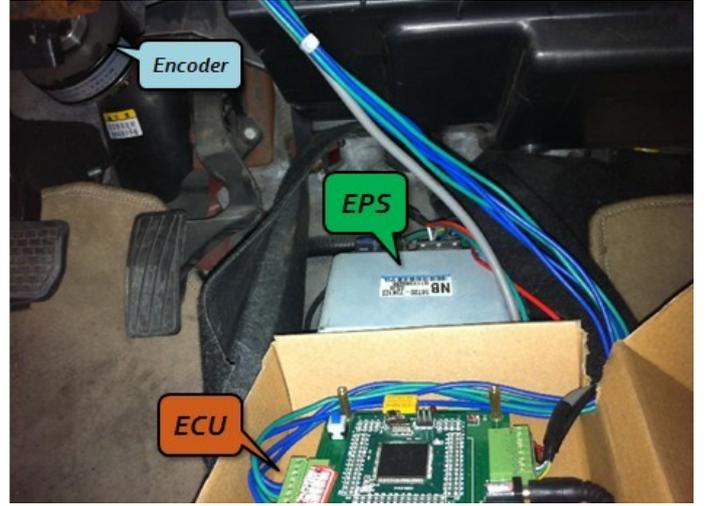


Fig. 7. Vehicle Platform

can be estimated as:

$$q_e = q_c + \begin{bmatrix} x(s_c) \\ y(s_c) \\ \theta(s_c) \\ k(s_c) \end{bmatrix} + \begin{bmatrix} x(s_a) \\ y(s_a) \\ \theta(s_a) \\ k(s_a) \end{bmatrix} = \begin{bmatrix} x_e(b, s_c, s_a) \\ y_e(b, s_c, s_a) \\ \theta_e(b, s_c, s_a) \\ k_e(b, s_c, s_a) \end{bmatrix} \quad (10)$$

where b is the angle rate (steering speed) (degrees per meter), and s_c, s_a correspond to the clothoid and arc. State q_e is given by:

$$x_e = x_c + \int_0^{s_c} \cos(g(b, s_c, s)) ds + \int_0^{s_a} \cos(f(b, s_c, s)) ds \quad (11)$$

$$y_e = y_c + \int_0^{s_c} \sin(g(b, s_c, s)) ds + \int_0^{s_a} \sin(f(b, s_c, s)) ds \quad (12)$$

$$\theta_e = \theta_c + \frac{\alpha(\varphi_c s_c + \frac{1}{2} b s_c^2)}{L} + \frac{\alpha(\varphi_c + b s_c)}{L} s_a \quad (13)$$

$$k_e = \frac{\tan(\varphi_c + b s_c)}{L} \approx \frac{\alpha(\varphi_c + b s_c)}{L} \quad (14)$$

α and φ_c are the Linearization coefficient and current turning angle of the front tyres (degree). and function $g(b, s_c, s), f(b, s_c, s)$ is given by:

$$g(b, s_c, s) = \theta_c + \frac{\alpha}{L} (\varphi_c s + \frac{1}{2} b s^2) \quad (15)$$

$$f(b, s_c, s) = \theta_c + \frac{\alpha}{L} (\varphi_c s_c + \frac{1}{2} b s_c^2) + \frac{\alpha}{L} (\varphi_c + \frac{1}{2} b s_c^2) s \quad (16)$$

Since our goal is to make the end state q_e as closer as possible to the aim state q_a . Then the issue turn out to be an optimization problem with constraint. Our objective function is:

$$C = \operatorname{argmin}_{b, s_1, s_2} (x_e - x_a)^2 + (y_e - y_a)^2 + \sigma(\theta_e - \theta_a)^2 \quad (17)$$

with the constraints:

$$|\varphi_c - b s_c| < \varphi_{max} \quad s_c, s_a \geq 0 \quad (18)$$

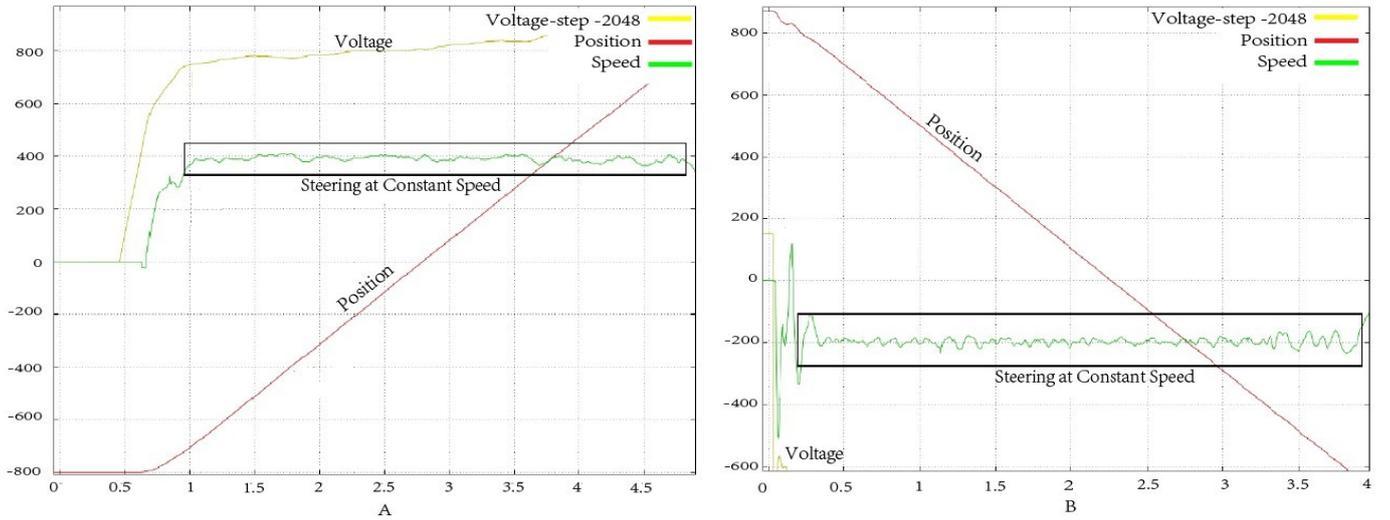


Fig. 8. Speed Control(A:speed=400 B:speed=200)

$$w_{min} \leq |bv| \leq w_{max} \quad (19)$$

where σ is a weight provided to choose whether the location or the orientation to be guaranteed. And φ_{max} is the maximum angle of the front tyres, v is the velocity of the vehicle and w_{min}, w_{max} stand for the minimum and maximum turning speed of the front tyres. For this optimization problem, simple gradient descent algorithm can achieve a good result.

V. EXPERIMENT RESULTS

The experimental platform is essentially a smart car(Suzuki Swift), which is equipped with perception sensors and original EPS system of the car. For automatic steering and tracking, we add a embedded control system in the car, which is showed in Fig. 7,

A Freescale 16-bit processor is chosen to be the central computation unity. Besides, the system have a 12-bit digital to analog converter, electronic relay and some other peripheral circuits. The reserved comport interface can receive the indications from the PC and implement the PID control law which is described above, then output the signal to EPS system and finally steer the wheel automatically.

In this section, we firstly perform the steering system test by giving orders to the embedded system from the PC and recording the feed-back signals of the encoder. Then a trajectory generated with [17] method is given as aim trajectory. Finally the tracking algorithm is performed in the trajectory tracking simulation process.

A. Steering System Test

Fig. 8 is the results of the steering system response to the given indications(speed=400 and 200). It is obviously that the steering speed in both A and B are constant because the *Position* line is visual linear and the *Speed* line is vibrating near a line parallel to the x axis. In fact, the only control variable is the

voltage which is given to the EPS system, and the relationship between the steering speed and the position is :

$$Position = \int (steeringspeed) dt \quad (20)$$

Theoretically the lag of the EPS system should be less than 50ms, but in Fig. 8(A), we can easily discover the problem that there may cost 150 – 200ms delay when the speed begin to increase, which can not be found in Fig. 8(B). Several tests suggest that this issue attribute to the kinematic characters of the car. We conclude a model that satisfy the equation:

$$a = k(T(u) - F_r(x) - F_t(v) - f) \quad (21)$$

at the condition $steeringspeed \neq 0$, if $steeringspeed = 0$, the equation holds only if $a > 0$. The k is constant and $T(u)$ is the output torque of the EPS motor, which is a function of input voltage u . The $F_r(x), F_t(v)$ and f are the aligning torque, tyre torque and frictional force respectively. $F_r(x)$ is a function of steering wheel position x and $F_t(v)$ is a function of car velocity v . The issue mentioned above is mainly caused by the frictional force f because when the car is motionless, the wheel can steer only if the the output force is larger than the static frictional force.

The real lag we should take into consideration is the response time of the steering system. A proper PID parameters could improve the problem but it must be adaptive. The same PID parameters in Fig. 8 exhibit two different response results. Fig. 8(A), the aim speed is 400(encoder units) and the static frictional force exists, so the system seems undershoot. However, in Fig. 8(B) the aim speed is smaller, the same parameters could bring overshoot and cause vibrating. According to our testing, the average response time is less than 250ms, which is an acceptable result when the car velocity is no more than 10m/s.

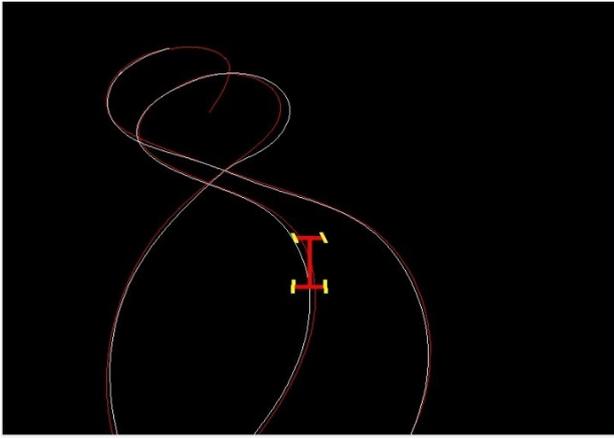


Fig. 9. Real-time Pure Pursuit Tracking

B. Tracking Results

The current tracking algorithm in our system is Pure Pursuit[7]. The look-ahead distance is a function of velocity, the higher the car is, the longer the look-ahead distance. In our experiment, the look-ahead distance should $\leq 3m$, otherwise the real path may vibrate. According our tests, we fitting a equation to describe the relationship between the tyre and the steering wheel:

$$\sigma_t = \frac{aE_s}{1000} + b \quad (22)$$

In spite of the assumption that the mechanism is linear, we find out that the $a = 0.4815 \pm 0.0250, b = 0.0026 \pm 0.0186$. E_s is the encoder feed-back units in the range $(-1000, 1000)$ and the boundary $-1000, 1000$ refer to the max turning angle φ_m . Fig. 9 is the tracking result of the PP algorithm.

In [17] method, a nonholonomic trajectory is generated via parametric iteration. The trajectory is curvature continuous, which is designed for car-like robot. In our simulation, this trajectory is regarded as goal trajectory as Fig. 10 shows, the maximum lateral error is $0.1m \sim 0.3m$ at the constant speed $5m/s$ because the optimal algorithm may converge to local minimum.

VI. FUTURE WORK

In this paper, we provide a simple steering system and the corresponding tracking algorithm. In the experiment and simulation, the results are promising. Since the tracking algorithm have not been added to our platform yet, future work will be concentrated on the combination of the steering and tracking system, and the response lag of the system will also be considered, a more accurate model can be established.

REFERENCES

[1] C. Urmson; et al. "A Robust Approach to High-Speed Navigation for Unrehearsed Desert Terrain[J]," *Journal of Field Robotics*. 2006, 23(8) : 467C508
 [2] Snider J M. "Automatic Steering Methods for Autonomous Automobile path Tracking[R]," Pittsburgh:Robotics Institute, Carnegie Mellon University, 2009.

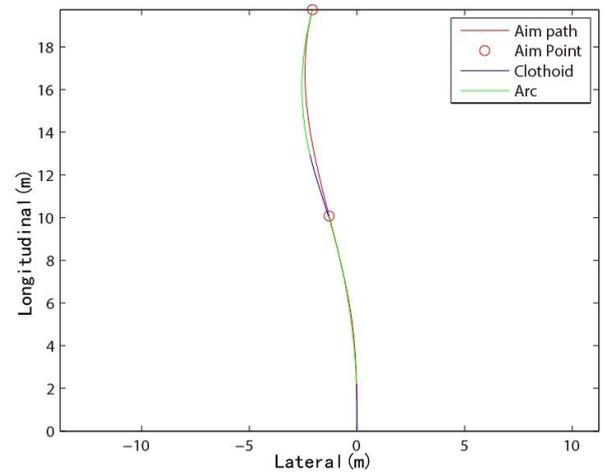


Fig. 10. Tracking Simulation

[3] S. Karaman and E. Frazzoli, "sampling-based algorithms for optimal motion planning," *International Journal of Robotics Research archive Volume 30 Issue 7*, June 2011. Pages 846-894
 [4] M. Likhachev, D. Ferguson, G. Gordon, A. Stentz, and S. Thrun, "Anytime Dynamic A*: An Anytime, Replanning Algorithm," in *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS 05-027)*, 2005.
 [5] D.Brandt, "Comparison of A* and RRT-Connect Motion Planning Techniques for Self-Reconfiguration Planning," *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems October 9-15, 2006*, 892 - 897. Beijing,China.
 [6] R.Mason, J.Radford, D.Kumar; et al. "The Golem Group/University of California at Los Angeles Autonomous Ground Vehicle in the DARPA Grand Challenge," *Journal of Field Robotics*. 2006, 23(8) : 527C553
 [7] O. Amidi. "Integrated mobile robot control," *Carnegie Mellon Univ. Robotics Institute. Technical Report CMU-RI-TR-90- 17*, 1990
 [8] X.Zhao, H.Chen, "Optimal Control of the Automatic Steering System[C]," *IEEE International Conference on Vehicular Electronics and Safety(ICVES)*, 2011. Beijing, China:2011:39-43.
 [9] J. M. Snider, "Automatic Steering Methods for Autonomous Automobile Path Tracking, tech. report CMU-RI-TR-09-08, Robotics Institute, Carnegie Mellon University, Pittsburgh, February, 2009.
 [10] "Manual of DFL vehicle EPS system instructions and maintainance ZhuZhou Elite Electro Mechanical CO., LTD
 [11] Ang, K.H., Chong, G., Li, Y., "PID control system analysis, design, and technology," *IEEE Trans. Control Systems Techn.*, vol. 13, 2005, 559-576.
 [12] R.J.E. Merry; M.J.G Van de Molengraft. "Velocity and Acceleration Estimation for Optical Incremental Encoders[J]," *Mechatronics*. 2010, 20(1):20C26
 [13] S. Wang; S. Wan. "Velocity and Acceleration Computations by Singledimensional Kalman Filter with Adaptive Noise Variance[J]," *Electrical Review*. 2012, 88(2):283C287
 [14] O. Baser; et al. "A Hybrid Method to Estimate Velocity and Acceleration using Low-Resolution Optical Incremental Encoders[C]," *International Conference on Signals and Electronic Systems*. 2010. 57 - 60.
 [15] V. Girbes, L. Armesto, and J. Tornero. "On Generating Continuous-Curvature Paths for Line Following Problem with Curvature and Sharpness Constraints," *IEEE International Conference on Robotics and Automation*. 2011. 6156 - 6161.
 [16] V. Girbes, L. Armesto, and J. Tornero. "Continuous-Curvature Kinematic Control for Path Following Problems," *IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2011. 4335 - 4340 .
 [17] Kelly, A. and Nagy, B. (2003). "Reactive nonholonomic trajectory generation via parametric optimal control," *International Journal of Robotics Research*, 22(7-8): 583C601.