Review of various control algorithms for autonomous vehicles

Aayushi Shrivastava, *Btech, IIT Roorkee*

***Abstract*—This paper reviews various path following and trajectory tracking methods to track a reference point on the path which represents the rate of change of parameter of the trajectory. The aim is to minimize the euclidean distance between the vehicle and above mentioned reference point. Moreover the other concerns would be to reduce the cross-track error and the heading error. The methods like pure pursuit and its variations are described. The Stanley method of path following and other trajectory tracking methods like control Lyapunov Based Design and its modifications are analyzed. This paper has also looked into MPC(Model Predictive Control) and its variations for applying it to trajectory tracking. Simulations were done on kinematic model of car in MATLAB to find the best method to achieve the objective.**

***Keywords*—control system, autonomous vehicle, trajectory tracking, pure pursuit, Lyapunov methods,Model Predictive Control.**

I. Introduction

In the age of automation and robots, self-driving car has become a domestic name. Therefore a lot of research is carried on all over the world. To automate a vehicle three major task are to be done which are perception, path planning, path tracking and low level control. Perception refers to the estimation of the position of car and its surrounding environment using various sensors, cameras and other techniques. The perception system (image,range data, sonars) as well as internal sensors(inclinometers, gyroscopes, acelerometers) and dead reckoning techniques localizes the vehicle in the world. Perception resembles like the senses of the vehicle, to check for obstacles, the position of the car in the surrounding world and its speed acceleration etc. Path planning means finding the best trajectory from initial position to the final point while avoiding collisions with stationary and dynamic obstacles and minimizing the cost. The cost is generally the distance travelled while considering the constraints of road and vehicle. This paper is mainly concerned with path/trajectory tracking. The objective is to follow a path or trajectory by taking into account the actual position and constraints imposed by the vehicle.

Vehicle control refers to the formation of control algorithms to track a reference path or trajectory and minimize the cross-track error and the heading error.The car is a non-holonomic vehicle: it has only two controls, but its configuration space has dimension 3. We can only control the linear velocity and the steering rate of the car by applying control on the throttle, brake and steer. Thus the Bicycle model is the best model which describes the kinematics of the car at slow and medium speeds. At high speed the dynamics of the vehicles play an important role because of lateral slip of the wheels. But for our application we require kinematics model only. Therefore bicycle model described below in the paper is used for all the controllers which are compared.

A. Bicycle model

The car is modelled as a bicycle where the linear velocity of the vehicle v_r is considered at the mid point of the axle of rear wheels[1]. The front wheel has extra degree of freedom. It can rotate about the axis perpendicular to ground plane. This models the steering of the vehicle. We have assumed that there is no slipping between the wheels and the road. Lateral displacement of the vehicle is impossible without the forward movement of the vehicle. The positions of the rear wheel depends on velocity and steering angle as

$$\dot{x_r} = v_r cos\theta$$

$$\dot{y_r} = v_r sin\theta$$

$$\dot{\theta} = w = \frac{v_r}{l} sin\delta$$

According to the non-holonomic constraint

$$v_r = v_f cos\delta$$

Every vehicle has a maximum and minimum turn radius which defines the maximum and minimum steering angle, δ_{max} and δ_{min}. Moreover there is also a limit on the translational velocity of the vehicle ; v_{min} and v_{max}. Generally the output of the controller is heading rate of the vehicle,w which is related to steering angle by the equation-

$$\delta = arctan\Big(\frac{wl}{v_r}\Big)$$

Hence,

$$w \in \Big[\frac{v_r}{l} tan\delta_{min}, \frac{v_r}{l} tan\delta_{max}\Big]$$

Based on this model which also known as ackermann model, the following controllers are analyzed and compared.

II. Path following controllers

Path tracking has been mostly performed with constant velocity. Thus, the path tracking algorithm implements a steering control law by using the error between the current estimated vehicle position/orientation and the path to follow. The inputs of the path tracker are the state of the vehicle ,i.e. position and orientation with respect to the path, and the output is the steering command to be executed by the low-level motion controllers. [4]

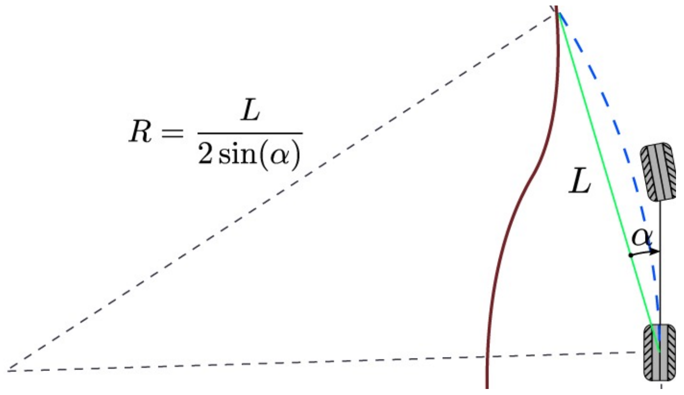

Fig. 1. Pure Pursuit controller

These are generally geometric controllers which exploit geometric relationships between the vehicle and the path, resulting in control law solutions to the path tracking problem. Examples are pure pursuit and rear/front wheel position based feedback.

A. Pure Pursuit Algorithm

In Pure Pursuit Algorithm, a point is chosen on the path which is to be followed at a look-ahead distance from the vehicle centre. The vehicle has to follow this point with constant velocity. The vehicle then moves in a circular arc to reach the look-ahead point. The turning rate of the vehicle, w is given by the control law[1]-

$$w = \frac{2v_r sin\alpha}{L}$$

The Look-ahead Distance is the main tuning parameter for the Pure Pursuit controller. The look ahead distance is how far along the path the robot should look from the current location to compute the angular velocity commands. Smaller look-ahead distance will lead faster convergence towards the path but this would lead to oscillations whereas larger look-ahead distance would lead more cross-track error without any oscillations. A general practice is to tune the look-ahead distance according to the linear velocity of the vehicle. Therefore look-ahead distance is generally scaled with the velocity. Additionally, the look-ahead distance is commonly saturated at a minimum and maximum value. Thus the steering angle becomes

$$\delta(t) = tan^{-}1\Big(\frac{2Lsin\alpha}{kv_r(t)}\Big)$$

Some variations of pure pursuit include Vector pursuit and CF Pursuit. Vector Pursuit takes into consideration orientation of the look-ahead point along with the look-ahead distance in the control law[2]. CF Pursuit used clothoids instead of circular arc for curve fitting so as to reduce the fitting error. It also uses fuzzy for selecting look-ahead distance. The curvature at three points on the curve at 6m, 9m and 12m were taken as the input to fuzzy controller to determine the optimal look-ahead distance[3].

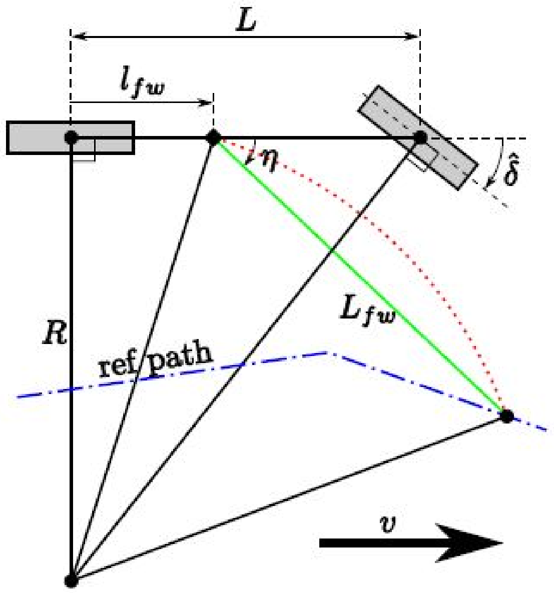

Fig. 2. MIT's Method

B. MIT's method

To increase the stability of the pure pursuit method, MIT used a modified method in 2007 DARPA Grand Challenge. In this method, the original point of local coordinate system is not located at the center of rear wheels; instead its located at the anchor point which at some distance ahead of the vehicle center. Therefore, the computational method of the steering angle is also different from Pure-Pursuit.

$$\delta = tan^{-1}\left(\frac{Lsin\eta}{\frac{L_{fw}}{2} + l_{fw}cos\eta}\right)$$

According to its analysis, this change will improve stability. This has been proved in their paper [5]. Moreover this is the most stable method of all path following algorithms. Another improvement relates to the look-ahead distance. The look-ahead distance will change with the command velocity. PI controller was used for the control of linear velocity. Look-Ahead distance cannot be varied with actual velocity since it varies slowly than the predicted. If the actual vehicle accelerates more slowly than predicted, the time-based command velocity would increase faster than the prediction with respect to the travel distance. The speed command is tied to the position of the vehicle with respect to the predicted path. Then, the steering command depends only on where the vehicle is, and is insensitive to the prediction error of the speed.[9]

$$L_{fw} = \begin{cases} 3 & v_{cmd} < 1.34 \\ 2.24v_{cmd} & 1.34 \leq v_{cmd} \leq 5.36 \\ 12 & otherwise \end{cases}$$

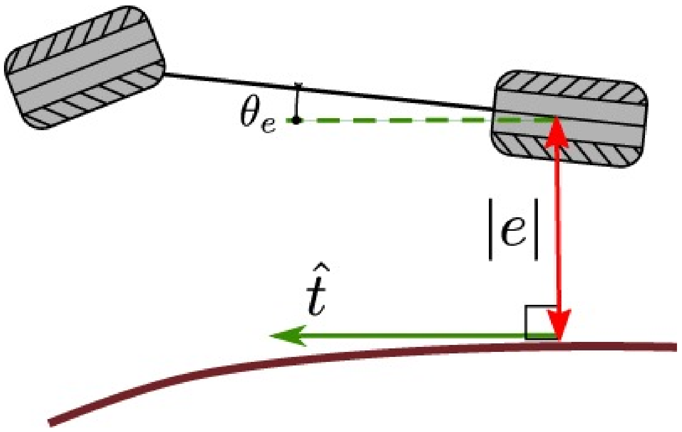

Fig. 3. Rear wheel position based feedback

C. Rear wheel position based feedback

Here, the cross track error or the distance between the vehicle's rear wheel and the point normal to it on the predicted path is used to formulate the control law.[1]

$$\dot{s} = \frac{v_r cos\theta_e}{1-\kappa(s)e}$$

$$\dot{e} = v_r sin\theta_e$$

$$\dot{\theta}_e = w - \frac{v_r\kappa(s)cos\theta_e}{1-\kappa(s)e}$$

s represents the parameter of the point on predicted path which is at the minimum distance from vehicle's rear wheel. θ_e is the heading error calculated by taking the difference of the heading of the vehicle and orientation at the s . κ represents the curvature of the path at s. Thus the control law becomes

$$w = \frac{v_r\kappa(s)cos\theta_e}{1-\kappa(s)e} - \big(k_\theta|v_r|\big)\theta_e - \Big(k_e v_r \frac{sin\theta_e}{\theta_e}\Big)e$$

k_e and k_θ are tuning parameters. An advantage of this control law is that stability is unaffected by the sign of v_r making it suitable for reverse driving. This leads to local exponential convergence with a rate independent of the vehicle speed so long as v_r = 0.

D. Stanley's Method

This approach was proposed and used in Stanford Universitys entry to the 2005 DARPA Grand Challenge. The approach is to take the front wheel position as the regulated variable. The control uses the variables $s(t)$, $e(t)$, and $\theta_e(t)$ as in the previous section, with the modification that e(t) is computed with the front wheel position as opposed to the rear wheel position.[6] The steering angle is given by

$$\delta = arcsin\big(-ke/v_f\big) - \theta_e$$

Thus, the error converges exponentially to zero. The parameter k determines the rate of convergence. Though this system works well for high speed but is least stable method. It is oscillatory and is not suitable for reverse driving.

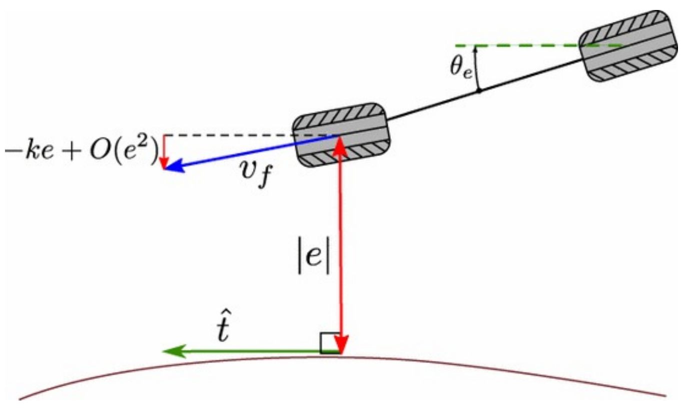

Fig. 4. Stanley's Method

III. Trajectory Tracking Algorithms

The following methods are generally based on control Lyapunov functions. Input to the vehicle are a reference posture $(x_r, y_r, \theta_r)'$ and reference velocities $(v_r, w_r)'$. The approach is to follow a point on reference path which has linear and angular velocity and hence changes with time.The point to be followed here is the $\dot{s}$ point where s represents the parameter of trajectory. Therefore the reference state/posture of the point is defined in terms of x_r, y_r and θ_r ; where θ_r is given by

$$\theta_r = tan^{-1}\frac{\dot{y}}{\dot{x}}$$

The linear and angular velocity of the reference point is defined by

$$v_r = \sqrt{\dot{x}^2 + \dot{y}^2}$$

$$w_r = \frac{\dot{x}\ddot{y} - \dot{y}\ddot{x}}{\dot{x}^2 + \dot{y}^2}$$

where v_r denotes the forward velocity, w_r corresponds to the angular velocity.[7]
The state or posture of the vehicle is given (x, y, θ). The linear and angular velocity of the vehicle is given by (v, w).

$$\dot{x} = vcos\theta$$

$$\dot{y} = vsin\theta$$

$$\dot{\theta} = w$$

We transform the error coordinates $[x_e, y_e, \theta_e]$ of the reference posture from the global coordinate frame to local coordinates fixed on the robot that is,

$$\begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} = \begin{bmatrix} cos\theta & sin\theta & 0 \\ -sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix}$$

Correspondingly,we obtain the following error dynamics

$$\dot{x_e} = wy_e - v + v_r cos\theta_e$$

$$\dot{y_e} = -vx_e + v_r sin\theta_e$$

$$\dot{\theta}_e = w_r - w$$

Thus we need to find controller in the form of

$$[v\ w]' = f(x_e, y_e, \theta_e, v_r, w_r)$$

such that for all initial states $[x_e(t_0)\ y_e(t_0)\ \theta_e(t_0)]'$ $\in R^3, \forall\, t_0 \geq 0,\ x_e(t)$ and $y_e(t)$ are uniformly bounded, and

$$\lim_{t\to\infty} x_e(t) = 0, \quad \lim_{t\to\infty} y_e(t) = 0, \quad \lim_{t\to\infty} \theta_e(t) = 2K\pi$$

A. Kanayama's Lyapunov Control Method

The control law is given by

$$v = v_r cos\theta_e + K_x x_e$$

$$w = w_r + v_r(K_y y_e + K_\theta sin\theta_e)$$

where K_x, K_y and K_θ are positive constants. The stability of the system by above control law is proved using Lyapunov's Direct Method. Kanayama and others have demonstrated that the system is stable for any combination of parameter values of $K_x, K_y,$ and K_θ. A larger K_x makes convergence faster and reduces a steady error x_e. However, it is not appropriate to have a time constant 1/K , comparable to the sampling time of the robots hardware. K_y and K_θ are determined by solving the error equation for critical damping condition.[7]

They adopted a simple algorithm of limiting the target velocities($v,\ w$) by constants (v_{max}, w_{max}) and the target accelerations (a, α) by constants (a_{max}, α_{max}) , where $a = \dot{v}$ is linear target acceleration and $\alpha = \dot{w}$ rotational target acceleration.

B. Lyapunov Method with Velocity Constraints

The control law is given by

$$v = v_r + \frac{c_1 x_e}{\sqrt{1 + x_e^2 + y_e^2}}$$

$$w = w_r + \frac{c_2 v_r(y_e cos\frac{\theta_e}{2} - x_e sin\frac{\theta_e}{2})}{\sqrt{1 + x_e^2 + y_e^2}} + c_3 sin\frac{\theta_e}{2}$$

where c_1, c_2 and c_3 are positive constants.The salient features of the proposed controller are as follows.First, it is simply based on feedback of the tracking errors in vehicles Frenet-Serret frame. Second, all or part of aforementioned velocity constraints can be satisfied by tuning of the design parameters. Third, the reference angular velocity is allowed to be piecewise continuous.

There are two assumptions[8] -

A1 $v_r(t)$ and $w_r(t)$ are bounded, i.e.,

$$v_r^{min} \leq v_r(t) \leq v_r^{max},$$

$$w_r^{min} \leq w_r(t) \leq w_r^{max}, \forall t \geq 0.$$

where $v_r^{min} > v^{min}, v_r^{max} < v_{max}$, and $w_r^{min} > w^{max}, w_r^{max} < w_{max}$.

A2 $v_r(t)$ is uniformly continuous in t.

Most current results included the term $v_r cos\theta_e$ in the controller design of the linear velocity v. However, this term makes the linear velocity v difficult to always satisfy the velocity constraint. Hence the above law was proposed.

C. Lyapunov-based control under persistency of excitation

The control law is given by

$$v = v_r cos\theta_e + K_x x_e$$

$$w = w_r + K_\theta \theta_e + v_r k_y y_e \phi(\theta_e)$$

where ϕ is the so-called sync function defined by

$$\phi(\theta_e) = \frac{sin(\theta_e)}{\theta_e}$$

The standing assumption is that either the forward or the angular reference velocities are persistently exciting.[8]

The main contribution of the paper is twofold: first, they establish uniform global asymptotic stabilization in the context of formation-tracking control under weak assumptions; secondly ,according to them, that was this controller is the first one which is a strict Lyapunov function proposed in the context of control of non-holonomic systems under a persistently excited reference velocities.[10]

The controller ensures uniform global asymptotic stability under a simple condition of persistency of excitation of either of the reference velocities, forward or angular. In particular, the controller applies to the difficult problem of following straight paths: null angular velocity and constant forward velocity.

D. z coordinate transformation Lyapunov function

let us consider the following change of coordinates and control variables $(x_e, y_e, \theta_e, v, w) \rightarrow (z_1, z_2, z_3, u_1, u_2)$ defined by

$$z_1 = x_e,$$

$$z_2 = y_e,$$

$$z_3 = tan\theta_e,$$

$$u_1 = vcos\theta_e - v_r,$$

$$u_2 = \frac{w - w_r}{cos^2\theta_e}$$

. Note that, around zero, this mapping is only defined when $\theta_e \in (\pi/2, \pi/2)$. In other words, the orientation error between the physical robot and the reference robot has to be smaller than $\pi/2$.

The linearization yields the simpler control law[11]

$$u_1 = -k_1|v_r|z_1,$$

$$u_2 = -k_2 v_r z_2 - k_3|v_r|z_3.$$

Since $u_1 \approx v - v_r$ and $u_2 \approx w - w_r$ near the origin, it is legitimate to wonder if the control

$$v = v_r - k_1 |v_r| x_e$$

$$w = w_r - k_2 v_r y_e - k_3 |v_r| tan\theta_e.$$

In fact, it is not difficult to verify, via a classical pole-placement calculation, that this control asymptotically stabilizes the origin of the linear system which approximates the system when v_r and w_r are constant, with $v_r \neq 0$.The control is in fact designed so that its tuning for the specific velocities $v_r = 1$ and $w_r = 0$ gives good results for all other velocities.

IV. Simulation and Results

The simulations of the four trajectory tracking methods were done in MATLAB. The trajectory to be followed was lissajous curve; eight figure defined by $x = acos(s)$ and $y = bsin(2s)$ where s is the parameter. The angular velocity of the vehicle is constrained with $\delta_{max} = 30°$ that is the maximum steering angle of the vehicle. The initial states are $[x_r(0)\ y_r(0)\ \theta_r(0)]' = [30.0\ 0.0\ \pi/2]'$ and $[x(0)\ y(0)\ \theta(0)]' = [30.0\ -5.0\ \pi/2]'$.

The cross-track error , heading error and tracking error was calculated. The cross-track error is defined as the minimum distance from the vehicle center to the path to be followed. Heading error refers to the difference between the heading of the vehicle and the angle of the tangent at the point to be followed. Whereas tracking error refers to the euclidean distance between the vehicle and the tracking point to be followed($\dot{s}$ point in our case).

A. Kanayama's Lyapunov Control Method

$$v = v_r cos\theta_e + K_x x_e$$

$$w = w_r + v_r(K_y y_e + K_\theta sin\theta_e)$$

By applying the above control law with the parameters $K_x = 20, K_y = 0.1$ and $K_\theta = 1.0$, we have the simulation results shown in Fig 5 - Fig 7. Figure 5 shows that the tracking errors. Fig 6 shows that the velocities v(t) and w(t). Figure 7 shows the reference trajectory and trajectory of the vehicle. These results show that the proposed controller is effective since the error is minimum.
From the results it can be seen that the errors are minimum except where the curvature is more. The general ratio of K_x and K_θ is approximately 20:1 and that of K_θ and K_y is approximately 10:1 . Larger the value of K_x faster the attainment of zero error(steady state). K_y and K_θ helps in reducing oscillations and the time constant of the system. The controller is effective in tracking the path with time varying linear and angular velocity.

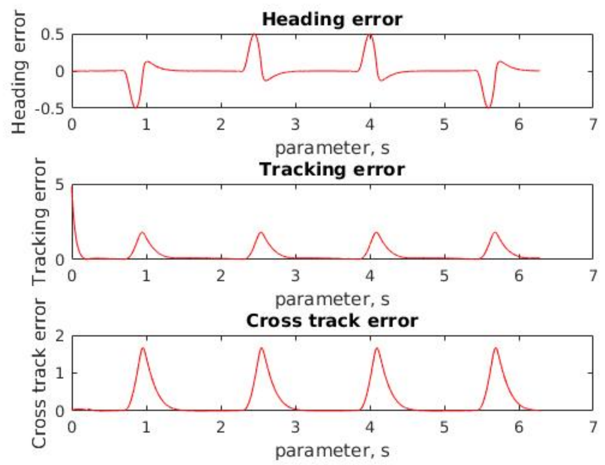

Fig. 5. A) Kanayama's Lyapunov Control Method : Tracking Errors

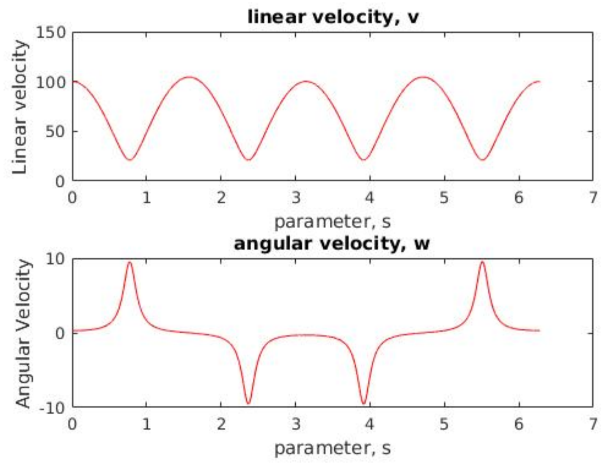

Fig. 6. A) Kanayama's Lyapunov Control Method : Linear and Angular Velocity

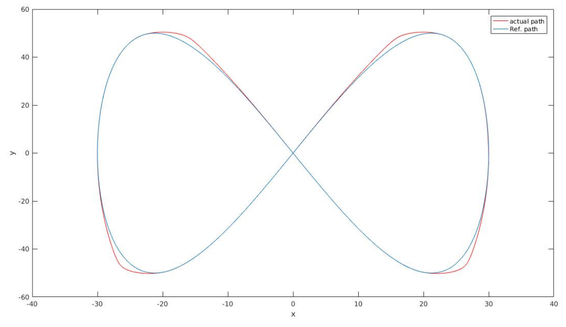

Fig. 7. A)Kanayama's Lyapunov Control Method : Trajectory

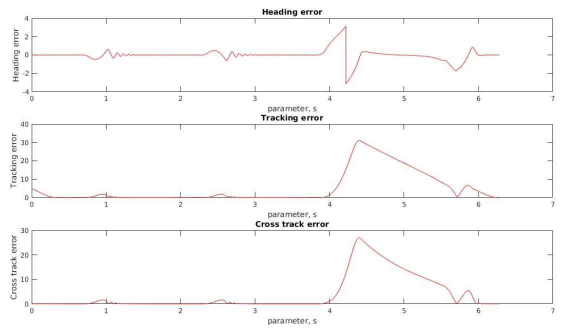

Fig. 8. B) Lyapunov Method with Velocity Constraints : Tracking Errors

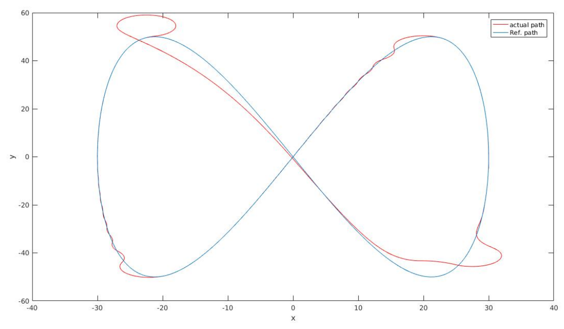

Fig. 10. B) Lyapunov Method with Velocity Constraints : Trajectory

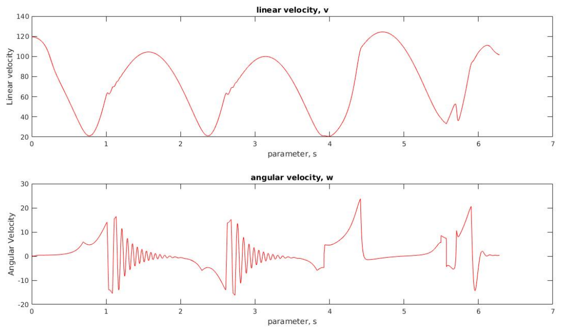

Fig. 9. B) Lyapunov Method with Velocity Constraints : Linear and Angular Velocity

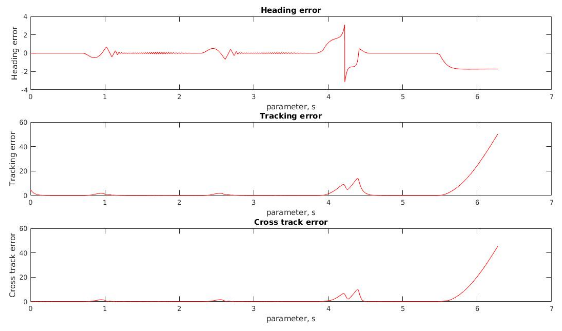

Fig. 11. C) Lyapunov-based control under persistency of excitation : Tracking Errors

B. Lyapunov Method with Velocity Constraints

$$v = v_r + \frac{c_1 x_e}{\sqrt{1 + x_e^2 + y_e^2}}$$

$$w = w_r + \frac{c_2 v_r (y_e cos\frac{\theta_e}{2} - x_e sin\frac{\theta_e}{2})}{\sqrt{1 + x_e^2 + y_e^2}} + c_3 sin\frac{\theta_e}{2}$$

By applying the above control law with the parameters $c_1 = 20, c_2 = 1.0$ and $c_3 = 40.0$, we have the simulation results shown in Fig 8 - Fig 10. Figure 8 shows that the tracking errors. Fig 9 shows that the velocities v(t) and w(t). Figure 10 shows the reference trajectory and trajectory of the vehicle. The figures show that the control law is ineffective in tracking the path with high curvature.

C. Lyapunov-based control under persistency of excitation

$$v = v_r cos\theta_e + K_x x_e$$

$$w = w_r + K_\theta \theta_e + v_r K_y y_e \phi(\theta_e)$$

$$\phi(\theta_e) = \frac{sin(\theta_e)}{\theta_e}$$

By applying the above control law with the parameters $k_x = 20.0,\ k_y = 5.0$ and $k_\theta = 40.0$, we have the simulation results shown in Fig 11 - Fig 13. Figure 11 shows that the tracking errors. Fig 12 shows that the velocities v(t) and w(t). Figure 13 shows the reference trajectory and trajectory of the vehicle. From the figure it can be observed that the errors accumulate over the path and the vehicle never reaches its destination. This is mainly due to the extra path travelled at one of the higher curvature turns. The controller works properly for small curvature path but has difficulty on high curvature. Moreover the control law is invalid when θ_e is zero. The function ϕ becomes invalid in that case.

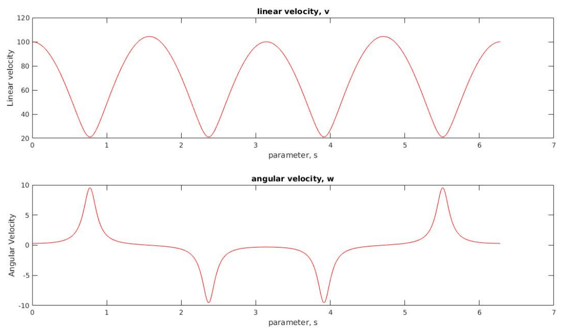

Fig. 12. C) Lyapunov-based control under persistency of excitation : Linear and Angular Velocity

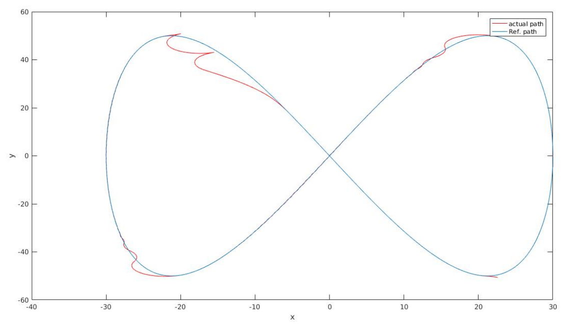

Fig. 13. C) Lyapunov-based control under persistency of excitation : Trajectory

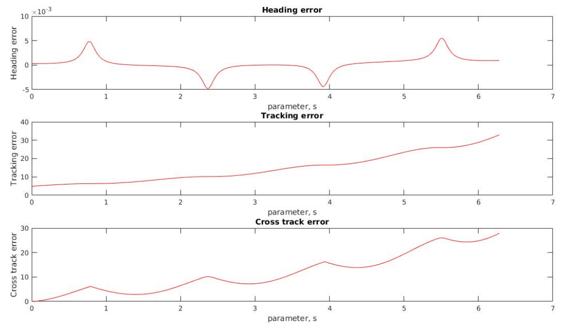

Fig. 14. D) z coordinate transformation Lyapunov function : Tracking Errors

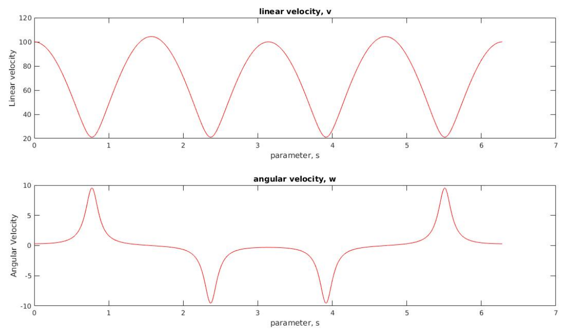

Fig. 15. D) z coordinate transformation Lyapunov function : Linear and Angular Velocity

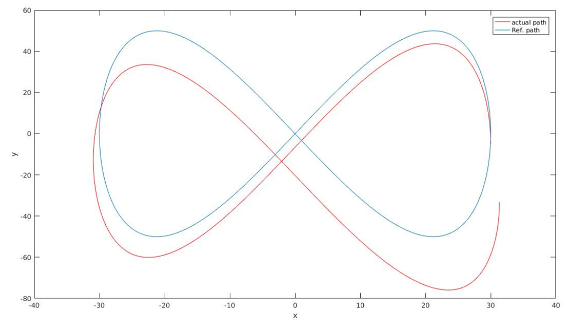

Fig. 16. D) z coordinate transformation Lyapunov function : Trajectory

D. z coordinate transformation Lyapunov function

$$v = v_r - k_1|v_r|x_e$$

$$w = w_r - k_2 v_r y_e - k_3|v_r|tan\theta_e.$$

By applying the above control law with the parameters $k_1 = 0.005, k_2 = 0.0$ and $k_3 = 0.005$, we have the simulation results shown in Fig 14 - Fig 16. Figure 14 shows that the tracking errors. Fig 15 shows that the velocities v(t) and w(t). Figure 16 shows the reference trajectory and trajectory of the vehicle.
As it is clear from the figures that the controller could not track the path since this controller is designed for constant linear and angular velocities whereas in our case both the velocities are time-varying. Therefore the cross-track error and tracking error never becomes zero.

Various simulations for all the control laws showed that most of the control law fail at the high curvature paths. We can conclude from the above discussion that Kanayam's Lyapunov Method control law is the best among these controllers since the errors are minimum. One drawback of these controllers is that the linear velocity has a large value in our case and by putting constraint on the linear velocity leads to poor tracking. The stabilty of all the control law is proved by Lyapunov Direct Method and Barbalats Lemma.

V. Conclusion

Acknowledgment

References

[1] B. Paden, M. p, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on Intelligent Vehicles,* vol. 1, no. 1, pp. 3355, March 2016.

[2] J. Wit, C. D. Crane, and D. Armstrong, "Autonomous ground vehicle path tracking," *Journal of Robotic Systems,* vol. 21, pp. 439-449, 2004.

[3] A. Ollero and G. Heredia, "Stability analysis of mobile robot path tracking,"in *Intelligent Robots and Systems* 95.*'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference* on, 1995, pp. 461-466.

[4] M. Samuel, M. Hussein, and M. B. Mohamad, "A review of some pure-pursuit based path tracking techniques for control of autonomous vehicle," *International Journal of Computer Applications,* 2016.

[5] Kuwata Y, Teo J, Karaman S, Fiore G, Frazzoli E, How J P. " Motion planning in complex environments using closed-loop prediction." *In Proceedings of AIAA Guidance, Navigation, and Control Confer ence and Exhibit (AIAA 2008)*; August 2008; Honolulu. 2008.

[6] M. D. Ventures, "Stanley: The robot that won the DARPA Grand Challenge," J. Field Robot., vol. 23, pp. 661692, 2006.

[7] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, "A stable tracking control method for an autonomous mobile robot," in *Proc. Int. Conf.Robot. Autom.,* 1990, pp. 384389.

[8] Z.-P. Jiang and H. Nijmeijer, ”Tracking control of mobile robots: A case study in backstepping,” *Automatica,* vol. 33, pp. 13931399, 1997.

[9] X. Yu, L. Liu, and G. Feng, ”Trajectory tracking for nonholonomic vehicles with velocity constraints,” *IFAC PapersOnLine,* vol. 48, no. 11,pp. 918923, 2015.

[10] Maghenem, M., Loria, A., Panteley, E.: Lyapunov-based ”Formation-Tracking Control of Nonholonomic Systems under Persistency of Excitation.” *10th IFAC Symposium on Nonlinear Control Systems (NOLCOS 2016),* Monterey, CA, United States (2016)

[11] 0. Khatib B. Siciliano. *Springer handbook of robotics.* Springer, 2008.

[12] J. Leonard, D. Barrett, J. How, S. Teller, M. Antone, S. Campbell, A. Epstein, G. Fiore, L. Fletcher, E. Frazzoli, A. Huang, T. Jones, O. Koch, Y. Kuwata, K. Mahelona, D. Moore, K. Moyer, E. Olson, S. Peters, C. Sanders, J. Teo, and M. Walter, ”Team MIT urban challenge technical report,” *MIT Computer Science and Artificial Intelligence Laboratory,* Tech. Rep., 2007.

[13] C. Katrakazas, M. Quddus, W.-H. Chen, L. Deka ”Real-time motion planning methods for autonomous on-road driving: state-of-the-art and future research directions” *Transport. Res.* Part C: Emer. Technol., 60 (2015), pp. 416-442

[14] D. Gonzalez, J. P erez, V. Milan es, and F. Nashashibi, ”A review of motion planning techniques for automated vehicles,” *IEEE Trans. Intell. Transp. Syst.*, vol. 17, pp. 11351145, 2016.