Stabilize System for Autonomous Vehicle/Wheel Robot

Xiangzhuo Ding[1] and Xupeng Ai[2]

***Abstract*— In this paper, we designed a stabilize system for autonomous vehicle or wheel robot. First, a 2-D dynamic vehicle model is built. In order to test the model's reliability, several experiments are conducted. These tests also guarantee that the model can perform complex motions, such as turning, obstacle avoiding and drift. Then, a PD control law is applied to this model in order to restore stable after tire burst. Obstacle avoiding control and curve trajectory control are also added to test if the control system can handle such complex and critical situations.**

I. INTRODUCTION

Autonomous vehicle control system is an integral part of intelligent vehicle control system. However, due to some accidents during the road test of autonomous vehicles, people are raising more and more concerns about the safety issue[1]. In other words, if we want to fully trust our computer to drive, it must be able to handle unexpected and critical situations. However, many researchers are still focusing on the algorithm level, for instance, how to identify pedestrians in advance through monitoring system (e.g., computer vision) and immediately evade. The traditional safety problem has been ignored, such as tire puncture, which is an unexpected problem when vehicle driving in the high way. In this paper, this problem is identified and PD control law is applied to make the vehicle controllable after tire burst occur. Many different situations are also simulated to test the system's obstacle avoiding ability. The results show our stabilize control system design can help vehicle model control direction after the tire burst.

II. DYNAMIC MODEL

In this section the dynamic model of the vehicle are built using Newton's Law. To simplify the model, aerodynamic is ignored and we only consider the forces from wheels.

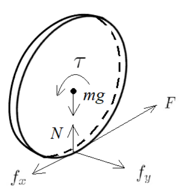

Fig. 1. Wheel model

As is shown in Fig. 1, the active force F is generated by engine and brake. It is evident that F is linearly dependent on the torque τ applied to the wheel, namely,

$$F = \frac{\tau}{r} \tag{1}$$

The vector f_x results from the rolling resistance and the vector f_y denotes the lateral reactive force. The model of friction is quite complicated since it is highly nonlinear and depends on many variables. By (Kozlowski et al., 2004), the following approximation of this equation is proposed[2]:

$$f_x = -\mu_1 mg\widehat{\mathrm{sgn}}(V_x) \tag{2}$$

$$f_y = -\mu_2 mg\widehat{\mathrm{sgn}}(V_y) \tag{3}$$

where,

$$\widehat{\mathrm{sgn}}(\sigma) = \frac{2}{\pi}\arctan(k\sigma) \tag{4}$$

μ_1 and μ_2 denote the coefficients of the rolling resistance and lateral friction respectively. V_x and V_y denote the wheel's forward velocity and transverse slip velocity. Eq. (4) is to make sure that when the model is static, its friction force is zero and when $k\sigma \to \infty$, $\widehat{\mathrm{sgn}}(\sigma) \to 1$, which means that the friction force is constant.

To consider the dynamic model of a vehicle robot, it is assumed that the robot is placed on a plane surface with the inertial orthonormal basis (X,Y,Z). A local coordinate frame denoted by (x,y,z) is assigned to the robot at its center of mass(COM). Since in this paper only the plane motion is considered, the $Z-$coordinate of COM is constant ($Z = const$).

Suppose that the robot moves on a plane with linear velocity expressed in the local frame as $v = \begin{bmatrix} v_x & v_y & 0 \end{bmatrix}^T$ and rotates with an angular velocity vector $\omega = \begin{bmatrix} 0 & 0 & \omega \end{bmatrix}^T$. If $q = \begin{bmatrix} X & Y & \theta \end{bmatrix}^T$ is the state vector describing generalized coordinates of the robot (i.e., the COM position and the orientation θ of the local coordinate frame with respect to the inertial frame), then $\dot{q} = \begin{bmatrix} \dot{X} & \dot{Y} & \dot{\theta} \end{bmatrix}^T$ denotes the vector of generalized velocities.

From Fig. 2, it can be noted that the variables $\dot{X}$ and $\dot{Y}$ are related to the coordinates of the local velocity vectors as follows:

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} \tag{5}$$

Until now, the coordinate frames are set and the force analysis of one steering wheel on its own coordinate frame is completed. Based on Eq. (5) we can get:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}^{-1} \begin{bmatrix} \dot{X} \\ \dot{Y} \end{bmatrix} \tag{6}$$

*MECE 6614 Final Project

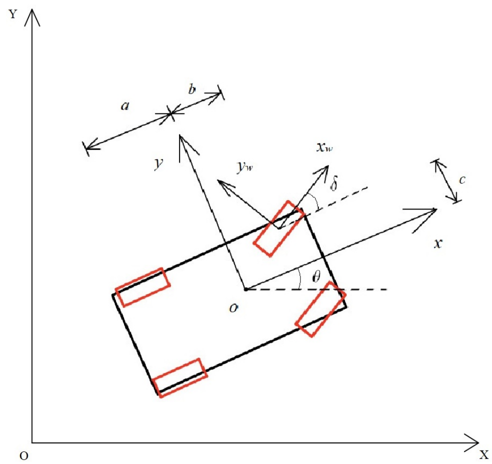

Fig. 2. Coordinate frames of the model

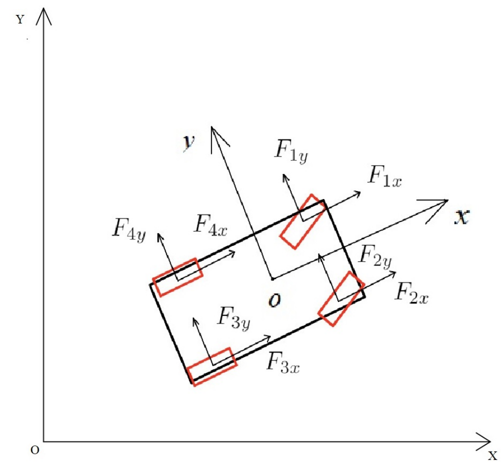

Fig. 4. Force analysis of the dynamic model

Eq. (6) gives the velocity of vehicle's COM in local frame. From that relationship we can get four wheels' forward velocity and transverse slip velocity which are shown in Eq. (7). $V_{1,2,3,4}$ denote the vehicle's left front, right front, right rear and left rear wheel's velocity respectively. Then friction forces can be calculated by Eq. (2) and Eq. (3).

$$\begin{bmatrix} V_{ix} \\ V_{iy} \end{bmatrix} = \begin{bmatrix} \cos\delta & -\sin\delta \\ \sin\delta & \cos\delta \end{bmatrix}^{-1} \begin{bmatrix} \dot{x} \pm \dot{\theta} c \\ \dot{y} \pm \dot{\theta} b \end{bmatrix} \tag{7}$$

The force distribution of left front steering wheel is shown in Fig. 3. It is obvious that the force vector for left front

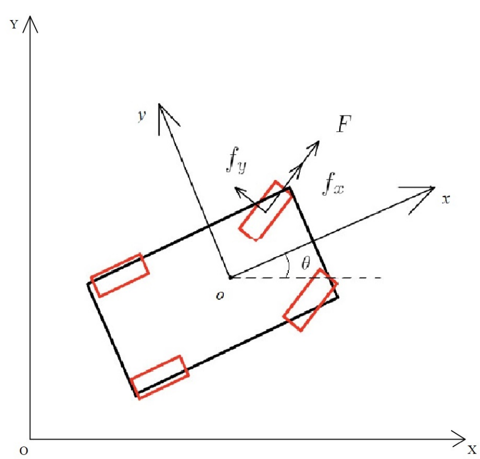

Fig. 3. Force analysis of the left front wheel

wheel is $\begin{bmatrix} F_1 + f_{1x} & f_{1y} \end{bmatrix}^T$. Based on the rotation matrix, each wheel's force distribution can be calculated by Eq. (8) and shown in Fig. 4.

$$\begin{bmatrix} F_{ix} \\ F_{iy} \end{bmatrix} = \begin{bmatrix} \cos\delta & -\sin\delta \\ \sin\delta & \cos\delta \end{bmatrix}^{-1} \begin{bmatrix} F_i + f_{ix} \\ f_{iy} \end{bmatrix} \tag{8}$$

According to the relationship between moment, angular acceleration and rotation inertia I, it is evident that

$$I\ddot{\theta} = (F_{2x} + F_{3x} - F_{1x} - F_{4x})c + (F_{1y} + F_{2y})b - (F_{3y} + F_{4y})a \tag{9}$$

The right side is the total moment around vehicle's COM.

Based on Newton's second law of motion, each wheel's accelerations in the body frame are:

$$\ddot{x} = \frac{\sum F_i x}{m} \tag{10}$$

$$\ddot{y} = \frac{\sum F_i y}{m} \tag{11}$$

Eq. (5) gives the robot's velocity in world frame. Taking the partial derivative over time, one can obtain the acceleration in the word frame.

$$\begin{bmatrix} \ddot{X} \\ \ddot{Y} \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \ddot{x} - \dot{y}\dot{\theta} \\ \ddot{y} + \dot{x}\dot{\theta} \end{bmatrix} \tag{12}$$

III. TIRE BURST SIMULATION

When the tire puncture occurs, there will be some chain reaction to the whole vehicle. To simplify the calculation, some approximation are made. It is assumed that our models left front wheel is punctured. When the puncture occurs, the coefficient of rolling resistance and the coefficient of friction of the punctured tire will increase significantly. Also we assume that the steering wheel will turn around 60 degree for 0.2 second. If the brakes do not work, the vehicle will instantly change direction and go off the road, which is shown in Fig. 5. However, if we apply full brake, we will completely lose control. The vehicle will deviate and dramatically change the moving direction. As is shown in Fig. 6, it is even more dangerous due to the potential risk of roll over.

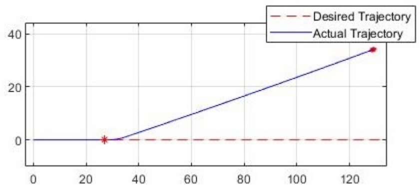

Fig. 5. The trajectory when no action is taken

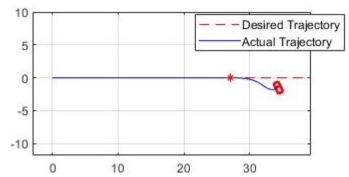

Fig. 6. The trajectory when full brake is applied

IV. CONTROL LAW

A. Stability Control

When a tire blown, a PD control system will intervene. The input is the steering angle δ of the front wheels. There are two errors needed to be controlled (Fig. 7), e_1 is the distance between COM and x axis in the world coordinate frame, where x axis is the desired trajectory. The other error e_2 is the angle between orientation of the vehicle and x axis. We use Eq. (13) to control the system where $k_{p_{1,2}}$ and $k_{d_{1,2}}$ are P control and D control gain respectively.

$$\delta = k_{p_1} e_1 + k_{p_2} e_2 + k_{d_1} \frac{\mathrm{d}e_1}{\mathrm{d}t} + k_{d_2} \frac{\mathrm{d}e_2}{\mathrm{d}t} \tag{13}$$

The actual and desired trajectories is shown in Fig. 8. Fig. 9 shows the wheel robot's velocity over time.

B. Obstacle Avoiding

A distance sensor is placed in front of the vehicle. It can detect obstacles within 20 meters in front of it. To simplify the calculation, we assume that the obstacle is a cube placed directly in front. The distance sensor can also detect the distance that needs to be slide off the original trajectory (Fig. 10). We still use PD control to do the obstacle avoiding task. However, in this situation, the distance error is length between COM and obstacle's width plus vehicle's width in the world coordinate frame. After avoiding the obstacle, the

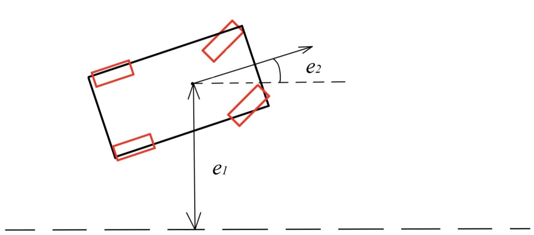

Fig. 7. Control law

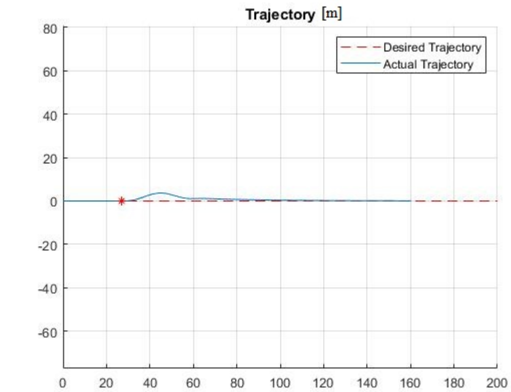

Fig. 8. Trajectory of stability control

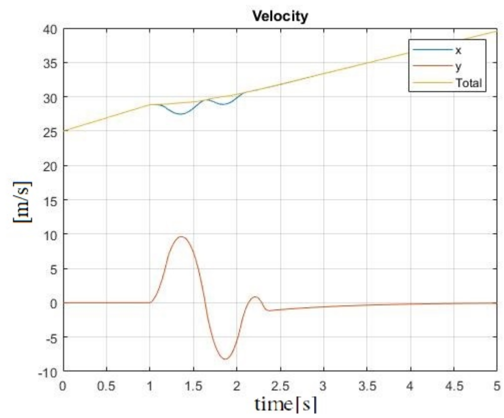

Fig. 9. Velocity-Time of stability control

model's desired trajectory is set to back to the original one. The desired trajectory is shown in Fig. 11 and Fig. 12 gives the velocity-time plot.

C. Curve Trajectory Control

Considering that after a tire of a vehicle has a blown out, the road ahead may not be straight. The vehicle must be able to turn without rushing out of the road. We designed a curve trajectory, which is set to be the vehicles desired moving path. First, the curve trajectory is sampled to several discrete points. Among all this points, one error is the distance between the vehicles COM and the closest sample point to it. The other one is the angle between the orientation of the vehicle and the direction between the chosen point and the

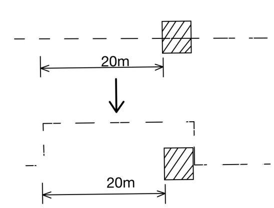

Fig. 10. Obstacle detection and trajectory altering

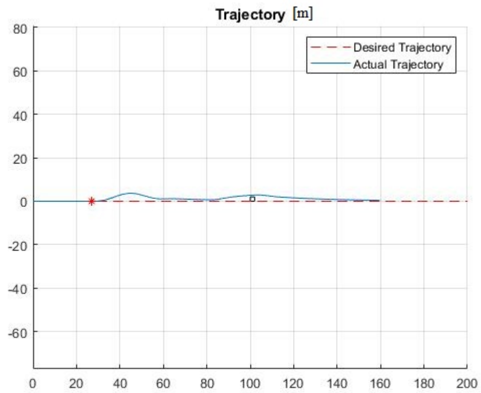

Fig. 11. Trajectory of obstacle avoiding

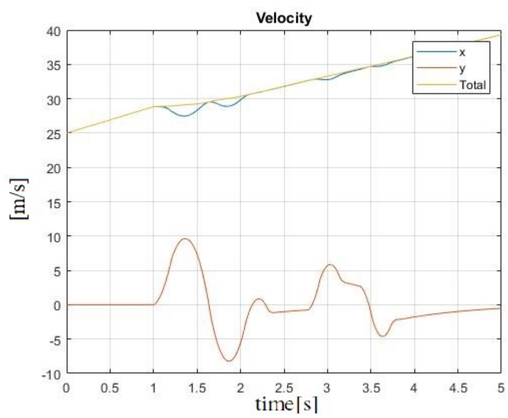

Fig. 12. Velocity-time of obstacle avoiding

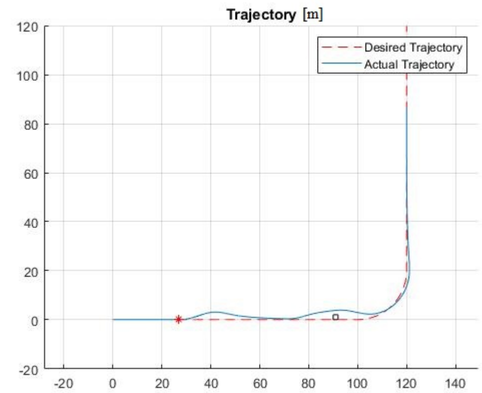

Fig. 14. Curve trajectory

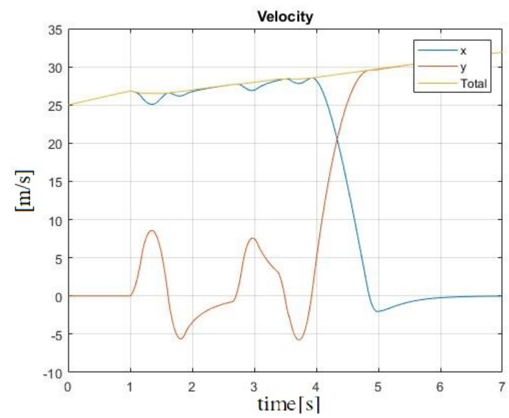

Fig. 15. Velocity-time of curve trajectory

point next to it (Fig. 13). Fig. 14 and Fig. 15 shows the performance of this control algorithm.

V. RESULTS

By analyzing velocity-time plots above, namely Fig. 9,12 and 15, it is evident that when a disturbance (i.e., tire burst) is introduced to the system, the system can restore stability in a very short time, which is just 2-3 seconds in our simulation. It means that the applied control law is appropriate and effective. Besides, when receiving feedback signals from distance sensor, our model can conduct complex operations like obstacle avoiding and curve trajectory control.

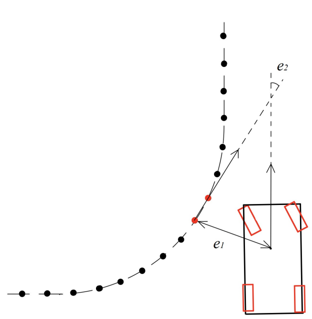

Fig. 13. Curve trajectory control

VI. CONCLUSIONS

In this paper, a dynamic model of autonomous vehicle or wheel robot is built and its reliability is tested. We have confirmed that it is accurate enough to implement control theory, which is designed to help vehicle remain stable after unexpected incident. We tested our system in several situations and the results are promising. We also provide a curve trajectory control method which in theory are suitable for any 2-D trajectory control problem.

However, the dynamic model we build is a simplified 2-D model, which means the roll and pitch angle are not being considered. We must admit that in real life, the weight distribution of each wheels can alter with the change of row and pitch angle. Thus, it is necessary to build a more accurate model in the future. We also plan to add brake control, which we believe can provide more robust to the system.

REFERENCES

[1] Frazzoli E, Dahleh MA, Feron E. Real-time motion planning for agile autonomous vehicles. Journal of guidance, control, and dynamics. 2002 Jan;25(1):116-29.

[2] Kozowski K, Pazderski D. Modeling and control of a 4-wheel skid-steering mobile robot. International journal of applied mathematics and computer science. 2004;14:477-96.