

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Control Laws Design and Validation of Autonomous Mobile Robot Off-Road Trajectory Tracking Based on ADAMS and MATLAB Co-Simulation Platform

Yang. Yi, Fu. Mengyin, Zhu. Hao and Xiong. Guangming
*School of Automation, Beijing Institute of Technology,
 China*

1. Introduction

Autonomous automobile technology is a rapidly developing field, with interest in both academia and industry. Outdoor navigation of autonomous vehicles, especially for rough-terrain driving, has already been a new research focus. DARPA Grand Challenge and LAGR program stand for the top development level in this research region. Rough-terrain driving research offers a challenge that the in-vehicle control system must be able to handle rough and curvy roads, and quickly varying terrain types, such as gravel, loose sand, and mud puddles – while stably tracking trajectories between closely spaced hazards. The vehicle must be able to recover from large disturbances, without intervention (Gabriel, 2007).

Since robotics autonomous navigation tasks in outdoor environment can be effectively performed by skid-steering vehicles, these vehicles are being widely used in military affairs, academia research, space exploration, and so on. In reference (Luca, 1999), a model-based nonlinear controller is designed, following the dynamic feedback linearization paradigm. In reference (J.T. Economou, 2000), the authors applied experimental results to enable Fuzzy Logic modelling of the vehicle-ground interactions in an integrated manner. These results illustrate the complexity of systematic modeling the ground conditions and the necessity of using two variables in identifying the surface properties. In reference (Edward, 2001), the authors described relevant rover safety and health issues and presents an approach to maintaining vehicle safety in a navigational context. Fuzzy logic approaches to reasoning about safe attitude and traction management are presented. In reference (D. Lhomme-Desages, 2006), the authors introduced a model-based control for fast autonomous mobile robots on soft soils. This control strategy takes into account slip and skid effects to extend the mobility over planar, granular soils. Different from above researches, which control the robots on the tarmac, grass, sand, gravel or soil, this chapter focuses on the motion control for skid-steering vehicles on the bumpy and rocklike terrain, and presents novel and effective trajectory tracking control methods, including the longitudinal, lateral, and sensors pan-tilt control law. Furthermore, based on ADAMS&MATLAB co-simulation platform, iRobot ATRV2 is

modelled and the bumpy off-road terrain is constructed, at the same time, trajectory tracking control methods are validated effectively on this platform.

2. Mobile robot dynamic analysis

To research on the motion of skid-steering robots, the 4-wheeled differentially driven robot, which is moving on the horizontal road normally without longitudinal wheel slippage, is analyzed for kinematics and dynamics. As shown in Fig. 1 (a), global Descartes frame and in-vehicle Descartes frame are established. In Global frame $OXYZ$, its origin O is sited on the horizontal plane where the robot is running, and Z axis is orthogonal to the plane; in in-vehicle frame $oxyz$, its origin o is located at the robot center of mass, z axis is orthogonal to the chassis of robot, x axis is parallel to the rectilinear translation orientation of the robot and y axis is orthogonal to the translation orientation in the plane of the chassis. The robot wheelbase is A and the distance between the left and right wheels is B . According to the relation between $OXYZ$ and $oxyz$, the kinematics equation is as follows.

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

(1)

The derivative of equation (1) with respect to time is

$$\begin{aligned} \begin{bmatrix} \ddot{X} \\ \ddot{Y} \end{bmatrix} &= \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \ddot{x} - \dot{y}\dot{\theta} \\ \ddot{y} - \dot{x}\dot{\theta} \end{bmatrix} \\ &= \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} a_x \\ a_y \end{bmatrix} \end{aligned}$$

(2)

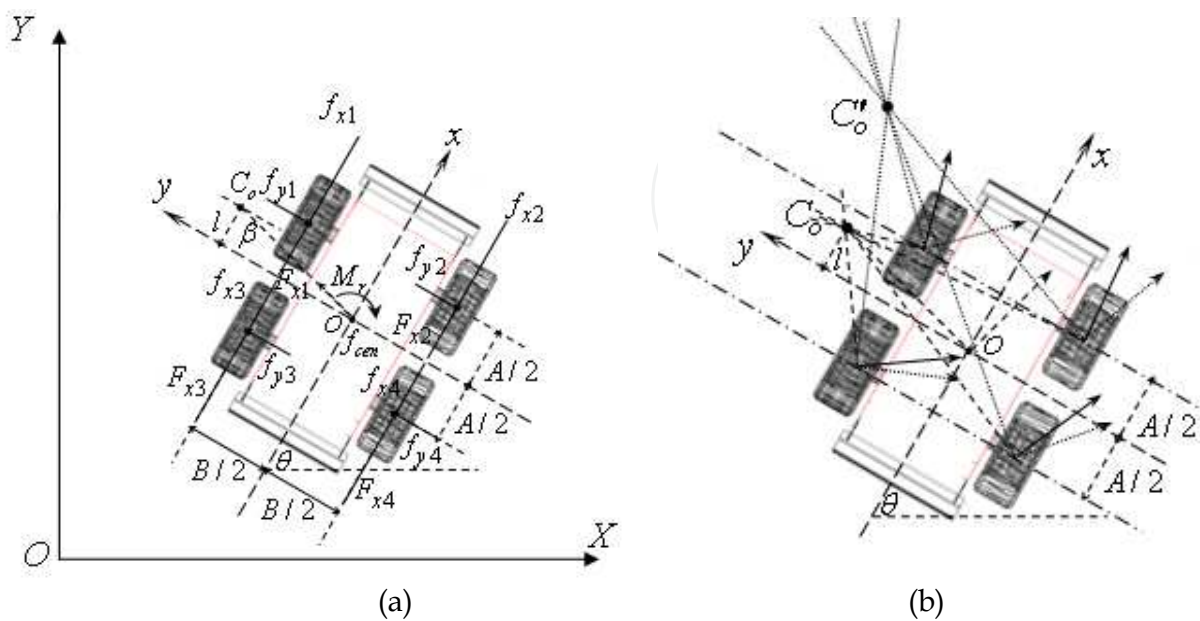


Fig. 1. Dynamics analysis of the robot and the motion constraint about C_o

In the above equations, $\dot{X}, \dot{Y}, \ddot{X}, \ddot{Y}$ represent the absolute longitudinal and lateral velocities, and the absolute longitudinal and lateral accelerations respectively; θ is the angle between x and X axes. $\dot{x}, \dot{y}, \dot{\theta}$ denote longitudinal, lateral and angular velocities in in-vehicle frame, respectively; a_x, a_y are the longitudinal and lateral acceleration respectively in in-vehicle frame. The equations of the robot motion are as follows:

$$\begin{cases} ma_x = \sum_{i=1}^4 F_{xi} - \sum_{i=1}^4 f_{xi} \\ ma_y = - \sum_{i=1}^4 f_{yi} \\ J\ddot{\theta} = (B/2)((-F_{x1} - F_{x3}) + (F_{x2} + F_{x4})) - M_r \end{cases} \quad (3)$$

where J is the moment of inertia of the robot; F_{xi} denotes the tractive force produced by the i -th wheel, and composes the outside tractive force $F_{outside}$ and the inside F_{inside} ; f_{xi} is the longitudinal resistance of the i -th wheel; f_{yi} is the lateral resistance of the i -th wheel; M_r is the resistive moment around the center of o .

During the turning, the robot is subject to the centripetal force f_{cen} , the equation of which is,

$$f_{cen} \cos \beta = f_{y3} + f_{y4} - f_{y1} - f_{y2} \quad (4)$$

Let μ_r and μ_s be the coefficient of longitudinal rolling resistance and the coefficient of lateral friction respectively. In (4), β is the angle between $\overline{C_o o}$ and y axis. Accordingly, when the centripetal acceleration is considered, assume $\beta \approx 0$, then the equation of the tractive force of the robot is,

$$\begin{cases} F_{outside} = \left(\frac{mg}{2} + \frac{hma_y}{B} \right) \mu_r + \frac{ma_y x_{co}}{2R} + \frac{\mu_s mg A}{4B} \left[1 - \left(\frac{a_y}{g\mu_s} \right)^2 \right] \\ F_{inside} = \left(\frac{mg}{2} - \frac{hma_y}{B} \right) \mu_r + \frac{ma_y x_{co}}{2R} - \frac{\mu_s mg A}{4B} \left[1 - \left(\frac{a_y}{g\mu_s} \right)^2 \right] \end{cases} \quad (5)$$

where h is the height of the robot center of mass, the distance between the robot center of mass and the ground, and R is the radius of the robot turning (J. Y. Wong, 1978).

In the context of the robot motion around instantaneous, fixed-axis rotation, the straight line motion of the robot can be referred to $R = \infty$. C_o is the instantaneous center of the rotation, and the coordinates of C_o can be expressed as

$$\begin{bmatrix} x_{C_o} & y_{C_o} \end{bmatrix}^T = \begin{bmatrix} -\dot{y} & \dot{x} \\ \dot{\theta} & \dot{\theta} \end{bmatrix}^T.$$

In Fig. 1 (b), the real line arrowheads represent the transfer orientations of the four wheels of the robot, that is C_o moves within the wheelbase range of the robot; the dashed arrowheads

stand for the transfer orientations of the four wheels of the robot, when C_o , that is C'_o , moves beyond the wheelbase range of the robot. Clearly, if C_o moves out of the wheelbase range, the lateral transfer orientations of four wheels are the same; as a result, the motion of the robot will be out of control. The motion constraint is as follows,

$$\begin{bmatrix} -\sin \theta & \cos \theta & l \end{bmatrix} \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\theta} \end{bmatrix} = 0 \quad (|l| < A/2) \quad (6)$$

where l is the distance from C_o to y axis. According to Fig. 1 (a) and equation (5), it is $l = Aa_y \cos \beta / 2\mu_s g$ that can be obtained. Therefore, the constraint of the motion, shown as follow, is imperative.

$$(\dot{x})^2 / R < \mu_s g \text{ or } (\dot{\theta})^2 R < \mu_s g \quad (7)$$

In this paper, a description of trajectory space (Matthew Spenko, 2006) is presented according to the robot's dynamic analysis, which is defined as the two-dimensional space of the robot's turning angular speed $\dot{\theta}$ and longitudinal velocity V_{long} , ($V_{long} = \dot{x}$). This kind of description is very useful during motion control. Consider inequalities (7): $(V_{long}(t))^2 / R < \mu_s g$ or $(\dot{\theta}(t))^2 R < \mu_s g$, it is easy to get $|V_{long}(t)| < \sqrt{\mu_s g R}$ or $|\dot{\theta}(t)| < \sqrt{\mu_s g / R}$, then take another constraint into consideration, $V_{long} = \dot{\theta} R$, now a figure of the robot's $(V, \dot{\theta})$ space (see Fig. 2.) can be obtained. The boundary curve in the figure satisfies $V_{long}(t) \cdot \dot{\theta}(t) = \mu_s g$. When the robot's $(V, \dot{\theta})$ state is in the shadow region, that's to say $V_{long}(t) \cdot \dot{\theta}(t) < \mu_s g$, the robot is safe, which means the hazardous situation like side slippage won't occur. As a result, inequalities (7) is crucial for control decision making.

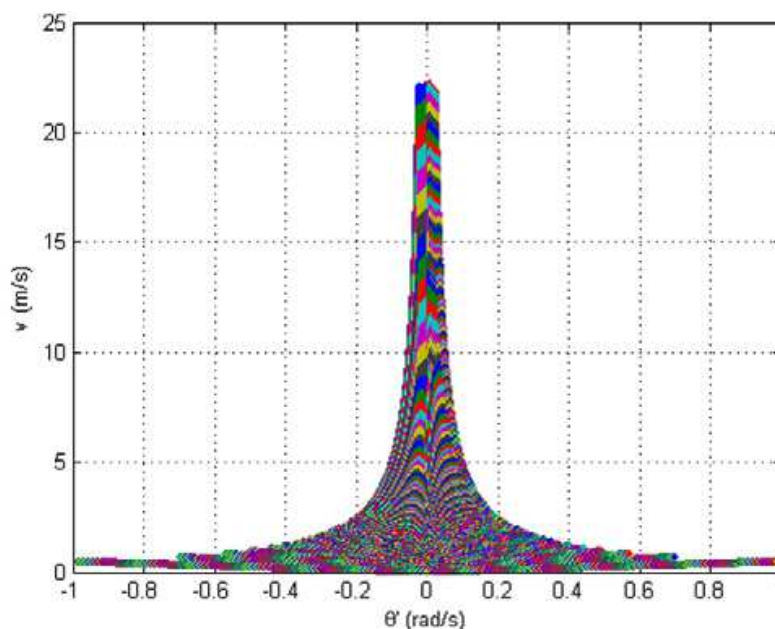


Fig. 2. The $(V, \dot{\theta})$ space of motion control of the robot. (In this figure, $\mu_s = 0.49$ and $g = 9.8 \text{ m/s}^2$)

3. Trajectory tracking control laws design

Autonomous mobile robot achieves outdoor navigation by three processes, including the environment information acquired by the perception module, the control decision made by the planner module, and the motion plan performed by the motion control module (Gianluca, 2007). Consequently, for safe and accurate outdoor navigation it is vital to harmonize the three modules performance. In this paper, the emphasis is focused on the decision of control laws of the robot, and these controlled objects include the longitudinal velocity, the lateral velocity, and the angles of sensor pan-tilts.

As shown in Fig. 3, in an off-road environment, the robot uses laser range finder (LRF) with one degree of freedom (DOF) pan-tilt (only tilt) to scan bumpy situation of the close front ground, and employs stereo vision with two DOF pan-tilt to perceive drivable situation of far front ground. With the data accessed from laser and vision sensors, the passable path can be planned, and the velocities of left and right robot's sides can be controlled to track the path, consequently, the robot off-road running is completed.

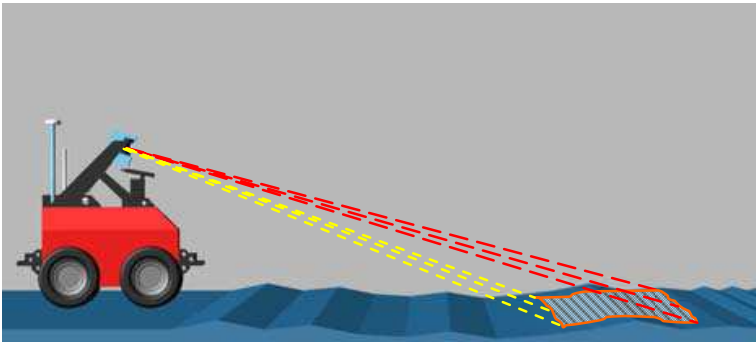


Fig. 3. Off-road driving of the robot

3.1 Longitudinal control law

In this section, a kind of humanoid driving longitudinal control law based on fuzzy logic is proposed. First, the effect factors to the longitudinal velocity are classified. As shown in Fig. 4,

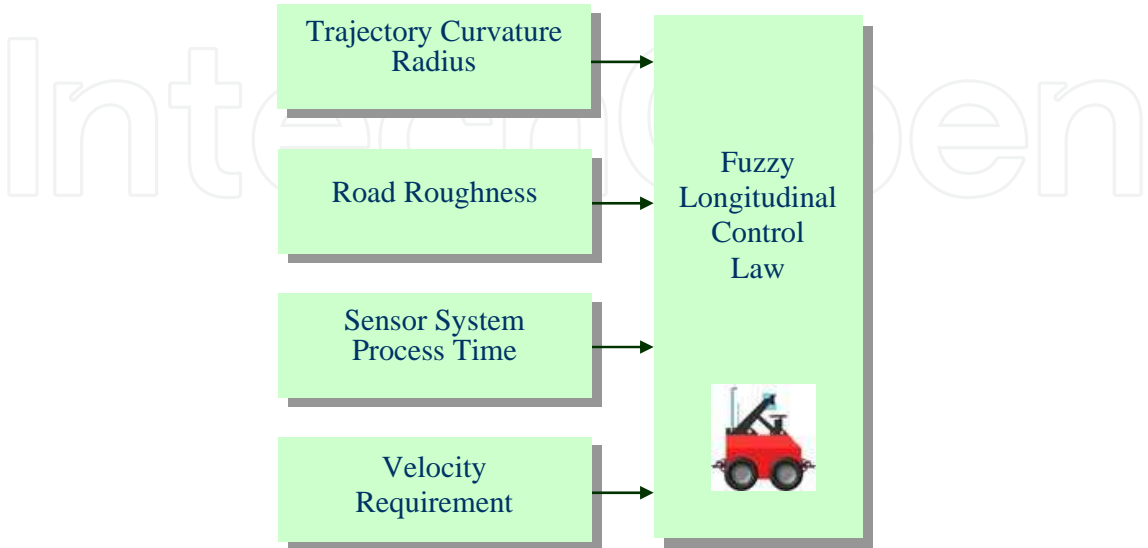


Fig. 4. The longitudinal control law of the robot

these factors include curvature radius of trajectory, road roughness, process time of sensors and speed requirement. Based on the four factors and the analysis of kinematics and dynamics, the longitudinal velocity equation can be given,

$$V_{long} = (\mu_{cv}(r_c) + \mu_{rr}(R_{rough}))\mu_t(T_{sen})\mu_v(V_{re})V_{re} \quad (8)$$

where V_{long} is longitude velocity command, μ_{cv} , μ_{rr} , μ_t and μ_v are the weight factors of the curvature radius of trajectory r_c , road roughness R_{rough} , processing time of sensors T_{sen} and speed requirement V_{re} respectively. The weight factors can vary within $[0,1]$.

3.2 Lateral control law

Autonomous vehicle off-road driving is a special control problem because mathematical models are highly complex and can't be accurately linearized. Fuzzy logic control, however, is robust and efficient for non-linear system control (Gao Feng, 2009), and a well-tested method for dealing with this kind of system, provides good results, and can incorporate human procedural knowledge into control algorithms. Also, fuzzy logic lets us mimic human driving behavior to some extent (José E. Naranjo, 2007). Therefore, the paper has presented a kind of novel fuzzy lateral control law for the robot off-road running.

Thereafter, the lateral control law is designed for the purpose of position tracking by adjusting navigation orientation to reduce position error.

Definition: When the robot moves toward the trajectory, the orientation angle of the robot is positive, whichever of the two regions, namely left error region and right error region of the tracking position, the robot is in. When the robot moves against the trajectory, the angle is negative. When the robot motion is parallel to the trajectory, the angle is zero.

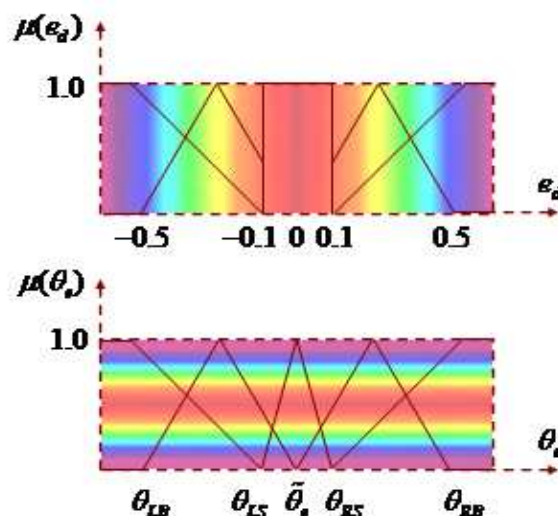


Fig. 5. Plots of membership functions of e_d and θ_e . The upper plot stands for $\mu(e_d)$. $[-0.1, 0.1]$ is the range of the center region, $\mu(e_d) = 1.0$ in this region. The lower plot expresses $\mu(\theta_e)$, which is indicated with triangle membership function. In this figure, the colorful bands represent the continuous changing course of e_d .

In the course of trajectory tracking of the robot, **position error** e_d and **orientation error** θ_e , which are the error values between the robot and the trajectory, are the inputs of the lateral

controller. First, the error inputs, including the position error and the orientation error, are pre-processed to improve steady precision of the trajectory tracking, and restrain oscillation of that. The error E can be written as $E = \bar{E} + \Delta E$, where \bar{E} is the result of smoothing filter about E , i.e.

$$\bar{E} = \sum_{i=n+1}^i e_i / n ;$$

ΔE forecasts the error produced in the next system sample time,

$$\Delta E = \sum_{i=n+1}^i k_{ei} e_i / n \quad (i > n).$$

As a result,

$$E = \sum_{i=n+1}^i (1+k_{ei}) e_i / n \quad (i > n), \quad n = 4, \quad k_{ei} = f_{ei}(\dot{x}, T_s),$$

T_s represents system sample time. Second, the error inputs, after being pre-processed, are fuzzyfication-processed. As shown in Fig. 5, degree of membership $\mu(e_d)$ and $\mu(\theta_e)$ are indicated in terms of triangle and rectangle membership functions:

Part One: the space of e_d is divided into LB , LS , MC , RS , RB . According to the result of the e_d division, the phenotype rules of the lateral fuzzy control law hold.

Part Two: the space of θ_e is also divided into LB , LS , MC , RS , RB . According to the result of the θ_e division, the recessive rules of the control law are obtained.

It is necessary to explain that the phenotype rules are the basis of the recessive rules, namely each phenotype rule possesses a group of relevant recessive rules. Each phenotype rule has already established their own expectation orientation angle $\tilde{\theta}_e$, that is to say the robot is expected to run on this orientation angle, $\tilde{\theta}_e$, and this angle is the very center angle of this group recessive rules. When the center angle, $\tilde{\theta}_e$, of each group of recessive rules is varying, the division of θ_e of each group recessive rules changes accordingly.

All of the control rules can be expressed as continuous functions: f_{LB} , f_{LS} , f_{MC} , f_{RS} , f_{RB} , and, consequently, the global continuity of the control rules is established.

The functions, f_{LB} , f_{LS} , f_{RS} , f_{RB} , which express the rule functions of the non-center region of e_d , are as follows:

$$\dot{\theta} = \frac{k_{\tilde{\theta}_e} k_t (\tilde{\theta}_e - \theta_e)}{T_s} + k_t \tilde{\theta}_{tr} \quad (9)$$

In (9), $\tilde{\theta}_e$ is the expectation tracking-angle in the current position-error region, and $\tilde{\theta}_e$ is monotonically increasing with e_d . $\tilde{\theta}_e$ can be given by this formula

$$\tilde{\theta}_e = \arctan \frac{k_{e_d} e_d}{V_{long}}, \quad k_{e_d} = \mu_{DL}(e_d) k_{DL} + \mu_{DS}(e_d) k_{DS},$$

where μ_{DL} stands for the large error region, including LB and RB , membership degree of e_d , and μ_{DS} represents the small error region, including LS and RS , membership degree of e_d . $\dot{\theta}_{tr}$ is the estimation of the trajectory-angle rate. $k_{\tilde{\theta}_e}$ can be worked out by this equation:

$$k_{\tilde{\theta}_e} = \mu_{\theta L}(\theta_e)k_{\theta L} + \mu_{\theta S}(\theta_e)k_{\theta S},$$

where $\mu_{\theta L}$ stands for the large error region, including LB and RB , membership degree of θ_e , and $\mu_{\theta S}$ denotes the small error region, including LS and RS , membership degree of θ_e . k_{DL} and $k_{\theta L}$ express the standard coefficients of the large error region of e_d and θ_e ; k_{DS} and $k_{\theta S}$ indicate the small error region of e_d and θ_e , respectively. k_t is the proportional coefficients of the system sample time T_s .

f_{MC} is deduced as follows:

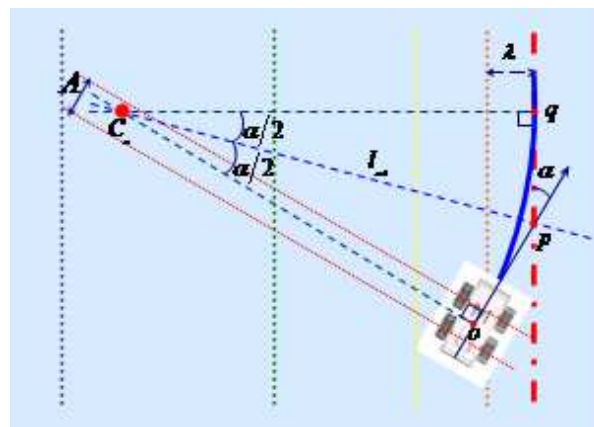


Fig. 6. Tracking trajectory of the robot. In this Figure, the red dashdotted line stands for the trajectory tracked by the robot. The different color dotted lines represent the boundaries of the different error regions of e_d .

When the robot moves into the *center* region at the orientation of α , the motion state of the robot can be divided into two kinds of situations.

Situation One: Assume that α has decreased into the rule admission angular range of *center* region, i.e. $0 \leq \alpha \leq \theta_{cent}$, where θ_{cent} , which is subject to (7), is the critical angle of *center* region. To make the robot approach the trajectory smoothly, the planner module requires the robot to move along a certain circle path. As the robot moves along the circle path in Fig. 6, the values of e_d and θ_e decrease synchronously. In Fig. 6, λ is the variety range of e_d in the *center* region. α is the angle between the orientation of the robot and the trajectory when the robot just enters the *center* region. $R \approx 2\lambda/\alpha^2$ can be worked out by geometry, and in addition, the value of α is very small, so the process of approaching trajectory can be represented as

$$\Delta\alpha = \alpha \frac{\Delta\lambda}{\lambda}.$$

Situation Two: When $\alpha < 0$ or $\alpha > \theta_{cent}$. If the motion decision from the planner module were the same as Situation One, the motion will not meet (7). According to the above

analysis, the error of tracking can not converge until the adjusted $\tilde{\theta}_e$ makes α be true of Situation One. Therefore, the purpose of control in Situation Two is to decrease θ_e . Based on the above deduction, f_{MC} is as follow:

$$\dot{\theta} = \frac{k_t(\tilde{\theta}_e - \theta_e)}{T_s} + k_t\ddot{\theta}_{tr} \tag{10}$$

Where

$$\tilde{\theta}_e = \frac{e_d\theta_{cent}}{\lambda},$$

λ is the variety range of e_d in the *center* region, $[-0.1m,0]$ or $[0,0.1m]$. $\dot{\theta}$ is the output of (9) and (10), at the same time, $\dot{\theta}$ is subject to (7), consequently, $(\dot{\theta})^2 R < \mu_s g$ is required by the control rules.

The execution sequence of the control rules is as follows:
First, the phenotype control rules are enabled, namely to estimate which error region (*LB*, *LS*, *MC*, *RS*, *RB*) the current e_d of the robot belongs to, and to enable the relevant recessive rules; Second, the relevant recessive rules are executed, at the same time, $\tilde{\theta}_e$ is established in time.
The lateral control law is exemplified in Fig. 7. In this figure, the different color concentric circle bands represent the different position error e_d . From the outermost circle band to the

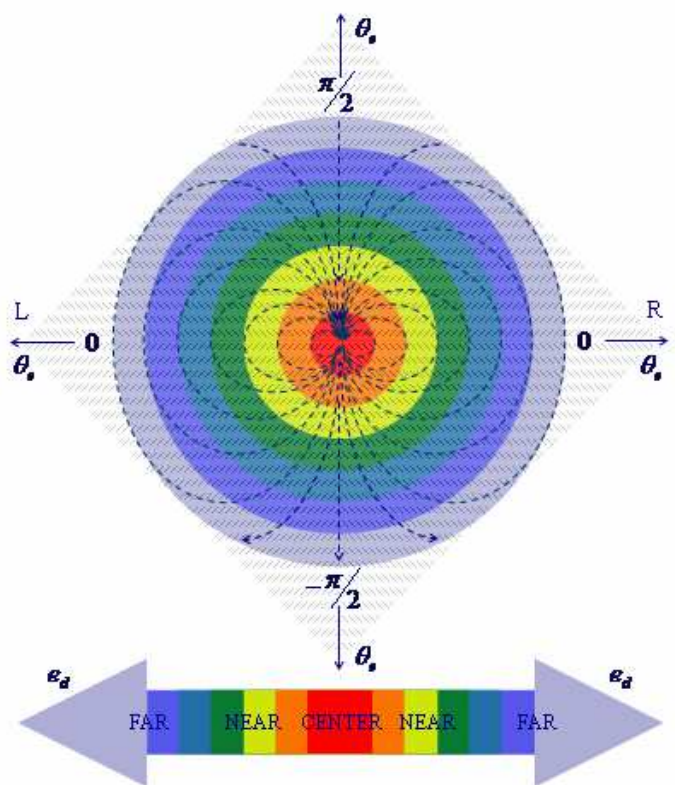


Fig. 7. Plot of the lateral control law of the robot. These dasheds stand for the parts of the performance result of the control law.

center round, the values of e_d is decreasing. The red center round stands for MC of e_d , that is the *center* region of e_d . At the center point of the red round, $e_d = 0$. According to the above definition, the orientation range of the robot is $(-\pi, \pi]$, and the two 0 degree axes of θ_e stand for the 0 degree orientation of the left and right region of the trajectory, respectively. At the same time, $\pi/2$ axis and $-\pi/2$ axis of θ_e are two common axes of the orientation of the robot in the left and right region of the trajectory. In the upper sub-region of 0 degree axes, the orientation of the robot is toward the trajectory, and in the lower sub-region, the orientation of the robot is opposite to the trajectory. The result of the control rules converges to the center of the concentric circle bands according to the direction of the arrowheads in Fig. 7. Based on the analysis of the figure, the global asymptotic stability of the lateral control law can be established, and if $e_d = 0$ and $\theta_e = 0$, the robot reaches the only equilibrium zero. The proving process is shown as follow:

Proof: From the kinematic model (see Fig. 8.), it can be seen that the position error of the robot e_d satisfies the following equation,

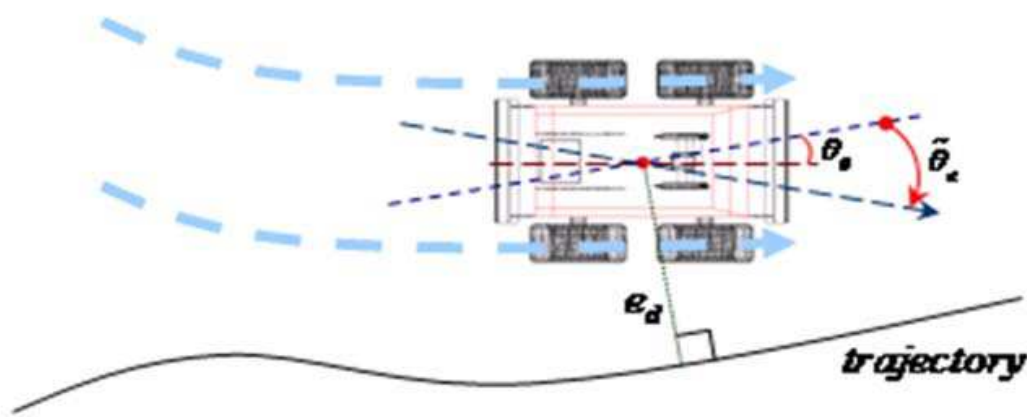


Fig. 8. Trajectory Tracking of the mobile robot

$$\dot{e}_d(t) = -V_{long}(t) \sin(\tilde{\theta}_e(t)) \quad (11)$$

a. When the robot is in the *non - center* region, a controller is designed to control the robot's lateral movement:

$$\tilde{\theta}_e(t) = \arctan \frac{k_{e_d} e_d(t)}{V_{long}(t)} \quad (12)$$

Combining Equations (11) and (12), we get

$$\dot{e}_d(t) = -V_{long}(t) \sin(\arctan(\tilde{\theta}_e(t))) = \frac{-k_{e_d} e_d(t)}{\sqrt{1 + \left(\frac{k_{e_d} e_d(t)}{V_{long}(t)} \right)^2}} \quad (13)$$

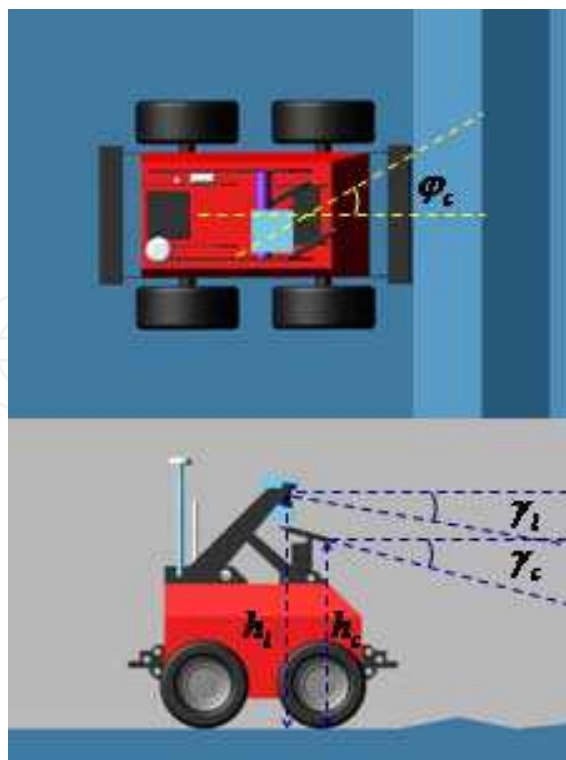


Fig. 9. LRF Pan-Tilt and Stereo Vision Pan-Tilt motion

As the sign of \dot{e}_d is always opposite that of e_d , e_d will converge to 0. In equation (11), $|\dot{e}_d(t)| \approx V_{long}(t)$, and $|\dot{e}_d(t)| \approx k_{e_d} e_d(t)$ can be formed by equation (13). Therefore the convergence rate of e_d is between linear and exponential. When the robot is far away from the trajectory, it's heading for trajectory vertically, then

$$\tilde{\theta}_e = \frac{\pi}{2}, \quad \dot{e}_d(t) = -V_{long}(t), \quad e_d(t) = -V_{long}(t) + e_d(t_0);$$

when the robot is near the trajectory, $e_d \approx 0$, then in equation (12),

$$\sqrt{1 + \left(\frac{k_{e_d} e_d(t)}{V_{long}(t)} \right)^2} \approx 1, \quad \dot{e}_d(t) = -k_{e_d} e_d(t).$$

According to equation (12), e_d and $\tilde{\theta}_e$ can converge to 0 simultaneously.

b. When the robot enters the *center* region, another controller is designed,

$$\tilde{\theta}_e(t) = \frac{e_d(t) \theta_{cent}}{\lambda} \quad (14)$$

Combining Equations (11) and (14), we get

$$\dot{e}_d = -V_{long} \sin\left(\frac{e_d \theta_{cent}}{\lambda}\right).$$

In this region, e_d is very small, and consequently, $\frac{e_d \theta_{cent}}{\lambda}$ will also be very small, and then

$$\sin\left(\frac{e_d \theta_{cent}}{\lambda}\right) \approx \frac{e_d \theta_{cent}}{\lambda}$$

is derived. Therefore,

$$\dot{e}_d = -V_{long} \frac{e_d \theta_{cent}}{\lambda} = -\frac{V_{long} \theta_{cent}}{\lambda} e_d, \text{ and then } e_d(t) = e_d(t_1) \exp\left\{-\frac{V_{long} \theta_{cent}}{\lambda} t\right\},$$

where t_1 is the time when the robot enters the *center* region. In other word, e_d converges to 0 exponentially. Then, according to

$$\tilde{\theta}_e(t) = \frac{e_d \theta_{cent}}{\lambda}, \tilde{\theta}_e(t) \text{ converges to } 0.$$

So the origin is the only equilibrium in the $(e_d, \tilde{\theta}_e)$ phase space.

3.3 LRF Pan-tilt and stereo vision pan-tilt control

Perception is the key to high-speed off-road driving. A vehicle needs to have maximum data coverage on regions in its trajectory, but must also sense these regions in time to react to obstacles in its path. In off-road conditions, the vehicle is not guaranteed a traversable path through the environment, thus better sensor coverage provides improved safety when traveling. Therefore, it is important for off-road driving to apply active sensing technology. In the chapter, the angular control of the sensor pan-tilts assisted in achieving the active sensing of the robot. Equation (15) represents the relation between the angles measured, i.e. φ_c , γ_c and γ_l , of the sensors mounted on the robot and the motion state, i.e. θ_e and \dot{x} , of the robot.

$$\begin{bmatrix} \varphi_c \\ \gamma_c \\ \gamma_l \end{bmatrix} = \begin{bmatrix} k_{\varphi_c} & 0 & 0 \\ 0 & k_{\gamma_c} & 0 \\ 0 & 0 & k_{\gamma_l} \end{bmatrix} \begin{bmatrix} \theta_e \\ \dot{x} \\ \dot{x} \end{bmatrix} \quad (15)$$

In (15), φ_c , γ_c are the pan angle and tilt angle of the stereo vision respectively. γ_l is the tilt angle of the LRF; k_{φ_c} , k_{γ_c} and k_{γ_l} are the experimental coefficients between the angles measured and the motion state, and they are given by practical experiments of the sensors and connected with the measurement range requirement of off-road driving. At the same time, the coordinates of the scanning center are $x_{ec} = x_c + h_c \cot \gamma_c \cos \varphi_c$, $y_{ec} = y_c + h_c \cot \gamma_c \sin \varphi_c$; and $x_{el} = x_l + h_l \cot \gamma_l$, $y_{el} = 0$. In the above equations, x_c , y_c , x_l , y_l , respectively, are the coordinates of the sense center points of the stereo vision and LRF in in-vehicle frame. As shown in Fig. 9, h_c and h_l are their height value, to the ground, accordingly.

The angular control and the longitudinal control are achieved by *PI* controllers, and they are the same as the reference (Gabriel, 2007).

4. Simulation tests

4.1 Simulation platform build

In this section, ADAMS and MATLAB co-simulation platform is built up. In the course of co-simulation, the platform can harmonize the control system and simulation machine system, provide the 3D performance result, and record the experimental data. Based on the analysis of the simulation result, the design of experiments in real world can become more reasonable and safer.

First, based on the character data of the test agent, ATRV2, such as the geometrical dimensions ($HLW65 \times 105 \times 80cm$), the mass value (118Kg), the diameter of the tire ($\Phi 38cm$) and so on, the simulated robot vehicle model is accomplished, as shown in Fig.10.



Fig. 10. ATRV2 and its model in ADAMS

Second, according to the test data of the tires of ATRV2, the attribute of the tires and the connection character between the tires and the ground are set. The ADAMS sensor interface module can be used to define the motion state sensors parameters, which can provide the information of position and orientation to ATRV2.

It is road roughness that affects the dynamic performance of vehicles, the state of driving and the dynamic load of road. Therefore, the abilities of overcoming the stochastic road roughness of vehicles are the key to test the performance of the control law during off-road driving. In the paper, the simulation terrain model is built up by Gaussian-distributed pseudo random number sequence and power spectral density function (Ren, 2005). The details are described as follows:

- Gaussian-distributed random number sequence $x(t)$, whose variance $\sigma = 18$ and mean $E = 2.5$, is yielded;
- The power spectral $S_X(f)$ of $x(t)$ is worked out by Fourier transform of $R_X(\tau)$, which is the autocorrelation function of σ ,

$$\begin{aligned} S_X(f) &= \int_{-\infty}^{+\infty} R_X(\tau) e^{-j2\pi f\tau} d\tau \\ &= T\sigma^2 \left(\frac{\sin \pi fT}{\pi fT} \right)^2 \end{aligned} \quad (16)$$

where T is the time interval of the pseudo random number sequence;

- Assume the following,

$$y(t) = x(t) * h(t) = \int_{-\infty}^{+\infty} x(\tau)h(t - \tau)d\tau \quad (17)$$

$$h(t) = \int_{-\infty}^{+\infty} H(f)e^{j2\pi ft}df \quad (18)$$

where $h(t)$ is deduced by inverse Fourier transform from $H(f)$, and they both are real even functions, then,

$$H(f) = \sqrt{\frac{S_Y(f)}{S_X(f)}} \quad (19)$$

$$\begin{aligned} y_k &= y(kT) \\ &= T \sum_{-M}^{+M} x(rT)h(kT - rT) = T \sum_{-M}^{+M} x_r h_{k-r} \end{aligned} \quad (20)$$

where $S_Y(f)$ is the power spectral of $y(t)$, y_k is the pseudo random sequence of $S_Y(f)$, $k=0,1,2,\dots,N$, and M can be established by the equation $\lim_{M \rightarrow \infty} h_m = h(MT) = 0$;

- d. Assign a certain value to the road roughness and adjust the parameters of the special points on the road according to the test design, and the simulation test ground is shown in Fig. 11.

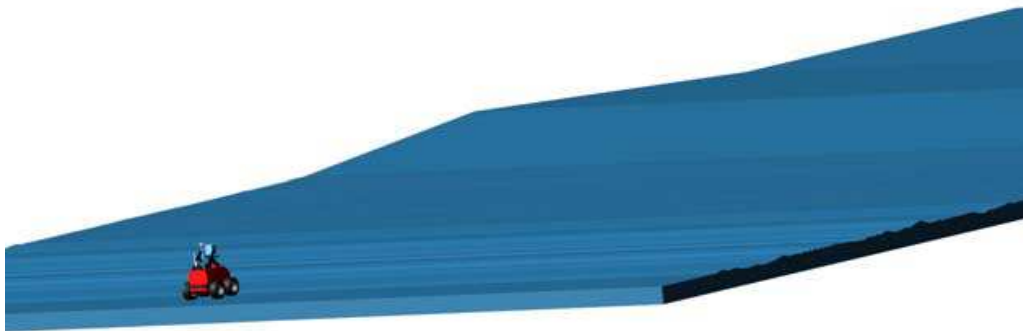


Fig. 11. The simulation test ground in ADAMS

4.2 Simulation tests

In this section, the control law is validated with the ADAMS&MATLAB co-simulation platform.

Based on the position-orientation information provided by the simulation sensors and the control law, the lateral, longitudinal motion of the robot and the sensors pan-tilts motion are achieved. The test is designed to make the robot track two different kinds of trajectories, including the straight line path, sinusoidal path and circle path. In Test One, the tracking trajectory consists of the straight line path and sinusoidal path, in which the wavelength of the sinusoidal path is $5\pi m$, the amplitude is $3m$. The simulation result of Test One is shown in Fig. 12. In Test Two, the tracking trajectory contains the straight line path and circle path, in which the radius of the circle path is $5m$. The simulation result of Test Two is shown in Fig. 13.

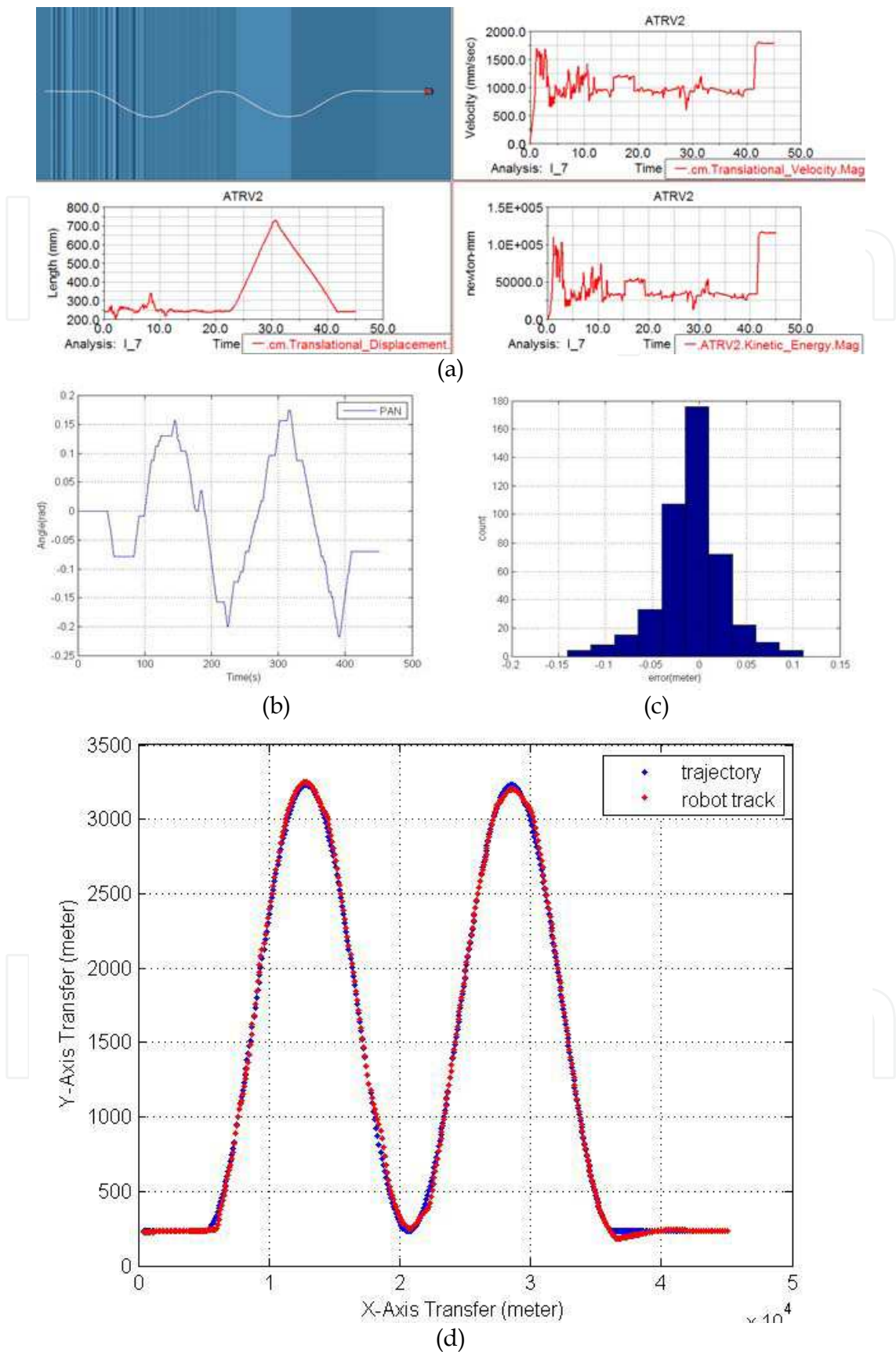


Fig. 12. Plots of the result of Test One ($T_s = 0.05s$)

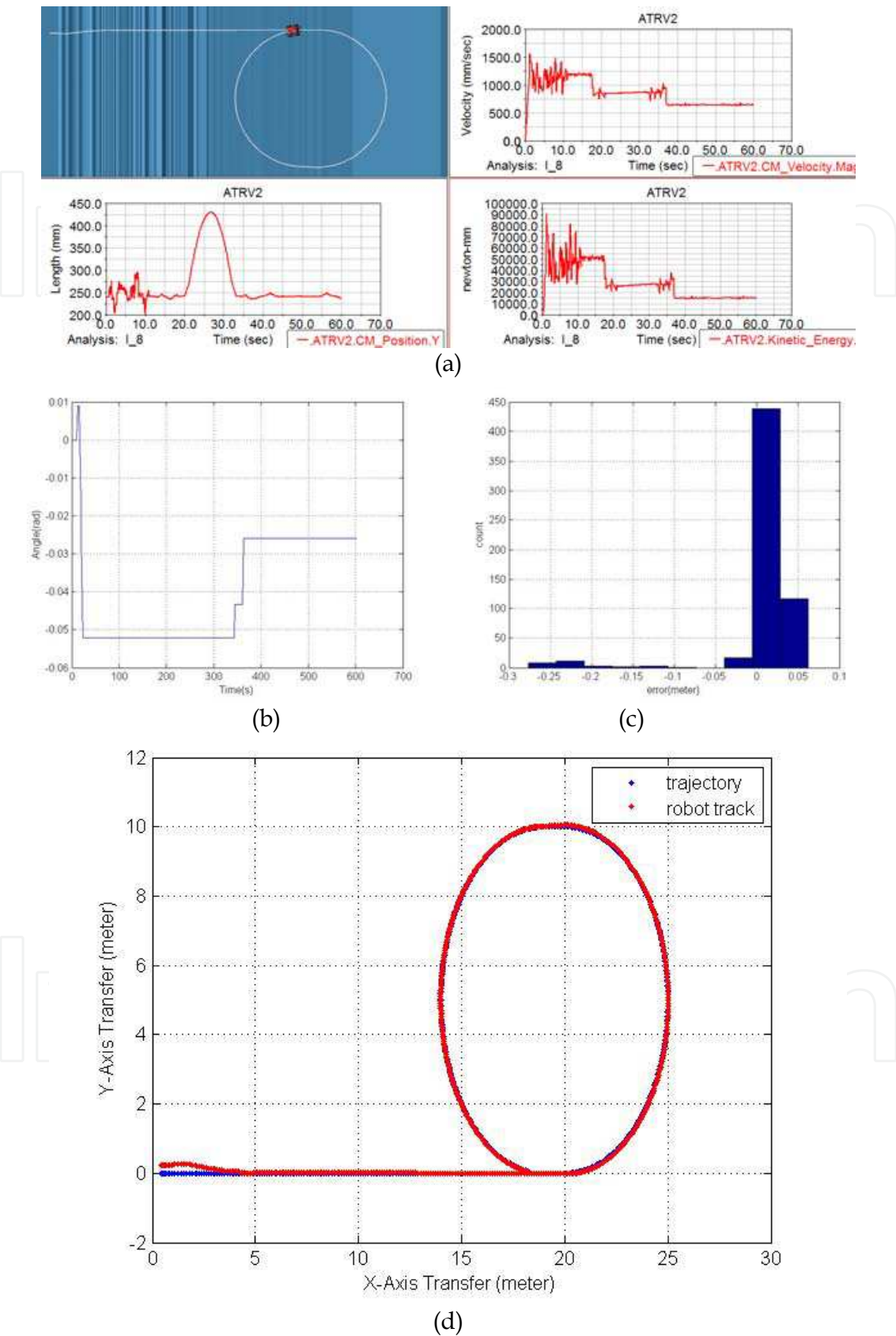


Fig. 13. Plots of the result of Test Two ($T_s = 0.05s$)

In Fig. 12, which is the same as Fig. 13, sub-figure **a** is the simulation data recorded by ADAMS. In sub-figure **a**, the upper-left part is the 3D animation figure of the robot off-road driving on the simulation platform, in which the white path shows the motion trajectory of the robot. The upper-right part is the velocity magnitude figure of the robot. It is indicated that the velocity of the robot is adjusted according to the longitudinal control law. In addition, it is clear that the longitudinal control law, whose changes are mainly due to the curvature radius of the path and the road roughness, can assist the lateral control law to track the trajectory more accurately. In Test One, the average velocity approximately is 1.2 m/s , and in Test Two, the average velocity approximately is 1.0 m/s . The bottom-left part presents the height of the robot's mass center during the robot's tracking; in the figure, the road roughness can be implied. The bottom-right part shows that the kinetic energy magnitude is required by the robot motion in the course of tracking. In Sub-figure **b**, the angle data of the stereo vision pan rotation is indicated. The pan rotation angle varies according to the trajectory. Sub-figure **c** is the error statistic figure of trajectory tracking. As is shown, the error values almost converge to 0. The factors, which produce these errors, include the roughness and the curvature variation of the trajectory. In Fig. 13 (**d**), the biggest error is yielded at the start point due to the start error between the start point and the trajectory. Sub-figure **d** is the trajectory tracking figure, which contains the objective trajectory and real tracking trajectory. It is obvious that the robot is able to recover from large disturbances, without intervention, and accomplish the tracking accurately.

5. Conclusions

The ADAMS&MATLAB co-simulation platform facilitates control method design, and dynamics modeling and analysis of the robot on the rough terrain. According to the practical requirement, the various terrain roughness and obstacles can be configured with modifying the relevant parameters of the simulation platform. In the simulation environment, the extensive experiments of control methods of rough terrain trajectory tracking of mobile robot can be achieved. The experiment results indicate that the control methods are robust and effective for the mobile robot running on the rough terrain. In addition, the simulation platform makes the experiment results more vivid and credible.

6. References

- D. Lhomme-Desages, Ch. Grand, J-C. Guinot, "Model-based Control of a fast Rover over natural Terrain," Published in the Proceedings of CLAWAR'06: Int. Conf. on Climbing and Walking Robots, Sept 2006.
- Edward Tunstel, Ayanna Howard, Homayoun Seraji, "Fuzzy Rule-Based Reasoning for Rover Safety and Survivability," Proceedings of the 2001 IEEE International Conference on Robotics & Automation, pp. 1413-1420, Seoul, Korea • May 21-26, 2001.
- Gabriel M. Hoffmann, Claire J. Tomlin, Michael Montemerlo, and Sebastian Thrun (2007). Autonomous Automobile Trajectory Tracking for Off-Road Driving: Controller Design, Experimental Validation and Racing. *Proceedings of the 2007 American Control Conference*, 2296-2301. New York City, USA.

- Gao Feng, "A Survey on Analysis and Design of Model-Based Fuzzy Control Systems," IEEE Transactions on Fuzzy Systems, Vol. 14, No. 5, pp. 676-697, 2006.
- Gianluca Antonell, Stefano Chiaverini, and Giuseppe Fusco. "A Fuzzy-Logic-Based Approach for Mobile Robot Path Tracking," IEEE Transactions on Fuzzy Systems, Vol. 15, No. 2, pp. 211-221, 2007.
- José E. Naranjo, Carlos González, Ricardo García, and Teresa de Pedro, "Using Fuzzy Logic in Automated Vehicle Control. IEEE Intelligent Systems," Vol. 22, No. 1, pp. 36-45, 2007.
- J.T. Economou, R.E. Colyer, "Modelling of Skid Steering and Fuzzy Logic Vehicle Ground Interaction," Proceedings of the American Control Conference, pp. 100-104, Chicago, Illinois June 2000.
- J. Y. Wong, "Theory of Ground Vehicles," John Wiley and Sons, New York, USA, 1978.
- Luca Caracciolo, Alessandro De Luca, and Stefano Iannitti, "Trajectory Tracking Control of a Four-Wheel Differentially Driven Mobile Robot," Proceedings of the 1999 IEEE International Conference on Robotics & Automation, pp. 2632-2638. Detroit, Michigan, USA.
- Matthew Spenko, Yoji Kuroda, Steven Dubowsky, and Karl Iagnemma, "Hazard avoidance for High-Speed Mobile Robots in Rough Terrain", Journal of Field Robotics, Vol. 23, No. 5, pp. 311-331, 2006.
- Ren Weiqun. Virtual Prototype in Vehicle-Road Dynamics System, Chapter Four. Publishing House of Electronics Industry, Beijing, China.

IntechOpen



Applications of MATLAB in Science and Engineering

Edited by Prof. Tadeusz Michalowski

ISBN 978-953-307-708-6

Hard cover, 510 pages

Publisher InTech

Published online 09, September, 2011

Published in print edition September, 2011

The book consists of 24 chapters illustrating a wide range of areas where MATLAB tools are applied. These areas include mathematics, physics, chemistry and chemical engineering, mechanical engineering, biological (molecular biology) and medical sciences, communication and control systems, digital signal, image and video processing, system modeling and simulation. Many interesting problems have been included throughout the book, and its contents will be beneficial for students and professionals in wide areas of interest.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Yang. Yi, Fu. Mengyin, Zhu. Hao and Xiong. Guangming (2011). Control Laws Design and Validation of Autonomous Mobile Robot Off-Road Trajectory Tracking Based on ADAMS and MATLAB Co-Simulation Platform, Applications of MATLAB in Science and Engineering, Prof. Tadeusz Michalowski (Ed.), ISBN: 978-953-307-708-6, InTech, Available from: <http://www.intechopen.com/books/applications-of-matlab-in-science-and-engineering/control-laws-design-and-validation-of-autonomous-mobile-robot-off-road-trajectory-tracking-based-on->

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2011 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen