Wheeled Mobile Robot Control Based on SVM and Nonlinear Control Laws*

Yong Feng[1,2], Huibin Cao[2], and Yuxiang Sun[2]

[1] Department of Automation, University of Science and Technology of China
Hefei, Anhui, 230026, China
fengyong@mail.ust.edu.cn
[2] Institute of Intelligent Machine, Chinese Academy of Science
Hefei, Anhui, 230031, China
hbcao@iim.ac.cn, sunyuxiang636@sohu.com

Abstract. Wheeled mobile robot control method based on SVM algorithm and nonlinear control laws is discussed in this paper. The control system includes two parts: nonlinear controller and SVM controller. Nonlinear controller's primary role is to obtain the desired velocity which can make the kinematics stable, SVM controller's primary role is to optimize the control parameters through on-line learning and track the desired velocity. The control method proposed in this paper is independent of the control object model, and has good generalization capability. Simulations illustrate quality and efficiency of this method.

Keywords: wheeled mobile robot, SVM, nonlinear control, tracking control.

1 Introduction

Wheeled mobile robot (WMR) can drive automatically by motion controller and surroundings sensors. In the motion control, WMR should be capable of performing trajectory tracking and stabilization. However, WMR is a nonholonomic dynamic system with intrinsic nonlinearty, and commonly with unmodeled disturbance and unstructured, unmodeled dynamics [1]. Conventionally, this control design relies on engineers to analyze the WMR system so as to synthesize the appropriate controller. But usually difficulties arise from absence of accurate model. Fuzzy control design may skip building the model but needs domain expert to construct the fuzzy rules. Neural networks offer exciting advantages such as adaptive learning, fault tolerance and generalization. In [2] and [3] an artificial neural network-based controller was developed by combining the feedback velocity control technique and torque controller. But the controller structure and the neural network-learning algorithm are very complicated and preparing appropriate training samples usually needs an existing controller. In this paper, we proposed a novel control method based on support vector

* This paper is partially supported by National Nature Science Foundation project China (Grant #60910005).

H. Deng et al. (Eds.): AICI 2011, Part III, LNAI 7004, pp. 462–471, 2011.

machine (SVM) and nonlinear laws. We first construct the nonlinear controller using direct Lyapunov method to stabilize the system, but its dynamic performance is bad, then we construct a SVM controller to optimize the control parameters to ensure the well dynamic performance using online samples.

This paper is organized as follows. In section 2 we will present the dynamic model of wheeled mobile robot. In section 3 we will describe a Lyapunov based nonlinear control method for asymptotic stability of kinematic equations. In section 4 we will describe SVM controller to optimize the control parameters. In section 5 we describe the structure of the control system. The simulation results are presented in section 6 and the conclusions are given in the last section.

2 Mobile Robot Dynamic Model

In this paper, we consider the two wheeled differential drive mobile robot (Fig. 1). Two independent analogous DC motors are the actuators of left and right wheels, while one or two free wheel casters are used to keep the robot stable. Point *C* is the center of axis of driving wheels, and θ is the orientation angle of robot in the inertial frame.

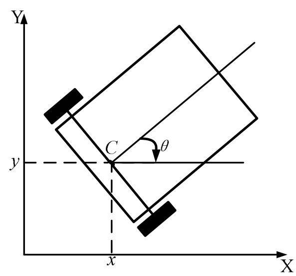

Fig. 1. Coordination of Differential Drive Mobile Robot

Pose vector of robot is defined as $q=[x,y,\theta]^T$, x and y are the coordination of point *C*. Neglecting the centripetal force, coriolis torque and gravity torque, the dynamic equation is given by

$$\begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I \end{bmatrix}\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\theta} \end{bmatrix} = \frac{1}{R}\begin{bmatrix} \cos\theta & \cos\theta \\ \sin\theta & \sin\theta \\ L & -L \end{bmatrix}\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} + \begin{bmatrix} \sin\theta \\ -\cos\theta \\ 0 \end{bmatrix}\lambda \tag{1}$$

Where τ_1 and τ_2 are input torques of left and right wheels respectively, m and I are the mass and inertia of robot, R is the radius of the wheel, L is the distance of rear wheels, λ is the Lagrange multipliers of constrained forces [4].

The nonholonomic constraint equation is written as

$$\dot{x}\sin\theta - \dot{y}\cos\theta = 0 \tag{2}$$

Assuming $\tau_l = (\tau_1 + \tau_2)/R$ and $\tau_a = L(\tau_1 - \tau_2)/R$, then the dynamic equation can be transformed to

$$\begin{cases} \ddot{x} = \tau_l \cos\theta / m + \lambda \sin\theta / m \\ \ddot{y} = \tau_l \sin\theta / m - \lambda \cos\theta / m \\ \ddot{\theta} = \tau_a / I \end{cases} \tag{3}$$

Where τ_l and τ_a are linear force and angular torque respectively.

Assuming v and w are the linear velocity and angular velocity of robot, the following transformation is obtained:

$$\dot{q} = g(q)\begin{bmatrix} v \\ w \end{bmatrix} \tag{4}$$

Where $g(q) = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix}$

Then the differential equation can be written as:

$$\ddot{q} = \dot{g}(q)\begin{bmatrix} v \\ w \end{bmatrix} + g(q)\begin{bmatrix} \dot{v} \\ \dot{w} \end{bmatrix} \tag{5}$$

Therefore

$$\begin{cases} \ddot{x} = -v\dot{\theta}\sin\theta + \dot{v}\cos\theta \\ \ddot{y} = v\dot{\theta}\cos\theta + \dot{v}\sin\theta \\ \ddot{\theta} = \dot{w} \end{cases} \tag{6}$$

Comparing Equation (3) and Equation (6), we can obtain

$$\begin{cases} \lambda \sin\theta / m + \tau_l \cos\theta / m = -v\dot{\theta}\sin\theta + \dot{v}\cos\theta \\ -\lambda \cos\theta / m + \tau_l \sin\theta / m = v\dot{\theta}\cos\theta + \dot{v}\sin\theta \\ \tau_a / I = \dot{w} \end{cases} \tag{7}$$

Multiplying the first part of Equation (7) by $\cos\theta$ and the second part by $\sin\theta$ and adding the result the following is obtained

$$\dot{v} = \tau_l / m, \quad \dot{w} = \tau_a / I \tag{8}$$

According to Equation (4), we can obtain:

$$\begin{cases} \dot{x} = v\cos\theta \\ \dot{y} = v\sin\theta \\ \dot{\theta} = w \end{cases} \tag{9}$$

Equation (8) and (9) are the equations of dynamic model and kinematic model equations.

3 Nonlinear Control Model

We can use nonlinear kinematic controller to stabilize the configuration variables. Tracking control of mobile robot is simply reduced to regularization problem of error variables in kinematic model. A path planner defines the reference trajectory as a time variant pose vector: $q_r = (x_r \ y_r \ \theta_r)^T$. This trajectory should satisfy not only the kinematic equations but also the nonholonomic constraint [5]:

$$\dot{x}_r = v_r \cos\theta_r, \quad \dot{y}_r = v_r \sin\theta_r, \quad \dot{\theta}_r = w_r, \quad \dot{x}_r \sin\theta_r = \dot{y}_r \cos\theta_r \tag{10}$$

The error dynamics is written independent of the inertial coordinate frame by Kanayama transformation [6]:

$$\begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix} \tag{11}$$

(x_e, y_e, θ_e) are the error variables in mobile coordinate system which is attached to the robot. Differentiating left hand side of Equation (11), (10) and (2) the error dynamics is written in the new coordinate system:

$$\begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\theta}_e \end{bmatrix} = \begin{bmatrix} v_r \cos\theta_e \\ v_r \sin\theta_e \\ w_r \end{bmatrix} + \begin{bmatrix} -1 & y_e \\ 0 & -x_e \\ 0 & -1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \tag{12}$$

Where $(v, w)^T$ is the control vector of the kinematic model.

We construct control vector using direct Lyapunov method. The constructive Lyapunov function is:

$$V = \frac{1}{2}\left(x_e^2 + y_e^2\right) + \left(1 - \cos\theta_e\right) \tag{13}$$

Time derivative of Equation (13) becomes:

$$\begin{aligned} \dot{V} &= v_r x_e \cos\theta_e - v_d x_e + v_r \sin\theta_e y_e + w_r \sin\theta_e - w_d \sin\theta_e \\ &= (v_r \cos\theta_e - v_d) x_e + \sin\theta_e (v_r y_e + w_r - w_d) \end{aligned} \tag{14}$$

Assuming v_d and w_d are the desired velocities to make the kinematics stable, they are chosen as follow to make $\dot{V}$ negative definite:

$$\begin{cases} v_d = v_r \cos\theta_e + k_x x_e \\ w_d = w_r + v_r y_e + k_\theta \sin\theta_e \end{cases} \tag{15}$$

Where k_x and k_θ are positive reals.

Substituting Equation (15) in Equation (14):

$$\dot{V} = -k_x x_e^2 - v_r k_\theta \sin^2 \theta_e \tag{16}$$

It's clear that $\dot{V}$ is only negative semi definite.

Using LaSalle principle, convergence of x_e, y_e and θ_e to zeros is guaranteed, so the closed loop system is globally asymptotically stable.

Control laws designed according Lyapunov principle can make the robot stabilization, but the dynamic performance is bad, and the robot can not track the path accurately under noisy environment. In order to optimize the dynamic performance of the robot, SVM algorithm is used to design another controller based nonlinear controller.

4 SVM Controller

Originally, SVM was developed for classification problems. It was then extended to regression estimation problems [7]. For regression problem, the basic idea is to map the data to a higher dimensional feature space, via a nonlinear mapping, and then to do the linear regression in this space [8]. Therefore given a training set of training samples $\{x_i, y_i\}_{i=1}^{l} \subset R^n \times R$, we introduce a nonlinear mapping $\varphi(\cdot): R^n \to R^h$, which maps the training samples to a new data set. In ε-insensitive support vector regression the goal is to estimate the following function:

$$\hat{f}(x) = w^T \varphi(x) + b \tag{17}$$

Where $w \in R^{n_h}$ is weight vector, $b \in R$ is threshold. $\hat{f}(x)$ can estimate input x which is not in training set, and give the output y.

Estimation problem can be described as the following optimization problem:

$$\begin{aligned} &\min_{w,b,\xi^*,\xi} J_\varepsilon(w,\xi^*,\xi) = \frac{1}{2} w^T w + \gamma \left\{ \sum_{i=1}^{N} \xi_i^* + \sum_{i=1}^{N} \xi_i \right\} \\ &s.t. \quad \begin{cases} y_i - w^T \varphi(x_i) - b \le \varepsilon + \xi_i^* & i = 1,\dots,N \\ -y_i + w^T \varphi(x_i) + b \le \varepsilon + \xi_i & i = 1,\dots,N \\ \xi_i^* \ge 0 & i = 1,\dots,N \\ \xi_i \ge 0 & i = 1,\dots,N \end{cases} \end{aligned} \tag{18}$$

Where ξ_i and ξ_i^* are slack variables and γ is a positive real constant. One obtains $w = \sum_{i=1}^{N} (\alpha_i^* - \alpha_i) \varphi(x_i)$ where α_i and α_i^* are the Lagrange multipliers related to the first and second set of constraints. The data points corresponding to non-zero values for $(\alpha_i^* - \alpha_i)$ are called support vectors.

Finally, one obtains the following model in the dual space

$$\hat{f}(x) = \sum_{i=1}^{N} (\alpha_i^* - \alpha_i) K(x_i, x) \tag{19}$$

Where the kernel function K corresponds to

$$K(x_i, x) = \varphi(x_i)^T \varphi(x) \tag{20}$$

One has several possibilities for the choice of this kernel function, including linear, polynomial, splines, RBF.

To the ε -insensitive loss function

$$J_{\varepsilon,p}(w, \xi^*, \xi) = \frac{1}{2} w^T w + \gamma \left\{ \sum_{i=1}^{N} (\xi_i^*)^p + \sum_{i=1}^{N} (\xi_i)^p \right\} \tag{21}$$

Where p=1 corresponds to Eq. (21), we employ a least squares version of the support vector method for function estimation (LS-SVM), it corresponds to p=2 and the following form of ride regression:

$$\begin{aligned} &\min_{w,b,\xi} J_{LS}(w,b,\xi) = \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_{i=1}^{N} \xi_i^2 \\ &s.t. \quad y_i = w^T \varphi(x_i) + b + \xi_i \end{aligned} \tag{22}$$

One defines the Lagrangian

$$J_{LS}(w,b,\xi;\alpha) = J_{LS}(w,b,\xi) - \sum_{i=1}^{N} \alpha_i (w^T \varphi(x_i) + b + \xi_i - y_i) \tag{23}$$

Where α_i are Lagrange multipliers, it can be positive or negative due to equality constrains as follows from the Kuhn-Tucker conditions. The conditions for optimality

$$\begin{cases} \partial J_{LS} / \partial w = 0 \rightarrow w = \sum_{i=1}^{N} \alpha_i \varphi(x_i) \\ \partial J_{LS} / \partial b = 0 \rightarrow w = \sum_{i=1}^{N} \alpha_i = 0 \\ \partial J_{LS} / \partial \xi_i = 0 \rightarrow \alpha_i = \gamma \xi_i & i = 1, \ldots, N \\ \partial J_{LS} / \partial \alpha_i = 0 \rightarrow w^T \varphi(x_i) + b + \xi_i - y_i = 0 & i = 1, \ldots, N \end{cases} \tag{24}$$

Equation (24) can be written as the solution to the following set of linear equations after elimination of w and ξ_i .

$$\begin{bmatrix} 0 & \vec{1}^T \\ \vec{1} & \Omega + \gamma^{-1} I \end{bmatrix} \begin{bmatrix} x \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix} \tag{25}$$

With $x = [x_1, \ldots, x_N]$, $y = [y_1, \ldots, y_N]$, $\vec{1} = [1, \ldots, 1]$, $\alpha = [\alpha_1, \ldots, \alpha_N]$, $\Omega_{ij} = K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j)$.

LS-SVM can be used in many applications in identification and control theory such as in the context of prediction error algorithms [9]. In this paper, we use LS-SVM because of the equality constraints in the problem formulation.

Considering Eq. (8) and tracking error, the following control laws are used to prepare tracking of v_d and w_d:

$$\begin{cases} \tau_l = m\dot{v}_d + k_{le} v_e + k_{ld} \dot{v}_e \\ \tau_a = I\dot{w}_d + k_{ae} w_e + k_{ad} \dot{w}_e \end{cases} \tag{26}$$

Where k_{le}, k_{ld}, k_{ae} and k_{ad} are the weights of v_e, $\dot{v}_e$, w_e and $\dot{w}_e$, they are unknown control parameters which will be estimated by LS-SVM.

In order to have a proper performance of SVM, we need to select as many samples as possible for training, however the dimension of SVM will greatly increase in the process of on-line training. Based on the aim of designing a controller which depends on current state of the nonlinear dynamic system, the training data collected earlier might not suit for real-time system, the large data set might lead to time consuming calculation. Therefore, sliding time window is constructed by N with sample time interval, then sample data is collected orderly from current to past. Moreover, a new data sample is collected while the oldest data being dropped. We assume that the nearest data can more properly describe the feature of the system than the oldest data.

For any given continuous real function $f(x)$ on compact set, for a large enough length of slide time window N combined with properly selected sampling time interval and any given $\varepsilon > 0$, there exists an SVM approximation function $\hat{f}(x)$ formed by (19) such that $\sup_{t\in[T,T-N+1]} \left| f(x) - \hat{f}(x) \right| \le \varepsilon$, where T denotes the current time[10].

5 Structure of Control System

The structure of the control system is shown in Fig.2. The control system includes two controllers: nonlinear controller and SVM controller. Nonlinear controller's primary role is to obtain the desired velocity v_d and w_d which can make the Kinematics stable, its inputs are the dynamic errors of reference position and actual position. SVM controller's primary role is to track the desired velocity, its inputs are $(\dot{v}_d, \dot{w}_d)$, (v_e, w_e) and $(\dot{v}_e, \dot{w}_e)$, its output are τ_l and τ_a (force and angular torque respectively).

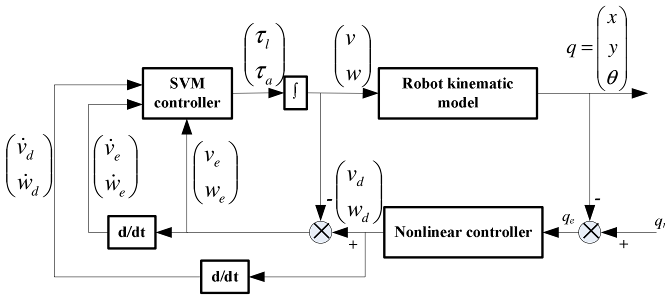

Fig. 2. Block diagram of Control System

6 Simulation

In order to validate the effectiveness of proposed method, simulations were carried out when the robot was disturbed by noise. The structure parameters of the robot are: m=8kg, $I = 2\text{kg}\cdot\text{m}^2$, R=20cm, and L=0.2m; the weights k_x and k_θ of the errors of the nonlinear controller are 0.4 and 0.2; the kernel function is RBF; the initial pose vector $(x, y, \theta)^T$ is $(0,0,0)^T$; the length of sliding time window is two seconds; the sampling frequency of pose position is 500Hz. When the reference track is line, sinusoid and circle, the results of simulation are showed in Fig. 3 to Fig. 5.

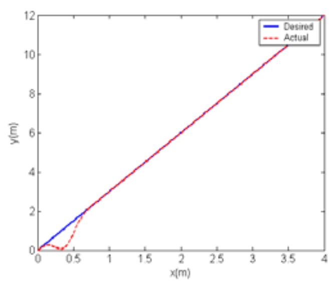

Fig. 3. Tracking of line

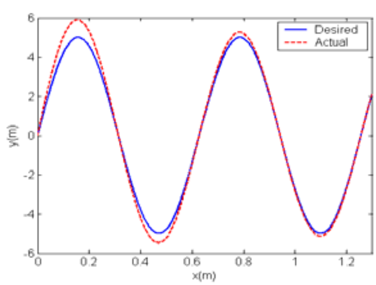

Fig. 4. Tracking of sinusoid

Error of tracking in the beginning is large because the parameters of SVM controller are not adaptive to the environment and the robot, then the error begin to decrease rapidly. The error will converge to zero if the time is long enough.

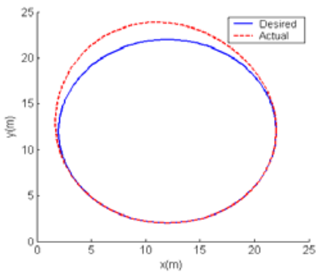

Fig. 5. Tracking of circle

7 Conclusion

In this paper, a novel control method was proposed for tracking of mobile robot. The controller includes two consecutive parts, one is nonlinear kinematic controller to obtain the desired velocity which can make the system stable and the other is the SVM controller to provide tracking of desired linear velocity and angular velocity. The main characteristic of the proposed controller is its robustness of performance against the environment disturbed by noise. Simulation results demonstrate that the system is able to track reference signals satisfactorily.

References

[1] Chen, Q., Ozguner, U., Redmill, K.: Ohio state university at the 2004 DARPA grand challenge: Developing a completely autonomous vehicle. IEEE Intelligent System 19(5), 8–11 (2004)

[2] Das, T., Kar, I.N., Chaudhury, S.: Simple neuron-based adaptive controller for a nonholonomic mobile robot including actuator dynamics. Neurocomputing 69(16-18), 2140–2151 (2006)

[3] Fierro, R., Lewis, F.L.: Control of a nonholonomic mobile robot using neural networks. IEEE Trans. Neural Networks 9(4), 589–600 (1998)

[4] Bloch, A.M., Reyhanoglu, M., McClamroch, N.H.: Control and stabilization of nonholonomic dynamic system. IEEE Transactions on Automatic Control 37(11), 1746–1757 (1992)

[5] Gholipour, A., Dehghan, S.M., Nili, M., Ahmadabadi: Lyapunov based tracking control of nonholonomic mobile robot. In: Proc. of 10th Iranian Conference on Electrical Engineering, pp. 262–269 (2002)

[6] Kanayama, Y., Kimura, Y., Miyazaki, F., Noguchi, T.: A stable tracking control scheme for an autonomous mobile robot. In: Proc. 1990 IEEE Int. Conf. Rob. Autom., pp. 384–389 (1990)

[7] Vapnik, V.: An overview of statistical learning theory. IEEE Trans. on Neural Networks 10(5), 988–999 (1999)

[8] Vapnik, V., Golowich, S.: Support vector method for function approximation, regression estimation, and signal processing. In: Advances in Neural Information Processing Systems, pp. 281–287 (1997)
[9] Suykens, J.A.K., Vandewalle, J., De Moor, B.: Optimal control by least squares support vector machines. Neural Networks 14(1), 23–35 (2001)
[10] Li, Z., Kang, Y.: Dyanmic coupling switching control incorporating Support Vector Machine wheeled mobile manipulators with hybrid joints. Automatic 46(5), 832–842 (2010)