Motion Control of an Omnidirectional Mobile Robot

Xiang Li and Andreas Zell

Wilhelm-Schickard-Institute, Departmant of Computer Architecture
University of Tübingen, Sand 1, 72076 Tübingen, Germany
{xiang.li, andreas.zell}@uni-tuebingen.de

Abstract. This paper focuses on the motion control problem of an omnidirectional mobile robot. A new control method based on the inverse input-output linearized kinematic model is proposed. As the actuator saturation and actuator dynamics have important impacts on the robot performance, this control law takes into account these two aspects and guarantees the stability of the closed-loop control system. Real-world experiments with an omnidirectional middle-size RoboCup robot verify the performance of this proposed control algorithm.

1 Introduction

Recently, omnidirectional wheeled robots have received more attention in mobile robots applications, because they have full mobility in the plane, which means that they can move at each instant in any direction without any reorientation [1]. Unlike nonholonomic robots, such as car-like robots, having to rotate before implementing any desired translation velocity, omnidirecitonal robots have higher maneuverability and are widely used in dynamic environments, for example, in the middle-size league of the annual RoboCup competition.

Most motion control methods of mobile robots are based on dynamic models [2–5] or kinematic models [6–8] of robots. A dynamic model directly describes the relationship between the forces exerted by the wheels and the robot movement, with the applied voltage of each wheel as the input and the robot movement in terms of linear and angular accelerations as the output. But the dynamic variations caused by the changes in the robot's inertia moment and perturbations from the mechanic components [9] make the controller design more complex. With the assumption that no slippage of wheels occurs, sensors have high accuracy and ground is planar enough, kinematic models are widely used in designing robots behaviors because of the simpler model structures. As the inputs of kinematic models are robot wheels velocities, and outputs are the robot linear and angular velocities, the actuator dynamics of the robot are assumed to be fast enough to be ignored, which means that the desired wheel velocities can be achieved immediately. However, the actuator dynamics limit and even degrade the robot performance in real situations.

Another important practical issue of robot control is actuator saturation. Because the commanding motor speeds of the robot wheels are bounded by the saturation limits, the actuator saturation can affect the robot performance, even destroy the stability of the controlled robot systems [10, 11].

This paper presents a motion control method for an omnidirectional robot, based on the inverse input-output linearization of the kinematic model. It takes into account not only the identified actuator dynamics but also the actuator saturation in designing a controller, and guarantees the stability of the closed-loop control system.

The remainder of this paper introduces the kinematic model of an omnidirectional middle-size Robocup robot in section 2; Path following and orientation tracking problems are solved based on the inverse input-output linearized kinematic model in section 3, where the actuator saturation is also analyzed; section 4 presents the identification of actuator dynamics and their influence on the control performance. Finally, the experiment results and conclusions are discussed in sections 5 and 6, respectively.

2 Robot Kinematic Model

The mobile robot used in our case is an omnidirectional robot, whose base is shown in Fig. 1. It has three Swedish wheels mounted symmetrically with 120 degrees from each other. Each wheel is driven by a DC motor and has a same distance L from its center to the robot's center of mass R.

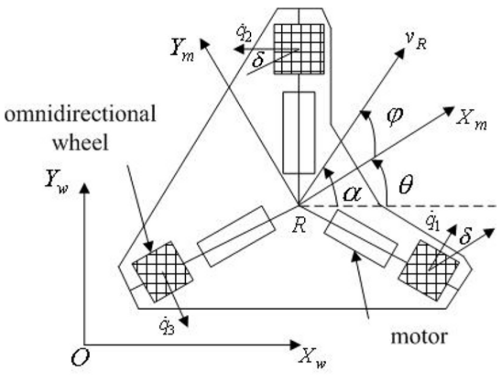

Fig. 1. Kinematics diagram of the base of an omnidirectional robot.

Besides the fixed world coordinate system $[X_w, Y_w]$, a mobile robot fixed frame $[X_m, Y_m]$ is defined, which is parallel to the floor and whose origin locates at R. θ denotes the robot orientation, which is the direction angle of the axis X_m in the world coordinate system. α and φ denote the direction of the robot translation velocity v_R observed in the world and robot coordinate system, respectively. The kinematic model with respect to the robot coordinate system is given by :

$$\mathbf{v} = \begin{bmatrix} \sqrt{3}/3 & -\sqrt{3}/3 & 0 \\ 1/3 & 1/3 & -2/3 \\ 1/(3L) & 1/(3L) & 1/(3L) \end{bmatrix} \dot{\mathbf{q}}, \tag{1}$$

where $\mathbf{v} = [\dot{x}_R^m \; \dot{y}_R^m \; \omega]^T$ is the vector of robot velocities observed in the robot coordinate system; $\dot{x}_R^m$ and $\dot{y}_R^m$ are the robot translation velocities; ω is the robot rotation velocity.

$\dot{\mathbf{q}}$ is the vector of wheel velocities $[\dot{q}_1\ \dot{q}_2\ \dot{q}_3]^T$, and $\dot{q}_i(i = 1, 2, 3)$ is the i-th wheel's velocity, which is equal to the wheel's radius multiplied by the wheel's angular velocity.

Introducing the transformation matrix from the robot coordinate system to the world coordinate system as

$$^{w}R_m = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, \tag{2}$$

the kinematic model with respect to the world coordinate system is deduced as:

$$\dot{\mathbf{x}} = \begin{bmatrix} \frac{2}{3}cos(\theta+\delta) & -\frac{2}{3}cos(\theta-\delta) & \frac{2}{3}sin\theta \\ \frac{2}{3}sin(\theta+\delta) & -\frac{2}{3}sin(\theta-\delta) & -\frac{2}{3}cos\theta \\ \frac{1}{3L} & \frac{1}{3L} & \frac{1}{3L} \end{bmatrix} \dot{\mathbf{q}}, \tag{3}$$

where $\dot{\mathbf{x}} = [\dot{x}_R\ \dot{y}_R\ \dot{\theta}]^T$ is the vector of robot velocities with respect to the world coordinate system; $\dot{x}_R$ and $\dot{y}_R$ are the robot translation velocities; $\dot{\theta}$ is the robot rotation velocity; δ refers to the wheel's orientation in the robot coordinate system and is equal to 30 degrees.

It is important to notice that the transformation matrix in the kinematic models is full rank, which denotes that the translation and rotation of the robot are decoupled, and guarantees the separate control of these two movements.

For the high level control laws without considering the wheel velocities, the kinematic model

$$\dot{\mathbf{x}} = G\mathbf{v} \tag{4}$$

is used in our control method, where the transformation matrix G is equal to $[^{w}R_m\ 0\ ;\ 0\ 1]$. Because G is full rank, the characteristics of decoupled movement is also kept.

3 Inverse Input-Output Linearization based Control

The trigonometric functions of angle θ in the transformation matrix G determine the nonlinearities of the kinematic model (4). Since the matrix G is full rank, this nonlinear model can be exactly linearized by introducing a simple compensator $C = G^{-1}$. The linearized system becomes $\dot{\mathbf{x}} = \mathbf{u}$ with a new input vector $\mathbf{u} = [u_1\ u_2\ u_3]^T$.

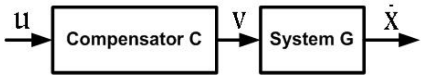

Fig. 2. Linearized system by the compensator C.

This linear system shown in Fig. 2 is completely decoupled and allows the controlling of the robot's translation and rotation in a separate way. When a controller K is designed based on this simple linear system, the controller of the original system is generated as CK. The overall control loop, which consists of the nonlinear system, the compensator and the controller, is shown in Fig. 3,

where $\mathbf{x}$ denotes the robot state vector $[x_R\ y_R\ \theta]^T$ and $\mathbf{x_d}$ is the desired state vector; x_R and y_R are robot position observed in the world coordinate system.

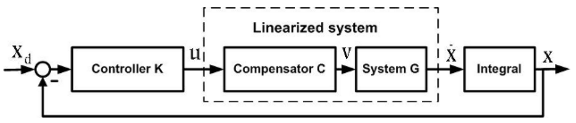

Fig. 3. Closed-loop control system.

Based on this input-output linearized system, path following and orientation tracking problems are analyzed with respect to the robot translation and rotation control in the following subsections. The influence of actuator saturation is also accounted to keep the decoupling between the translation and rotation movements.

3.1 Path Following Control

As one high-level control problem, path following is chosen in our case to deal with the robot translation control. The path following problem is illustrated in Fig. 4. P denotes the given path. Point Q is the orthogonal project of R on the path P. The path coordinate system x_tQx_n moves along the path P and the coordinate axes x_t and x_n direct the tangent and normal directions at point Q, respectively. θ_P is the path tangent direction at point Q.

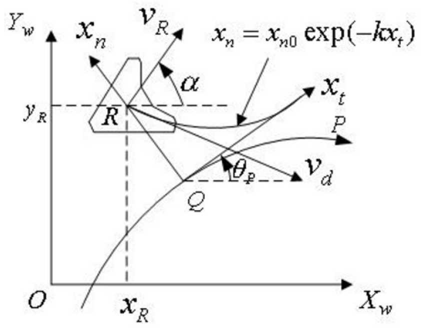

Fig. 4. Illustration of the path following problem.

Based on the above definitions, the path following problem is to find proper control values of the robot translation velocity v_R and angular velocity $\dot{\alpha}$ such that the deviation distance x_n and angular error $\tilde{\theta}_R = \alpha - \theta_P$ tend to zero.

To solve this problem, a Lyapunov candidate function

$$V = \frac{1}{2}K_d x_n^2 + \frac{1}{2}K_\theta \tilde{\theta}_R^2 \tag{5}$$

can be considered, where K_d and K_θ are positive constants. The time derivation of V results in

$$\dot{V} = K_d x_n \dot{x}_n + K_\theta \tilde{\theta}_R \dot{\tilde{\theta}}_R. \tag{6}$$

Mojaev [12] presents a simple control law based on the deviation x_n, where R is controlled to move along an exponential curve and to converge to the axis x_t. The exponential curve is expressed as

$$x_n = x_{n_0} exp(-kx_t), \tag{7}$$

where x_{n_0} is the initial deviation and the positive constant k determines the convergence speed of the deviation. Differentiating (7) with respect to x_t, we get the tangent direction of the exponential curve as

$$\tilde{\theta}_R = \arctan(\frac{dx_n}{dx_t}) = \arctan(-kx_n). \tag{8}$$

Therefore, for a non-zero constant desired velocity v_d, the translation velocity of robot in the coordinate system $x_t Q x_n$ results in

$$\dot{x}_n = v_d sin\tilde{\theta}_R, \tag{9}$$

$$\dot{x}_t = v_d cos\tilde{\theta}_R. \tag{10}$$

Substituting the time derivative of $\tilde{\theta}_R$ into (6), we get

$$\dot{V} = K_d x_n \dot{x}_n + kK_\theta \arctan(-kx_n)\frac{-\dot{x}_n}{1+(kx_n)^2} < 0, \tag{11}$$

because $x_n\dot{x}_n = x_n v_d \sin(\arctan(-kx_n)) < 0$ and $\dot{x}_n \arctan(kx_n) < 0$. This solution of $\dot{V}$ guarantees the global stability of the equilibrium at $x_n = 0, \tilde{\theta}_R = 0$, which means this control law solves the path following problem.

Transforming the robot velocity into the world coordinate system, we get the control values of the linearized system as

$$u_1 = v_d \cos\alpha, \tag{12}$$

$$u_2 = v_d \sin\alpha, \tag{13}$$

where $\alpha = \tilde{\theta}_R + \theta_P$.

The input of controller (12) and (13) is the deviation distance between point R and the given path, which normally can be directly obtained by the sensors on the robot. Moreover, the deviation converges smoothly to zero with the speed controlled by parameter k, which can be chosen according to the performance requirement.

3.2 Orientation Tracking

Unlike a car-like wheeled robot, the orientation of an omnidirectional robot can be different from the direction of the robot translation velocity by any angle φ. This relationship is denoted as $\alpha = \theta + \varphi$. That means the robot orientation can track any angle when the robot is following a given path. Based on the linearized model, the orientation tracking task is to find a suitable u_3, which is equal to the robot rotation velocity ω, such that

$$\lim_{t\longrightarrow\infty}(\theta_d(t) - \theta(t)) = 0, \tag{14}$$

where $\theta_d(t)$ is the desired orientation.

As the system between input variable u_3 and output variable θ is an integrator, a commonly used PD controller can be designed to fulfill the orientation tracking task.

3.3 Actuator Saturation

Based on the inverse input-output linearization, the translation and rotation of an omnidirectional robot can be easily achieved in a separate way. This linearization is with respect to the input-output relationship, which requires the internal parts having sufficient capability to achieve the desired inputs. However, the power of the robot motors is bounded and the actuators will saturate when the commanding velocities are too large. The presence of actuator saturation can influence the decoupling between robot translation velocity and rotation velocity, such that the system performance and stability is severely impacted. Therefore, it is necessary to deal with the actuator saturation in the controller design.

For our omnidirectional robot, the maximal velocity of each wheel is limited by $\dot{q}_m$, namely $|\dot{q}_i| \leq \dot{q}_m$. Substituting the above control values from equations (12) (13) and u_3 into the inverse kinematic models (2) and (1), the wheel velocities are computed as:

$$\begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \begin{bmatrix} v_d cos(\alpha - \theta - \delta) + Lu_3 \\ -v_d cos(\alpha - \theta + \delta) + Lu_3 \\ v_d sin(\theta - \alpha) + Lu_3 \end{bmatrix}, \tag{15}$$

To achieve orientation tracking based on the above path following control, the desired translation velocity's magnitude V_d is assumed to be not bigger than $\dot{q}_m$. Substituting $\dot{q}_m$ into (15),

$$|v_d cos(\alpha - \theta - \delta) + Lu_3| \leq \dot{q}_m \tag{16}$$

$$|-v_d cos(\alpha - \theta + \delta) + Lu_3| \leq \dot{q}_m \tag{17}$$

$$|v_d sin(\theta - \alpha) + Lu_3| \leq \dot{q}_m, \tag{18}$$

the lower and upper boundary of u_3 with respect to each wheel (l_{b_i} and u_{b_i}, $i = 1, 2, 3$) can be calculated as follows,

$$l_{b_1} = -\dot{q}_m - v_d cos(\alpha - \theta - \delta) \leq Lu_3 \leq \dot{q}_m - v_d cos(\alpha - \theta - \delta) = u_{b_1} \tag{19}$$

$$l_{b_2} = -\dot{q}_m + v_d cos(\alpha - \theta + \delta) \leq Lu_3 \leq \dot{q}_m + v_d cos(\alpha - \theta + \delta) = u_{b_2} \tag{20}$$

$$l_{b_3} = -\dot{q}_m - v_d sin(\theta - \alpha) \leq Lu_3 \leq \dot{q}_m - v_d sin(\theta - \alpha) = u_{b_3}. \tag{21}$$

Then the dynamic boundary values of u_3 are computed as

$$\begin{aligned} l_b &= max(l_{b_1}, l_{b_2}, l_{b_3})/L \\ u_b &= min(u_{b_1}, u_{b_2}, u_{b_3})/L, \end{aligned} \tag{22}$$

where l_b and u_b are the low and up boundary.

Considering the saturation function

$$x_2 = \begin{cases} u_b, if & x_1 > u_b \\ x_1, if & l_b \leq x_1 \leq u_b \\ l_b, if & x_1 < l_b, \end{cases} \tag{23}$$

and its gain characteristics illustrated in Fig. 5, we can take the saturation function as a dynamic gain block k_a, which has maximum value one and converges to zero when the input saturates. Then the closed-loop system of controlling the robot orientation is as shown in Fig. 6, in which a PD controller is used to control the robot orientation converging to the ideal θ_d,

$$\omega = k_1(e_\theta + k_2\dot{e}_\theta), \tag{24}$$

where $e_\theta = \theta_d - \theta$, k_1 and k_2 are the proportional and derivative gains, respectively. It can be obtained that the closed-loop has only one pole $\frac{-k_a k_1}{1+k_a k_1 k_2}$ and one zero $-1/k_2$. Therefore, when k_2 and k_1 are positive, the stability of the closed-loop system can be guaranteed whenever k_a decreases.

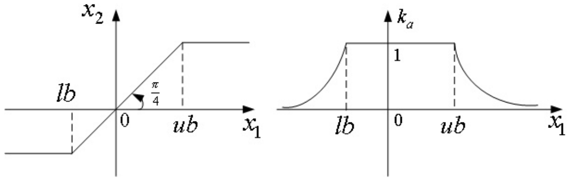

Fig. 5. Saturation function and its gain characteristics.

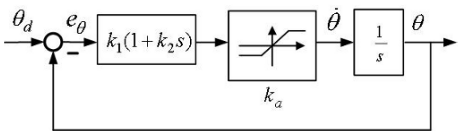

Fig. 6. Closed-loop of robot orientation control.

4 Actuator Dynamics

The results in the last section are only practical when we assume that the low level actuator dynamics is faster than the kinematics, or the delay of actuator dynamics can be ignored. Therefore, it is necessary to analyze the actuator dynamics and take it into account when designing a controller. In the following subsections, the actuator dynamics is identified based on the observed input-output data, and its influence on the robot motion control is presented.

4.1 Actuator Dynamics Identification

The system identification problem is to estimate a model based on the observed input-output data such that a performance criterion is minimized. Because the full rank transformation matrix in the low level dynamics model (1) denotes the outputs $\dot{x}_R^m$, $\dot{y}_R^m$ and ω are not relevant, we identify the actuator models for these three values. The inputs of the actuator models are required velocity values($\dot{x}_{R_c}^m$, $\dot{y}_{R_c}^m$ and ω_c), and the outputs are corresponding measured values. As one commonly used parametric model, ARMAX is chosen as the identified model, which has the following structure

$$A(z)y(t) = B(z)u(t - n_k) + C(z)e(t), \tag{25}$$

$$A(z) = 1 + a_1 z^{-1} + ... + a_{n_a} z^{-n_a}, \tag{26}$$

$$B(z) = 1 + b_1 z^{-1} + ... + b_{n_b} z^{-n_b+1}, \tag{27}$$

$$C(z) = 1 + c_1 z^{-1} + ... + c_{n_c} z^{-n_c}. \tag{28}$$

n_k denotes the delay from input $u(t)$ to output $y(t)$. $e(t)$ is white noise. z is the shift operator resulting in $q^{-1}u(t) = u(t-1)$. n_a, n_b and n_c are the orders of polynomials $A(z)$, $B(z)$ and $C(z)$, respectively. To choose the optimal parameters of this model, we use the prediction error method, which is to find the optimal n_k and parameters of $A(z)$, $B(z)$ and $C(z)$ such that the prediction error E is minimized, namely

$$[A(z), B(z), C(z), n_k]_{opt} = argmin \sum_{t=1}^{N} E^2 \tag{29}$$

$$E = y_o(t) - A^{-1}(z)(B(z)u(t - n_k) + C(z)e(t)), \tag{30}$$

where $y_o(t)$ denotes the measured output data.

The system identification toolbox of Matlab has been used to identify the actuator dynamics model. Figures 7, 8 and 9 show the optimal parameters and the comparison between models outputs and measured outputs with respect to the actual inputs.

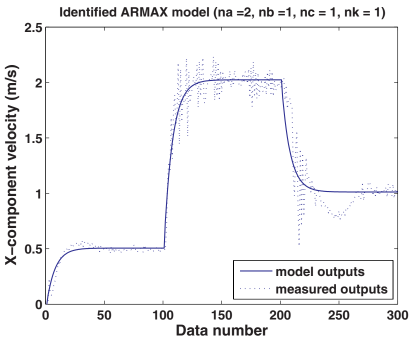

Fig. 7. Identified model for $\dot{x}_R^m$.

To coincide with the robot's continuous model, the identified models are transformed from discrete ones into continuous ones using the 'zoh'(zero-order hold) method,

$$\dot{x}_R^m = \frac{8.7948(s + 58.47)}{(s + 73.66)(s + 6.897)} \dot{y}_{R_c}^m, \tag{31}$$

$$\dot{y}_R^m = \frac{2.4525(s + 48.83)(s + 6.185)}{(s + 28.45)(s^2 + 6.837s + 25.97)} \dot{y}_{R_c}^m, \tag{32}$$

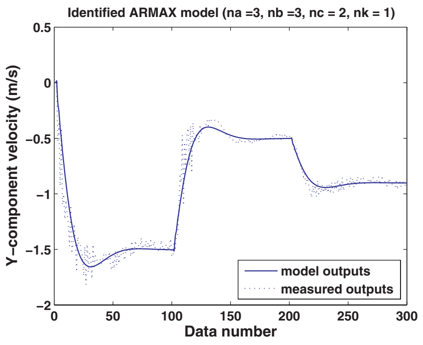

Fig. 8. Identified model for $\dot{y}_R^m$.

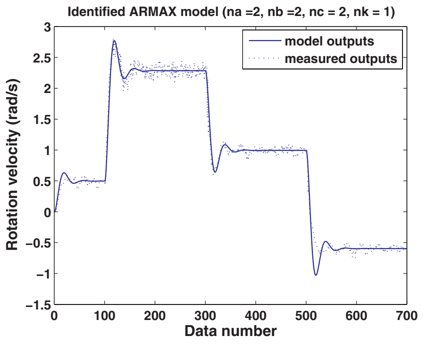

Fig. 9. Identified model for ω.

$$\omega = \frac{1.667(s + 45.37)}{(s^2 + 6.759s + 76.11)}\omega_c. \tag{33}$$

4.2 Actuator Influence

With consideration of the actuator, the whole structure of the control system is shown in Fig. 10, where $\mathbf{v}_c = [\dot{x}_{Rc}^m \ \dot{y}_{Rc}^m \ \omega_c]$ is the commanding robot velocity vector with

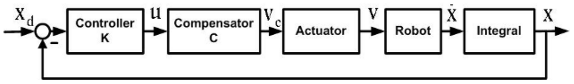

Fig. 10. Closed-loop control system including actuator dynamics.

respect to the robot coordinate system. Because the poles of the actuators dynamics (31) and (32) have negative real parts, these two systems are stable. That means there exits a finite short time t^*, after which the real velocities $\dot{x}_R^m$ and $\dot{y}_R^m$ can converge to the desired ones $\dot{x}_{Rc}^m$ and $\dot{y}_{Rc}^m$, and the inputs u_1 and u_2 begin to take effect. Therefore, the above path following law can also guarantee the robot approach to the reference path, although during t^* the deviation distance x_n and angular error $\tilde{\theta}_R$ may increase.

In the orientation tracking control, as the dynamic system (33) adds another two poles to the closed-loop system, shown in Fig. 11, the controller parameters decided in the above section may result the system losing the stability.

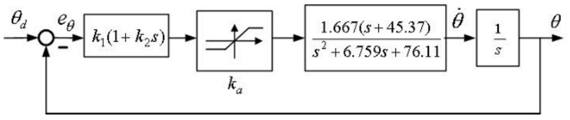

Fig. 11. Closed-loop of robot orientation control including actuator dynamics.

By setting the positions of poles and zeros of the closed-loop system with the locus technique, we obtain that the conditions $k_1 > 0$ and $k_2 > 0.0515$ can guarantee the closed-loop system's stability, even when the actuators saturate. Fig. 12 shows the root locus of an open-loop system in the critical situation with $k_2 = 0.0515$, where all the poles of the closed-loop system locate in the left-half plane whatever positive value $k_a k_1$ is. Otherwise, when k_2 is less than 0.0515, the root locus may cross the imaginary axis, and the poles of closes-loop system may move to the right-half plane when k_a goes to zero.

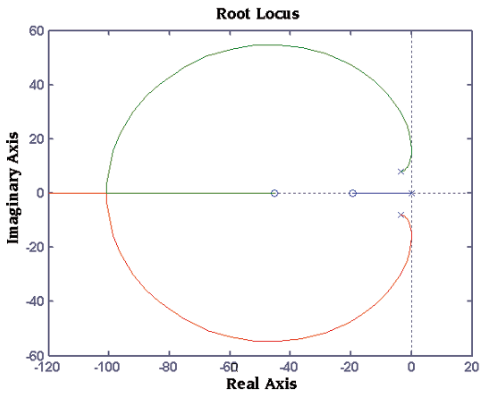

Fig. 12. Root locus of the open-loop model.

5 Experiment

The control algorithm discussed above has been tested in our robot laboratory having a half-field of the RoboCup middle size league. The omnidirectional robot is shown in Fig. 13.

An AVT Marlin F-046C color camera with a resolution of 780×580 is assembled pointing up towards a hyperbolic mirror, which is mounted on the top of the omnidirectional robot, such that a complete surrounding map of the robot can be captured. A

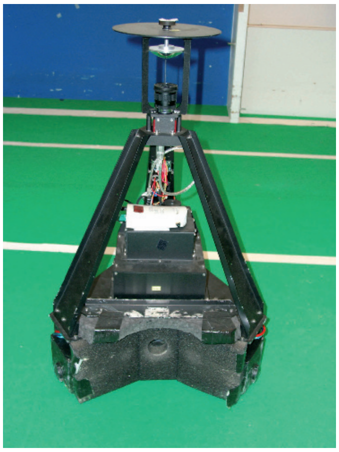

Fig. 13. The real omnidirectional robot.

self-localization algorithm described in [13] based on the 50 Hz output signal of the camera gets the robot's position in the play field in real time. The wheels are driven by three $60W$ Maxon DC motors and the maximum wheel velocity is $1.9m/s$. Three wheel encoders measure the real wheel velocities, which are steered by three PID controllers.

An eight-shaped path is adopted as the reference path, while its geometrical symmetry and sharp changes in curvature make the test challenging. With a scale variable s, the chosen eight-shaped path is calculated as

$$\begin{aligned} x_r &= 1.8sin(2s) \\ y_r &= 1.2sin(s), \end{aligned} \tag{34}$$

The robot was controlled to follow the eight-shaped path with a constant translation velocity $v_d = 1m/s$, and the parameters of our control algorithm were chosen as $k = 2.5,\ k_1 = 4.15,\ k_2 = 3$. The first experiment selected the path tangent direction θ_p as the desired robot orientation. Figures 14, 16, 18 and 20 show us that the proposed control method steers the robot center R converging to the given path and the robot orientation tracking the desired ones with acceptable errors, where the actuator saturation did not appear. In order to check the influence of the actuator saturation, the second experiment selected the desired robot orientation as

$$\theta_d = \theta_P + 0.9c_P v_d^2, \tag{35}$$

where c_P is the path curvature at point Q. The results illustrated in figures 15, 17, 19 and 21 show us that the robot center R converges to the given path, even though the wheels velocities come in the saturation when the path turns sharply.

6 Conclusions

In this paper a new motion control method for an omnidirectional robot is presented. This approach is based on the inverse input-output linearized robot kinematic model,

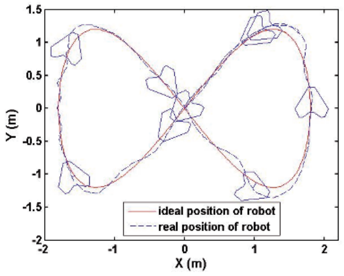

Fig. 14. Reference path and robot path.

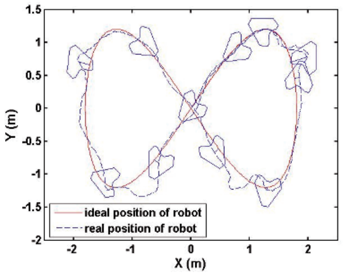

Fig. 15. Reference path and robot path.

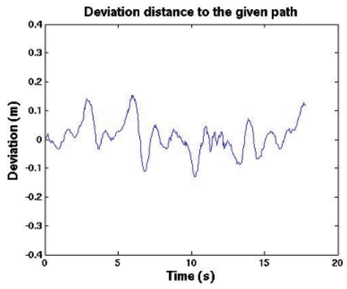

Fig. 16. Distance error.

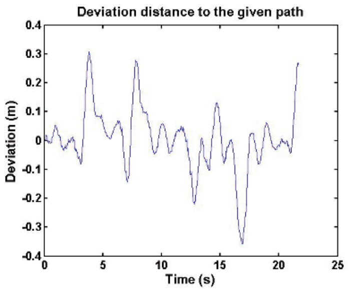

Fig. 17. Distance error.

which completely decoupled the robot translation and rotation. The robot translation is steered to follow a reference path, and the robot rotation is controlled to track the desired orientation. Because the actuator dynamics and saturation can greatly affect the robot performance, they are taken into account when designing the controller. With the Lyapunov stability theory, the global stability of the path following control law has been proven. The locus technique is used to analyze and choose the suitable parameters of the PD controller, such that the robot orientation can converge to the desired one even when the wheels velocities saturate.

In real-world experiments, the robot was controlled to follow an eight-shaped curve with a constant translation velocity of $1m/s$, and to track sharp changing orientations. The results show the effectiveness of the proposed control method in the case of both actuator saturation and non-saturation.

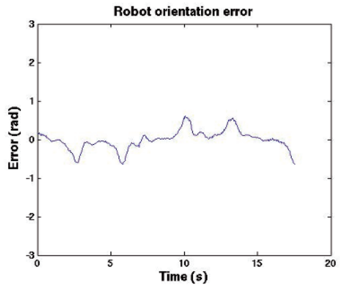

Fig. 18. Orientation error.

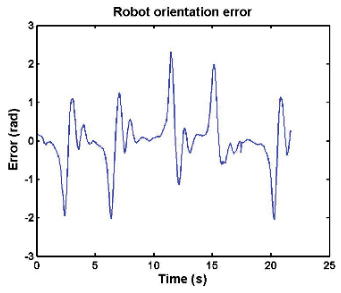

Fig. 19. Orientation error.

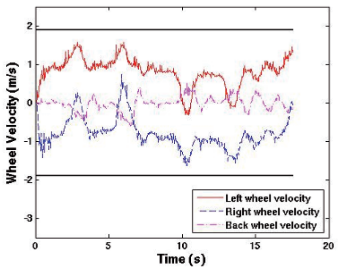

Fig. 20. Real wheel velocities.

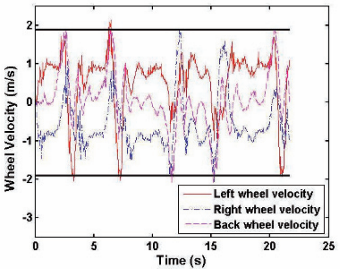

Fig. 21. Real wheel velocities.

References

1. Campion, G., Bastin, G., D'Andréa-Novel, B.: Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. In: IEEE Transactions on Robotics and Automation. Volume 12. (1996)
2. Watanabe, K.: Control of an omnidirectional mobile robot. In: KES'98, 2th International Conference on Knowledge-Based Intelligent Electronic Systems. (1998)
3. Liu, Y., Wu, X., Zhu, J.J., Lew, J.: Omni-directional mobile robot controller design by trajectory linearization. In: ACC'03, Proceeding of the 2003 American Control Conference. (2003)
4. Purwin, O., Andrea, R.D.: Trajectory generation and control for four wheeled omnidirectional vehicles. In: Robotics and Autonomous Systems. Volume 54(1). (2006)
5. Tsai, C.C., Huang, H.C., Wang, T.S., Chen, C.M.: System design, trajectory planning and control of an omnidirectional mobile robot. In: 2006 CACS Automatic Control Conference. (2006)
6. Muir, P.F., Neuman, C.P.: Kinematic modeling for feedback control of an omnidirectional wheeled mobile robot. In: Autonomous Robot Vehicles, Springer-Verlag (1990)
7. Terashima, K., Miyoshi, T., Urbano, J., Kitagawa, H.: Frequency shape control of omnidirectional wheelchair to increase user's comfort. In: ICRA'04, Proceedings of the 2004 IEEE International Conference on Robotics and Automation. (2004)
8. Rojas, R., Förster, A.G.: Holonomic Control of a Robot with an Omni-directional Drive. BöttcherIT Verlag, Bremen (2006)
9. Scolari Conceição, A., j. Costa, P., Moreira, A.: Control and model identification of a mobile robot's motors based in least squares and instrumental variable methods. In: MMAR'05, 11st International Conference on Metgids abd Models in Automation and Robotics. (2005)
10. Indiveri, G., Paulus, J., Plöger, P.G.: Motion control of swedish wheeled mobile robots in the presence of actuator saturation. In: 10th annual RoboCup International Symposium. (2006)
11. Scolari Conceição, A., Moreira, A., j. Costa, P.: Trajectory tracking for omni-directional mobile robots based on restrictions of the motor's velocities. In: SYROCO'06, 8th International IFAC Symposium on Robot Control. (2006)
12. Mojaev, A., Zell, A.: Tracking control and adaptive local navigation for nonholonomic mobile robot. In: Proceedings of the IAS-8 conference. (2004)
13. Heinemann, P., Rueckstiess, T., Zell, A.: Fast and accurate environment moddelling using omnidirectional vision. In: Dynamic Perception, Infix (2004)