

# A Posture Control for Two Wheeled Mobile Robots

Hyun-Sik Shim and Yoon-Gyeong Sung

**Abstract:** In this paper, a posture control for nonholonomic mobile robots is proposed with an empirical basis. In order to obtain fast and consecutive motions in realistic applications, the motion requirements of a mobile robot are defined. Under the assumption of a velocity controller designed with the selection guidance of control parameters, the algorithm of posture control is presented and experimentally demonstrated for practicality and effectiveness.

**Keywords:** nonholonomic mobile robot, posture control, soccer robot

## I. Introduction

It has been a topic to design the wheeled mobile robot controller due to the control redundancy that an ordinary mobile robot possesses only two degrees of freedom for motion control, while its posture is described by three degrees of freedom. For the entire system dynamics, the kinematic relation is nonlinearly coupled. Moreover, the mobile robot is described by a non-holonomic system with its unintegrable constraint equation. Hence, a variety of approaches have been proposed including various techniques of automatic control [1], [2].

Kanayama *et al.* [3] presented the use of reference and current postures for vehicle control, the use of a local error coordinate system, and a PI control algorithm for linear and rotational velocities. A novel control algorithm for determining the robot's linear and rotational velocities based on Lyapunov theory is described in [4]. The algorithm has a convergence problem so that if the reference linear velocity is zero, convergence cannot be achieved. Thus, it cannot be applied to the parking problem. Shim and Kim [5] suggested a stable tracking control method for a non-holonomic vehicle to resolve the convergence problem. Aicardi *et al.* [6] presented a control algorithm which uses polar coordinates and its stability was proven with the Lyapunov stability theory. For convenient analysis and motion control design, the algorithms [4], [5], [6] were reported with numerical simulations. Recently, Yang and Kim [7] presented an algorithm with sliding mode control with the inclusion of robot dynamics. Those algorithms were evaluated under constraint conditions in practical illustrations.

For the experiment, most of the algorithms were sluggish due to the limitations of motor output, velocity controller performance, etc. A controller in this paper is presented with an empirical motivation by taking the robot kinematics at the controller design stage as simple as possible. Contrary to the former control design approaches with a mathematical basis, the control algorithm is designed by focusing on the given specifications of the robot's desired trajectory and physical limitations. The stability of the proposed controller is verified with the Lyapunov stability theory in the reference [8]. With the worst case of scenario, the effectiveness and robustness of the posture control are demonstrated through experiments.

The remainder of the paper is organized as follows. In Section 2, the notations used and the kinematic model are defined. In Section 3, a control law for motion control is designed. In Section 4, experimental results illustrate the effectiveness of the scheme. Lastly, the conclusion is given.

Fig. 1. Coordinate system of the mobile robot (subscript "c" refers to the "current state").

## II. Kinematics of the mobile robot

A mobile robot is on a 2-D plane in a global cartesian coordinate. There are global coordinates  $X - Y$  and local coordinates  $x - y$  attached to the center of the robot body as shown in Figure 1. The robot has three degrees of freedom in its relative positioning which is represented by the *posture*,

$$\mathbf{p} = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} \quad (1)$$

where, the heading direction  $\theta$  is taken counterclockwise from the  $X$ -axis to the  $x$ -axis. The angle  $\theta$  indicates the orientation of the robot or the wheels. The set  $(x \ y)$  represents the robot *position*. In Figure 1, the vector  $(x_c \ y_c \ \theta_c)^T$  corresponds to the current posture of the robot.

The robot's motion is controlled by its *linear velocity*  $v$  and *angular velocity*  $\omega$ , which are functions of time. The robot kinematics are defined by the Jacobian matrix  $J(\theta)$ ,

$$\dot{\mathbf{p}} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = J(\theta)\mathbf{q} = \begin{pmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{pmatrix} \mathbf{q} \quad (2)$$

where,  $\mathbf{q} = (v \ \omega)^T$ .

As mobile robot postures in control implementation, there are two postures such as the *reference*  $\mathbf{p}_r = (x_r \ y_r \ \theta_r)^T$  which is the target posture and the *current posture*  $\mathbf{p}_c = (x_c \ y_c \ \theta_c)^T$  which is the true posture at the current moment. By transforming the position of the robot in cartesian coordinates into polar coordinates, the posture error is expressed as

$$\begin{pmatrix} x_{ec} \\ y_{ec} \\ \theta_{ec} \end{pmatrix} = \begin{pmatrix} \cos\theta_c & \sin\theta_c & 0 \\ -\sin\theta_c & \cos\theta_c & 0 \\ 0 & 0 & 1 \end{pmatrix} (\mathbf{p}_r - \mathbf{p}_c) = \begin{pmatrix} r_e \cos\phi_e \\ r_e \sin\phi_e \\ \theta_e \end{pmatrix} \quad (3)$$

where,  $r_e > 0$  is the distance error,  $\theta_e = \theta_r - \theta_c$  is the orientation error with respect to  $\mathbf{p}_c$ , and  $\phi_e$  is the angle measured from the robot's principle axis to the distance error vector  $r_e$ . The posture velocity error [6] in polar coordinates is expressed as

$$\begin{cases} \dot{r}_e &= -v_c \cos \phi_e \\ \dot{\phi}_e &= -\omega_c + v_c \frac{\sin \phi_e}{r_e} \\ \dot{\theta}_e &= -\omega_c \end{cases} \quad (4)$$

### III. Posture control

For the control based on robot kinematics, a control structure is composed of a velocity controller and a posture controller as shown in Fig. 2. If a reference posture is given, the controllers command the mobile robot to engage in the reference posture. The posture controller let the robot chase the reference posture through a velocity command  $\mathbf{q}_m$ . The velocity controller then makes the robot follow the command  $\mathbf{q}_m$ . However, in the velocity controller, the dynamic model of the mobile robot is needed. Moreover, computational burden follows in implementing this scheme. Considering these issues, a posture controller is presented under the assumption that the velocity controller is designed using a control scheme such as adaptive control [9], variable structure control [10], or nonlinear feedback control [11]. In this paper, if the velocity controller designed is under such as friction, unknown disturbances or motor saturation, it could assume that a command velocity becomes the robot's velocity:  $\mathbf{q}_m = \mathbf{q}_c$ .

Fig. 2. Control diagram with posture controller.

#### A. Motion Requirements

The motion specifications based on the posture velocity error in Eq.(4), to follow a trajectory or a velocity command, are given as follows:

- (1) If the traveling distance to the reference posture is large, then move fast. Reduce speed as it comes closer.
- (2) It should take the arriving time to arrive at a final posture.
- (3) If the angle error  $\phi_e$  is large, the robot's linear velocity should be reduced so that the robot will have a traveling distance.
- (4) If the reference posture is far, it needs to reduce the distance error first.
- (5) The robot should meet the reference posture immediately.

These specifications require the posture controller to generate the best trajectory with a posture error. The posture controller is designed using the vector  $(r_e \phi_e \theta_e)^T$  and the velocity command to the robot is generated based on the above specifications.

Fig. 3. Control parameter assignment to the robot's location.

For the first and second specifications, a linear velocity controller with  $v_m = k_r r_e$  is considered. This equation is similar to the equation proposed by Aicardi *et al.* [6]. When the distance is very large, the velocity command will be very high and

the robot velocity will be saturated. Instead of  $r_e$ , a new variable  $r'_e$  is introduced to take care of such a situation:

$$r'_e = \begin{cases} \frac{\pi}{2}, & r_e \geq R_m \\ \frac{\pi r_e}{2R_m}, & \text{otherwise.} \end{cases}$$

A 'sine' relation instead of a linear one is employed so that the velocity becomes higher even with a smaller  $r_e$ :

$$v_m = k_r \sin r'_e.$$

For the third specification, the angle error  $\phi_e$  is used in controlling the linear velocity. To apply an inverse proportional relationship for  $\phi_e$ , the following equation is employed:

$$v_m = k_r \sin r'_e \cos \phi_e.$$

The following variables are defined for limitations:

$$d = \begin{cases} 1, & -\Phi_m \leq \phi_e \leq \Phi_m \\ -1, & \phi_e < -\pi + \Phi_m \text{ or } \phi_e > \pi - \Phi_m \\ 0, & \text{otherwise,} \end{cases}$$

$$\phi'_e = \begin{cases} \phi_e - \pi, & \phi_e > \frac{\pi}{2} \\ \phi_e + \pi, & \phi_e < -\frac{\pi}{2} \\ \phi_e, & \text{otherwise,} \end{cases}$$

$$\theta'_e = \begin{cases} \theta_e - \pi, & \theta_e > \frac{\pi}{2} \\ \theta_e + \pi, & \theta_e < -\frac{\pi}{2} \\ \theta_e, & \text{otherwise} \end{cases}$$

where,  $\Phi_m$  is an arbitrary constant between 0 and  $\frac{\pi}{2}$ . By introducing  $d$ ,  $\phi'_e$  and  $\theta'_e$ , it is possible to make the robot move in the back and forth directions because the front and rear sides of a mobile robot are not distinguishable. Thus, the linear velocity controller law with a new parameter diagram in Fig. 3 is proposed as

$$v_m = k_r d \sin r'_e \cos \phi'_e \quad (5)$$

where,  $k_r$  is a positive constant with the unit of velocity and  $d$  refers to the direction of the robot so that the robot moves in a forward direction with  $d = 1$  and in a backward direction with  $d = -1$ . Since there is no difference between the front and rear sides of a mobile robot, only the regions  $r_e \geq 0, -\frac{\pi}{2} \leq \phi_e \leq \frac{\pi}{2}$  and  $-\frac{\pi}{2} \leq \theta_e \leq \frac{\pi}{2}$  are considered. For each of the distance errors  $r_e$  and angle errors  $\phi_e$ , four regions are classified as in Fig. 4.

Fig. 4. Four classifications to  $r_e$  and  $\phi_e$ . Region\_0 ( $r_e \leq R_e$ ), Region\_1 ( $-\Phi_m \leq \phi_e \leq \Phi_m$  and  $R_e \leq r_e \leq R_a$ ), Region\_2 ( $-\Phi_m \leq \phi_e \leq \Phi_m$  and  $r_e \geq R_a$ ) and Region\_3 ( $\Phi_m < \phi_e \leq \frac{\pi}{2}, -\frac{\pi}{2} \leq \phi_e < -\Phi_m$  and  $r_e > R_e$ ).

In order to regulate the angle and orientation errors, the angular velocity is uniquely employed in this paper. Based on the design concept above, a relation equation could be expressed as

$$\omega_m = k_\theta (\sin \phi'_e + \sin \theta'_e)$$

where  $k_\theta$  is a positive constant with the unit of angular velocity. When the reference posture is at a distance, the angle error  $\phi_e$  should be considered more to meet the fourth specification. As the robot is approaching the reference posture, the orientation error  $\theta_e$  will be the important factor to meet the 4th specification. Therefore, the angular velocity control law is deliberately proposed as

$$\omega_m = k_\theta(\epsilon \sin \phi'_e + (1 - \epsilon) \sin \theta'_e) \quad (6)$$

where, a variable  $\epsilon$  is defined as

$$\epsilon = \begin{cases} 0 & r_e \leq R_e \\ 0.5 - \alpha \text{Sgn}(\sin \theta'_e) \text{Sgn}(\sin \theta'_e + \sin \phi'_e) & R_e < r_e < R_a \\ 1 & \text{otherwise} \end{cases}$$

where,  $0 < R_e \leq R_a \leq R_m$ ,  $0 < \alpha < 0.5$  and, the function  $\text{Sgn}(x)$  is 1 if  $x$  is positive and is -1 otherwise.

The control parameters for the proposed control law are  $k_r$ ,  $k_\theta$ ,  $R_m$ ,  $R_e$ ,  $R_a$  and  $\Phi_m$ . Various trajectories can be generated according to their values. However, the trajectories do not differ as much from each other as those in [4], [5], [6]. This is possible because the control laws for  $v_m$  and  $\omega_m$  are designed based on the design specifications.

$k_r$  and  $k_\theta$  are the variables limited by the maximum velocity of the robot. To the contrary, the values of  $k_r$  and  $k_\theta$  should be higher to meet the 2nd specification. To tackle this problem, the robot's linear velocity  $v$  and angular velocity  $\omega$  should be limited within the shadow areas as shown in Fig. 5, as a robot's wheel velocity is limited by its maximum velocity  $V_{max}$ . That is, the robot should be controlled by the values within this area. If a high linear velocity  $v_m$  is given, the applied angular velocity  $\omega_c$  should be limited to a smaller value than the control command  $\omega_m$  in the calculation. Therefore,  $k_r$  and  $k_\theta$  must be limited to

$$k_r + Lk_\theta \leq V_{max} \quad (7)$$

where,  $L$  is the radius of the robot body.

Fig. 5. Usable velocity area of wheel and robot velocities.

$R_m$  is a deceleration factor for linear velocity. If it is large, the velocity is reduced even when the robot is too far away. A high value reduces slippage and makes it more stable than a smaller  $R_m$ . It will, however, take more time to reach the target point.

$\Phi_m$  should be decided because it is related to the trajectory. If it is close to  $\Phi_m = 0^\circ$ , the robot motion will be "turning" or "going straight". However, if it is close to  $\Phi_m = 90^\circ$ , the traveling distance will be large. Therefore,  $\Phi_m$  is an important factor which is related to the allowable error range (maximum  $R_e$ ).

$R_a$  is the radius of the circular area where the control law has all three error terms. It can generate similar trajectories with smaller differences according to  $\Phi_m$ . If it has a small value, the trajectory is close to a straight line and the orientation error is reduced at the reference posture. If it has a large value, the trajectory is smooth and it may take a longer time to get to the target point.

$R_e$  takes an allowable value based on the system performance and its maximum value depends on  $\Phi_m$ . If a small value is used for the same, the arriving time will be much longer.

In the implementation of the motion control, there are several control parameters such as  $k_r$ ,  $k_\theta$ ,  $R_m$ ,  $R_e$ ,  $R_a$ , and  $\Phi_m$  so that the motion trajectory depends on the parameters. In Figs. 6 and 7, the variation of the motion trajectory are shown according to the parameters  $R_a$  and  $\Phi_m$ .

Fig. 6. Schematic trajectory variation to  $\Phi_m$ .

Fig. 7. Schematic trajectory variation to  $R_a$ .

## IV. Experimental evaluation

### 1. Experimental evaluation

In order to illustrate the effectiveness and practicality of the proposed method with a PID (Proportional Integral Derivative) velocity controller as a worst case of scenario, the posture control law is implemented with a mobile robot, sized  $7.5\text{cm} \times 7.5\text{cm} \times 7.5\text{cm}$ , designed for robot soccer games [12], [13]. The soccer robot employed has no body frame, but has 2 PCBs with surface mounting devices. The robot has an 80C196KC microprocessor running at 20MHz, and two wheels driven by two DC motors with two LM629 motion controllers. The LM629 chip is used as a velocity controller which receives only a reference velocity as the input signal and implicitly generates the required torque according to the reference velocity via the L298 driver chip. The maximum speed of the robot is about  $125 \text{ cm/sec}$ .

Fig. 8. Control block diagram of a vision based mobile robot.

The sampling time of the posture control is  $33 \text{ msec}$  which is depending on the vision processing time. The sampling time of the velocity controller is  $410 \mu\text{sec}$ . The command velocities ( $v, \omega$ ) from the posture controller are utilized to get the right and left wheel velocities ( $V_R, V_L$ ) as

$$\begin{aligned} V_R &= v + L\omega \\ V_L &= v - L\omega \end{aligned}$$

where,  $L$  is the radius of the robot body. The host computer transmits the wheel velocities to the motion controller of the robot. However, accurate motion information is not provided to the robot system because the command velocity of the motor is updated by  $1 \text{ cm/sec}$  and the locating accuracy of the vision system is limited to  $1 \text{ cm} \sim 2 \text{ cm}$ . Therefore, the proposed algorithm will be demonstrated under a nonideal situation with the built-in PID velocity controller on the LM629.

In this paper,  $R_e$  is chosen as  $2 \text{ cm}$  due to the locating accuracy of the vision information. This is a suitable choice because it is more than the maximum error of  $1.607 \text{ cm}$  when  $\Phi_m = 70^\circ$ . The reference posture is given as  $(100, 50, 0^\circ)$ . Fig. 9 shows the robot trajectories according to the values of  $R_a$ . The smaller  $R_a$  is, the faster the robot can move to its reference posture. It is shown that the orientation error is largely

reduced when the robot is close to the reference posture. Fig. 10 shows the robot trajectories according to  $R_m$ . There is little difference between the two trajectories since it works only when the robot is close to the reference posture. However, it takes 3.465 sec for  $R_m = 10$  and 6.1 sec for  $R_m = 50$  to get the reference posture so that it is faster when  $R_m$  is small.

Fig. 9. Robot trajectories to the values of  $R_a$  ( $k_r = 60$ ,  $\Phi_m = 70^\circ$ ,  $R_m = 20$ ,  $k_\theta = 10$ ,  $p_r = (112, 72, 0^\circ)$ ).

Fig. 10. Robot trajectories to the values of  $R_m$  ( $k_r = 60$ ,  $\Phi_m = 70^\circ$ ,  $R_a = 10$ ,  $k_\theta = 10$ ).

Fig. 11 shows the robot trajectories for various values of  $k_r$ .  $k_r$  refers to the maximum linear velocity of the robot with unit *cm/sec*. If the maximum velocity is high, the robot will quickly arrive at the reference posture. However, if the maximum velocity is very high, the trajectories could not achieve the design specifications.

Fig. 11. Robot trajectories to the values of  $k_r$  ( $R_a = 10$ ,  $\Phi_m = 70^\circ$ ,  $R_m = 20$ ,  $k_\theta = 10$ ).

In Fig. 12, the robot trajectories are shown through the 7 via-points determined in advance as a more general robot motion control. The via-points are located at  $(60, 47, 0^\circ)$ ,  $(115, 70, 45^\circ)$ ,  $(120, 48, 0^\circ)$ ,  $(113, 17, -45^\circ)$ ,  $(66, 13, 0^\circ)$ ,  $(13, 20, 0^\circ)$ , and  $(16, 80, 90^\circ)$  from 'Start' to 'Via 6'. The posture control could generate a smooth motion trajectory in both a continuous reference posture and a discrete reference posture.

Fig. 12. Consecutive robot trajectory through 7 via-points ( $k_r = 60$ ,  $R_a = 10$ ,  $\Phi_m = 70^\circ$ ,  $R_m = 20$ ,  $k_\theta = 10$ ).

## V. Conclusion

In this paper, a posture control is proposed with empirical motivation. The control law was derived based on the given design specifications for the desired trajectory. In the experiment, no matter how the velocity controller may be constructed, an ideal controller will not be readily designed because velocity and acceleration limits exist resulting from limitations on the motor specifications. Thus, the control algorithm was proposed and experimentally demonstrated for effectiveness and applicability under the worst case scenario. The proposed algorithm was successfully employed and has shown excellent performance for the international robot soccer game MiroSot'98 for a vision based soccer robot system held in France.

## References

[1] C. C. de Wit, H. Khenouf, C. Samson, and O.J. Sordalen, *Nonlinear Control Design for Mobile Robots*, Recent Trends in Mobile Robots; World Scientific Series in Robotics and Automated Systems vol. 11, Yuan F. Zheng(ed.), Singapore, 1993.

- [2] E. P. Silva, F. L. Pereira, and J. B. Sousa, On the Design of a Control Architecture for an Autonomous Mobile Robot, *Advances in Intelligent Autonomous Systems(IAS)*, Kluwer Academic Publishers, 1998.
- [3] Y. Kanayama, A. Nilipour, and C. Lelm, "A locomotion control method for autonomous vehicle," *Proc. IEEE Int. Conf. Robotics and Automat.*, pp. 1315-1317, 1989.
- [4] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, "A stable tracking control method for an autonomous mobile robot," *Proc. IEEE Int. Conf. Robotics and Automat.*, pp. 384-389, 1990.
- [5] H.-S. Shim, and J.-H. Kim, "A stable locomotion control for a non-holonomic mobile robot," *Proc. of the Asian Control Conference*, pp. 639-642, Tokyo, Japan, July, 1994.
- [6] M. Aicardi, G. Casalino, A. Balestrino, and A. Bicchi, "Closed loop smooth steering of unicycle-like vehicles", *Proc. IEEE Int. Conf. on Decision and Control*, pp. 2455-2458, 1994.
- [7] J.-M. Yang and J.-H. Kim, "Sliding mode control for trajectory tracking of nonholonomic wheeled mobile robots," *IEEE Transactions of Robotics and Automation*, vol. 15, no. 3, pp. 578-587, June, 1999.
- [8] H. S. Shim, Design and Implementation of Multi-Robot Cooperation System Using BIOS:Its Application to Robot Soccer, Ph.D Thesis, KAIST, 1998.
- [9] H.-S. Shim and J.-H. Kim, "Robust adaptive control for nonholonomic wheeled mobile robot," *Proc. IEEE Int. Conf. on Industrial Technology*, Guangzhou, China, pp. 245-248, 1994.

Hyun-Sik Shim

Received his B.S., M.S., and Ph.D. degrees in Electrical Engineering, Korea Advanced Institute of Science and Technology, 1991, 1993, 1998, respectively. He is working for LG Traffic Information System. His research interests are mobile robots, intelligent traffic control.

He is a member of IEEE, KITE, and ICASE.

- [10] H.-S. Shim, J.-H. Kim, and K. Koh, "Variable structure control for nonholonomic wheeled mobile robot," *Proc. IEEE Int. Conf. on Robotics and Automat.*, Nagoya, Japan, pp. 1694-1699, 1995.
- [11] Y. Zhao and M. Reyhanoglu, "Nonlinear control of wheeled mobile robots," *Proc. IEEE Int. Conf. Robotics and Automat.*, pp. 1967-1973, 1992.
- [12] J.-H. Kim, H.-S. Shim, H.-S. Kim, M.-J. Jung, I.-H. Choi and K.-O. Kim, "A cooperative multi-agent system and its real time application to robot soccer," *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Albuquerque, New Mexico, pp. 638-643, 1997.
- [13] H.-S. Shim, H.-S. Kim, M.-J. Jung, I.-H. Choi, J.-H. Kim, and J.-O. Kim, "Designed Distributed Control Architecture for Cooperative Multi-agent System and Its Real-Time Application to Soccer Robot," *Journal of Robotics and Autonomous System*, Ed. J.-H. Kim, vol.21, no.2, pp. 149-165, September, 1997.

#### Yoon-Gyeong Sung

Received his B.S., M.S., and Ph.D. degrees in Mechanical Engineering, Yeungnam University in 1982, Yonsei University in 1985, University of Alabama in 1997, respectively. He had worked for Korea Atomic Energy Institute. Since 2000, He is in School of Mechanical Engineering, Yeungnam University as a professor. His research interests are robotics, smart structural control, multibody dynamics, nonlinear control, and machine vision. He is a member of ASME, KSME, and ICASE.