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

3,900 Open access books available
116,000 International authors and editors
120M Downloads

154 Countries delivered to
TOP 1% Our authors are among the 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
A Robust Nonlinear Control for Differential Mobile Robots and Implementation on Formation Control

Jie Wan and Peter C. Y. Chen
Department of Mechanical Engineering, National University of Singapore

1. Introduction

The past three decades witness a long-term intensive interest in researching a variety of control methods on wheeled mobile robots (WMRs), although the general nonholonomic systems have been investigated for more than one and half a century (Kolmanovsky & McClamroch, 1995). It is well-known that usually WMRs are characterized by non-integrable kinematic constraints, namely the nonholonomic constraints. The consequence is that these constraints rule out the possibility of direct application of standard control theories, such as linear control system theory in this field. Furthermore, as pointed out in a landmark paper (Brockett, 1983), nonholonomic systems cannot be stabilized by continuously differentiable, time invariant, state feedback control laws. To cope with the challenges arising in nonholonomic system control, a great number of approaches have been proposed and some selections of the vast amount of published literature are reflected in the survey paper (Kolmanovsky & McClamroch, 1995) and the book (Dixon et al., 2001) and in chapters (7 - 9) of the book (Wit & Siciliano, 1996). Several controls from experiment perspectives are examined and implemented in the work (Luca et al., 2001). Central to the WMR motion control are the tracking control problems. Normally there are two categories of tracking control: posture tracking and point tracking. The former aims to achieving stably tracking a moving reference posture (i.e., position and orientation) while the latter only concerns about position tracking. Nonlinear feedback control strategies (Wit & Sordalen, 1992), (Godhavn & Egeland, 1997), (Kanayama et al., 1990), (Closkey & Murray, 1997), (Teel et al., 1995) are often favored in dealing with tracking control problem to compensate disturbances and uncertainties although open-loop control laws are also workable (Murray & Sastry, 1990), (Lafferiere & Sussmann, 1991), (Braquaud & Latombe, 1989), (Brockett, 1981). In recent years, more and more attention have been drawn to the robustness of the controller in presence of uncertainties. In (Aguiar et al., 2000), the authors address the regulation control of WMR with parametric modelling uncertainty using Lyapunov functions. A robust new Kalman-based active observer controller for path following was proposed (Coelho & Nunes, 2005) in circumstances of uncertainties and disturbances. And the paper (Dixon et al., 1999) presents a controller robust to parametric uncertainty and additive bounded disturbance in the dynamic model through the use of a dynamic oscillator. An autonomous multi-robot system comprises a group of (often homogeneous) robots, each has a certain degree of mobility and autonomy. Research interests in unmanned autonomous robots have been growing significantly in recent years, due to the potential that this type of
robotic systems will be able to perform a variety of tasks in environments inaccessible or too dangerous to humans. One basic problem concerning multi-root systems is formation control, whereby a group of robots maintain a certain (usually 2D) geometry while in concerted motion. When encountering obstacles (either static or dynamic), the group must maneuver to avoid them while maintaining the overall formation geometry whenever possible. In this chapter, we studied the robustness of a nonlinear feedback control law based on a kinematic model. A generic kind of feedback control laws which cover some commonly used methods and the associated stability are quickly reviewed in a new perspective by invoking Lyapunov stability theorems. A simple fact is then unveiled that the stability proof actually depends on perfect mathematical manipulation which requires some terms of the differential equations to be cancelled. However, in real world, the perfect will be ruined under some circumstances and the robustness issue has to be investigated. Invariance principles, rather than Lyapunov stability theorems, are the major tools in dealing with the imperfect cases in which no term cancellation can be reached. The stability issue and the robustness of this control law as well is analyzed and the results lead to stable zones for each given set of controller gains. It is found that under certain circumstances the closed-looped system may fail in reaching the desired control objectives and performance. Such insights into the stability zone for a given set of controller gains make it possible to improve the controller by choosing proper controller gains. This new robust control can overcome the drawbacks of the previous commonly used counterpart. Guidelines on designing improved control law are also provided to facilitate real implementation. The merits and benefits of this new control are also highlighted through comparisons with its prototype. The analysis shows that except the more robustness, the new control law is entitled to faster response if the controller gains are properly chosen. Matlab simulation results which show the benefit are presented. Implementation of the proposed new control law on real robots is another major work of this chapter. An implementation on multiple robot formation control is reported including overview of the whole system structure, description of the robots used in the experiments, the vision system which is used for acquisition of the real-time position information of a group of robots moving on a test bed, and the background noise analysis of the vision system and so on. In the experiments, a group of mobile robots are requested to follow their corresponding visual leaders to form certain geometric patterns. A triangle formation with three robots and a square formation with four robots are demonstrated. The velocities/headings of robots during the formation control and other information are summarized in figures. The downside of robot locomotion mechanism, which is based on step motor and its effects on real implementation, such as misstep and dead zone are discussed. Suggestions and further improvements are discussed at the end of this chapter. This chapter is organized as follows: in Section 2, the stability problem is reviewed in a new perspective. Section 3 deals with formulation the problem of robustness. In Section 4 robustness analysis and its benefits are addressed while the effectiveness of the new proposed robust control law is verified via simulation in Section 5. Implementation and experiments with multiple robots on the application of formation control are addressed in Section 6, followed by conclusions summarized in Section 7.

2. Problem statement

We consider the point tracking problem for a wheeled mobile robot which is depicted in Figure 1. In this scenario, a wheeled mobile robot is supposed to track a series of goal points denoted by the symbol $q_g$ on a segment, which is a smooth curve in the world frame.
Fig. 1. Illustration of a wheeled mobile robot and its goal point \( q_g \), which may be moving on a segment (smooth curve) in the world frame.

Referring to this figure, intuitively we refer to notations \( r \) and \( \phi \) as "distance to target" and "misalignment angle" respectively.

As shown in Figure 2, in order to facilitate modelling kinematics of the wheeled mobile robot in a polar coordinate, specially we assign the origin to be the goal point (on the segment) for the robot to track. In this figure, a differential mobile robot, together with the associated notations, is illustrated in a polar coordinate. The separation between point \((x, y)\) and center of each wheel is represented by \( C_d \), which is a constant parameter for a given model of real robot. The heading of the robot is \( \theta \) while its translational velocity and angular velocity are denoted by \( v \) and \( \omega \) respectively. Notice that throughout this chapter, both \( \phi \) and \( \theta \) are defined in the domain \([-\pi, \pi]\).

Referring to Figure 2, motion of a differential mobile robot can be described by

\[
\begin{bmatrix}
\dot{\theta} \\
\dot{x} \\
\dot{y}
\end{bmatrix} = \begin{bmatrix}
0 & 1 & 0 \\
0 & \cos(\phi) & 0 \\
\sin(\phi) & 0 & 1
\end{bmatrix} \begin{bmatrix}
v \\
\omega
\end{bmatrix}.
\]

(1)

To link this model with the notations in polar coordinate, we can calculate \( r \) and \( \phi \) as

\[
r = \sqrt{x^2 + y^2},
\]

\[
\phi = \pi + \theta - \phi,
\]

respectively.

A kinematic model of a differential robots in the polar coordinate can be derived as follows:

\[
\begin{bmatrix}
\dot{\theta} \\
\dot{r} \\
\dot{\phi}
\end{bmatrix} = \begin{bmatrix}
0 & 1 & 0 \\
-\cos(\phi) & 0 & 1 \\
\frac{1}{\sin(\phi)} & 0 & 1
\end{bmatrix} \begin{bmatrix}
v \\
\omega
\end{bmatrix}.
\]

(2)
Fig. 2. Representation of a wheeled mobile robot in the polar coordinate frame with the origin $O'$ (i.e., point $q_g$ on Figure 1) being its goal point to track and the associated notations.

Detailed derivation procedures of Equation (2) can be found in the work (Lee et al., 1999) and are omitted for the sake of brevity in this chapter. This model is similar to the ones used in chapter 3 of (Siegwart & Nourbakhsh, 2004). From this model, specifically we have the relationship between $\dot{r}$, $\dot{\phi}$ and $v$, $\omega$ as

$$
\begin{bmatrix}
\dot{r} \\
\dot{\phi}
\end{bmatrix} =
\begin{bmatrix}
\frac{1}{2} \sin(\phi) & 0 \\
-\cos(\phi) & 1
\end{bmatrix}
\begin{bmatrix}
v \\
\omega
\end{bmatrix}.
$$

(3)

Now we consider to derive a general form of control laws which can stabilize the robot in the sense of Lyapunov stability theorems. It requires that both $r$ and $\phi$ tend to zero as time $t \to \infty$. One possible way is to choose control laws which lead to diagonalization of the matrix on the right hand side of Equation (3). To this end, we let $v$ and $\omega$ being

$$
\begin{bmatrix}
v \\
\omega
\end{bmatrix} =
\begin{bmatrix}
g_1(r, \phi) & 0 \\
0 & g_2(r, \phi)
\end{bmatrix}
\begin{bmatrix}
r \\
\phi
\end{bmatrix} +
\begin{bmatrix}
0 \\
-g_1(r, \phi) \sin(\phi)
\end{bmatrix},
$$

(4)

where $g_1(r, \phi)$ and $g_2(r, \phi)$ are certain unexplicit functions to be determined. Then by substituting the above equations into Equation (3), we can rewrite the Equation (3) into

$$
\begin{bmatrix}
\dot{r} \\
\dot{\phi}
\end{bmatrix} =
\begin{bmatrix}
\frac{1}{2} \sin(\phi) & 0 \\
-\cos(\phi) & 1
\end{bmatrix}
\begin{bmatrix}
g_1(r, \phi) & 0 \\
0 & g_2(r, \phi)
\end{bmatrix}
\begin{bmatrix}
r \\
\phi
\end{bmatrix} +
\begin{bmatrix}
0 \\
-g_1(r, \phi) \sin(\phi)
\end{bmatrix}.
$$

(5)

A family of possible functions $g_1(r, \phi)$ and $g_2(r, \phi)$ can be chosen as follows:

$$
g_1(r, \phi) = K_1 r^n \phi^{2q} (\cos(\phi))^{2p+1},
g_2(r, \phi) = -K_2 \phi^s,
$$

(6)

where $n = 0, 1, 2, \cdots$, $p = 0, 1, 2, \cdots$, $q = 0, 1, 2, \cdots$ and $s = 0, 1, 2, \cdots$. 

www.intechopen.com
Accordingly Equation (4) can be rewritten into the following form:

\[ v = K_1 r^{n+1} \phi^{2q} (\cos(\phi))^{2p+1}, \]
\[ \omega = -K_3 r^n \phi^{2q} \sin(\phi) (\cos(\phi))^{2p+1} - K_2 \phi^{2s+1}. \]  

(7)

**Proposition 1** The family of control laws given in Equation (7) asymptotically stabilizes a differential robot on its goal point. \( \square \)

**Proof:** The proof for this proposition is pretty straightforward by constructing a Lyapunov function candidate as

\[ V = \frac{1}{2} r^2 + \frac{1}{2} \phi^2. \]  

(8)

Simplifying Equation (5), we reach at

\[
\begin{bmatrix}
\dot{r} \\
\dot{\phi}
\end{bmatrix} = 
\begin{bmatrix}
-g_1(r,\phi) \cos(\phi) \\
\frac{1}{2}g_1(r,\phi) + g_2(r,\phi)
\end{bmatrix}
\begin{bmatrix}
r \\
\phi
\end{bmatrix} + 
\begin{bmatrix}
0 \\
-g_1(r,\phi) \sin(\phi)
\end{bmatrix}
\]
\[ = \begin{bmatrix}
-rg_1(r,\phi) \cos(\phi) \\
g_1(r,\phi) \sin(\phi) + \phi g_2(r,\phi)
\end{bmatrix} + \begin{bmatrix}
0 \\
-g_1(r,\phi) \sin(\phi)
\end{bmatrix}
\]
\[ = \begin{bmatrix}
r g_1(r,\phi) \cos(\phi) \\
\phi g_2(r,\phi)
\end{bmatrix}. \]  

(9)

Based on the results from Equation (9) and Equation (6), the first time derivative of \( V \) can be calculated readily as

\[
\dot{V} = r \dot{r} + \dot{\phi} \phi
\]
\[ = -r^2 g_1(r,\phi) \cos(\phi) + \phi^2 g_2(r,\phi)
\]
\[ = -K_1 \phi^{2q} r^{n+2} (\cos(\phi))^{2p+2} - K_2 \phi^{2s+2} \leq 0, \]  

(10)

thus completes the proof. \( \square \)

It should be noted that the general control law represented in Equation (7) can theoretically asymptotically stabilize the robot at its goal point. However the term \( \phi^{2q} \) will greatly slow down the system response. Therefore for the sake of practical considerations, \( q = 0 \) is preferred. Let us focus on the control with simple structure. Obviously if we let \( n = p = q = s = 0 \), then the Equation (7) can be reduced into the simplest form as shown below:

\[ v = K_1 r \cos(\phi), \]
\[ \omega = -K_1 \sin(\phi) \cos(\phi) - K_2 \phi. \]  

(11)

However, it is noted that control in Equation (11) is actually not the “simplest”. We can adopt another family of possible functions \( g_1(r,\phi) \) and \( g_2(r,\phi) \) as: \( g_1(r,\phi) = K_1 r^n \phi^{2q} \), and \( g_2(r,\phi) = -K_2 \phi^{2s} \), where \( n = 0,1,2,\cdots, p = 0,1,2,\cdots \) and \( s = 0,1,2,\cdots \). Therefore a more simplified control law can be found as follows:

\[ v = K_1 r, \]
\[ \omega = -K_1 \sin(\phi) - K_2 \phi, \]  

(12)

by simply letting \( n = q = s = 0 \). The control law presented by Equation (12) is used in (Baillieul, 2005) with preliminary analysis results.
3. Robustness problem

Beneath the control law stated in Equation (11) exists a fundamental challenge, although the stability issue seems to be affirmatively guaranteed by Lyapunov stability theorem. We already noticed that the technique of term cancellation (i.e., \( g_1(r, \phi) \sin(\phi) \)) is used when simplifying Equation (9). One may be interested in the following question: what if the ideal cancellation fails and the gain \( K_1 \) in \( v \) and \( \omega \) does not match with each other? To put it in details, it is to consider an alternative to the control law in Equation (11) as follows:

\[
\begin{align*}
    v &= K_1 r \cos(\phi), \\
    \omega &= -K_3 \sin(\phi) \cos(\phi) - K_2 \phi,
\end{align*}
\]

where \( K_3 \) is another independent variable and it may not be equal to \( K_1 \). In other words, it is equivalent to ask: “will the closed-loop system be stable if an alternative control represented in Equation (13) rather than the one in Equation (11) is applied to the system?’. 

In real world, there are numerous factors contributing to the such kind of "gain mismatching”. Take the digital control for example, truncation error of numerical calculation of triangle functions of \( \phi \) is unavoidable. More than that, in terms of real outputs of physical actuator, this “mismatching gain” phenomena may happen from time to time. To explain it, let \( v_L, v_R \) denote the tangent velocities of each wheel about the centers of rotation.

\[
\begin{align*}
    v &= \frac{v_L + v_R}{2}, \\
    \omega &= \frac{v_R - v_L}{2C_d},
\end{align*}
\]

where \( C_d \) is the displacement from the point \((x, y)\) to each wheel. We can establish the relationship between the vector \([v \ \omega]^T\) and \([v_L \ v_R]^T\) as follows:

\[
\begin{align*}
    \begin{bmatrix} v \\ \omega \end{bmatrix} &= \frac{1}{2} \begin{bmatrix} 1 & 1 \\ C_d & -C_d \end{bmatrix} \begin{bmatrix} v_L \\ v_R \end{bmatrix}, \quad (15) \\
    \begin{bmatrix} v_L \\ v_R \end{bmatrix} &= \begin{bmatrix} 1 & C_d \\ 1 & -C_d \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}. \quad (16)
\end{align*}
\]

The ideal case of control law Equation (11) is based on the assumption that we can make the equations

\[
\begin{align*}
    v_L &= K_1 r \cos(\phi) - C_d (K_1 \sin(\phi) \cos(\phi) + K_2 \phi), \\
    v_R &= K_1 r \cos(\phi) + C_d (K_1 \sin(\phi) \cos(\phi) + K_2 \phi),
\end{align*}
\]

strictly hold for each moment during the operation. However, in the real world, this turns out to be unrealistic. Apart from external disturbances, there are many factors that can ruin the perfect diagnosing shown in aforementioned context. For instance, each motor have different electro-mechanical characteristics. And each motor has its own nonlinearities (e.g. saturation) and so on. So in dynamic scenarios, we only have the real velocities \( v_L’ \) and \( v_R’ \) instead of the ideal counterparts \( v_L \) and \( v_R \). It means that in the real world, we have the following relationship,

\[
\begin{align*}
    v' &= (v_L' + v_R')/2, \\
    \omega' &= (v_R' - v_L')/(2C_d). \quad (17)
\end{align*}
\]
Both $v'$ and $\omega'$ can be transformed into uncertainties in $K_1$ and $K_2$. To simplify the analysis, we consider uncertainties of $K_1$ caused by mismatching of $\omega$ with respect to $v$, which is the case with control in Equation (13). Substituting Equation (13) into Equation (2) in Part I of this chapter, we obtain

$$
\begin{bmatrix}
\dot{r} \\
\dot{\phi}
\end{bmatrix} = \begin{bmatrix}
-\cos(\phi) & 0 \\
\frac{1}{2} \sin(\phi) & 1
\end{bmatrix} \begin{bmatrix}
K_1 r \cos(\phi) \\
- K_3 \sin(\phi) \cos(\phi) - K_2 \phi
\end{bmatrix} \\
- K_1 (\cos(\phi))^2 r \\
- K_2 \phi - \left(\frac{K_3 - K_1}{2}\right) \sin(2\phi)
\end{bmatrix}.
$$  \hspace{1cm} (18)

Through studying the stability of the closed-loop system described by Equation (18), we are entitled to investigating the robustness of the alternative control law given in Equation (13).

4. Robustness analysis

4.1 Stable zone

We refer to the model in Equation (18) as the real closed-loop system model. Then our problem is to analyze the stability and robustness of this real-world model. We can decompose this model into two subsystems as follows.

$$
\dot{r} = - K_1 (\cos(\phi))^2 r,
\phi = - K_2 \phi - \left(\frac{K_3 - K_1}{2}\right) \sin(2\phi).
$$

Obviously except the special case with $\cos(\phi(t)) \equiv 0$, $r(t)$ is at least asymptotically convergent to zero. As to $\phi(t)$, the situation is more complicated.

Let $K_4 = (K_3 - K_1)/2$, then we have

$$
\phi = - K_2 \phi - K_4 \sin(2\phi)
$$  \hspace{1cm} (19)

As to the subsystem denoted by Equation (19), construct a Lyapunov candidate as $V = \frac{1}{2} \phi^2$. The derivative of $V$ with respect to time is

$$
\dot{V} = \phi \dot{\phi} = - K_2 \phi^2 - K_4 \phi \sin(2\phi).
$$  \hspace{1cm} (20)

As shown by the closed-loop system equation in Equation (18), this system is time invariant indicating that LaSalle’s theorem is applicable. Therefore we are motivated to find out the invariant set $\Sigma$, which leads to negative $\dot{V}$ represented in Equation (20). The invariant set $\Sigma$ can be calculated according to the following equation:

$$
\Sigma = \{(K_1, K_2, K_3) | \dot{V} < 0\}.
$$

To this end we let $\dot{V} = 0$, then we have to make either $\phi = 0$ or $\phi = - K_4 / K_2 \sin(2\phi)$.

To find out the solution of $\phi = - K_4 / K_2 \sin(2\phi)$ for $\phi \in [0, \pi]$, we perform numerical calculation in Matlab environment. There are two scenarios: either $K_4 / K_2 \geq 0$ or $K_4 / K_2 < 0$. The illustration of different solutions when $K_4 / K_2 > 0$ is shown in Figure 3 while the case with $K_4 / K_2 < 0$ is shown in Figure 4. The calculation shows that:

- if $K_4 / K_2 \geq 0$ when $0 \leq K_4 / K_2 < c_1$, equation $\phi = - K_4 / K_2 \sin(2\phi)$ has only one solution, i.e., $\phi = 0$.

www.intechopen.com
Fig. 3. Illustration of different solutions with $K_4/K_2 > 0$.

- if $K_4/K_2 < 0$ when $c_2 < K_4/K_2 < 0$, equation $\phi = -K_4/K_2 \sin(2\phi)$ has only one solution, i.e., $\phi = 0$.

where $c_1$ and $c_2$ are constants. The numerical calculations offer approximation values of $c_1$ and $c_2$ as $c_1 \approx 2.30$ and $c_2 \approx -0.50$.

To sum up, the ratio $K_4/K_2$ should be within the range $(c_2, c_1)$ to make subsystem Equation (19) asymptotically stable. Or in other words, the relationship among $K_1, K_2, K_3$ to make subsystem Equation (19) stable is: $K_1 + 2c_2 K_2 < K_3 < 2c_1 K_2 + K_1$ ($K_2 > 0$) or $2c_1 K_2 + K_1 < K_3 < K_1 + 2c_2 K_2$ ($K_2 < 0$).

In practise, $K_2$ is usually chosen to be positive. So we can further simplify the conclusions above. In this case, the whole stable range of $K_3$ is:

$$K_1 + 2c_2 K_2 < K_3 < 2c_1 K_2 + K_1.$$  \hspace{1cm} (21)

The stable zone is shown in Figure 5. The stable zone is the whole wedge and is separated by a plane with $K_3 = K_1$. The upper part of this wedge has the property of $K_4 > 0$ while the lower part with $K_4 < 0$. Compared with the nominal sets of parameters in the plane $K_3 = K_1$, $K_4/K_2 < 0$.

Fig. 4. Illustration of different solutions with $K_4/K_2 < 0$. 

www.intechopen.com
Fig. 5. Illustration of stable zone with practical concerns in the case when $K_1 > 0$ and $K_2 > 0$. Note that the whole zone is separated by a plane $K_3 = K_1$, i.e., $K_4 = 0$.

the difference of those two parts of the zone is that the system response will be different as revealed by Proposition 3.

4.2 A new robust control and its benefits

Proposition 2 If $K_4 \geq 0$, namely, $K_3 - K_1 \geq 0$, $r(t)$ is exponentially convergent to zero for arbitrary initial $\phi_0$. □

The proof is pretty straightforward and hence it is omitted here for the sake of brevity.

Proposition 3 A set of $(K_1, K_2, K_3)$ in the upper part of the wedge in Figure 5 expedites the response of $\phi$ if $\phi_0 \in (0, \pi/2)$. □

Proof: It is noted that equation

$$\dot{\phi} = -K_2\phi - K_4\sin(2\phi)$$

has a unique solution on time interval $[0, t_1]$ for any $t_1 > 0$ because

$$f(\phi) = -K_2\phi - K_4\sin(2\phi),$$

is locally Lipschitz. Let $p(t) = \phi^2(t)$, then

$$p(t) = 2\phi\dot{\phi} = -2K_2\phi^2 - 2K_4\phi\sin(2\phi) \leq -2K_2\phi^2 = -2K_2p(t).$$

Let $q(t)$ be the solution of the differential equation

$$\dot{q}(t) = -2K_2q(t),$$

where $q(0) = \phi(0)$ then we arrive at

$$q(t) = \phi^2(0)e^{-2K_2t}.$$
According to comparison principle, the solution $\phi(t)$ is defined for all $t \geq 0$ and satisfies

$$|\phi(t)| = \sqrt{p(t)} \leq |\phi(0)|e^{-K_2t}, \forall t \geq 0,$$

thus completes the proof.

According to the proposition above, we can deliberately choose $K_3 \geq K_1$ to make the system more robust. Specifically we can design control laws according to guidelines as follows:

1. As revealed in Figure 5, $K_2$ should not be too close to zero as the bigger $K_2$, the wider zone between upper bound and lower bound.
2. To maximize the stability zone for a given set of $(K_1, K_2, K_3)$, it is desirable to choose $K_3 = K_1 + (c_1 + c_2)K_2$. In other words, $(K_1, K_2, K_3)$ is within the plane in the middle of upper bound and lower bound as illustrated in Figure 5.
3. To obtain comparatively large stability zone for a given set of $(K_1, K_2, K_3)$ while keep the converging rate from being negatively affected, it is desirable to choose $K_3 = K_1 + c_1K_2$.

5. Simulation study

5.1 Mismatching $K_3$ and $K_1$

A simulation in Matlab is designed to show two cases of mismatching $K_3$ and $K_1$. In case one, initial conditions are set to be $\phi_0 = 1$ rad and $r_0 = 1$ and in case two $\phi_0 = \pi$ rad and $r_0 = 1$. The nominal gains are chosen as $K_1 = 20$ and $K_2 = 1$. Suppose there is $-6\%$ deviation of $K_3$ with respect to $K_1$, i.e., $K_3 = 18.8$ in case one and a positive $24\%$ deviation of $K_3$, i.e., $K_3 = 24.8$ in case two.

The simulation results of system response are depicted in Figure 6 with (a), (b) for case one and (c), (d) for case two. From this figure, it is obvious that in those two cases $\phi(t)$ fails to approach to zero due to $-6\%$ and $24\%$ deviation of $K_3$ respectively. In other words, this is because both cases break the constraint described by Equation (21).

![Fig. 6. Illustration of mismatching $K_3$ and $K_1$. In (a) and (b) initial conditions are $\phi_0 = 1$ rad and $r_0 = 1$ and gain $K_1 = 20$, $K_2 = 1$ and $K_3 = 18.8$ (i.e., $-6\%$ deviation). And in (c) and (d) initial conditions are $\phi_0 = \pi$ rad and $r_0 = 1$ and gain $K_1 = 20$, $K_2 = 1$ and $K_3 = 24.8$ (i.e., $24\%$ deviation).](www.intechopen.com)
A Robust Nonlinear Control for Differentialy Mobile Robots and Implementation on Formation Control

5.2 Effects of $K_3$ on System Response

In this simulation we investigate the effects of mismatching $K_3$ on system response through simulation. Initial conditions are set to be $\phi_0 = 1$ rad and $r_0 = 1$ and gain $K_1 = 20$ and $K_2 = 1$. We vary the value of $K_3$ with respect to $K_1$. Refer to the stable zone illustrated in Figure 5, we deliberately choose several sets of $(K_1, K_2, K_3)$ from upper part, separation plane ($K_1 = K_3$) and lower part respectively. According to design guidelines, in this experiment, we choose $K_3 = 22.30, 21.80$ from upper part and $K_3 = 20$ for the nominal case and $K_3 = 19.2$ from the lower part. The results are depicted in Figure 7. From this figure, it is noticed that compared with $K_3 = 20$, a set in upper part of the wedge in Figure 5 contributes to expediting the system’s response while a set in lower part of the wedge will negatively affect the system’s response. The most significant effects of mismatching $K_3$ is on the converging rate of term $\phi(t)$. Since they are all within the stable wedge, both $r(t)$ and $\phi(t)$ approach to zero as time $t \to 0$.

To compare the energy needed for each controller, we define a function $J_n$ which is described by

$$J_n = \int_0^t \left( v^2(\tau) + \omega^2(\tau) \right) d\tau.$$  

In this simulation, the integral of the norm squared of the actual velocity signals for each controller is shown in Table 1. From the figures shown in this table, the control laws recommended by design guidelines seem to be more efficient than the nominal case with $K_3 = K_1$ and the one with negative deviation ($K_3 = 19.2$). And there is no significant difference between the two recommended control laws, i.e., $K_3 = 21.80$ and $K_3 = 22.30$ respectively.

Table 1. Comparison of the integral of the norm squared of the velocity input signals $\int_0^t (v^2(t) + \omega^2(t)) dt.$
6. Implementation and experiment

6.1 Overview of implementation
One picture of the real implementation is presented by Figure 8. In this figure, a 4m by 2.8m wooden test bed offers the field for a group of mobile robots. The MRKIT mobile robots presented in Figure 8 with on-board infrared sensors and compass, which are used in the experiments consist of the main platform to verify algorithms. Each robot has two independently controlled wheels driven by stepper motors. A GPS system is simulated by a vision system comprising vision frame grabber, CCD color camera with lens, a working station, and wireless communication modules. Two web-cam are mounted on the ceiling and can be used for robot tracking or video recording and only one is showed in Figure 8. The main parts of this implementation are connected as shown by Figure 10.

6.2 Parameters of MRKIT mobile robots
Each wheel of MRKIT robot is driven independently with step motor being controlled by on-board micro-controller. The velocity of wheel is controlled via PWM waveform and is determined by an internal time interval \( T \) in the micro-controller. The relationship between the velocity \( V \) of a wheel and \( T \) can be represented as

\[
V = \frac{D \pi}{NP} \cdot \frac{T}{10^{-6}}
\]

where \( D = 54 \text{ mm} \) is the diameter of the wheel; \( N = 400 \) is the step of motor per revolution; \( P \) is the time (second) per step and

\[
P = \frac{T}{2.5} \cdot 10^{-6}
\]

Fig. 8. Picture of real robots, test bed(on the floor), CCD color camera with wide-angle lens and one web-cam (mounted on the ceiling).
Finally we arrive at

\[ V = \frac{1060.288}{T} \text{ m/s}, \]

and \( T \) is a 16-bit integer which can be set in micro-controller. Due to the finite length of \( T \) and physical limitations of motor, \( V \) has a minimum \( V_{\text{min}} = 0.0162 \text{ m/s} \) and a maximum \( V_{\text{max}} = 0.3 \text{ m/s} \).
6.3 Vision system: resolution and noise analysis

A vision system, comprising of CCD camera, lens, frame grabber and application program as shown in Figure 10, is developed for the tracking of mobile robots and detecting position/orientation of them. Its resolution is largely determined by the resolution of the CCD camera and the optical system. In the experiment, the CCD camera is mounted on a bracket fixed on the ceiling. Due to the limitation of ceiling’s height, the viewable area on the test bed is of 1800mm by 2480mm. The CCD camera has a resolution of 576 by 768 pixels. We designate the x and y coordinate to the vision system and therefore we can calculate the real resolution of the vision system. The calculation results show that on the x axis, the resolution is 3.229mm per pix while on the y axis, 3.177mm per pix.

To identify the robot’s position and orientation, a color pad is attached on the top of a robot as shown in Figure 9. Each color pad has two different color circles aligned in a line. Each color circle has a diameter of 65mm. One circle is painted blue and another one is yellow. The center of each circle can be calculated through the image processing hardware and software, namely the frame grabber and the corresponding vision processing software running on working station. We can use coordinates of the centers of the two color circles to calculate the position of the robot’s center and its orientation as well. Let \( (x_a, y_a) \) and \( (x_b, y_b) \) denote the measured coordinates of the center of yellow circle and blue circle respectively. Hence, the coordinate of the robot’s center can be represented as \( \left( \frac{x_a + x_b}{2}, \frac{y_a + y_b}{2} \right) \).

The measurement of position of each color circle is a resultant of its real position and the error signal together with noise. The position error is incurred by the hardware of the system. For instance the field is not even and can end up with position error. Another sample of the source of the error signal can be the optical system. The distortion of the lens on the margin of the viewable area is relatively salient and such distortion in fact affects the accuracy of the measurement. Roughly the measurement of position can be expressed in the following equation:

\[
X_m = x_r + x_e + x_n,
\]

where \( x_r \) is the real position; \( x_e \) is the system error and \( x_n \) the noise. It is of interest and practically importance to know the noise level of the measured signal. For any static robot on the test bed, its real position and system error are always constant and contribute no variation to the mean value of \( X_m \) and to the variance either. From this observation, it helps to sample the measurement for a certain period and then use the spectrum analysis tools to get the information of the noise signal. One convenient way is to use the FFT technique. It is well known that Microsoft Windows is not a real time operation system. For the purpose of FFT, it is required to evenly sample the data. To solve this conflict, a high resolution timer without accumulation error is in need. In this experiment, the multi-media timer is used. It is a high resolution timer with high accuracy and resolution while demands of the system resource are relatively low. We set the sampling rate to be 500 Hz. A period of 2 minutes is used to sample the data and the error signal along x-axis is presented in Figure 11. Figures of error signal along y-axis and the associated orientation error signal are shown in Figure 12 and 13 respectively. Then we apply the FFT technique to analyze its frequency components. It turns
out that the noise signals on x, y and orientation all show on the feature of Gaussian noise. The analysis results show that $\delta_x = 0.142$ pix, $\delta_y = 0.154$ pix $\delta_\theta = 0.0122$ radius. Obviously compared with vision system’s resolution, the noise level of position signal is relatively low.

6.4 Experiment-1: triangle formation of three robots

A scalable formation control scheme is introduced in (Ge & Fua, 2005). The idea is that, instead of being attracted to a predetermined point, each robot is to be attracted to the corresponding segment, and once there, move along the segment to distribute themselves along the trench in order to form a formation by maintaining the desired position in relation to other robots. To briefly review this idea, Figure 14 is presented to show the segments and the robots which are supposed to fall into certain assigned segment. Usually a segment $S$ is a curve defined by some smooth (i.e., at least twice-differentiable) function in $\mathbb{R}^3$ that passes through one or two formation vertices. And a robot will arrive at the nearest point on the segment and then move along the curve of the segment.

In this experiment, suppose that assignment mechanism of segment is known and initially all robots are static. Three straight lines are assigned to three robots respectively. For the first
Fig. 13. Angular error signal with sample rate $f = 500Hz$.

8 seconds, each robot will try to approach the nearest point on the segment and then three virtual points moving along segments are assigned to each robot. Those three virtual points form a triangle pattern and will stop at the vertices of segments. The velocity of virtual points are set to be 20 pix per second. During the process of formation, velocities and headings of each robot are depicted in Figure 15 - 18. Snapshots of video (taken by web-cam) are shown in Figure 19-22. The controller parameters are set to be $K_1 = 0.1$, $K_3 = 0.12$ and $K_2 = 1.0$. From those figures that all the robots are attracted to the segment for the first 8 seconds and later on form the triangular pattern while moving forward.

6.5 Experiment-2: square formation of four robots

In this experiment, four robots which are initially randomly scattered are required to form a square pattern. Two straight lines are assigned. For the first 3 seconds, each robot will try to approach the corresponding nearest point on the segment and then try to approach to four virtual points moving along segments are assigned to each robot. Those three virtual points form a triangle pattern and will stop at the vertices of segments. The velocity of virtual points are set to be 20 pix per second. During the process of formation, velocities and headings of each robot are depicted in Figure 23 - 27. Snapshots of video (taken by web-cam) are shown in Figure 28-30. The controller parameters are set to be $K_1 = 0.1$, $K_3 = 0.12$ and $K_2 = 1.0$. From those figures that all the robots are attracted to the segment for the first 3 seconds and later on form the square pattern while moving forward.

![Segment](image.png)  ![Robot](image.png)  ![Vertices](image.png)

Fig. 14. Illustration of segments and robots falling into assigned segments.
6.6 Limitation of locomotion

Each wheel of MRKIT utilizes an independent step motor for locomotion. There are two outstanding drawbacks which impede implementation. As indicated by Equation (11), the desired velocity is proportional to the distance to goal point (i.e., \( r \)). If a robot is initially placed far away from its goal point, the desired velocity will be relatively high. However, step motor usually is weak on its maximum starting speed and starting torque. If the gain \((K_1, K_2, K_3)\) is too high, a robot initially at standstill will immediately miss its step at the very beginning of formation control. The other shortcoming arises from the minimum speed of step motor. Due to the limitation of minimum speed, a wheeled mobile robot in fact can not reach a fixed goal point. Instead it will stop moving once it enters certain range with respect to its goal point. It results in a dead zone to which the robot is prohibited. To reduce dead zone, higher gain is demanded and thus increases the risk of missing steps. Trade-off has to be done for real implementation. To overcome such downside of locomotion, other motors with high starting torque such as permanent magnet brushless DC motor cater for real implementation.

Fig. 15. Velocity of robot 1 during 3-robot triangle formation control.

Fig. 16. Velocity of robot 2 during 3-robot triangle formation control.
7. Conclusion

In this chapter we first consider the stability issue of a generic form of nonlinear feedback control based on kinematic model in polar coordinate in a novel perspective. Some commonly used controls can be derived from this general form of a family of stable control laws. In addition to the commonly used stability analysis based on Lyapunov stability theorems, in this chapter we employ LaSalle’s invariance theorem to investigate the robustness of a point tracking controller. Then the robustness problem of the control law (Lee et al., 1999) is investigated and successfully solved. Thanks to LaSalle’s invariance theorem and Lyapunov’s stability theorem as well, we are able to unveil the whole stable range of controller gains when velocity and angular velocity of each driving motor are not exactly what they are supposed to be in the real world. This study yields some exciting conclusions on converging rate than the counterpart. Based on the robustness analysis, we can propose useful and handy guidelines in determining controller gains and consequently a new robust control law is proposed. Improvement to this control law is achieved from the analysis results and is verified in Matlab.

Fig. 17. Velocity of robot 3 during 3-robot triangle formation control.

Fig. 18. Headings of all robots during 3-robot triangle formation control.
Fig. 19. Snapshot of initial conditions of 3-robot triangle formation control at $t = 0$.

Simulation. Implementation with real robots has been done to demonstrate the application to multiple robot formation control. An implementation of multi-robot formation control has been done and discussed in details.

8. References


Fig. 20. Snapshot of 3-robot triangle formation control at ($t = 8s$).
Fig. 21. Snapshot of 3-robot triangle formation control at \((t = 16s)\).


Fig. 22. Snapshot of 3-robot triangle formation control at \((t = 24s)\).
A Robust Nonlinear Control for Differentialy Mobile Robots and Implementation on Formation Control

Fig. 23. Velocity of robot 1 during 4-robot square formation control.


Fig. 24. Velocity of robot 2 during 4-robot square formation control.
Fig. 25. Velocity of robot 3 during 4-robot square formation control.

2000.


Fig. 26. Velocity of robot 4 during 4-robot square formation control.
A Robust Nonlinear Control for Differentially Mobile Robots and Implementation on Formation Control


---

Fig. 27. Headings of all robots during 4-robot square formation control.

Fig. 28. Snapshot of 4-robot square formation control at (t = 0s).
Fig. 29. Snapshot of 4-robot square formation control at \( t = 6s \).

Fig. 30. Snapshot of 4-robot square formation control at \( t = 12s \).
This book is a collection of 29 excellent works and comprised of three sections: task oriented approach, bio inspired approach, and modeling/design. In the first section, applications on formation, localization/mapping, and planning are introduced. The second section is on behavior-based approach by means of artificial intelligence techniques. The last section includes research articles on development of architectures and control systems.

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