

Analysis on Nonlinear Feedback Controls for Differential Mobile Robots and its Application to Multi-robot Formation Control – Part Two

Jie Wan

Department of Mechanical Engineering
Faculty of Engineering, National University of Singapore.
wanjie@nus.edu.sg

Peter Chen

Department of Mechanical Engineering
Faculty of Engineering, National University of Singapore.
mpechenp@nus.edu.sg

Abstract—This is the second part of the whole paper and in this part, the robustness of a nonlinear control law for mobile robot is examined under Invariance Principles. It provides insight into the stable zone for a given set of controller gains and make possible a more robust controller under some circumstances. Improvement to this control law is achieved from the analysis results and is verified in Matlab simulation. Implementation with real robots has been done to demonstrate the application to multiple robot formation control.

Index Terms—robust control, vision system, formation control

I. INTRODUCTION

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-robot 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 the first part of this paper, the characteristics of two similar nonlinear feedback control laws of differential mobile robots were investigated in a comparative framework. Evolutions of robot's heading under those two controls are studied. Then uniqueness of robot's trajectory with respect to gain ratio and characteristics of trajectory curvature are revealed and lead to the concept of "critical gain ratio". In this part, we are going to investigate the robustness of one of the nonlinear control law and apply it to multiple robotic formation control.

This part is organized as follows: in Section II, the problem is presented. Section III deals with the robustness analysis and its effectiveness is verified via simulation in Section IV. Implementation and experiments with multiple robots on the application of formation control are addressed in Section V, followed by conclusions in section VI.

II. PROBLEM STATEMENT

Regarding the control law stated in previous part given by

$$\begin{aligned} v &= K_1 r \cos(\phi), \\ \omega &= -K_1 \sin(\phi) \cos(\phi) - K_2 \phi, \end{aligned} \quad (1)$$

one may be interested in the question: what if the gain K_1 in v and ω does not match? Consider an alternative to the control law in Equation (1) as follows:

$$\begin{aligned} v &= K_1 \cdot r \cos(\phi), \\ \omega &= -K_3 \sin(\phi) \cos(\phi) - K_2 \phi, \end{aligned} \quad (2)$$

where K_3 is not necessarily equal to K_1 . Then it is equivalent to say: "will the closed-loop system be stable if an alternative control represented in Equation (2) rather than the one in Equation (1) 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 ϕ 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.

$$\begin{aligned} v &= \frac{v_L + v_R}{2}, \\ \omega &= \frac{v_R - v_L}{2C_r}, \end{aligned} \quad (3)$$

where C_r 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:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ \frac{1}{C_r} & \frac{1}{C_r} \end{bmatrix} \begin{bmatrix} v_L \\ v_R \end{bmatrix}, \quad (4)$$

$$\begin{bmatrix} v_L \\ v_R \end{bmatrix} = \begin{bmatrix} 1 & C_r \\ 1 & -C_r \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}. \quad (5)$$

The ideal case of control law Equation (1) is based on the assumption that we can make the equations

$$\begin{aligned} v_L &= K_1 r \cos(\phi) - C_r (K_1 \sin(\phi) \cos(\phi) + K_2 \phi), \\ v_R &= K_1 r \cos(\phi) + C_r (K_1 \sin(\phi) \cos(\phi) + K_2 \phi), \end{aligned}$$

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,

$$\begin{aligned} v' &= (v'_L + v'_R)/2, \\ \omega' &= (v'_R - v'_L)/(2C_r). \end{aligned} \quad (6)$$

Both v' and ω' 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 ω with respect to v , which is the case with control in Equation (2). Substituting Equation (2) into Equation (2) in Part I of this paper, we obtain

$$\begin{aligned} \begin{bmatrix} \dot{r} \\ \dot{\phi} \end{bmatrix} &= \begin{bmatrix} -\cos(\phi) & 0 \\ \frac{1}{r} \sin(\phi) & 1 \end{bmatrix} \begin{bmatrix} K_1 r \cos(\phi) \\ -K_3 \sin(\phi) \cos(\phi) - K_2 \phi \end{bmatrix} \\ &= \begin{bmatrix} -K_1(\cos(\phi))^2 r \\ -K_2 \phi - (\frac{K_3 - K_1}{2}) \sin(2\phi) \end{bmatrix}. \end{aligned} \quad (7)$$

Through studying the stability of the closed-loop system described by Equation (7), we are entitled to investigating the robustness of the alternative control law given in Equation (2).

III. ROBUSTNESS ANALYSIS

A. Stable Zone

We refer to the model in Equation (7) 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.

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

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

$$\dot{\phi} = -K_2 \phi - K_4 \sin(2\phi) \quad (8)$$

As to the subsystem denoted by Equation (8), construct a Lyapunov candidate as $V = \frac{1}{2}\phi^2$. The derivative of V with respect to time is

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

As shown by the closed-loop system equation in Equation (7), this system is time invariant. It means that LaSalle's theorem is applicable. Therefore we are motivated to find out the invariant set $\Sigma = \{(K_1, K_2, K_3) | \dot{V} < 0\}$. 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 1 while the case with $K_4/K_2 < 0$ is shown in Figure 2. 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., $\phi = 0$.
- 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 $c_2 \approx -0.50$.

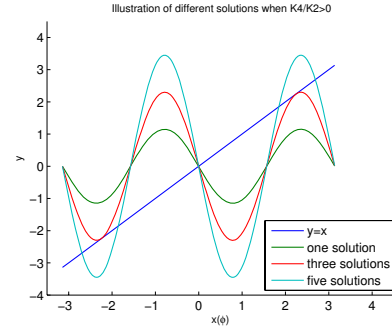


Fig. 1. Illustration of different solutions with $K_4/K_2 > 0$.

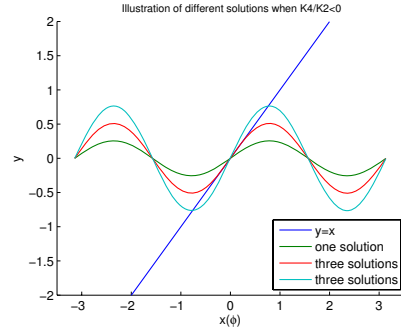


Fig. 2. Illustration of different solutions with $K_4/K_2 < 0$.

To sum up, the ratio K_4/K_2 should be within the range (c_2, c_1) to make subsystem Equation (8) asymptotically stable. Or in other words, the relationship among K_1, K_2, K_3 to make subsystem Equation (8) stable is: $K_1 + 2c_2K_2 < K_3 < 2c_1K_2 + K_1$ ($K_2 > 0$) or $2c_1K_2 + K_1 < K_3 < K_1 + 2c_2K_2$ ($K_2 < 0$).

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:

$$K_1 + 2c_2K_2 < K_3 < 2c_1K_2 + K_1. \quad (9)$$

The stable zone is shown in Figure 3. 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 $K_3 = K_1$, the difference of those two parts of the zone is that the system response will be different as revealed by Proposition 2.

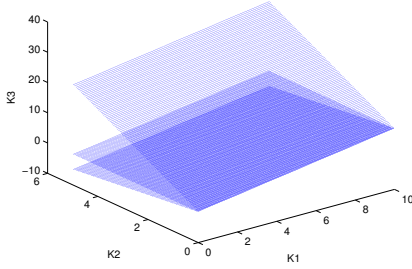


Fig. 3. Illustration of stable zone with practical concerns in the case when $K_1 > 0$ and $K_2 > 0$. Note that the whole zone is separated by a plane $K_3 = K_1$, i.e., $K_4 = 0$.

B. Improvements and Control Design Guidelines

Proposition 1: If $K_4 \geq 0$, namely, $K_3 - K_1 \geq 0$, $r(t)$ is exponentially convergent to zero for arbitrary initial ϕ_0 .

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

Proposition 2: A Set of (K_1, K_2, K_3) in the upper part of the wedge in Figure 3 expedites the response of ϕ 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

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

Let $q(t)$ be the solution of the differential equality $\dot{q}(t) = -2K_2q(t)$, 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 3, 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 3.

- 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 $K_3 = K_1 + c_1K_2$.

IV. SIMULATION STUDY

A. 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.

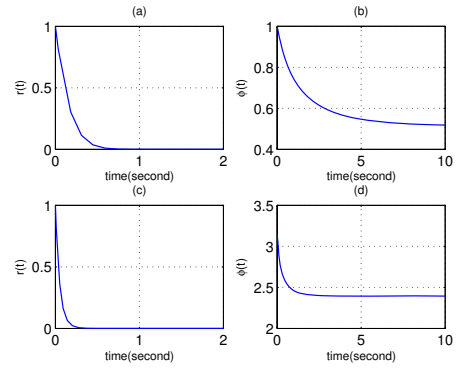


Fig. 4. Illustration of mismatching K_3 and K_1 . In (a) and (b) initial conditions are $\phi_0 = 1$ rad and $r_0 = 1$ and gain $K_1 = 20$, $K_2 = 1$ and $K_3 = 18.8$ (i.e., -6% deviation). And in (c) and (d) initial conditions are $\phi_0 = \pi$ rad and $r_0 = 1$ and gain $K_1 = 20$, $K_2 = 1$ and $K_3 = 24.8$ (i.e., 24% deviation).

The simulation results of system response are depicted in Figure 4 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 (9).

B. 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 3, 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 $K_3 = 22.30, 21.8$ 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 5. From this figure, it is noticed that compared with $K_3 = 20$, a set in upper part of the wedge in Figure 3 contributes to expediting the system's response while a set in lower part of

the wedge will negative 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 $t \rightarrow 0$.

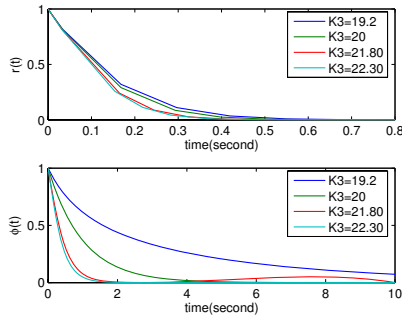


Fig. 5. Illustration of the effects of mismatching K_3 on system response. Initial conditions are $\phi_0 = 1$ rad and $r_0 = 1$ and gain $K_1 = 20$, $K_2 = 1$ and $K_3 = 19.2, 20, 21.80, 22.30$ respectively.

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

$$J_n = \int_0^t (v^2(\tau) + \omega^2(\tau)) d\tau. \quad (10)$$

In this simulation, the integral of the norm squared of the actual velocity signals for each controller is shown in Table I. 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 not significant difference between the two recommended control laws, i.e., $K_3 = 21.80$ and $K_3 = 22.30$ respectively.

TABLE I
COMPARISON OF THE INTEGRAL OF THE NORM SQUARED OF THE VELOCITY INPUT SIGNALS $\int_0^{30} (v^2(t) + \omega^2(t)) dt$.

$K_3 = 19.2$	$K_3 = 20$	$K_3 = 21.80$	$K_3 = 22.30$
331.8	129.8	69.78	63.87

V. IMPLEMENTATION AND EXPERIMENT

A. Overview of Implementation

One picture of the real implementation is presented by Figure 6. 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 6 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 6. The

main parts of this implementation are connected as shown by Figure 8.



Fig. 6. Picture of real robots, test bed(on the floor), CCD color camera with wide-angle lens and one web-cam(mounted on the ceiling).

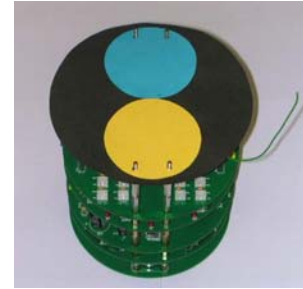


Fig. 7. Picture of one MRKIT mobile robot with on-board color pads.

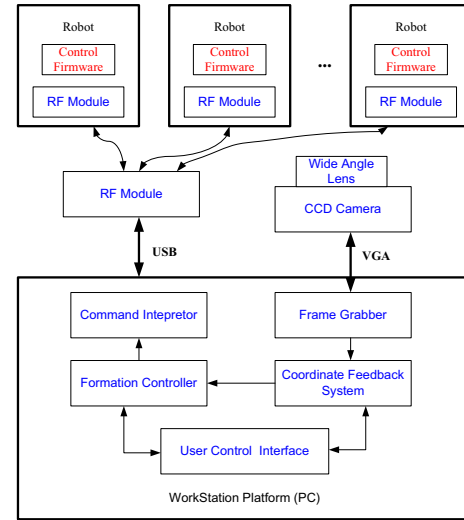


Fig. 8. Illustration of connection of the whole implementation.

B. 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

$$V = \frac{D\pi}{NP},$$

where $D = 54$ mm is the diameter of the wheel; $N = 400$ is the step of motor per revolution; P is the time (second) per step and $P = \frac{T \cdot 10^{-6}}{2.5}$. Finally we arrive at $V = \frac{1060.288}{T}$ 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 $V_{min} = 0.0162$ m/s and a maximum $V_{max} = 0.3$ m/s.

C. Vision System: Resolution and Noise Analysis

A vision system, comprising of CCD camera, lens, frame grabber and application program as shown in Figure 8, 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 7. 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 $(\frac{x_a+x_b}{2}, \frac{y_a+y_b}{2})$. Meanwhile the orientation of the robot can be calculated as $\theta = \cos^{-1}(\frac{x_a-x_b}{\rho})$, where $\rho = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}$.

The measurement of the 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 affect 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 accuracy and resolution of 1 millisecond while the system resource used is relatively small. We set the sampling rate to be 500 Hz. A period of 2 minutes is used to sample the data and for example, the error signal along x -axis is presented in Figure 9 (figures of error signal along y -axis the associated orientation error signal are omitted for sake of brevity). Then we apply the FFT technique and 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.

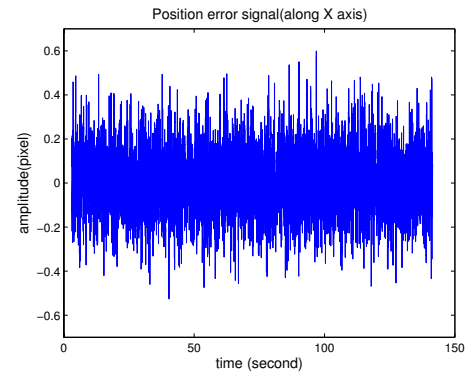


Fig. 9. Position error signal along x -axis with sample rate $f = 500$ Hz.

D. Experiment: Triangle Formation of Three Robots

A scalable formation control scheme is introduced in [GF05]. 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 10 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 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

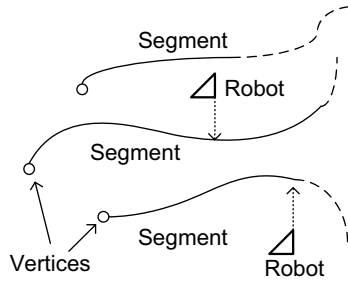


Fig. 10. Illustration of segments and robots falling into assigned segments.

straight lines are assigned to three robots respectively. For the first 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. Snapshots of video (taken by web-cam) are shown in Figure 11-15. 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.

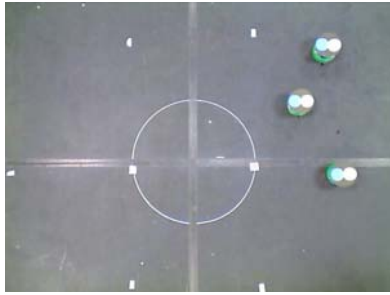


Fig. 11. Snapshot of initial conditions of 3 robots when $t = 0$.

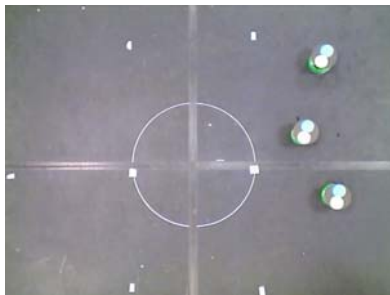


Fig. 12. Snapshot of 3 robots when falling into segments ($t = 4s$).

VI. CONCLUSION

In this part of the paper, we studied the robustness of a nonlinear feedback control law based on kinematic model. It provides insight into the stability zone for a given set of

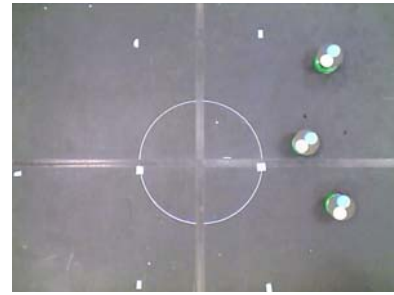


Fig. 13. Snapshot of 3 robots arrived at the nearest points on the segments ($t = 8s$).

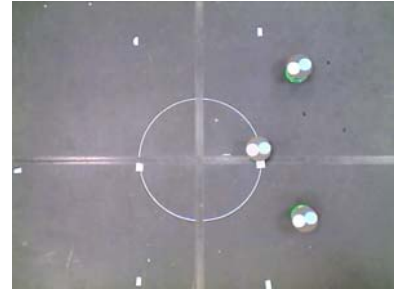


Fig. 14. Snapshot of 3 robots when moving to vertices of segments in a triangle pattern ($t = 16s$).

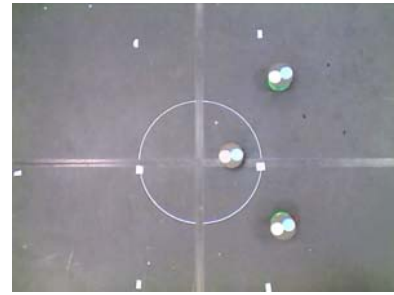


Fig. 15. Snapshot of 3 robots when moving to vertices of segments in a triangle pattern ($t = 24s$).

controller gains and it is found that under certain circumstances the closed-looped system may fails in reaching the desired control objectives and performance. Based on the robustness analysis, we are able to improve the controller design by choosing proper controller gains K_3 when K_1 and K_2 are given. This new control can overcome the drawbacks of the common nonlinear control law discussed. Guidelines of the improved control law are provided to facilitate real implementation. Several real robots have been used to demonstrate its application in multiple robot formation control.

REFERENCES

- [GF05] S. S. Ge and C.-H. Fua. Queues and artificial potential trenches for multi-robot formations. *IEEE Transactions on Robotics*, 21(4):646–656, Aug. 2005.