SUT-CARG Car-Like Robots: Their Electronics and Control Architecture

SUCHART PUNPAISARN, SARAWUT SUJITJORN†
School of Electrical Engineering, Institute of Engineering
Suranaree University of Technology
111 University Avenue, Muang District, Nakhon Ratchasima, 30000
THAILAND
Suchart_lek@hotmail.com http://www.sut.ac.th/engineering/electrical/carg/
†corresponding author: sarawut@sut.ac.th

Abstract: This article presents the development of the SUT-CARG car-like robot system. CARG car-like robots consist of four mobile robot agents and are built as integrated systems with IR, sonar, ultrasonic, vision sensors and controller modules. Hardware details are given. The software interface module for commanding the robots is described. The article also presents the details of robot following control which utilizes geometrical control approach based on vision and image processing. Colour tracking and flocking of the robot agents are demonstrated (VDO available online http://www.sut.ac.th/engineering/electrical/carg/).

Key-Words: Car-Like Robot, Visual Servo, Image Processing, Colour Tracking, Robot Following Control, Flocking.

1 Introduction

In the last few years, multivehicle platforms have been developed by several universities and research labs. Some high-level controls were accomplished by using remote workstations. Researchers at Brigham Young University used a multivehicle testbed for cooperative robotic research in MAGICC lab [1]. Their initial system consisted of five nonholonomic robots with an onboard Pentium-based computer and wireless communications. Cornell University's RoboFlag testbed [2] included several small robots with local control loops functioning cooperatively to achieve a common goal. Designed for indoor experiments, these robots used an overhead camera for localization. Each robot carried a simple microcontroller unit controlling the actuators onboard. Some of the robot systems involved hovercraft vehicles such as the Caltech multivehicle wireless testbed and the University of Illinois HoTDec [3]. Other testbeds featured unmanned ground vehicles and unmanned aerial vehicles, such as MIT's multivehicle testbed [4], the University of Pensylvania's multiple autonomous robots (MARS) testbed [5,6]. The University of Toronto Institute for Aerospace had successfully demonstrated the practical use of cyclic pursuit as a distributed control strategy for multiwheeled-robot systems. The MARS platform used ground vehicles based on the Tamiya TXT-1 chassis, which was used in COMET for coordinating control algorithms [7]. The VSB-Technical University of Ostrava had developed embedded control system for the soccer robot with better performance and safety function [8].

In this article, Suranaree University of Technology-Control & Automation Research Group (SUT-CARG) presents a car-like robots platform to be used for future research as a testbed for intelligent control algorithms. A CARG car-like robot consists of four mobile robot agents. Each agent is equipped with an onboard embedded microcontroller interfaced with several sensors, and a wireless network access module enabling all robots to be controlled through a computer. CARG car-like robots are built as integrated systems of available hardware and software modules. Software pieces have be coded in embedded C. The paper describes the integration of those existing technologies. Experiments on robot following control, colour tracking and flocking of the robots are described and illustrated. In addition, robot models suitable for design are presented with simulation results.

2 Hardware, Modelling and Simulation

CARG car-like robots consist of 4 mobile agents based on the TLT-1 (Tamiya, a Little Truck), a one-eighteenth scale radio-controlled (RC) 4x4 Pickup Truck from TAMIYA, Inc. Each truck is modified to be a member of the agents aimed for an operation in a laboratory with flat terrain. Each robot can

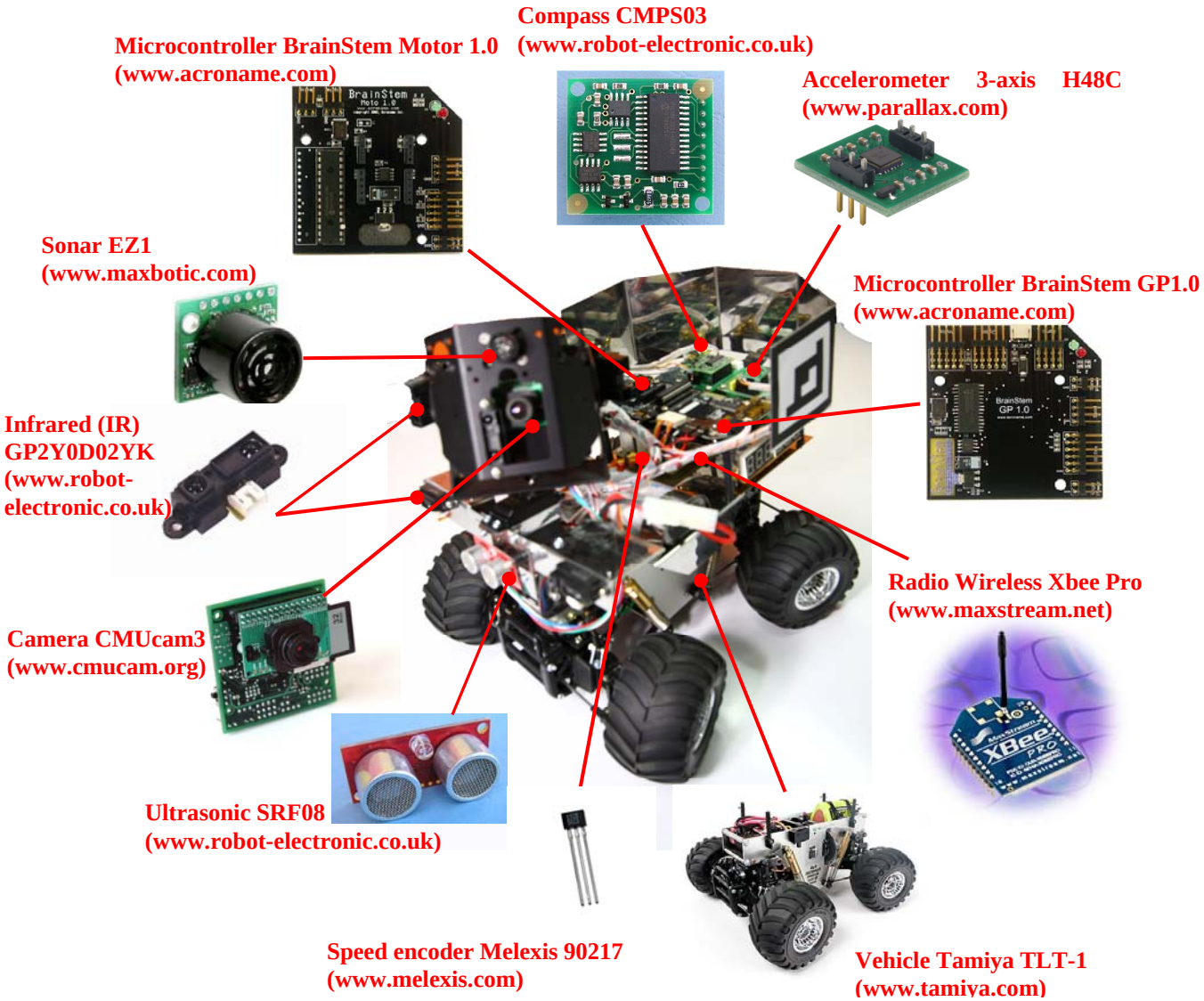


Fig. 1 CARG car-like robot.

support a load of 3 kg for various electronic devices including sensors, controller boards and other accessories. The list of the robot's components and associated vendors can be found in Fig. 1. Each CARG car-like robot uses various sensors that are implemented in the robot control system with the mechanical specifications shown in Table 1. The robot is recommended for laboratory experiments but robust enough for outdoor operations.

Referring to the Xbee pro, the wireless radio module is used to communicate with the onboard computer for signal monitoring and data recording. The Xbee pro consumes low energy from a 3.5 V battery and provides high speed communication (up to 250 kbps). Onboard sensors consist of two infrared sensors (GP2Y0D02YK, GP2Y0D21Y0F), an ultrasonic ranger (SRF08), a sonar (EZ1), an electronic compass (CMPS03), and a camera (CMUcam3), which are coordinated and interfaced via a GP1.0 microcontroller. This GP1.0 microcontroller acquires the robot motion data via

Table 1 CARG car-like robot's mechanical specification.

Dimensions	Length	30 cm
	Width	22 cm
	Height	25 cm
Weight	Load (Max)	3 kg
	Unload	1.5 kg
Max Speed		1.6 m/s
Max Torque		55 N/m
Carry Capacity		2.5 kg
Min Steering Radius	2 wheel steering	80.5 cm
	4 wheel steering	30 cm

each sensor on a 5,600 baud sampling rate.

The CMUcam3 is used for vision purposes, when the CARG car-like robots attempt to perform group behaviours, for instance flocking [9], formation of robot [10], multi-robot cooperation and communication systems [11]. The camera is also used to identify, and assist on the robot following control as well as estimate their positions and orientations. Moreover, the CARG car-like robot applies CMUcam3 in various tasks, for instance color tracking (CARG Hunting Robot), recognition and position information, obstacle detection, and route identification. The CMUcam3 functions independently on its own microcontroller. For more information, readers may visit the website www.sut.ac.th/engineering/electrical/carg.

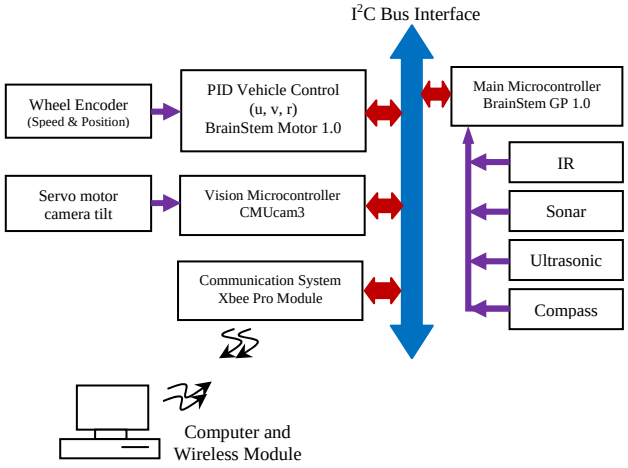


Fig. 2 CARG car-like robot control units.

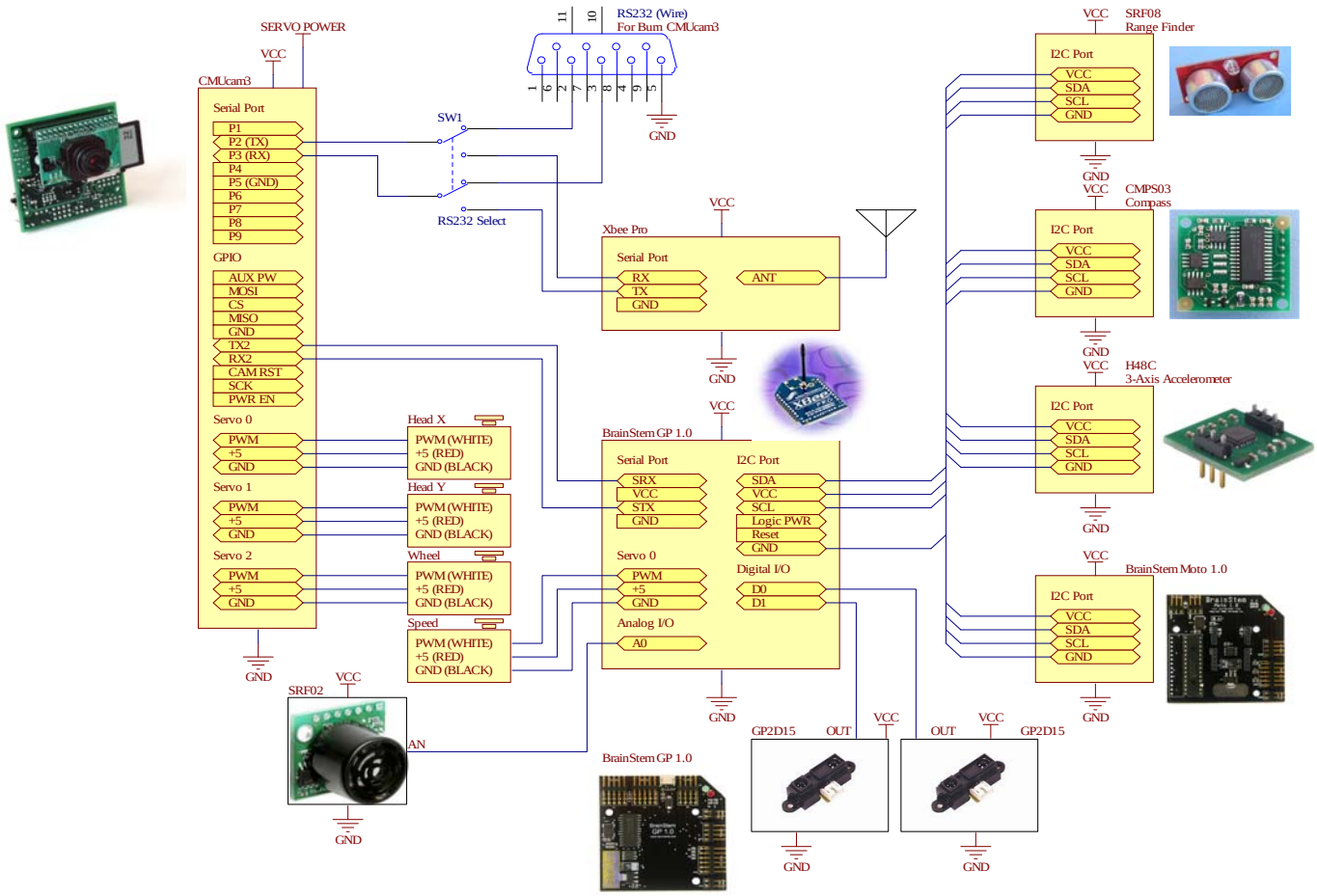


Fig. 3 CARG car-like robot schematic diagram.

Two infrared (IR) arrays (GP2Y0D21Y0F, 24 cm range and GP2Y0D02YK, 80 cm range) are mounted on the front part of the CARG robot to detect the ranges of obstacles. Each IR sensor gives its output in binary proportional to the detected distance against the obstacle. The sensor transmits the distance information via I^2C bus in realtime.

One ultrasonic range finder (SRF08) is mounted on the front part of the robot, another one on the rear part. The sensor offers a measurement range of 0-6 m with a wide viewing angle of 60-degree beaming. The measurement range is set to 0-1.5 m for our CARG robots. These sensors communicate with the GP1.0 microcontroller board through an I^2C interface bus.

MaxSonar-EZ1 is a very compact and versatile sonar distance sensor. It outputs range information in three forms: TTL serial, PWM and 0-2.5 V analog. The MaxSonar-EZ1 has a scanning rate of 20 Hz (50mS). Interfacing modes are selective as serial I^2C (0 to Vcc with 9,600 baud rate), analog (Vcc/512)/inch and PWM (147 μs/inch), respectively. CARG robot has one MaxSonar-EZ1 on its front. To achieve a wider beaming angle, it is possible to mount two more of these sonar sensors.

An electronic compass (CMP03) is specifically used as an aid to navigation and useful for the flocking movement control. It is sensitive to the earth magnetic field.

Eventhough an optical encoders is extremely insensitive to noise, it is not suitable for an environment subject to daylight. A hall effect sensor, Melexis 90217, is chosen and modified to work with 6 magnets to provide a better resolution for the CARG car-like robot. By measuring wheel rotation with an appropriate signal conditioning circuitry, the sensor is capable of detecting the translational and rotational speeds of the robot.

The CARG car-like robot control system is designed to provide a simple interface between sensors and microcontrollers. The control structure of the robot is represented by the diagram shown in Fig. 2. Some of the sensors (IR, sonar, ultrasonic range finder and electronic compass) are connected directly to the main controller (GP1.0) for realtime data acquisition. Three controller modules (motor controller, CMUcam3 and wireless communication modules) are connected to the main controller through the I^2C bus interface. The schematic diagram in Fig. 3 represents the interfacing of these components. For details of the signals and component specifications interested readers should refer to the vendor's document.

Modelling and simulation: CARG robot dynamic model is analyzed based on 3 degree of freedom (DOF) of a nonlinear system. This case study refers to the references [12, 13] that offer the models relating the longitudinal movement (u), the lateral movement (v) and the yaw angle (r). The robot acceleration and velocity vectors of the fundamental dynamic model describing the behaviour of a robot is given by equations (1) – (3). The controller design problem becomes finding the control δ_T and T_δ so that the system converges to the equilibrium position.

$$\dot{u} = vr + \frac{u^2(fk_1 - k_2)}{M} + \frac{\delta_T}{M} \quad (1)$$

$$\dot{v} = -ur - \frac{(c_f + c_r)v}{Mu} + \frac{(bc_r - ac_f)r}{Mu} + \frac{T_\delta}{M} \quad (2)$$

$$\dot{r} = -\frac{Mfhur}{I_z} - \frac{(ac_f - bc_r)v}{I_z} - \frac{(b^2c_r + a^2c_f)r}{I_z u} + \frac{aT_\delta}{I_z} \quad (3)$$

Using the models (1-3) with the following initial conditions: $u(0)$ = 10 m/s, $v(0)$ = 0.5 m/s, $r(0)$ = 0.05 rad/s, $\delta(0)$ = 0 rad and T(0) = 0 N, the simulation results as shown in Fig. 4 can be obtained. The models will be further used for our control system design to be based on a new architecture of ANNs.

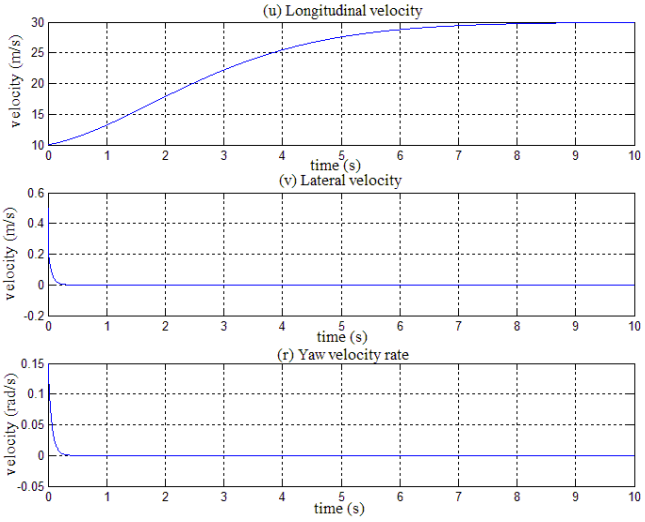
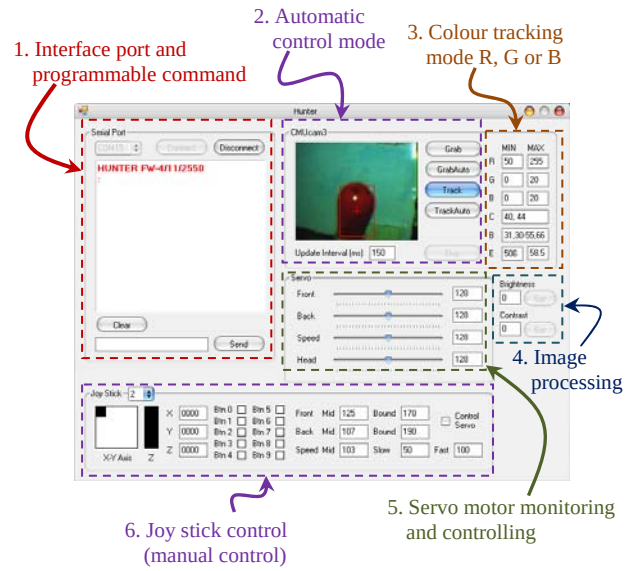


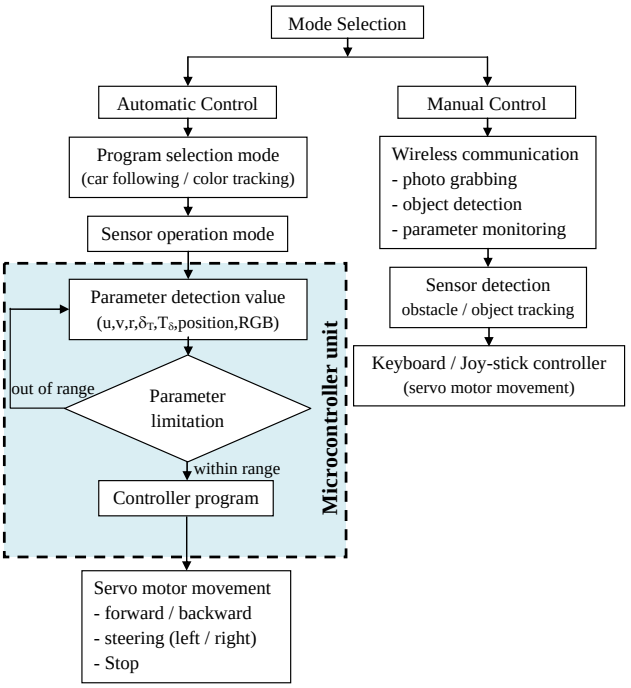
Fig. 4 u, v and r converge to equilibrium.

3 Software Interface Module

Control and interface program (CIP): Control software and image processing programs are written in C. To coordinate the multi robot network, a piece of software called "control and interface program" (CIP) for man-machine interface has been developed. There are monitoring and commanding



(a)



(b)

Fig. 5 Control and interface program (a) user interface window, (b) modes of CIP functions.

windows for a user to access and control the CARG robot. As in Fig. 5(a) the user can command the robot through the computer's keyboard, mouse or joy-stick. In SUT-CARG lab, the CARG robot can be connected to a wireless access point and run on a laptop/desktop computer with wireless capability. There are two models of operation (automatic and manual modes) which is user selective. The diagram in Fig. 5(b) represents the functions of the CIP.

Operation modes: The CIP has several features that allow the user to configure the robot's levels of autonomy. Through a wireless communication module, the user can set the following requirements for robot: longitudinal and turning velocities, photo grabbing, colour tracking mode, brightness and contrast control. Under an automatic control mode, one of the robots can be setup as a master or a slave for the robot following mode. Furthermore, a journey-path can be assigned to each robot via the command window of the CIP. Flocking movement mode is demonstrated by 4 robots agent in Fig. 6.

Fig. 6 CARG robots in flocking mode.

4 CARG Robot's Control System

Obstacle avoidance and steering: Obstacle avoidance is a prime necessity of an autonomous robotic system [14]. The obstacle avoidance behaviour can be seen as a part of a hierarchical hybrid system [15] which combines both continuous and discrete events.

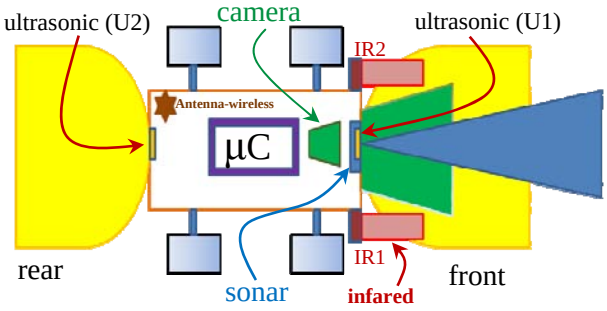


Fig. 7 Sensors' locations.

In this work, some simple heuristic rules in accordance with sensors' specifications are used to plan robots' trajectories. The obstacle avoidance will be operated once the sensor detects an object being closer than a preset distance. Fig. 7 shows the locations of all sensors used for the obstacle detection of the CARG robot. There are two ultrasonic sensors (U1-U2) for 0-1.5 m detection at front and rear, a Z1 sonar for wide angle and 0-2.5 m detection, and two infrared sensors (IR1-IR2) for 0.28 m short range detection for the front wheels of the robot.

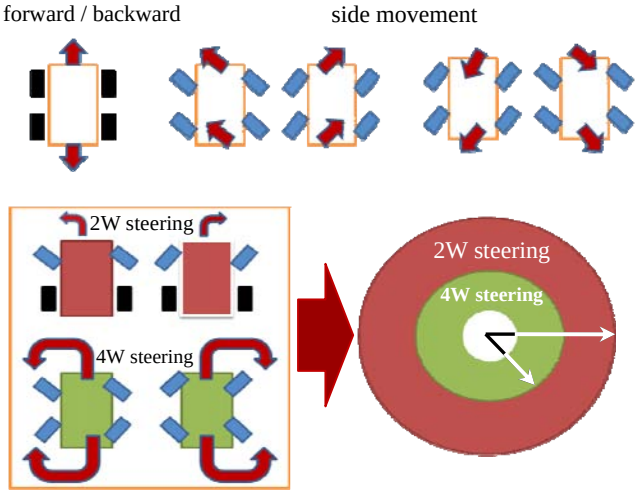


Fig. 8 CARG robots' steering capability.

A CARG robot has an improved steering ability by using a four-wheel steering instead of a front-wheel steering. This reduces the steering radius to 30 cm, about 63% shorter than the previous one of 80.5 cm. Fig. 8 depicts all possible movements of each robot and its steering capability.

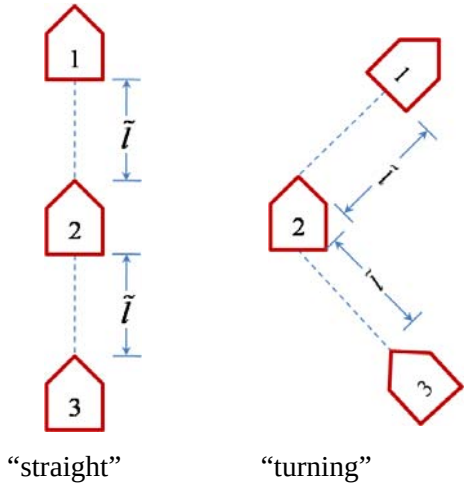


Fig. 9 Robot following patterns.

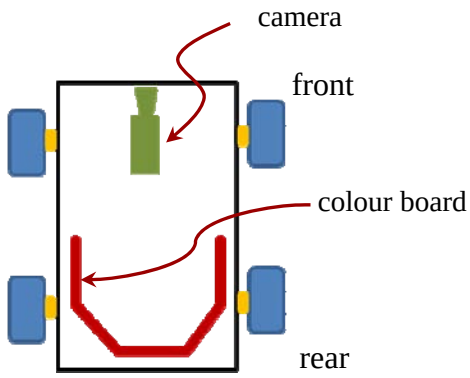
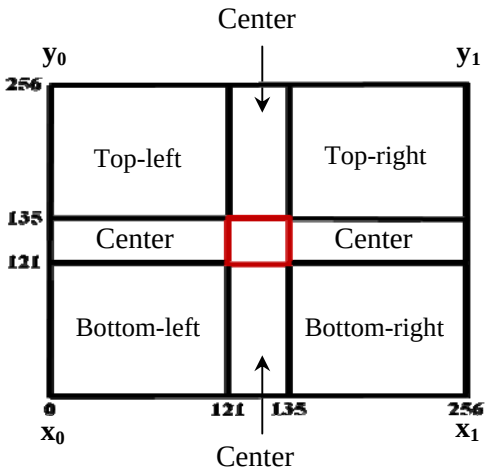
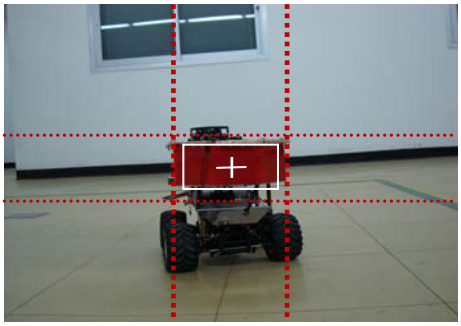


Fig. 10 Top view of a car-like robot



(a)



(b)

Fig. 11 (a) Vision grasped by the CMUcam3 camera, (b) segmented vision for control purposes.

Robot following control: In robot following control, the distance, $\tilde{l}$, between two robots is maintained as shown by the diagrams in Fig. 9. It is accomplished by a visual servo control system. There is a colour board fixed on the top of the rear part of each car-like robot. The shape of the board

can be seen from the picture in Fig. 10. The control described in this section needs proper vision and image processing as well as geometrical computing. These require appropriate algorithms and software. Recently, new geometric concepts in mathematical and computational morphology are applied in robotic navigation [16], and geometry projective visual artificial landmark for robot localization [17].

Image processing: As seen from the diagram in Fig. 10, there is a camera installed at the front of each robot. The camera is a pinhole type with the maximum resolution of 352× 288 pixels at 50 frames per second. The SUT-CARG robots utilize an available vision of 256 × 256 pixels (image size) at 12 frames per second. The vision grasped by the camera is shown in Fig. 11(a) by the white lines with its center indicated by the cross. The vision as depicted in Fig. 11(b) is segmented according to our control and image processing approaches.

The image processing algorithm must be able to locate the center of the image grasped. The algorithm listed in the embedded C language for the CMUcam3 camera is given in Fig. 12. Referring to the list, the main loop computes the coordinate (x,y) of the center point designated as C_x and C_y. It utilizes a special function "pixel (x,y)_colour" of the embedded C. The returned values of R, G, and B must be bounded within their corresponding minimum and maximum values denoted by Rmin, Rmax, Gmin, Gmax, Bmin, and Bmax, respectively.

```
//variable initialization
. . .
Num = 0
x0 = MAX_x , y0 = MAX_y
x1 = 0 , y1 = 0
C_x = 0 , C_y = 0
// loop to compute the coordinate (x,y) of the center point.
Seq_count = 0
For y = 0 to y = MAX_y
    For x = 0 to x = MAX_x
            R = pixel(x,y).red      //  identify red color with "R".
            G = pixel(x,y).green    //  identify green color with "G".
            B = pixel(x,y).blue     //  identify blue color with "B".
            If((Rmin ≤ R < Rmax) and
              (Gmin ≤ G < Gmax) and
              (Bmin ≤ B < Bmax)) then
                  Seq_count = Seq_count + 1
                  If (Seq_count ≥ NOISE_FILTER) then
                        Num = Num + 1
                        If (x0 ≤ x) then x0 = x
                        If (y0 ≤ y) then y0 = y
                        If (x1 ≥ x) then x1 = x
                        If (y1 ≥ y) then y1 = y
                        C_x = C_x + x      //  updating the coordinate C_x.
                        C_y = C_y + y      // updating the coordinate C_y.
                  Endif
            Else
                  Seq_count = 0     // reset counter.
            Endif
        Next x
Next y
. . .
End
```

Fig. 12 Image processing algorithm listed in C.

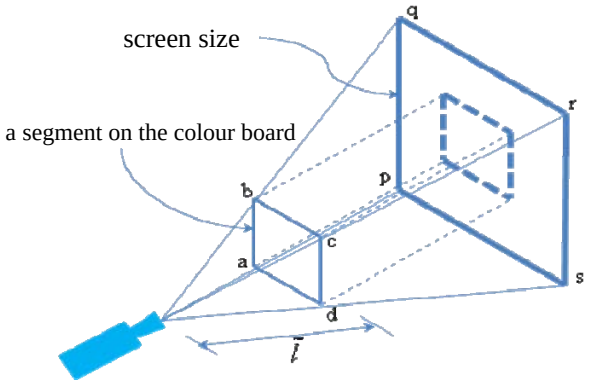
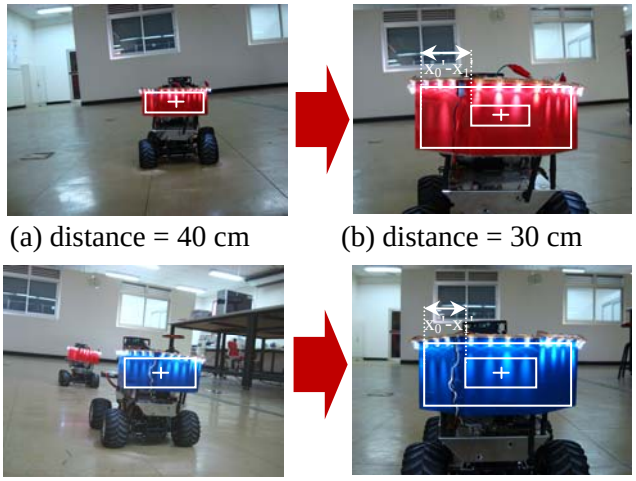


Fig. 13 Estimation of $\tilde{l}$.



(a) distance = 40 cm (b) distance = 30 cm

(c) distance = 39 cm (d) distance = 31 cm

Fig. 14 Visions grasped and estimated $\tilde{l}$.

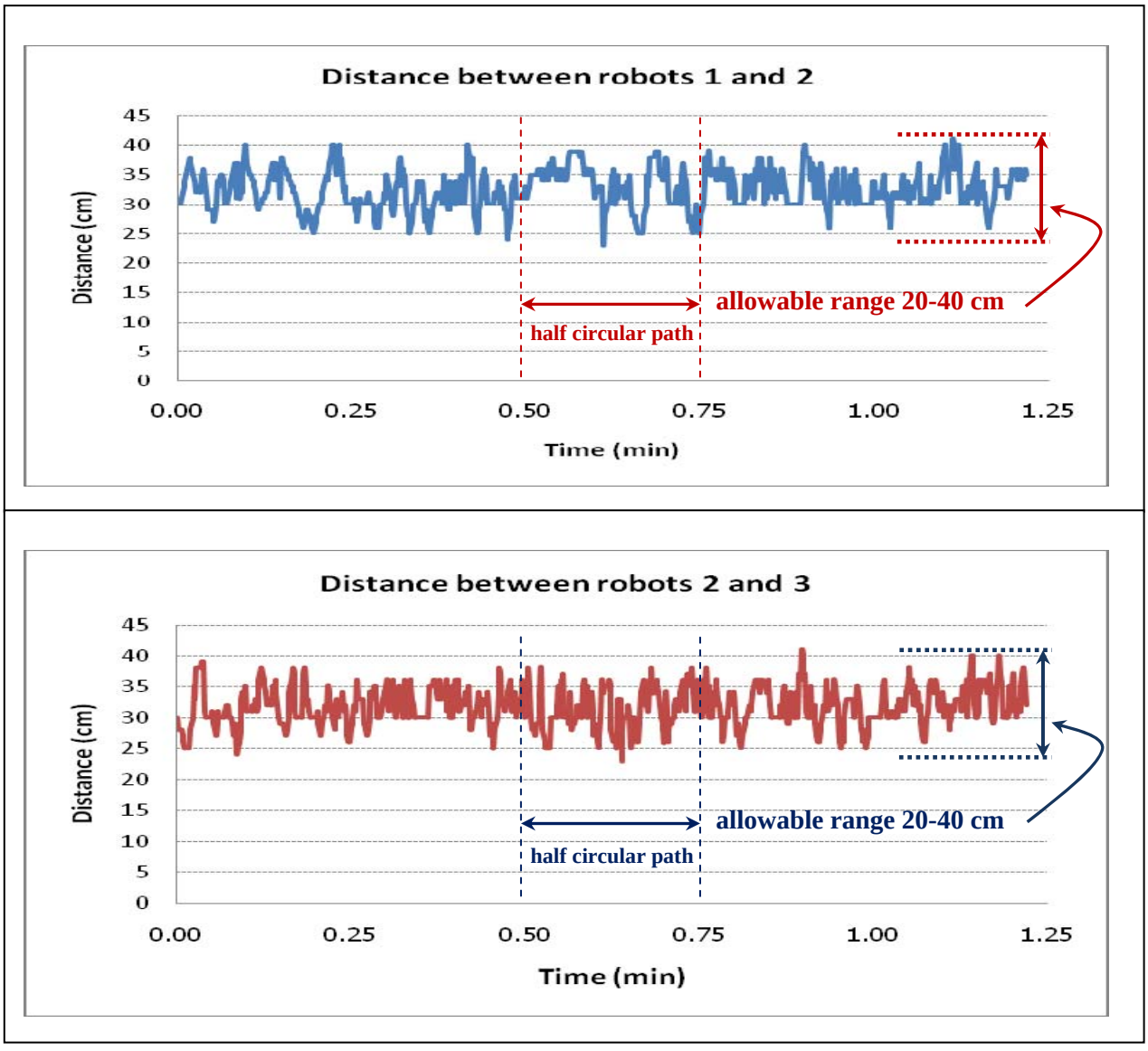


Fig. 15 $\tilde{l}s$ while robots in motions.

The meaningful pixels are counted by the counter "Seq_count". The last if-then block calculates the coordinates C_x and C_y providing that the count is greater than a threshold denoted by "NOISE_FILTER". The threshold is set according to the colour shade and the picture size.

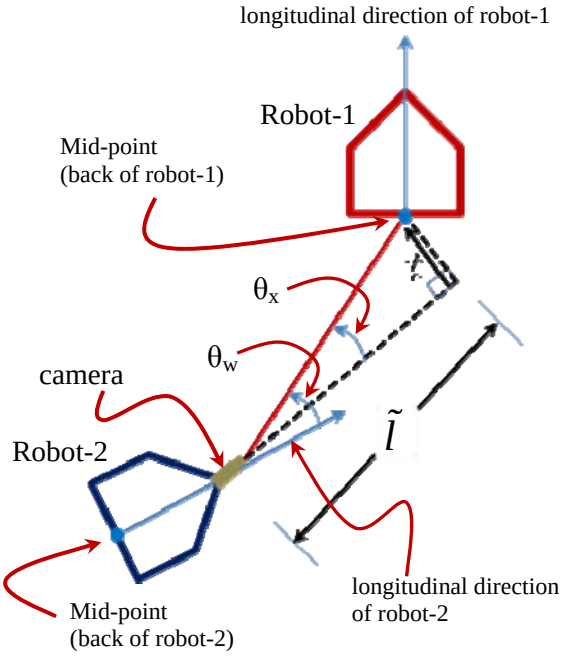


Fig. 16 Representation of car following turning control.

Robot separation control: To achieve a desired robot following pattern, the distance $\tilde{l}$ has to be maintained. Since an image-based servo control at this stage is not so accurate as a conventional sensor-based control, our control has to consider an allowable range of $\tilde{l}$. The implementation employs the $\tilde{l}$ range of 20-40 cm.

The $\tilde{l}$ can be estimated by some geometrical computing. This is represented by the diagram in Fig. 13 in which the rectangle pqrs is the known full capacity screen size of the camera. The vision grasped by the camera while the robot is moving is the rectangle abcd. The $\tilde{l}$ can be estimated proportionally to the area of the abcd-shape related to that of the pqrs-shape. Fig. 14 illustrates the sizes of the images grasped by the camera sensor for various distances. Fig. 15 shows the $\tilde{l}$ values between two following robots in motions.

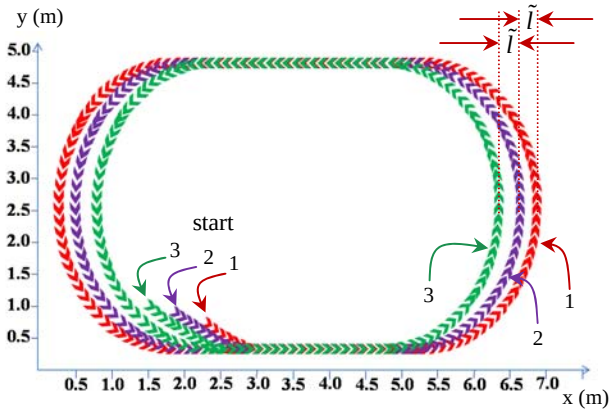


Fig. 18 Travelling courses.

Turning control: To aid an understanding, consider the diagram shown in Fig. 16 in which robot-1 is the leading robot, and robot-2 is the following one. For the robot-2 to follow the robot-1, the front wheels of the robot-2 must be turned by the angle θ_w, and the camera must be panned by the angle θ_x. The traverse displacement of the robot-2 is about X_c. $+X_c$ is reserved for left turning while $-X_c$ for right turning. The turning control can be achieved by implementing the simple algorithm listed in Fig. 17.

```
// finding the average values for the coordinate of center points.
// C_x and C_y are returned by the image processing algorithm.
// num is number of data.
C_x = C_x / num
C_y = C_y / num
Density = num / ((x1 – x2) * (y1 – y2)) * 100          // % density
      If (θx ≥ C_servo + err_margin) then
         (pan camera to the left by θx) and
         (turn front wheels to the left by θw)
      If (θx ≤ C_servo - err_margin) then
         (pan camera to the right by θx) and
         (turn front wheels to the right by θw)
```

Fig. 17 Turning control algorithm.

The algorithm begins with calculating the average values of C_x and C_y, then calculating the percent density. Regarding to this, the %density is discarded if it is low (≤ 60%), and the image processing algorithm explained earlier is repeated until an appropriate %density is found. Two logical statements provide decision making for camera panning control and wheel turning control. C_servo is defined by an angular position of the servo motor that matches with the center of the vision grasped (indicated by the white cross in Fig. 11) by the camera. This visual servo control can be achieved by panning the camera with some error margins allowance. At low level, camera panning and wheel turning utilize separate feedback control loops of DC-servo motors with corresponding PI-controllers.

Referring to the measured travelling courses of 3 robots in Fig. 18, robot-1 is the leading robot followed by robot-2 and robot-3, respectively. The three robots move nicely at an approximate speed of 16 m/min in straight and half circular courses. A VDO clip showing the robot flocking motion is available on website http://www.sut.ac.th/engineering/electrical/carg/.

5 Conclusions

SUT-CARG car-like robots have been built from the integration of available hardware and software modules. CARG-robots consist of 4 robot agents that can be independently controlled by the CIP software. CIP has been developed to control flocking and color tracking through a selection on a screen display. The software has been coded in embedded C for the particular hardware component described in the paper. Control of the robots utilizes computer vision, image processing and geometrical control approach. The robots exhibit satisfactory robot following patterns. However, the work still needs further improvement on visual-based robot following control to achieve a more accurate separation distance ($\tilde{l}$). Also in the near future, an ANN control system will be implemented on the CARG-robots platform for more flexible algorithms and control capabilities.

6 Acknowledgements

The authors wish to thank Suranaree University of Technology for the research grants as well as expenses for the conference participation.

References:

[1] T.W. McLain and R.W. Beard, Unmanned air vehicle testbed for cooperative control experiments, *Proc. American Control Conf.*, Boston, MA, 2004, Vol. 6, pp.5327-5331.

[2] R.D. Andrea and M. Babish, The RoboFlag testbed, *Proc. American Control Conf.*, Boston, MA, 2004, Vol. 1, pp. 656-660.

[3] Z. Jin et. al., MVWT-II: The second generation Caltech multivehicle wireless testbed, *Proc. American Control Conf.*, Boston, MA, 2004,Vol. 6, pp. 5321-5326.

[4] E. King et. al., Coordination and control experiments on a multi-vehicle testbed, *Proc. American Control* Conf., Boston, MA, 2004, Vol. 6, pp. 5315-5320.

[5] L. Chaimowicz, B.Grocholsky, J.F. Keller, V. Kumar, and C.J. Taylor, Experiments in multi-robot air-ground coordination, *Proc. IEEE Int. Conf. Robotics Automation,* New Orleans, LA, 2004, Vol. 4, pp. 4053-4058.

[6] A. Joshua et. al., Experiments in multi-robot coordination, *Robotics and Autonomous Systems*, 2006, Vol. 54, pp. 265-275.

[7] D. Cruz, J. Mc Clintock, B. Perteet, O.A.A. Orqueda, Y. Cao and R. Fierro, A Multivehicle platform for research in networked embedded systems, *IEEE Control Systems Magazine*, 2007, Vol. 27, pp. 58-78.

[8] J. Kotzian, M. Zdenek and S. Vilem Jr., Embedded control system for the mobile robot, *WSEAS Trans. on Systems*, Vol. 4, No. 12, December, 2005, pp. 2261-2268.

[9] R. Olfati-Saber, Flocking for multi-agent dynamic systems: Algorithms and theory, *IEEE Trans. Automat. Control*, 2006, Vol. 51, No. 3, pp. 401-420.

[10] R. Fierro, A.Das, V. Kumar, and J.P. Ostrowski, Hybrid control of formations of robots, *Proc. IEEE Int. Conf. Robotics Automation*, Seoul, Korea, 2001, pp. 157-162.

[11] R. Fierro, L. Chaimowicz, and V. Kumar, Multi-robot cooperation, *Autonomous Mobile Robots: Sensing, Control, Decision Making and Applications (Control Engineering series)*, New York: Taylor & Francis, 2006, ch. 11, pp. 417-459.

[12] A. Alloum, A. Charara and H. Machkour, Parameters nonlinear identification for vehicle's model, *Proc. IEEE, Conf. Control Applications*, Hartford, CT, 1997, pp. 505-510.

[13] J.R. Zhang, A. Rachid and S.J. Xu, Velocity controller design for automatic steering of vehicles, *Proc. American Control*

Conference, Arlington, VA, 2001, pp. 696-697.

[14] J. Borenstein and U. Raschke, Real-time obstacle avoidance for non-point mobile robot, *SME Trans. Robot. Res.*, 1992, Vol 2, pp. 2.1-2.10.

[15] R.Alur et. al., Hierarchical hybrid modeling of embedded systems, *Embedded Software, (LNCS series 2211)*, New York: Springer-Verlag, 2006, pp. 14-31.

[16] A. Aqarwal, Y. Bakopoulos, T. Raptis, Y. Doxaras, E. Kotsealos and S. Kouremenos, New geometric concepts in mathematical and computational morphology, *WSEAS Trans. on Information Science and Applications*, Vol. 3, No. 10, October, 2006, pp. 1976-1984.

[17] A. Gonzalez-Lorence, P. M. Gardunno, R. de los Seqovia and J. Armando, Geometry-projective visual landmarks for robot localization, *WSEAS Trans. on Computers*, Vol. 5, No. 2, February, 2006, pp. 463-468.

Notation List

M = Mass of the full vehicle, (kg)
T = Traction or braking force (N)
I_z = Initial moment around z-axis, ($kg.m^2$)
a = Distance from front tyres to CG, (m)
b = Distance from rear tyres to CG, (m)
f = Rotating friction coefficient
g = Acceleration of gravity force, (m/s^2)
h = Height from CG to road, (m)
$\tilde{l}$ = Distance between two robots in following pattern, (m)
c_f = Cornering stiffness coefficients of front tyres, (N/rad)
c_r = Cornering stiffness coefficients of rear tyres, (N/rad)
k_1 = Aerodynamic lift parameters, ($N.s^2/m^2$)
k_2 = Aerodynamic drag parameters, ($N.s^2/m^2$)
δ = Wheel steering angle, (rad)