

## **DISTRIBUTED CONTROL ARCHITECTURE OF AN OMNI-DIRECTIONAL AUTONOMOUS GUIDED VEHICLE**

**N.S. Tlale**

Department of Industrial and Systems Engineering  
University of Pretoria, South Africa  
[nkgatho.tlale@up.ac.za](mailto:nkgatho.tlale@up.ac.za)

### **ABSTRACT**

Omni-directionality is the ability of a mobile robot to move instantaneously in any direction. This paper describes the wheel and controller designs of a Mecanum-wheeled, autonomous guided vehicle (AGV) for reconfigurable manufacturing systems. Mecanum wheels use slip developed between rollers and surface, surface and ground, to achieve omni-directionality. An advantage of omni-directional robotic platforms is that they are capable of performing tasks in congested environments such as those found in factory workshops, narrow aisles, warehouses, etc. Controller Area Network (CAN) is implemented as a distributed controller to control motion and navigation tasks of the developed robot. The design of the distributed controller is described and its performance analyzed. This increases the reliability and functionality of the mobile robot.

### **OPSOMMING**

Die artikel beskryf wiel - en beheerontwerpe van 'n veelrigting mobiele robot. Die robot is 'n selfstandigbeheerde voertuig vir gebruik by vervaardigingstelsels met veranderbare konfigurasie. Die ontwerp van die robot en bypassende beheerstelsel word beskryf en ontleed teen die agtergrond van bewegings - en navigeertake. Die betroubaarheid en funksionering van die sisteem word beoordeel.

## 1. INTRODUCTION

Automated guided vehicles (AGV) that are controlled in real-time have become an integral part of modern reconfigurable manufacturing systems. They are used extensively in flexible manufacturing systems to move parts and to orient them as required [1]. Many designs of omni-directional or near omni-directional vehicles have been proposed. These can generally be dealt with by two approaches: conventional wheel and special wheel designs.

Conventional wheels are mechanically simple, have high load capacity and high tolerance to work surface irregularities. However, due to their non-holonomic nature, they are not truly omni-directional. Designs have been proposed to achieve near omni-directional mobility using conventional wheels. The most common designs are those using steered wheels [2]. Vehicles based on this design have at least two active wheels, each of which has both driving and steering actuators. However, this type of system is not truly omni-directional because it needs to stop and re-orient its wheels to the desired direction whenever it needs to travel in a trajectory with non-continuous curvatures [3]. Omni-directionality is the ability of an AGV to move instantaneously in any direction from any configuration, and not necessarily changing its orientation.

Most special wheel designs are based on a concept that achieves traction in one direction and allow passive motion in another, thus allowing greater flexibility in congested environments [4]. One of the more common omni-directional wheel designs is that of the Mecanum wheel, invented in 1973 by Bengt Ikon, an engineer with the Swedish company Mecanum AB [5]. Many of the other commonly current designs are based on Ikon's original concept. The mobile robot developed in this paper uses Mecanum wheels to achieve omni-directionality. Mecanum wheels give the AGV the ability to change direction of its motion instantaneously without changing its orientation.

Currently, there are three approaches to AGV navigation in the literature: model-based, sensor-based and hybrid approaches. Model-based approaches use accurate description of the environment to generate obstacle-free paths. Techniques for model-based path generation include: road mapping, cell decomposition and potential fields [6]. All of these methods can generate a path from an initial-point to an end-point using a model of the environment. However, it is usually difficult to obtain an accurate model of the environment. Sensor-based approaches execute control commands based on sensor data.

Central controllers complicate the development of the control systems for mobile robots because:

- many software components that interact with each other have to be developed. They sometimes do not behave in the manner expected;
- wiring of the robot's different systems to one central controller is complicated;

- if the central controller fails, all the other sub-systems fail. For safety and production costs reasons, this is not desirable; and
- troubleshooting of errors is difficult.

Distributed controllers do not have the disadvantages of central controllers. With the cost of microprocessors being reduced, it is becoming cheaper to develop distributed control systems for manufacturing applications. Distributed implementation has many potential advantages over centralized ones such as improved performance, optimized resource utilization, reduced cabling, as well as enhanced fault tolerance and modularity [7]. In this paper, a Controller Area Network (CAN) is implemented as a distributed controller of a Mecanum-wheeled AGV. CAN uses a two-wired bus to connect the distributed controller modules of individual wheels and sensor systems.

Disadvantages of distributed control systems are the delays introduced in the system as a result of distributed control architecture for control tasks that are time critical. This includes problems that are concerned with timing, such as lag effect of zero-order hold (ZOH) and problems with respect to motion control which are:

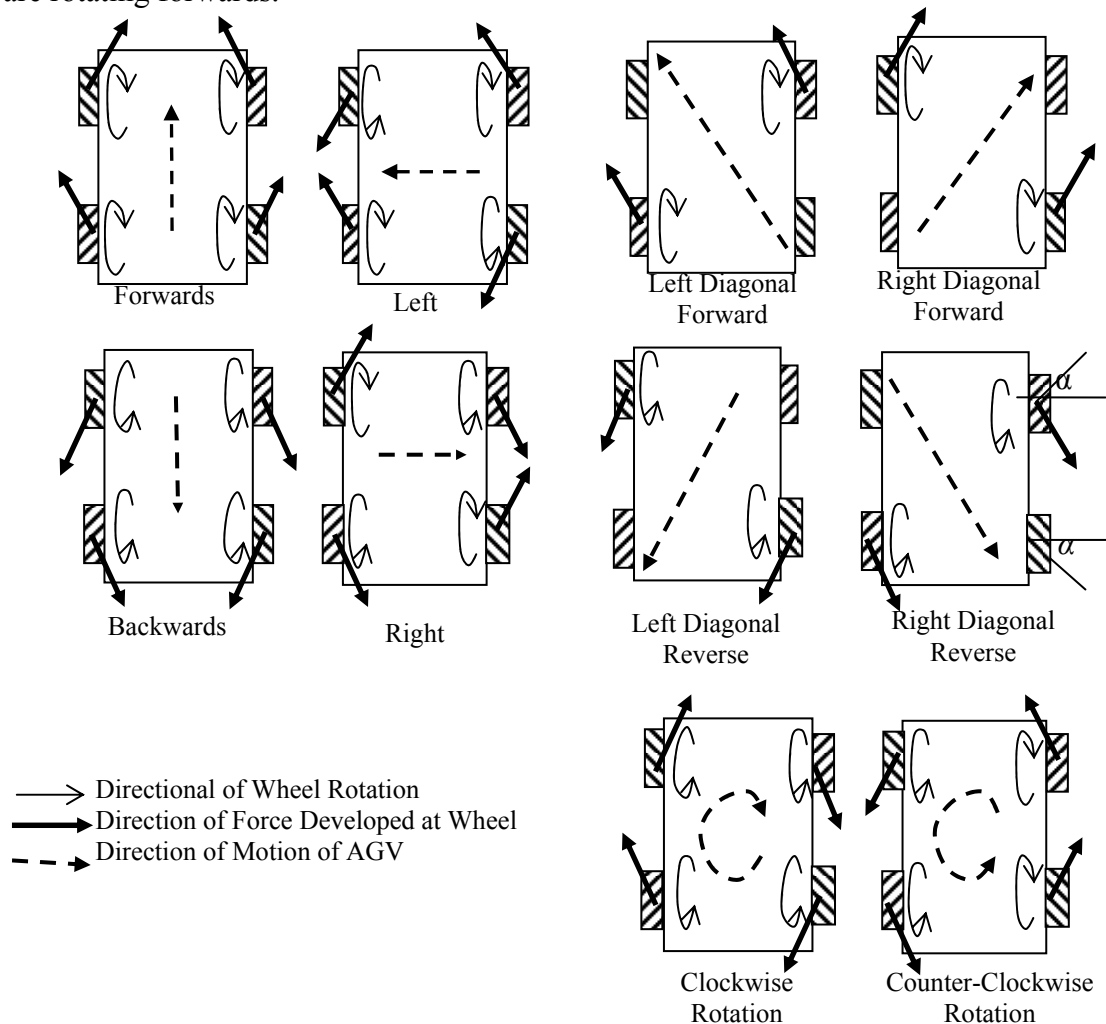
- transmission control delay  $\tau_{icd}(k)$ , which is the total time taken from when the message or signal is transmitted on the bus to the time the actuator responds. It can be broken-down into its components depending on the system design. Control delay reduces a controller's phase margin by introducing a phase displacement and results in deteriorated control stability;
- jitters, which are unintentional time-variations in the duration of a time interval, e.g., a sampling period. Larger jitters can lead to vacant sampling. Depending on the persistence in the variation of  $\tau_{icd}(k)$  and the dynamics of the controlled system, the performance or stability of the controller can suffer; and,
- transient errors, which are the loss or corruption of data or signals due to transient or permanent software and hardware failures. Such abnormal conditions can be thought of as prolonged sampling interval or actuation delay, causing varying control delays or open loop control.

Constant delays as well as the lag effects can be easily compensated for in discrete-time control design. Compensating for time-variations, which may be stochastic, is much more difficult. These problems necessitate the importance of a temporally predictable (i.e. real-time) computer system that guarantees the time-invariant assumptions of control design. Alternatively, the problems of time variations can also be partially tackled in control design, e.g., by using robust control so that deviations from nominal timing can be tolerated [7].

The requirement of high reliability and high functionality of the mobile robot necessitates the use of a distributed controller to be used because of the reasons given above. Controller area network (CAN), MCP2515 CAN controller from Microchip, is implemented as stand-alone distributed controller modules (or nodes) on the control network [8]. PIC18F442 microcontrollers are used as local controllers on the network's controller modules.

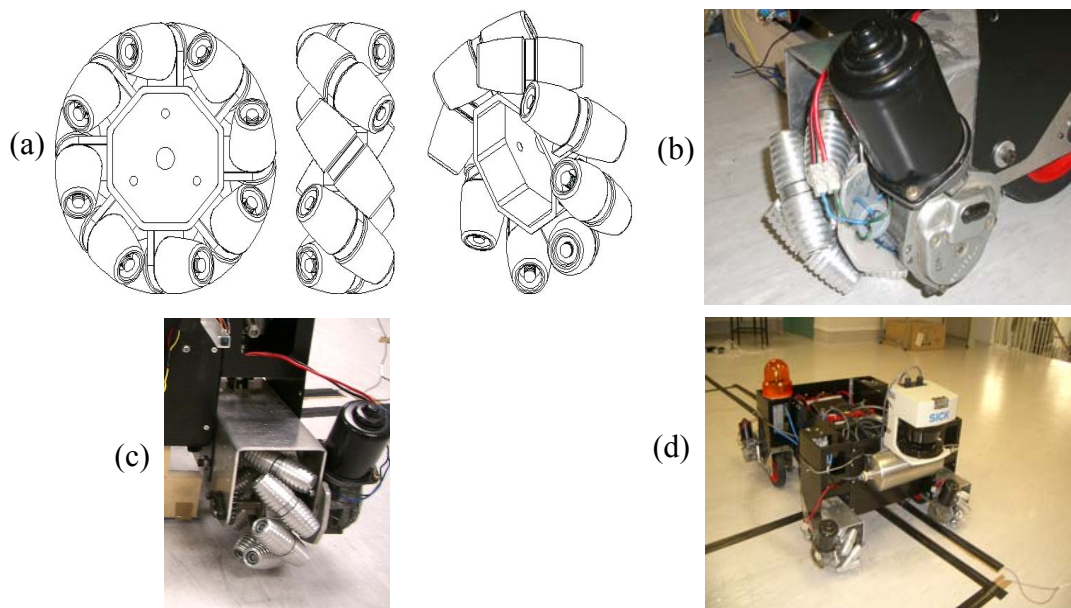
## 2. MECHANICAL WHEEL DESIGN

The developed AGV design uses both normal and Mecanum wheels. Mecanum wheels consist of a number of rollers around the circumference of the wheel hub. The rollers are orientated at some angle,  $\alpha$ , from the axis of rotation of the wheel. Rollers can rotate about their own axis. For the developed AGV, an angle  $\alpha$  of  $45^\circ$  is used. All the four pairs of Mecanum wheels are independently controlled. The angle,  $\alpha$ , controls the direction of the force that is developed between the wheel and ground surface. Forces developed on the wheels are perpendicular to the angle  $\alpha$ . When the wheels are mounted on the AGV as shown in figure 1, the forces causing motion that are developed on the AGV can cause the AGV's motion to have the following directions: forward, backward, left, right, left diagonal reverse, right diagonal reverse, left diagonal forward, right diagonal forward, clockwise and counter clockwise. The direction of the developed motion depends on the wheel orientation with respect to the AGV body and the rotation of the wheel. The orientation of the wheels on the AGV is such that the vehicle is moving forwards when all the wheels are rotating forwards.



**Figure 1: Directional control of a Mecanum wheeled AGV.  
Right-hand rule used to determine the direction of rotational speed of wheels  
Roller angle  $\alpha$  is also shown**

To control the direction of the translational and rotational speeds of the AGV, different combinations of Mecanum wheels' rotations are used. The ability of the AGV is fully exploited in this paper by implementing a distributed controller that has nodes dedicated to controlling only one Mecanum wheel. For example, to move the robot to the left, only two right wheels can be rotated against each other outwardly, or only two left wheels can be rotated against each other inwardly, or both the left and right pairs of wheels can be used and rotated in the manner as explained (refer figure 1). When the hardware on a certain wheel node fails, for example wheel node for the front left wheel, only the right wheel can be used to steer the vehicle to the left. Any desired orientation of the mobile robot can be achieved using the same technique. The slip developed between the rollers and the ground, and between the rollers and the wheel, gives the mobile robot the ability to change direction of its motion without changing its orientation. When the vehicle is moving on low friction surfaces such as surfaces covered with low friction plastics, carpeted surfaces, or surfaces that have some moisture or lubrication, considerable slip is encountered, and the AGV cannot be controlled effectively. It sometimes becomes dangerous for the AGV to operate in such conditions. It is necessary to detect slip, and to apply relevant control strategies to get the vehicle safely away from such surfaces.



**Figure 2: (a) Mecanum wheel design with centrally mounted rollers which can rotate about their own axis to reduce friction (b) and (c) Set of Mecanum and normal wheels controlled by pneumatic cylinder (d) The developed mobile robot**

Mecanum wheel design that uses rollers that are held in the middle is implemented. The advantage of this design is that the wheels produce less friction while driving on general surfaces (refer figure 2 (a)). The Mecanum wheel design for this project is based on this improved wheel design. With free rotating rollers any combination of forward, sideways and reverse movement is possible with less friction [8]. Mecanum

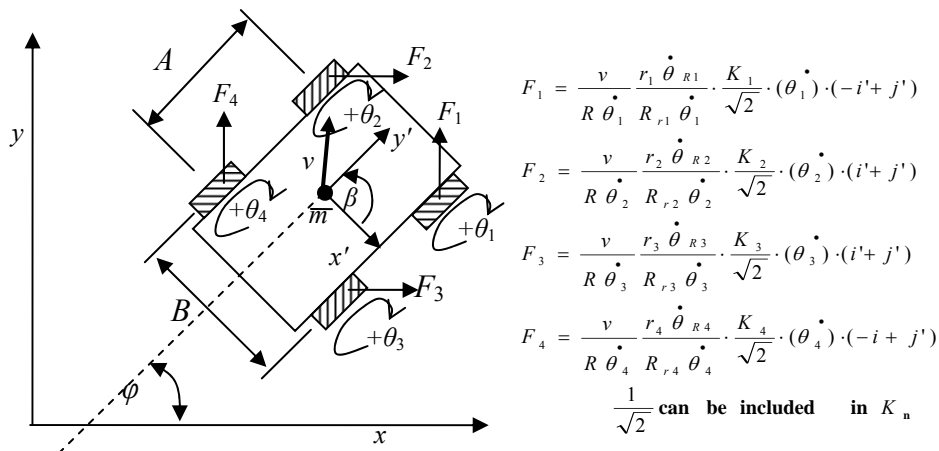
wheels are used to steer the robot in the desired direction, mostly when the robot is moving on smooth surfaces, such as those found indoors. AGV motion on rough terrains, such as those found in outdoor environments, is facilitated by a second set of normal wheels. Each Mecanum wheel has an accompanying normal wheel, both attached to the same wheel frame (refer figures 2 (b) and (c)). Pneumatic cylinders are used to change from a Mecanum wheels set to a normal set of wheels (and vice versa) in outdoor environments and indoor environments respectively. Figure 2 (d) shows the developed AGV.

### 3. MATHEMATICAL MODEL OF MECANUM WHEELED AGV

Let  $xyz$  be the reference coordinate axes, and  $x'y'z'$  be the body-attached coordinate axes as shown in figure 3. The force causing motion of the mobile robot, which is developed on each individual Mecanum wheel,  $F_n$ , can be determined to be:

$$F_n = \frac{v}{R} \frac{r_n \dot{\theta}_{Rn}}{\dot{\theta}_n} \cdot K_n \cdot (\pm \dot{\theta}_n) \cdot (\pm \cos \alpha i' \pm \sin \alpha j') \tag{1}$$

where  $n$  is the wheel number,  $v$  is the value of resultant translational velocity of AGV,  $R$  is the



**Figure 3: Kinematic model of Mecanum wheeled AGV. The forces that cause motion which are generated by the rotation of the wheel and the rollers,  $F_n$ , are shown**

radius of rotation of rollers on wheel  $n$ ,  $\dot{\theta}_{Rn}$  is the average rotational speed of rollers on wheel  $n$ ,  $r_n$  is average radius of rollers on wheel  $n$ ,  $K_n$  is the wheel constant dependant on; number of rollers per wheel, friction coefficient between rollers and ground surface and the robot's mass,  $i'$  and  $j'$  are unit vectors of body-attached coordinates axis  $x'y'z'$ .  $\dot{\theta}_n$ , the rotational speed of wheel  $n$ , is taken as positive in the direction shown in the Figure 3 i.e. when all the wheels are causing a forward movement of the mobile robot. If  $\alpha = 45^\circ$ , then  $K_n$  can include the effect of  $\alpha$  because

$\sin 45^0 = \cos 45^0 \cdot \frac{v}{R\dot{\theta}_{Rn}}$  is the slip between the rollers and the hub of wheel  $n$ ,

while  $\frac{r_n\dot{\theta}_{Rn}}{R_n\dot{\theta}_n} = s_{rgn}$  is the slip between the rollers and the ground on wheel  $n$ .

The direction of the resultant force from all the wheels is the same as the direction of the resultant velocity of the mobile robot. The driving velocity developed on the wheel number  $n$ ,  $v_n$ , can be found as:

$$v_n = s_{rhn} \cdot s_{rgn} \cdot K_n \cdot K_{wn} \left\{ (\pm \dot{\theta}_n)(\pm \cos \alpha i' \pm \sin \alpha j') \right\} \quad (2)$$

where  $K_{wn}$  is a constant for converting wheel force into wheel velocity, and is determined as the gain from the encoder reading to the linear wheel displacement of wheel  $n$ . The encoder gain  $K_{wn}$  is defined as:

$$K_{wn} = \frac{r_n}{h_n N_n} \quad (3)$$

where  $r_n$  is the radius of wheel  $n$ ,  $h_n$  is the encoder resolution (pulses/rad) of wheel  $n$  and  $N_n$  is the gear ratio from motor shaft to wheel  $n$ .

The resultant velocity of the mobile robot,  $v$ , can be determined to be:

$$v = \left\{ (s_{rh1}s_{rg1}K_1K_{w1}\dot{\theta}_1)(-i'+j') + (s_{rh2}s_{rg2}K_2K_{w2}\dot{\theta}_2)(i'+j') \right\} + \left\{ (s_{rh3}s_{rg3}K_3K_{w3}\dot{\theta}_3)(-i'+j') + (s_{rh4}s_{rg4}K_4K_{w4}\dot{\theta}_4)(i'+j') \right\} \quad (4)$$

Simplifying,

$$v = (-s_{rh1}s_{rg1}K_1K_{w1}\dot{\theta}_1 + s_{rh2}s_{rg2}K_2K_{w2}\dot{\theta}_2 + s_{rh3}s_{rg3}K_3K_{w3}\dot{\theta}_3 - s_{rh4}s_{rg4}K_4K_{w4}\dot{\theta}_4)i' + (s_{rh1}s_{rg1}K_1K_{w1}\dot{\theta}_1 + s_{rh2}s_{rg2}K_2K_{w2}\dot{\theta}_2 + s_{rh3}s_{rg3}K_3K_{w3}\dot{\theta}_3 + s_{rh4}s_{rg4}K_4K_{w4}\dot{\theta}_4)j' \quad (5)$$

Simplifying further,

$$\begin{aligned}
 v &= (-s_{rh1}s_{rg1}K_1K_{w1}\lambda_1 + s_{rh2}s_{rg2}K_2K_{w2}\lambda_2 \\
 &\quad + s_{rh3}s_{rg3}K_3K_{w3}\lambda_3 - s_{rh4}s_{rg4}K_4K_{w4}\lambda_4)i' \\
 &\quad + (s_{rh1}s_{rg1}K_1K_{w1}\lambda_1 + s_{rh2}s_{rg2}K_2K_{w2}\lambda_2 \\
 &\quad + s_{rh3}s_{rg3}K_3K_{w3}\lambda_3 + s_{rh4}s_{rg4}K_4K_{w4}\lambda_4)j' \\
 &= ai' + bj'
 \end{aligned} \tag{6}$$

In equation (4),  $\lambda_n$  is the reading of rate of encoder pulses on the wheel  $n$ . The direction of the resultant velocity of the AGV can be determined to be:

$$\begin{aligned}
 \beta &= \arctan(v) \\
 &= \arctan\left\{\frac{b}{a}\right\}
 \end{aligned} \tag{7}$$

Therefore, the kinematics equations of the developed mobile robot with respect to the reference coordinates system can be written as:

$$\begin{aligned}
 \dot{x}(t) &= v(t) \cdot \cos(\varphi(t) + \beta(t)) \\
 \dot{y}(t) &= v(t) \cdot \sin(\varphi(t) + \beta(t)) \\
 \dot{\varphi}(t) &= \frac{v(t)}{A} \cdot \cos(\beta(t)) \quad \text{for } \pm\theta_1, \mp\theta_2, \theta_3 = \theta_4 = 0 \text{ or} \\
 &\quad \text{or } \pm\theta_3, \mp\theta_4, \theta_1 = \theta_2 = 0 \text{ or} \\
 &\quad \text{or } \pm\theta_1, \mp\theta_2, \pm\theta_3, \mp\theta_4 = 0 \\
 \text{or } \dot{\varphi}(t) &= \frac{2 \cdot v(t)}{A} \cdot \cos(\beta(t)) \quad \text{for } \pm\theta_1, \mp\theta_4, \theta_2 = \theta_3 = 0 \text{ or} \\
 &\quad \text{or } \pm\theta_2, \mp\theta_3, \theta_1 = \theta_4 = 0 \text{ or} \\
 &\quad \text{or } \pm\theta_1, \mp\theta_4, \pm\theta_2, \mp\theta_3 = 0
 \end{aligned} \tag{8}$$

where  $v(t)$  is the resultant velocity of the AGV and  $\varphi(t)$  indicates its posture.  $s_h$ ,  $v_w$  and  $\beta$  are measured using the wheel encoders and the inertial sensor. The model is only true for constant rotational velocities of the wheels. The assumption is made that no brakes are applied at any wheel when the vehicle is moving or any wheel of the vehicle is being powered.

The above equations are used as guides to achieve control of the developed Mecanum-wheeled AGV so that the functionality of the AGV can be maintained when some distributed sensor/wheel nodes fail. The equations are useful in determining the following:

- Which combination of wheel(s) should be used to achieve specific motion(s) and orientation(s) of the AGV?
- What should be the direction(s) of the rotational speed(s) of the wheel(s) used



- to achieve specific motion(s) and orientation(s) of the AGV?
- Which other alternative wheel(s) can be used to achieve the same motion(s) and orientation(s) of the AGV in case some wheel node(s) have failed?
- Detection of wheel slip using the inertial sensor and the wheel encoders.

#### **4. CONTROLLER AREA NETWORK FOR DISTRIBUTED CONTROL ARCHITECTURE**

Control architecture of the developed Mecanum-wheeled AGV is divided into time-critical and event-based control strategies. Control functions such as navigation and motion planning are time-critical, while control functions such as path planning and object avoidance are event-based.

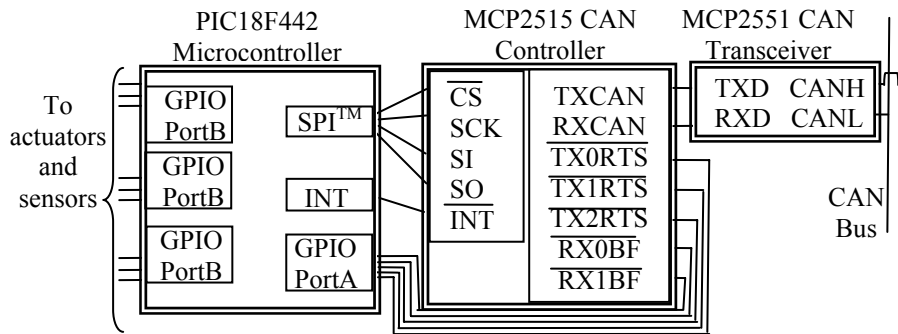
In order to achieve reliable and adequate control of the mobile robot, CAN is implemented. CAN comprises of only layers 2 (Data Link Layer) and 1 (Physical Layer) of the International Standardization Organisation/Open Systems Interconnect (ISO/OSI) hierarchical layered structure [10], [11]. CAN allows time-critical sensor information and control signals to be transmitted with determinism [11]. Another advantage of using CAN is that its two-wire bus offers noise immunity because it uses differential voltage lines on the bus. CAN version 2.0 B specification, which can run at 125 KBits/s of data transfer is implemented. Figure 4 is a schematic of the hardware that is implemented on a CAN node.

The CAN message data frame which is used is the standard data frame, which consists of eleven message identifier bits and eight data bytes. All CAN protocol specifications are implemented by the MCP2515 CAN controller. These include CAN message frames, CAN bus and message error routines, CAN bus and bit timing logic and CAN message transmission and reception on bus. It interfaces with the local microprocessor via Serial Peripheral Interface (SPI<sup>TM</sup>). In order to initiate communication on the SPI Interface, the PIC holds the **CS** pin (bold letter indicates an inverted signal) low until communication has been completed. Instruction and data are then driven out on the SO line (which is connected to the SI pin of MCP2515) on the rising edge of oscillator (SCK) pin. Data from the MCP2515 is then clocked in on the SI line on the falling edge of the SCK pin.

The last message to be sent on the CAN bus indicates the current AGV's task. CAN message identifiers indicate the priority of the message. Message priority is assigned according to descending values of the message identifiers. Figure 5 indicates specific messages that each node of the implemented CAN controller can initiate. General messages that can be initiated at each CAN node include: messages about the status of the error registers of each node, map

building messages, status of local sensors and local actuators. The mobile robot is controlled by synchronizing the software control programs on different CAN nodes that control and monitor the AGV's sensor and actuators respectively. Each CAN node has the ability to synchronize the distributed software control programs on different CAN nodes. The developed mobile robot can be controlled by any CAN node. This is desirable in case one node fails, the rest of the nodes can control the

robot adequately to complete the desired tasks.



**Figure 4: CAN node hardware implemented on nodes.**

Functions of each CAN node are to:

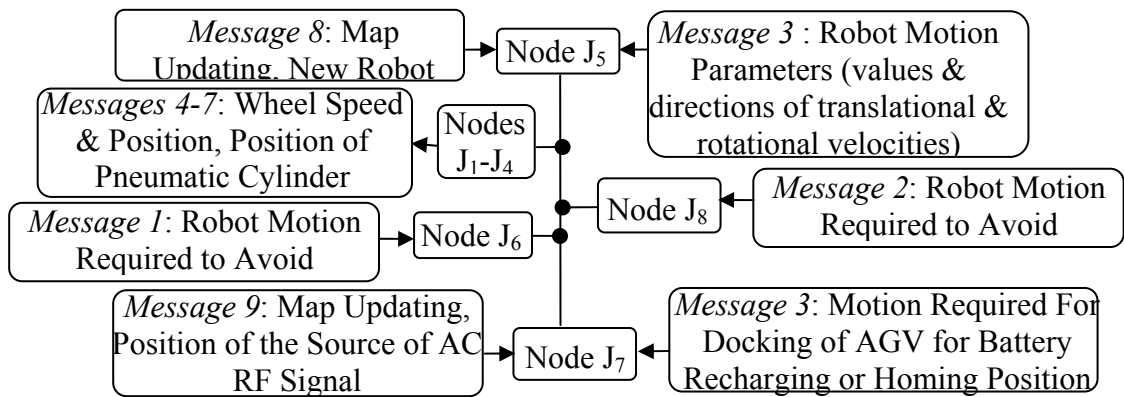
- transmit and receive messages to and from the CAN bus respectively; and/or
- control the local actuators using the feedback information from local sensors and/or remote sensors; and/or
- monitor the status of local sensors and/or send the local sensors' status on the CAN bus

The local microprocessor controls the local actuators and monitors the local sensors. It can control remote actuators and monitor remote sensors by sending and receiving correctly addressed messages on the CAN bus.

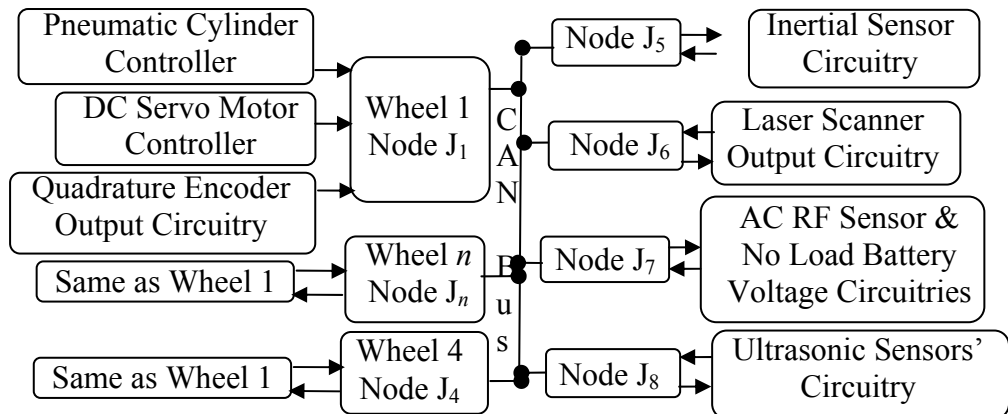
Four CAN nodes are used as wheel nodes. Each wheel node controls a set of wheels that is comprised of a normal wheel and a Mecanum wheel. A set of wheels are driven by a DC motor, which is in turn driven by a servo driver and a local microprocessor. Quadrature encoders are used as feedback sensors for the wheels. Nodes 1-4 are used to control each of the four sets of wheels. Nodes 5-8 are used to monitor outputs of the sensory circuits needed to achieve collision free and reliable working of the mobile robot. The first sensor node monitors the output from an inertial sensor. It is used to measure the translational and rotational motion of the AGV. It is also be used to measure the slip developed on the mobile

robot's wheels by comparing its output with the output of the wheels' encoders. The second sensor node is used to monitor the output of laser ranging systems, LMS291-S05 from SICK. This information is used for obstacle detection and avoidance. A third node is used for monitoring the AC RF sensor and the no load battery voltage circuitries. It is implemented to guide the robot to follow an RF wired signal buried on the ground i.e. for guided path following. This is used to guide the robot to the home position or the battery charging station. No load battery voltage sensor is used as a measure of the current power capacity of the battery. The fourth sensory node is used to monitor ultrasonic sensor which is used for obstacle detection and avoidance

in case the laser scanner node fails.



**Figure 5: Specific message structure of CAN nodes for Mecanum wheeled AGV**



**Figure 6: Sensory, actuation and distributed control architecture implementing MCP2515 CAN controller**

The nodes that are used on the distributed control architecture have the same general competency levels of controlling the AGV's motion and making the AGV to navigate without collision. This is different from the architecture proposed by Brooks, where each node or architectural layer has a different competency [12]. The advantage of this approach is that the system can be updated and controlled from any node. In the case that some nodes fail during the robot mission, other nodes have the competency to control the mobile robot in a safe manner. The nodes differ with specific competency levels, depending on whether nodes monitor a sensor or control an actuator. The functional structure of the different nodes can be seen in figure 6.

## 5. RESULTS AND DISCUSSION

The laser scanner is the default object detection sensor. In case a node attached to the laser scanner fails, the ultrasonic sensor is used as the backup object detection sensor. If both of these nodes fail, the robot is stopped until the CAN bus functionality of either one node has been fully recovered. If the node monitoring the inertial sensor fails, the node attached to the AC RF sensor circuitry can be used to guide the robot from one point to another. However, the AGV's motion is only restricted to the area where the AC RF signal can be detected until the node monitoring the inertial sensor has been fully recovered.

In case one of the nodes controlling the wheels fails, equation (8) is used to determine the appropriate combination of wheels which can produce the desired motion of the mobile robot. At least two wheels are required to achieve useful motion of the AGV. Tests were done with two wheel CAN nodes disabled, and either the laser scanner node or the ultrasonic sensor node disabled. Results indicate that the Mecanum wheeled AGV using distributed CAN nodes as implemented in this project, can adequately carry out its tasks when the nodes fail in that way. This represents the worst-case scenario. However, the motion of the AGV using only two wheel nodes is not as efficient as when using all four-wheel nodes.

There are three possible kinds of two wheels combinations that can simulate failure behaviour of two wheel nodes of CAN controller for the developed mobile robot as explained above. These are:

- i) Two wheel nodes on the same side go into the bus-off nodes (e.g.  $\theta_2$  and  $\theta_4$ ), and while the other two nodes on the other side are still error active or passive (e.g.  $\theta_1$  and  $\theta_3$ ). Equation 5 then becomes:

$$v = (-s_{rh1}s_{rg1}K_1K_{w1}\dot{\theta}_1 + s_{rh3}s_{rg3}K_3K_{w3}\dot{\theta}_3)i' + (s_{rh1}s_{rg1}K_1K_{w1}\dot{\theta}_1 + s_{rh3}s_{rg3}K_3K_{w3}\dot{\theta}_3)j' \quad (9)$$

The direction of rotational and translational velocities of the AGV can be seen in Figure 8 (a). The AGV loses all the diagonal translational motions.

Two diagonally opposite wheel nodes go into the bus-off nodes (e.g.  $\theta_2$  and  $\theta_3$ ), and while the other two diagonally opposite nodes are still error active or passive (e.g.  $\theta_1$  and  $\theta_4$ ). Equation 5 then becomes,

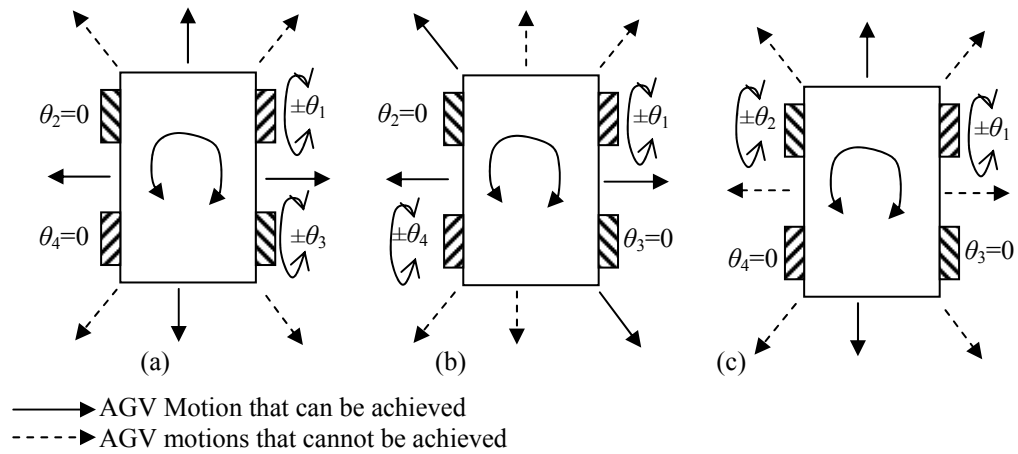
$$v = (-s_{rh1}s_{rg1}K_1K_{w1}\dot{\theta}_1 - s_{rh4}s_{rg4}K_4K_{w4}\dot{\theta}_4)i' + (s_{rh1}s_{rg1}K_1K_{w1}\dot{\theta}_1 + s_{rh4}s_{rg4}K_4K_{w4}\dot{\theta}_4)j' \quad (10)$$

The direction of rotational and translational velocities of the AGV can be seen in Figure 8 (b). The AGV loses all the lateral translational motions. The

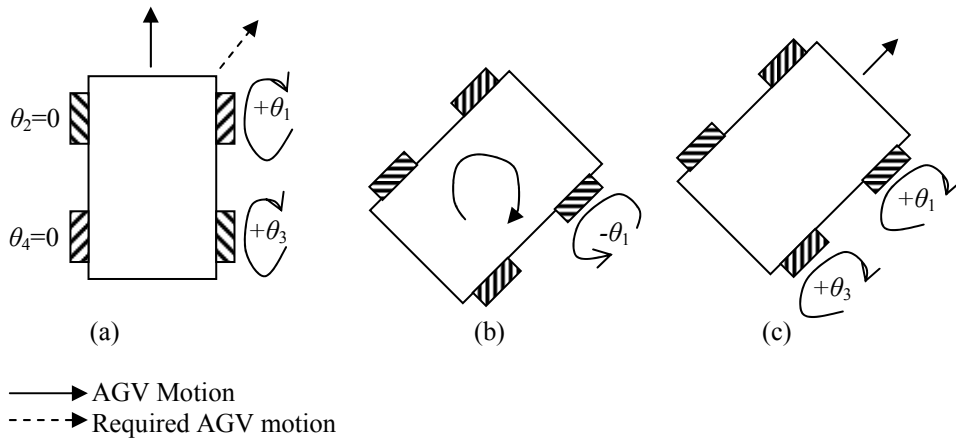
resultant AGV motions are not effective because they produce a lot of friction. This state is highly undesirable and the robot must not function in this state for prolonged periods.

- ii) Two back wheel nodes go into the bus-off nodes (e.g.  $\theta_3$  and  $\theta_4$ ), and while the two front nodes are still error active or passive (e.g.  $\theta_1$  and  $\theta_2$ ). Equation 5 then becomes

$$v = (-s_{rh1}s_{rg1}K_1K_{w1}\dot{\theta}_1 + s_{rh2}s_{rg2}K_2K_{w2}\dot{\theta}_2)i' + (s_{rh1}s_{rg1}K_1K_{w1}\dot{\theta}_1 + s_{rh2}s_{rg2}K_2K_{w2}\dot{\theta}_2)j' \tag{11}$$



**Figure 7: Directions of translational and rotational motions that can be achieved by any combination of two wheel nodes and other sensor nodes including one of the collision avoidance sensor node**



**Figure 8: Typical strategy implemented to achieve required AGV movement when some nodes fail. In this case a lateral motion to the front left is required, but wheel nodes 2 and 4 have failed. The AGV rotates clockwise using only wheel node 1 until the required angular turn has been achieved, then it moves forwards in that direction using wheel nodes 1 and 3**

The direction of rotational and translational velocities of the AGV can be seen in figure 8 (c). The AGV loses all the some lateral translational motions

By using the above equations and equations 7 and 8, the translational and rotational speed of the robot can be controlled to achieve the desired tasks, maintain partial omni-directionality and achieve collision free motion as shown in figure 7.

Typical strategy that is implemented in order to achieve the required AGV's movement when some node(s) fail for case (i) can be seen in figure 8. The figure also illustrate the advantage of distributed control system and the justification of the development of the mathematical functions of the developed AGV. In this case a lateral motion to the front left is required, but wheel nodes 2 and 4 have failed. The previous AGV motion was to the front. The AGV rotates clockwise using only wheel node 1 until the required angle of turn has been achieved. Then, it moves forwards in that direction using wheel nodes 1 and 3.

## 6. CONCLUSION

A Mecanum-wheel AGV has been developed. The controller of the developed AGV implements distributed CAN controller. Distributed CAN controller improves the functionality and reliability of the developed AGV by using a distributed controller structure. The distributed controller architecture uses independent control nodes for each wheel of the vehicle, and independent monitoring nodes for each sensor of the developed robot. All the distributed nodes communicates on the CAN bus in order to achieve reliable working of the vehicle.

Real-time critical tasks such as controlling of actuators and monitoring of sensors are controlled by various dedicated CAN nodes. This minimizes the effect of control delay. Only the event-based tasks are synchronized through CAN bus. CAN bus timing has no noticeable effect on the real-time control of the developed robot.

## 7. REFERENCES

- [1] **Kalpakjian, S. and Schmid, S. R.**, *Manufacturing Engineering and Technology*, Prentice-Hall, USA, 2000.
- [2] **Borenstein, J., Everett, H. R., Feng, L.**, *Navigating Mobile Robot*, AK Peters, Wellesley, Massachusetts, 1996.
- [3] **Dubowsky, S., Genot, F., Godding, S., Kozono, H., Skwersky, A., Yu, H. and Yu, L.**, PAAM – *A robotic Aid to the Elderly for Mobility Assistance and Monitoring: A Helping-Hand for the Elderly*, IEEE International Conference on Robotics and Automation, 2000.
- [4] **Stephen L. Dickeson and Brett D. Lapin**, *Control of an Omni-Directional Robotic Vehicle With Mecanum Wheels*, IEEE Proceedings of National Tele-Systems Conference, 1991, pp 323 –328.
- [5] **West, M. and Asada, H.**, 1997, *Design of Ball Wheel Mechanisms for Omni-directional Vehicles With Full Mobility and Invariant Kinematics*. Journal of Mechanical Design, Vol. 119, 1997, pp153-161.
- [6] **Tzafetas, E. S., Mathieu, J. D. and Tzafetas, S.**, *Overview of Experimental*

*Evaluation of Potential Field-Based Robot Navigation*, Advanced Robotics Beyond 2000: The 29<sup>th</sup> International Symposium on Robotics, N. E. C., Birmingham, UK, 27<sup>th</sup> April 1998.

- [7] **Chen, D.-J.** (2001), *Architecture for Systematic Development of Mechatronics Software Systems*, Licentiate Thesis, ISSN 1400 -1179, Mechatronics Lab, Department of Machine Design Royal Institute of Technology, KTH, Stockholm, Sweden
- [8] <http://www.microchip.com/>, retrieved 08/09/2005.
- [9] **Badve, A. A.**, *All Terrain Omni-directional Autonomous Mobile Robot*, Master's Thesis, Massey University, Auckland, New Zealand, 2003.
- [10] International Standard ISO 11898 (1993), "Road Vehicles – Interchange of Digital Information – Controller Area Network (CAN) for High Speed Communication.", ISO Reference Number ISO 11898: 1993(E), First Edition 1993 – 11 -15.
- [11] International Standard ISO 11519 -2 (1993), "Road vehicles – Low Speed Serial Data Communication – Part 2: Low-speed Controller Area Network (CAN).", ISO Reference Number ISO 11519-2:1994 (E), First Edition, pp. 11 -15.
- [12] **Brooks, R. A.**, *A Roboust Layered Control System for a Mobile Robot*, In IEEE Journal of Robotics, Vol. 2 No. 1, March 1986, pp. 14-23.

