openUP (March 2007)

On distributed mechatronics controller for omni-directional autonomous guided vehicles

Nkgatho Sylvester Tlale

Nkgatho Sylvester Tlale, *Department of Industrial and Systems Engineering,* ***University of Pretoria,*** *Pretoria, South Africa*

[**Figures and Tables at the bottom of the document**]

Purpose – In this paper, two omni-directional mobile vehicles are designed and controlled implementing distributed mechatronics controllers. Omni-directionality is the ability of mobile vehicle to move instantaneously in any direction. It is achieved by implementing Mecanum wheels in one vehicle and conventional wheels in another vehicle. The control requirements for omni-directionality using the two above-mentioned methods are that each wheel must be independently driven, and that all the four wheels must be synchronized in order to achieve the desired motion of each vehicle.

Design/methodology/approach – Distributed mechatronics controllers implementing Controller Area Network (CAN) modules are used to satisfy the control requirements of the vehicles. In distributed control architectures, failures in other parts of the control system can be compensated by other parts of the system. Three-layered control architecture is implemented for; time-critical tasks, event-based tasks, and task planning. Global variables and broadcast communication is used on CAN bus. Messages are accepted in individual distributed controller modules by subscription.

Findings – Increase in the number of distributed modules increases the number of CAN bus messages required to achieve smooth working of the vehicles. This requires development of higher layer to manage the messages on the CAN bus.

Research limitations/implications – The limitation of the research is that analysis of the distributed controllers that were developed is complex, and that there are no universally accepted tool for conducting the analysis. The other limitation is that teh mathematical models of the mobile robot that have been developed need to be verified.

Practical implications – In the design of omni-directional vehicles, reliability of the vehicle can be improved by modular design of mechanical system and electronic system of the wheel modules and the sensor modules.

Originality/value – The paper tries to show the advantages of distributed controller for omni-directional vehicles. To the author's knowledge, that is a new concept.

1 Introduction

1.1 Omni-directional mobile vehicle

Automated guided vehicles (AGVs) that are controlled in real-time have become an integral part of modern reconfigurable manufacturing systems. They are used extensively in flexible manufacturing systems (FMS) to move parts and to orient them as required (**Kalpakjian and Schmid, 2000**). Many designs of omni-directional or near omni-directional vehicles have been proposed. These can generally be broken into 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 (**Borenstein *et al.*, 1996**). 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 the vehicle needs to stop and re-orient its wheels to the desired direction whenever it needs to travel in a trajectory with non-continuous curvatures (**Dubowsky *et al.*, 2000**). One of the omni-directional mobile vehicles presented in this paper uses four independently steered and driven conventional wheels.

Most special wheel designs are based on a concept that achieves traction in one direction and allows passive motion in another, thus allowing greater flexibility in congested environments (**West and Asada, 1997**). One of the more common omni-directional wheel designs is that of the Mecanum wheel, invented in 1973 by Bengt Ilon, an engineer with the Swedish company Mecanum AB. Many of the other commonly currently designs are based on Ilon's original concept. Another the mobile vehicle developed in this paper uses Mecanum wheels to achieve omni-directionality. Mecanum wheels give the mobile vehicle the ability to change its direction of motion without changing its orientation.

1.2 Central and distributed control systems

Central controllers complicate the development of the control systems for mobile vehicles 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 robots' 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.
- Troubleshooting of errors is difficult.

Distributed controllers do not have the disadvantages of central controllers. With the cost of microprocessor being reduced, it is becoming cheaper to develop distributed control systems for manufacturing applications. Distributed implementation has many potential advantages over centralized one such as improved performance, optimized resource utilization, reduced cabling, as well as enhanced fault tolerance and modularity (**Barrett and Lafortune, 2000**).

Controller Area Network (CAN) is a distributed communication protocol/system that uses a two-wired bus to connect the distributed controller modules. The controller modules can be for individual wheels and sensor systems. The requirement of high reliability and high functionality for mobile vehicles necessitates the use of distributed controllers because of the reasons given above. CAN, MCP2515 CAN controller from Microchip, is implemented as stand-alone distributed controller modules (or nodes) on the control network (**Microchip Com., 2005**). PIC18F442 microcontrollers are used as local controllers on the modules of the CAN network.

2 Mechanical design, controller and sensor architecture

Omni-directionality in the first AGV that implements Mecanum wheels is achieved by implementing a number of angled rollers around the circumference of the wheel. The rollers are orientated at some angle, α, from the axis of rotation of the wheel and they can rotate about their own axis. A constant angle α of 45° is used in the design of the Mecanum wheel of the AGV that is presented. A constant angle ensures that the contact point between the wheels and the ground is not changed. Each Mecanum wheel is independently controlled. With free rotating rollers any combinations of forward, sideways and reverse movement are possible with less friction (**Badve, 2003**). Omni-directionality in the second AGV is achieved by implementing four conventional wheels that are independently steered and driven. The developed AGVs have simultaneous and independent control of their rotational and transitional capabilities.

From the control point of view, each design has some challenges. For Mecanum wheels, a lot of friction between the rollers, wheel hubs and the ground is created. Slippage can result, which can results in unstable control. For conventional wheeled vehicle, significant sliding and friction of the wheels may be generated under heavy payloads, or when vehicle is equipped with wide tyres. The reason for this is that steering requires rotation of the wheels around a vertical axis. In both vehicle types, any slight error in their control or actuation can result in wheel slippage, accumulation of positioning errors and unstable control. Autonomous control of the vehicles requires knowledge of mathematical models of the vehicles, or implementation of some form of intelligent controllers, e.g. fuzzy-logic based controller (refer to next section).

For a Mecanum wheeled AGV, control of the vehicle speed and direction are achieved by implementing different combinations of the wheels and the directions of their rotational speeds. The rotational speeds of the wheels being used during the manoeuvre are kept constant. Distributed control architecture exploits the requirement that each wheel must be independently controlled. For example, to move the robot to the left, two right wheels can be rotated against each other outwardly, or two left wheels can be rotated against each other inwardly, or both the left and right pairs of wheels can be simultaneously rotated in the directions explained above (**Figure 1**). Other desired orientations of the mobile vehicle 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 vehicle its ability to change direction of its motion without changing its orientation.

For the omni-directional AGV that implements conventional wheels, its speed and directional controls are achieved by controlling the wheels' steering angles and the value of the rotational speeds of the wheels. Any direction of vehicle motion can be achieved. **Figure 1** shows the wheel directions of a conventional wheeled AGV and the corresponding motions that can be achieved with a Mecanum wheeled AGV. **Figure 2** shows the concepts of the wheel designs and the developed platforms of the mobile vehicles.

3 Kinematic model of Mecanum wheeled AGV

It has been indicated in the previous section that autonomous control of omni-directional vehicles requires knowledge of mathematical models of the vehicles. Let xyz be the reference coordinate axes, and $x'y'z'$ be the body-attached coordinate axes for a Mecanum-wheeled vehicle as shown in **Figure 3**. The forces that cause motion of the mobile vehicle, which are the forces developed on the individual actuated Mecanum wheels, F_n, can be determined to be: **Equation 1** where n is wheel number, v is resultant translational velocity of mobile vehicle, R is radius of wheel $n,{}_{n}\bullet$ is rotational speed of wheel n, R_{rn} is length from axis of rotation of wheel n to axis of rotation of rollers on wheel $n,{}_{Rn}\bullet$ is 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 $x'y'z'$. ${}_{n}\bullet$ 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 vehicle. If α is constant, then K_n can include the effect of α. In our case, because sin 45°=cos 45°. **Equation 2** is the slip between the rollers and the hub of wheel n, while **Equation 3** is the slip between the rollers and the ground on wheel n.

For a Mecanum-wheeled vehicle, the resultant direction of the vehicle velocity is the same as he resultant force developed by the wheels. With further analysis, it can be shown that the kinematics equations of the Mecanum-wheeled mobile vehicle with respect to the reference coordinates system can be written as (**Lazic, 2002**): **Equation 4** where $v(t)$ is the resultant velocity, β is the direction of the resultant velocity and $\varphi(t)$ is the posture of the vehicle.

For the four wheel steered vehicle, let $x''y''z''$ be body-attached coordinate frame with its origin at the mass-center, m'_i. Then, the velocities $v'_i(t)$ of the wheel-surface contact points are related to the velocity $v''(t)$ of the body-attached coordinate frame as: **Equation 5** where $\varphi.(t)$ is the angular velocity of the vehicle's frame coordinate system $x''y''z''$ with respect to the fixed coordinate system xyz, k is the unit vector in the z direction and P_i is the contact point between the ground and the wheel. A further analysis of the vehicle dynamics will results in the following kinematics equation for the four-wheel steered vehicle: **Equation 6** where $v'_x(t)$ and $v'_y(t)$ are the components of the resultant vehicle velocity, β' is the direction of the resultant velocity and $\varphi'(t)$ is the posture of the vehicle.

The above models can be used in conjunction with the distributed controller architecture in order to achieve improved, reliable control of mobile vehicles. In case of failure in some nodes of the distributed controller, other nodes can be used to compensate for the nodes which have failed. This minimises the number of degrees of freedom that are lost when nodes fail, and to keep the vehicle's omni-directionality ability to some extend. The models can be further used to limit the effects the accumulation of errors during navigation. Sources of errors can be incorrect physical parameters such as incorrect wheel diameter, incorrect distances between the wheels, encoder readings that do not correspond to the actual displacement of the robot for different reasons such as uneven floors, wheel slippage, limited sampling rate of the digital controller and resolution, etc.

4 Distributed control architecture for mobile vehicles

Brooks (1986) introduced the principles underlying the design of distributed control architecture for mobile vehicles. His distributed architecture is based on distributed software layers that have different competencies. This allows easy extensions of the software control program at a latter stage, and robust system operation (**Figure 4**). It thus becomes possible to modify previously implemented control competency patterns without changing the existing control program structure through suppression and/or inhibition of specific software elements in the relevant layers. If the control functions of a particular layer fail, then the behaviour patterns of the other layers still work properly. However, if this software control architecture is implemented in a central controller, then failure of the controller will lead to failure of the whole system. Layers of Brook's control architecture are best implemented on distributed controller modules of a distributed controller. The control software architecture that is implemented in the two omni-directional mobile vehicles presented in this paper is the same as the Brook's architecture.

Another important aspect of mobile vehicle control is trajectory control, which can be embedded in layers 0-5 of the Brook's architecture. Embedding trajectory control in each distributed module of the distributed controller ensures that all the distributed controller modules have the same competency of controlling the mobile vehicle's motion. General trajectory control consists of standard feedback loop around the mobile vehicle. Feedback signals are provided by sensors such as inertial sensors, object detection sensors, wheel encoders, CCD sensors, etc. depending on the application and the level of autonomy of the mobile vehicle. All the distributed controller modules are updated with the feedback signals from sensors. Feedback sensors are themselves distributed on the controller network on different distributed controller modules. The reference to the controller is provided by the execution module, which simply, at some predefined rate picks the next reference point from the trajectory table. The planning module calculates the time history of references. Occasionally, there can be feedback from some sensor system reporting events like mission accomplished or re-planning needed. The feedback loop from some sensor system to the execution module enables the recalculation/re-planning/adjustment of the reference depending on how close the vehicle is to the target position or path. The planning module has status information from the system and environment regarding the mission, obstacles, etc. (**Figure 5**).

The purpose of the execution module is to provide references to the controller based on the state of the system and each controller's sampling instance. It is initialised by history of reference points from the trajectory generation module and its inputs include motion observations. It changes the references itself or the update rate based on the system state and saturation in actuators. It may switch to semi-autonomous control while, for instance, passing an obstacle. Its output is reference to motion controller. It may be interrupted by the planning module (**Bak, 2001**).

Trajectory generation module is re-executed in case of new information regarding target or obstacles. It is initialised by the task criteria, e.g. object avoidance. Its inputs are observations of the environment (obstacles), initial and target position, velocities and mission status. It calculates smooth history of references or simply defines the target of the motion along with constraints on the trajectory. Its outputs are history of references and target. It is event-based and its sampling occurs when mission is accomplished or re-planning is needed.

Some of the fundamental questions concerning distributed control of systems with communicating, closed-loop controllers include (**Bak, 2001**; **Teneketzis, 1996**):

- Which communication nodes (or distributed controllers) should know what and when?
- Which communication nodes should communicate with which nodes?
- When should distributed controllers communicate?
- What should distributed controllers communicate?
- How do the control delays affect the performance of the system?
- How do the loops of distributed controllers affect the stability of the system?
- How does the loops affect the performance of the system?
- How is such a system designed to achieve desired performance?

Questions 1-4 are answered by implementing a broadcast communication structure for globally distributed variables. Globally distributed variables are determined as those variables that are needed at each communication node (or in each distributed controller) for the correct functioning of the mobile vehicle. They are event-based or eventgenerated variables such as notification of task completion. The messages to be implemented can be determined by using techniques such as finite state machines and Petri-nets. Questions 5-8 are answered by analysing the dynamics of the system to be controlled. This approach is only suitable for system with few distributed modules. Higher communication layers for the management of ingress and egress of communication messages are needed for systems comprising of many distributed modules, such as the one developed for a modular reconfigurable robot (**Zhang *et al.*, 2001**). A theoretical analysis of distributed supervisory control with communicating controllers is covered by **Barrett and Lafortune (2000)**.

Delays introduced in distributed control affect performance of time-critical tasks. This includes problems that are concerned with timing, such as lag effect of zero-order hold and problems with respect to motion control. Constant delays as well as the lag effects can be easily compensated in discrete-time control design. Compensating time-variations, which may be stochastic, is much more difficult. These problems are solved by implementing the intelligent closed-loop controller at each distributed module to control local processes that are time-critical. 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 (**Chen, 2001**).

5 Controller area network, distributed mechatronics controller

AGVs that are presented in this paper implement CAN, MCP2515 CAN controller from Microchip, as a stand-alone distributed controller. PIC18F442 microcontrollers are used as local controllers on the distributed modules of the CAN network. Local microprocessors control local and/or remote actuators and monitor local and/or remote sensors by sending correctly addressed messages on the CAN bus. Four CAN nodes, each controlled by a local microprocessor, are used as wheel nodes for each omni-directional AGV to control functioning of vehicles' wheels. Nodes 1-4 are used to control each of the four sets of wheels. Nodes 5-8 are used to monitor outputs from sensory circuits that are needed to achieve collision free and reliable working of the mobile vehicle. Node 5 monitors the output from inertial sensors. They are used to measure the translational and rotational motion of the AGV. They can also be used to measure the slip developed on the mobile vehicles' wheels by comparing their outputs with the outputs of the wheels' encoders. The human machine interface functionality, or access to CAN network, can be placed on any CAN node. Other sensors such as vision sensor, temperature sensor, infrared (IR) sensors, etc. can be added to existing nodes or additional CAN nodes depending on the application of AGV and the required level of autonomy.

Distributed nodes are designed so that they have the same general competency levels of controlling the AGVs' motion and making the AGVs to navigate without collision. This is different from the architecture proposed by **Brooks (1986)**, where each node or architectural layer has a different competency. 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 vehicle's mission, other nodes have the competency to control the mobile vehicle 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**.

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; and
- monitor the status of local sensors and/or send the local sensors' status on the CAN bus.

Control tasks of the omni-directional AGVs presented in this paper are divided into time-critical and event-based control tasks. Control functions such as navigation and motion planning are time-critical, while control functions such as path planning and object avoidance are event-based. Time critical functions are controlled on the local CAN node (real-time section on **Figure 5**). CAN messages are used to update the status information of distributed variables such as wheel encoder value at some pre-determined time intervals, or when requested by a remote controller. They are used for synchronization of the robot's task as well, such as notification of tasks. The last message to be sent on the CAN bus indicates the current robot tasks (or the criteria, ref **Figure 5**). CAN message identifiers indicate the priority of the messages. Message priority is assigned according to descending values of the message identifiers. **Figure 7** shows the 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 vehicle is controlled by synchronizing the software control programs on different CAN nodes that control and monitor the vehicle's sensors and actuators, respectively. Each CAN node has the ability to synchronize the distributed software control programs on different CAN nodes. The developed mobile vehicle 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.

6 Discussion and conclusion

Two omni-directional mobile vehicles that use different mechanism to achieve omni-directionality were designed. CAN is used as a distributed controller to take the advantage that each wheel of the developed omni-directional mobile vehicles is independently driven. CAN facilitates robust communication between embedded, mechatronics controllers. The distributed control nodes must be properly synchronized so that communication network is optimized.

Petri-nets and other methods can be used to analyse the behaviour and performance of the developed distributed control system. The theory on the analyses of distributed controllers is still in its infancy, and analytical solutions for designing distributed systems are still difficult to achieve.

Equation 1

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

Equation 2

$$\frac{v}{R\dot{\theta}_{Rn}} = s_{rhn}$$

Equation 3

$$\frac{r_n\dot{\theta}_{Rn}}{R_{rn}\dot{\theta}_n} = s_{rgn}$$

Equation 4

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

Equation 5

$$v_i'(t) = v''(t) + \dot{\varphi}'(t)\cdot k\times\overrightarrow{\overline{m'}}P_i \quad i = 1, \ldots, 4 \qquad (3)$$

Equation 6

$$
\begin{aligned}
v'_x(t) &= \sum_{i=1}^{4} \left(v'_{ix}(t) + \dot{\varphi}'(t)\cdot \overline{m'}P_{iy}\right)\\
&= v'(t)\cdot \cos(\varphi'(t)+\beta'(t)),\\
v'_y(t) &= \sum_{i=1}^{4} \left(v'_{iy}(t) - \dot{\varphi}'(t)\cdot \overline{m'}P_{ix}\right)\\
&= v'(t)\cdot \sin(\varphi'(t)+\beta'(t)), \qquad\qquad (4)\\
\dot{\varphi}'(t) &= \frac{4\cdot v'(t)\cdot \cos\beta'(t)}{\sum_{i=1}^{4}\overline{m'}P_{ix}} + \frac{4\cdot v'(t)\cdot \sin\beta'(t)}{\sum_{i=1}^{4}\overline{m'}P_{iy}}\\
\text{where }\ \beta'(t) &= \arctan\left(\frac{v''_y(t)}{v''_x(t)}\right)
\end{aligned}
$$

Figure 1 Directional control of omni-directional AGVs implementing Mecanum wheels and conventional wheels. Direction of rotational speeds of wheels and Mecanum wheels' roller angle α shown

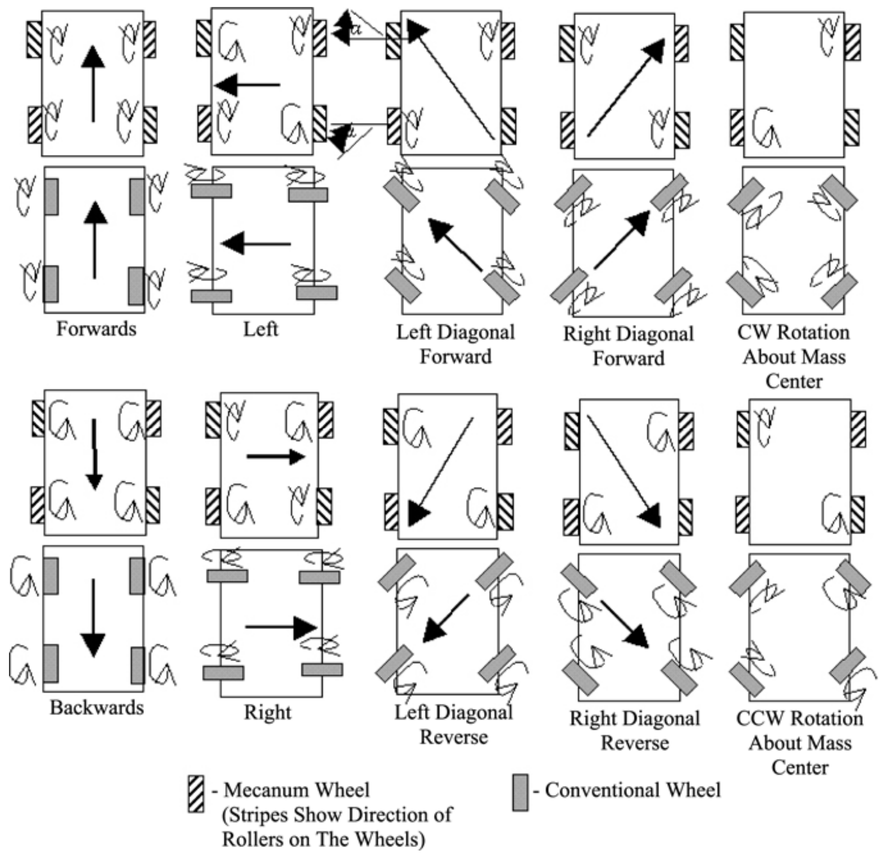
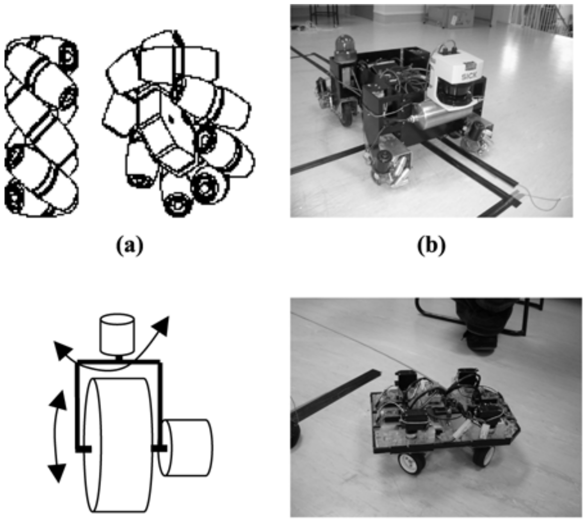


Figure 2 (a) Mecanum wheel design; (b) Mecanum wheeled omni-directional AGV; (c) concept of omni-directionality using conventional wheel; (d) conventional wheeled omni-directional AGV



(a) (b)

(c) (d)

Figure 3 Kinematics models of two omni-directional AGVs

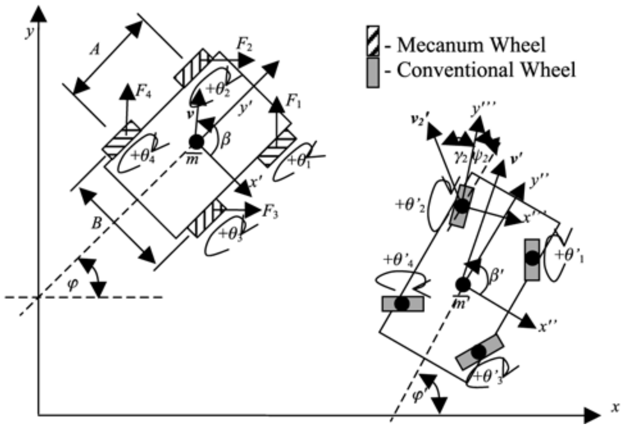


Figure 4 Brook's robot control architecture

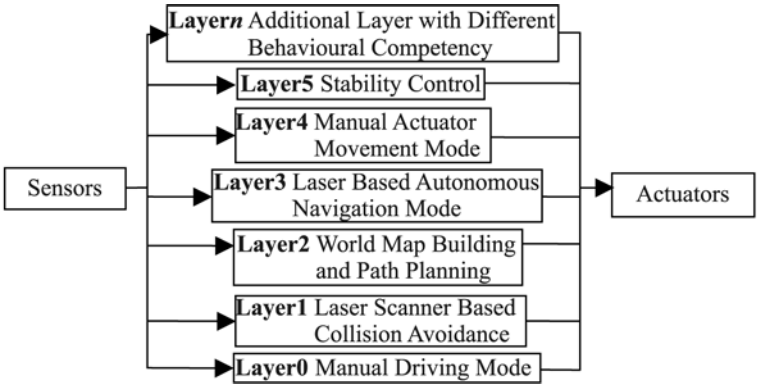
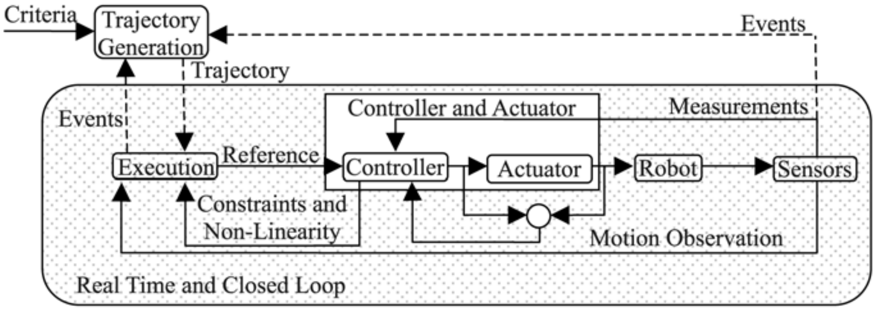
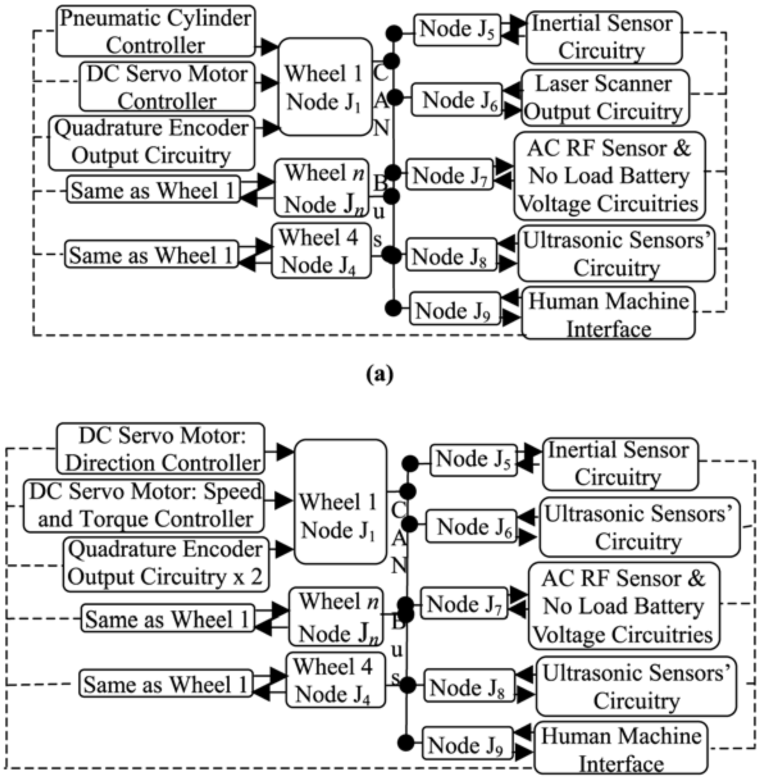


Figure 5 Overview of general generation/execution scheme for sensor – and event-based trajectory control



Source: Bak (2001)

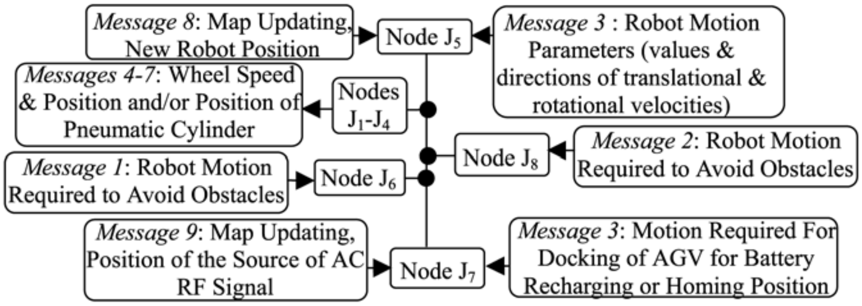
Figure 6 Sensory, actuation and distributed control architecture implementing MCP2515 CAN controller (a) Mecanum wheeled AGV; (b) conventional wheeled AGV



(a)

(b)

Figure 7 Specific message structure of CAN nodes for Mecanum wheeled AGV



References

Badve, A.A. (2003), "All terrain omni-directional autonomous mobile vehicle", Massey University, Auckland.

Bak, M. (2001), "Control of systems with constraints", Department of Automation, Technical University of Denmark, Lyngby.

Barrett, G., Lafortune, S. (2000), "Decentralized supervisory control with communicating controllers", *IEEE Trans. on Automatic Control*, Vol. 45 No.9, pp.1620-38.

Borenstein, J., Everett, H.R., Feng, L. (1996), *Navigating Mobile Vehicle*, AK Peters, Wellesley, MA.

Brooks, R.A. (1986), "A roboust layered control system for a mobile vehicle", *IEEE Journal of Robotics*, Vol. 2 No.1, pp.14-23.

Chen, D-J. (2001), "Architecture for systematic development of mechatronics software systems", Mechatronics Lab, Department of Machine Design Royal Institute of Technology, KTH, Stockholm.

Dubowsky, S., Genot, F., Godding, S., Kozono, H., Skwersky, A., Yu, H., Yu, L. (2000), "PAAM – a robotic aid to the elderly for mobility assistance and monitoring: a helping-hand for the elderly", paper presented at IEEE International Conference on Robotics and Automation.

Kalpakjian, S., Schmid, S.R. (2000), *Manufacturing Engineering and Technology*, Prentice-Hall, Englewood Cliffs, NJ.

Lazic, N. (2002), "Optimal vehicle dynamics – yaw rate and side slip angle control using 4-wheel steering", Department of Automatic Control Lund Institute of Technology, Sweden.

Microchip Com (2005), available at: www.microchip.com/ (accessed 15 January 2005).

Teneketzis, D. (1996), "On information structures and nonsequential stochastic control", *CWI Quarterly*, Vol. 9 No.3, pp.241-60.

West, M., Asada, H. (1997), "Design of ball wheel mechanisms for omni-directional vehicles with full mobility and invariant kinematics", *Journal of Mechanical Design*, Vol. 119 pp.153-61.

Zhang, Y., Roufas, K., Yim, M. (2001), "Software architecture for modular self-reconfigurable robots", paper presented at IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS).

Langson, W., Alleyne, A. (1999), "Multivariable bilinear vehicle control using steering and individual wheel torques", *Journal of Dynamic Systems, Measurement and Control*, Vol. 121 pp.631-7.

Makatchev, M., McPhee, J.J., Tso, S.K., Lang, S.Y.T. (1999), "System design, modelling, and control of a four-wheel-steering mobile vehicle", *Proc. of the 19th Chinese Control Conference, Hong Kong*, pp.759-63.

Matsumto, N., Tomizuka, M. (1992), "Vehicle lateral velocity and yaw rate control with two independent control inputs", *Tnas. of the ASME*, Vol. 114 pp.606-13.

Yu, S-H., Moskwa, J.J. (1994), "A global approach to vehicle control: coordination of four wheel steering and wheel torques", *Journal of Dynamic Systems, Measurement and Control*, Vol. 116 pp.375-90.

Corresponding author

Nkgatho Sylvester Tlale can be contacted at: nkgatho.tlale@up.ac.za