# ADAPTIVE MOTION CONTROL FOR A FOUR WHEEL STEERED MOBILE ROBOT

DETLEF HOLGER PLANTENBERG

A thesis submitted in partial fulfilment of the requirements of The Nottingham Trent University for the degree of Doctor of Philosophy

October 2000

# Abstract

For adaptive motion control of an autonomous vehicle, operating in a generally structured environment, position and velocity feedback are required to ascertain the vehicle location relative to a reference. Whilst the literature offers techniques for guiding vehicles along external references, autonomous vehicles should be able to navigate between despatch locations without the need to rely on external guidance systems. Considerations of the vehicle stability and manoeuvrability favour a vehicle design with four independently steered wheels.

A new motion control methodology has been proposed which utilises the geometric relationship of the angular displacements and the rotations of the wheels to estimate the longitudinal and lateral motions of the vehicle. The motion controller consists of three building blocks: the motion control system comprising the position tracking and the motion command generation; the electronic system comprising a data acquisition system and proprietary power electronics; the mechanical system which includes an undercarriage enabling permanent contact of the wheels with the floor. The components have been designed not only to perform optimally in their specific functions but also to ensure full compatibility within the integrated system.

For reliable deduction of the wheel rotations with a high degree of accuracy a dedicated data acquisition interface has been developed, which enables data to be captured in parallel from four optical encoders mounted directly on the wheel axles. Parallel sampling of the angular wheel position and parallel actuation of all steering motors improves the accuracy of the system state and gives a higher degree of certainty.

Considering only circular motion of the vehicle, a method for calculating the steering angles and wheel speeds based on the complex notation is presented. By cumulating the displacement vectors of the vehicle motion and the location of the centre of rotation between consecutive samples of the controller, the path of the vehicle is estimated. The difference between the nominal and the deduced centre of rotation is determined to minimise deviations from the reference trajectory and to allow the controller to adapt to changes in the road/tyre interface characteristics.

The individual mechanical and electronic components have been assembled and tested. Additionally, the performance of the electronic interface has been evaluated on a purpose built test-bed. For the experimental validation of the methodology, a simple method of mapping the centre of curvature with a pen mounted at the nominal centre of rotation has been proposed. Experiments have been conducted with both the steering angles fixed to their theoretical values for the nominal centre of rotation and with the proportional steering controller enabled. The results from the latter method have shown a significantly reduced deviation from the nominal centre of rotation.

The data captured of the angular wheel positions and steering angle settings has been analysed off-line. Good agreement is obtained between the deduced and the actual centres of rotation for the measurements averaged over 1.5 seconds. A number of different centres of rotation have been investigated and the required steering angles to compensate for the deviation have been plotted to form a control surface for the motion controller. The deviation between the estimated and the actual centre of curvature was less than 1.6% and adequate results could be obtained with the proportional steering controller.

The instinct of constructiveness, which is one of the chief incentives to artistic creation, can find in scientific systems a satisfaction more massive than any epic poem. Disinterested curiosity, which is the essence of almost all intellectual effort, finds with astonished delight that science can unveil secrets which might have seemed for ever undiscoverable...

Bertrand Russel

## Acknowledgements

I like to foremost thank my supervisor Dr. Eugene Lai for his guidance and invaluable support throughout the research project.

I like to extend my thanks to Dr. Frank Heitplatz for his advice, encouragement, and the many discussions we had.

I like to thank Dr. Martyn Jeffries for his help and input and the discussions.

Furthermore, I like to thank Doreen Corlett for her support throughout the research programme and the technical staff from the Department of Mechanical and Manufacturing Engineering at the Nottingham Trent University, in particular Ralph James and Jeremy Keeling for their help building the experimental mobile robot.

I also like to thank everyone else who has contributed directly and indirectly to this project but has not been mentioned before.

# Contents

# List of Figures

# List of Tables

# Nomenclature

| | |
|---|---|
| $CoC$ | Centre of Curvature – the axis the vehicle is rotating about [m] |
| $dCR$ | deduced Centre of Rotation – the centre of rotation deduced from the angular wheel displacements [m] |
| $nCR$ | nominal Centre of Rotation – the nominal axis the vehicle is commanded to rotate about [m] |
| $sCR$ | set Centre of Rotation – the point at which the wheel axle extensions intersect [m] |
| $v_c$ | vehicle velocity – the velocity of the vehicle measured at the geometric centre with respect to the wheelbase and track [m/s] |
| $v_{cn}$ | nominal vehicle velocity – the velocity of the vehicle at the geometric centre with respect to the wheelbase and track [m/s] |
| $v_w$ | wheel velocity – the velocity of individual wheels [m/s] |
| $le$ | length – the distance between the centres of the front and rear wheels; also referred to as the wheelbase of the vehicle [m] |
| $wi$ | width – the distance between the centres of the left and the right wheel; also referred to as the track of the vehicle [m] |
| $\boldsymbol{CW}$ | matrix of the wheel position vectors [m] |
| $\vec{R}$ | position vector to the nominal centre of rotation [m] |
| $\delta_i$ | steering angle of the i-th wheel [deg] |
| $T_V$ | Trace of the vehicle centre [m] |
| $T_{CoC}$ | Trace of the centre of curvature [m] |
| $T_{CR}$ | Trace of the centre of rotation [m] |
| $\psi$ | Yaw angle [deg] |
| $\psi_c$ | Cumulated yaw angle [deg] |
| $\kappa$ | Vehicle float angle [deg] |
| $\eta$ | Road/tyre interface factor |

# 1 Introduction

In the manufacturing industry, substantial investments are made each year to improve the quality and efficiency of production processes. Flexible Manufacturing Systems and Machining Centres have been introduced to minimise production overheads by reducing set-up and turnaround times. As a complementary measure, effective material transfer systems linking production islands are essential to maximise the utilisation of machines. However, unlike production machines, the development of material transport systems has received relative little attention.

When designing and implementing an internal transport system, the following three main points need to be considered: basic functions, flexibility, and adaptability to the existing building structure. Traditionally, manual transfer provides the best match to the three requirements where pallet jacks, pallet trucks or forklifts can be used for handling heavy loads. For high throughput, continuous transfer of materials between fixed locations, permanent installations such as conveyor belts, roller or chain conveyors can offer a cost-effective solution. However, conveyor systems often require high capital investment and they lack flexibility especially when despatch locations need to be altered periodically.

Alternative solutions that may provide the required flexibility are wheeled mobile platforms and purposely built vehicles for cargo handling applications. Automated Guided Vehicles (AGVs) are load carriers that transfer objects from one location to another. The use of these systems in factory environments is on the increase and is projected to increase dramatically in the next decade and the already emerging 'workerless' manufacturing facilities can be seen as an indicator for this new trend. However, the requirements for AGV systems are becoming more demanding due to the increasing complexity of factory floor environments. This has resulted in a need for AGV systems with augmented flexibility and reliability. In particular, it is becoming more important to be able to dynamically alter the AGV job queue and the path (Bostel et al., 1994).

Perhaps the most fundamental capability required for navigation by a mobile robot is that of self-localisation, that is recognition of the current location with reference to a known datum. Without the ability to reliably identify locations, a mobile robot will inevitably become lost, and therefore be unable to move purposefully between target locations. The topic is usually divided into two related sub-problems, namely *position tracking*, which assumes that the initial position of the robot is known, and *global localisation*, which entails being abe to *re-localise* under global uncertainty, i.e. to recover from becoming lost (Duckett and Nehmzow, 1999).

Figure 1 shows the layout of a possible manufacturing plant containing stores, a material collection point, a despatch area, several machining centres and a manufacturing cell.

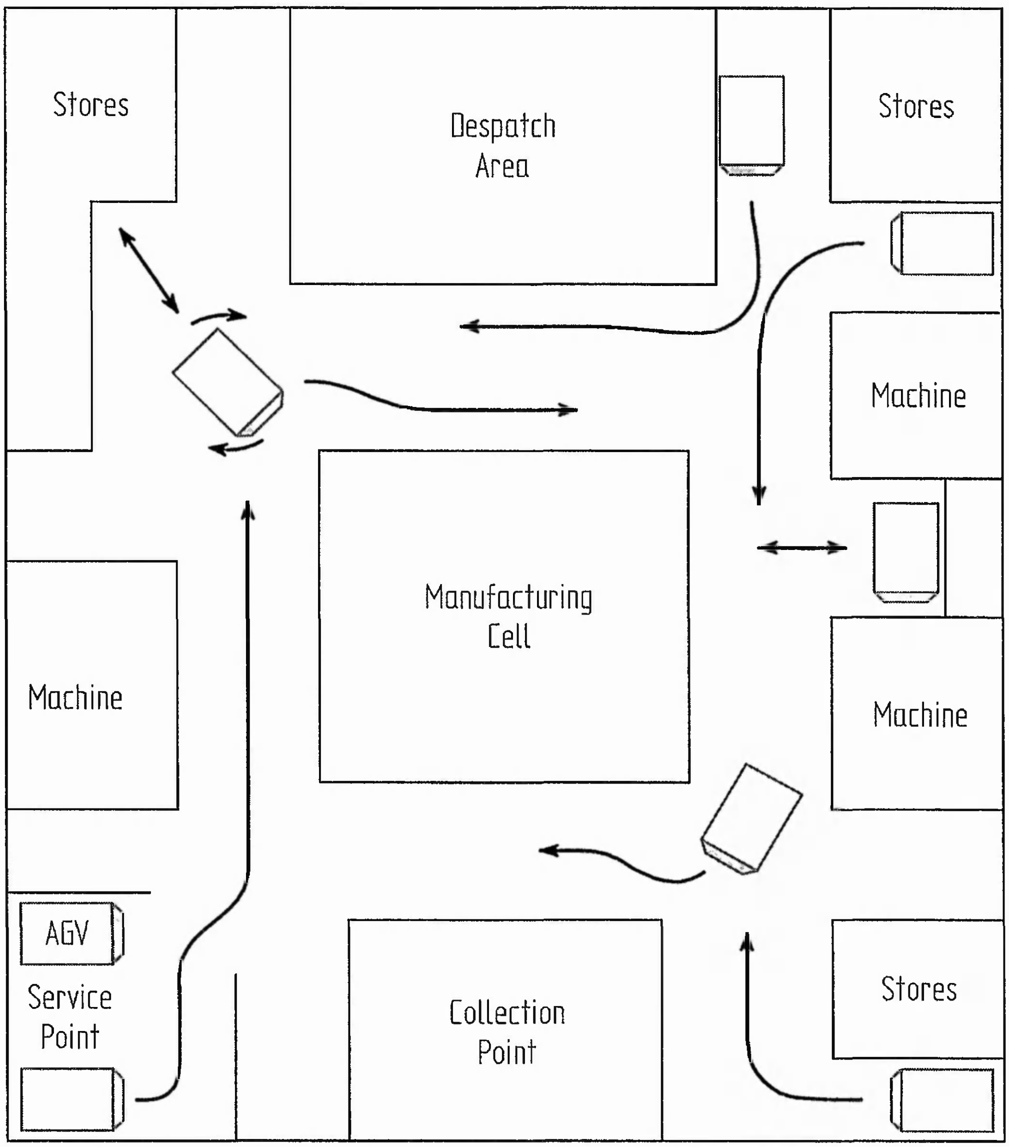


**Figure 1** A possible factory layout when utilising AGVs for material transfer

To facilitate a smooth running of the production operation, materials need to be transferred efficiently from stores or collection point to different locations on the shop floor; between different manufacturing or machining cells; and from these cells to the despatch area. During the life cycle of a factory, the facility layout can be expected to undergo frequent minor changes; replacement of machinery is a typical example. For an operation that involves frequent changes in despatch locations and varying demand levels, AGVs offer a suitable means to meet the required functionality and flexibility.

Ideally, an AGV should be capable of responding autonomously to changes in its operational environment and to different demand levels. Distribution and collection of materials from various locations require that individual tasks be scheduled to achieve maximum utilisation of a transportation system, taking into account the availability of AGVs, their current positions and the despatch locations.

In most cases, an AGV assigned the task of material transport would be expected to determine the shortest path to a specified destination and if congestion or obstacles were encountered, the search for alternative routes would precede the execution of the allocated task. The degree of flexibility and adaptability demanded by such a free ranging AGV implies that the vehicle would need to have the capability to navigate freely between manufacturing islands, without the need to rely on external guidance systems.

Furthermore, efficient positioning of an AGV relative to a machining centre or manufacturing cell for loading or unloading purposes requires access to several different docking strategies. A highly manoeuvrable vehicle greatly simplifies such docking operations in a confined space. To achieve the desired manoeuvrability, four main steering modes (Figure 2) have been identified: front, rear, parallel, and anti-parallel steering.

The transition between the different steering modes requires that the steering angle of each wheel must be independently adjustable - a condition often described as independent steering mode. Manoeuvrability of such wheeled platforms is improved if all wheels are steered independently. It is fair to say that all the aforementioned steering modes can be regarded as subsets of the independent steering mode, which is an essential feature to enable the evaluation of the road/tyre interaction characteristics in this project.

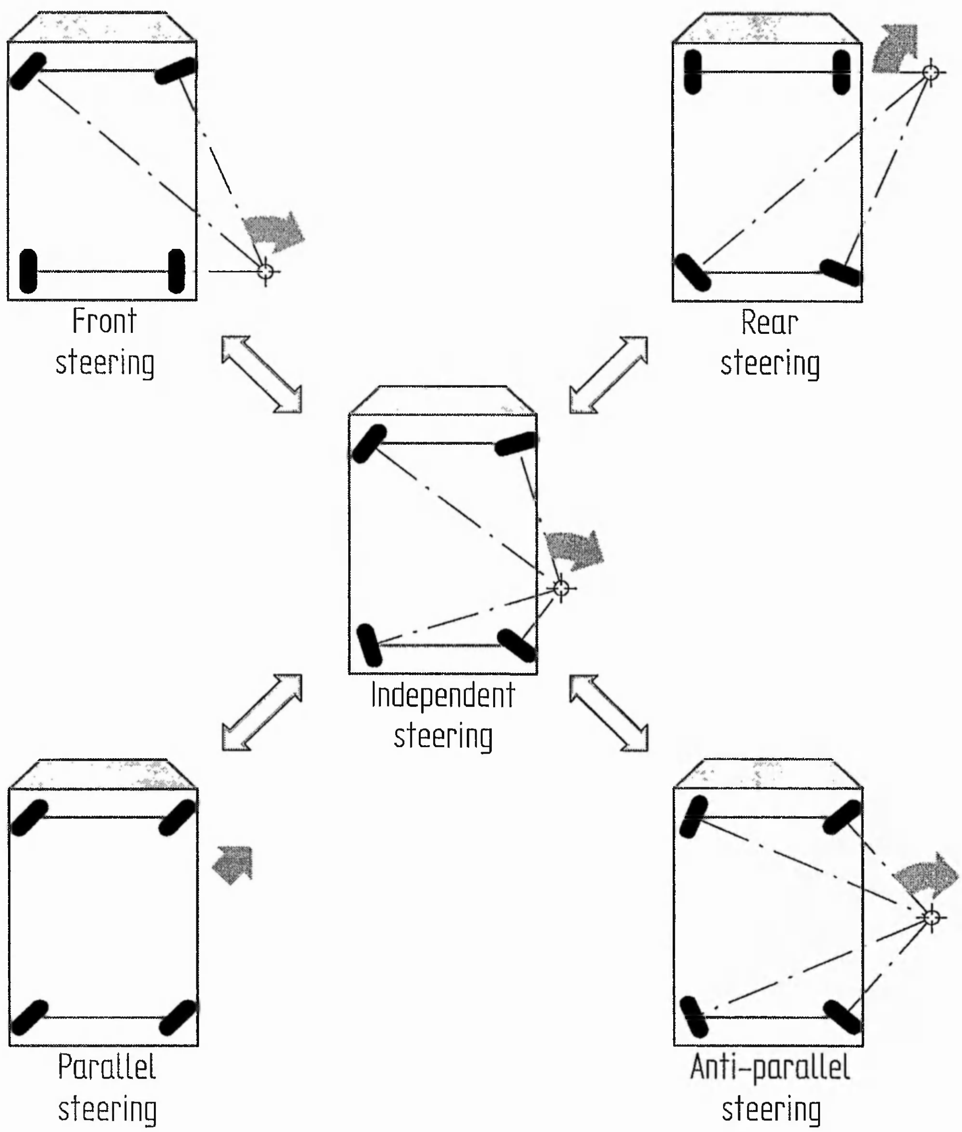


**Figure 2** Steering modes for four wheel vehicles

In order to achieve independence of the vehicle from external guidance systems one method is to rely on a close match of the vehicle's motion to a prescribed trajectory. It follows that two sets of key conditions must be satisfied: compatibility of the motion of all wheels; a functional relationship between the steering and driving rates. In the case of a four wheel steered vehicle, four steering and four traction parameters are to be controlled, which must satisfy one transcendental compatibility condition (Alexander and Maddocks, 1989). Furthermore, from the eight controllable parameters, only one velocity and two steering angles are independent. The five remaining, essentially redundant wheel parameters can be calculated so that the compatibility condition is satisfied. If there is an error in any of these parameters, such that the steering or driving modes are incompatible, slippage would occur.

Slippage may also occur due to longitudinal forces (i.e. traction or braking) or lateral forces from changes in the vehicle's path, resulting in centrifugal acceleration while cornering. In order to minimise the vehicle's deviation from a prescribed trajectory slippage or road/tyre interaction needs to be quantified and taken into account.

Attempts have been made (Matsumoto and Tomizuka, 1990) to incorporate the forces acting on the vehicle into the motion controller. The integration of the lateral, longitudinal and yaw acceleration has been utilised together with the differential driving torque to derive forces acting on the vehicle. For an AGV operating on the shop floor under high-load conditions, at low speeds and tight corners the signal to noise ratio of the measurements that can reasonably be achieved has unveiled some limitations of this approach. Any uncertainty should be avoided for vehicle control and safety reasons.

In order to incorporate localised road/tyre interaction into a motion controller, a reliable method for quantifying the vehicle's interaction with the road needs to be developed. The knowledge of the longitudinal and lateral motion of the vehicle while manoeuvring will enable appropriate traction and steering commands to be produced. This in turn allows the vehicle to compensate for localised changes of surface characteristics and thus improves its motion control.

Using the inverse kinematics of the actual wheel parameters, the correlation with the ideal rolling parameters will allow the deduction of road/tyre interaction characteristics, providing no skidding occurs (i.e. the saturation of the available friction is avoided). Several mathematical models describing the kinematics of standard passenger cars or four-wheel steered vehicles have been published (Cherry et al., 1995, Potter et al., 1996, Thanjavur and Rajagopalan, 1997), but most of them have neglected the important contribution of friction characteristics in the equations of motion. This is partly due to the lack of information describing the road/tyre interaction, and partly to the complexity involved in determining friction characteristics experimentally. Consequently, it is common to assume a constant friction coefficient when analysing the kinematics of a vehicle.

Realistic road/tyre interaction characteristics can only be deduced experimentally and hence a model vehicle is required to allow access to all the relevant parameters. An experiment designed to trace the nominal centre of rotation of the vehicle when it is subjected to different centrifugal forces would allow a range of road/tyre characteristics to be deduced during steady cornering. Evaluating the experiments conducted under different operating conditions can demonstrate the usefulness of the method for an autonomous vehicle motion control system.

**Aims**

The aims of the present investigation are:

- To develop a reliable method for quantifying road/tyre interaction
- To apply this knowledge to enable the determination of remedial actions for adaptive motion control of an autonomous vehicles (AVs)

This project aims to improve the understanding of the motion of a four-wheel independently steered mobile platform, particularly when cornering. In view of the difficulties associated with currently employed methods, which determine forces from measurements of acceleration and consequently the deduction of the vehicle's motion, an alternative method is needed for quantifying road/tyre interaction and determining the motion of an autonomous vehicle. In the present investigation the term 'road' refers to industrial surfaces and 'tyres' imply solid rubber tyres.

The proposed method is based on the consideration of a tyre's angular displacement when it is subjected to significant external lateral forces (e.g. during cornering). A tyre subjected to a lateral force elongates the path travelled in comparison to the effective path. This displacement of a wheel may be used as a measure of the lateral displacement of the vehicle. Thus a measure of the vehicle's lateral motion could be deduced by considering the geometric relations governing the resulting angular displacement under various operating conditions. Once the lateral displacement has been ascertained, this can be incorporated into the intelligent motion control system so as to minimise the deviation between the prescribed trajectory and the vehicle's path. In this manner the time interval between updates to a fixed reference can be increased. This will also allow the vehicle to adapt to localised changes of the road/tyre characteristics.

The knowledge of the interaction of the vehicle with the road will enable the formation of an empirical model, which, in turn, provides information of how to optimise steering and traction commands in order to minimise deviations from the reference path.

**Objectives**

- Review existing motion control models for wheeled mobile platforms
- Development of a kinematics model to cater for four wheel independently steered autonomous vehicles
- Design and manufacture of a four wheel independently steered model vehicle
- Development of measurement techniques for the characterisation of road/tyre interaction
- Development of a data acquisition interface module capable of sampling angular wheel displacements in a parallel manner
- Development of an undercarriage to ensure all wheels of the vehicle are in permanent contact with the road surface
- Experimental quantification of road/tyre interaction
- Experimental validation of the proposed methodology

**Structure of the Thesis**

This thesis presents the work undertaken in the research programme and is divided into ten chapters. Chapter 2 details a literature review covering the currently used models for vehicle kinematics and highlighting some of the control methods used for wheeled mobile platforms. Chapter 3 presents the adaptive motion control methodology and identifies the required components. The mechanical design for the new methodology is described and Chapter 4 and the details of the electronic interfaces including a data acquisition interface is described in Chapter 5.

In Chapter 6 the method of determining the road/tyre interaction geometrically, based on accurate measurements of displacement and velocity vectors is described. In Chapter 7 the integration of the individual component and the practical implications and aspects are given. Chapter 8 presents the experiments undertaken and the results obtained which are discussed in Chapter 9 and Chapter 10 concludes the report with the main outcomes of the research programme.

# 2 Literature Review

This chapter reviews the published literature related to path planning, modelling of vehicle kinematics including friction characteristics and control strategies currently used for Automated Guided Vehicles (AGVs). In addition, aspects from the automotive industry applicable to vehicles operating in a shop floor environment are highlighted.

Prior to the 1990s, AGV research has mainly concentrated on various aspects of the optimisation of path planning and task scheduling in order to improve vehicle utilisation and hence reduce costs. More recent research employs artificial intelligence tools such as fuzzy logic, neural networks or genetic algorithms as an adaptive method to optimise paths and predicts most likely occurrences of events in an attempt to counteract bottlenecks in an early stage. The methods developed are usually applicable to both road vehicles and AGVs for shop floor application. One major contributor to the development of Automatic Vehicle Control (AVC) technology is the US Program on Advanced Technology for the Highway (PATH), which is a tripartite partnership between US government, academia, and industry. Based on their experience through involvement in the PATH program, Shladover et al. (1991) suggested that both lateral and longitudinal control functions must be integrated to achieve complete vehicle motion control. The work on the lateral control function within the PATH program is based on the concept of a close co-operation between an intelligent vehicle and lateral position references installed along a roadway. These devices supply the vehicle with preview information about the changes in the forthcoming road geometry.

## 2.1 Vehicle Guidance Systems

Similar to the AVC technology developed under the PATH program an implementation of a four wheel independently steered driverless platform has been described by Albrecht (1995). The transportation system is used at a port to transport containers from the crane loading stations to relevant storage and vice versa. The high speed CPU of the SIMATIC PLC is used for both the geometric calculation of the desired steering angle of each wheel and the control of its position according to the selected steering mode. The onboard obstacle detection unit and an external navigation system are linked to the vehicle's controller so as to correct any deviation from the desired path and to alter the motion commands accordingly.

Durrant-Whyte (1996) described a similar design of an autonomous guided vehicle for transporting cargo containers in a port environment. The vehicles operate in a reasonably well-structured environment where absolute position estimation and control are fundamental requirements. The navigation system is based on millimetre-wave radar sensors detecting typically three of the 150 beacons distributed at known locations and reliably achieves accuracies of better than 3 cm over the complete port area. The kinematics model of the vehicle is central for both the navigation and the control. Although the vehicle dynamics are important at the anticipated speeds, these have been ignored because they become very difficult to model when combined with the dynamic effects of the hydraulic drive train. The dynamic effects can be minimised by using an overpowered drive system and a lookup table of maximum accelerations. The wheels on each axle are mechanically linked and the equations of motion are derived using a *double bicycle* model. The system architecture is based on 11 transputers and explicit software support for parallel processing.

A hydraulically powered four-wheel steering system for a vehicle with payloads between 20 - 50 tonnes is described by Entao et al. (1996). The steering position of each wheel is feed to a single-piece processor, which controls the proportional valves of the steering system. All four wheels turn simultaneously in a range of 180 degree so that the vehicle body can move in any direction. They stated that if the synchronism error of the steering angles of any wheel exceeds a certain threshold the vehicle may either be unable to move or even damage the vehicle's structure.

More basic approaches (e.g. embedded wire and triangulation methods) are currently used as AGV guidance systems on the shop floor. Attempts to automate material handling systems began in the early 1950's when trucks were used to tow rolling loads. The trucks follow inductive wires, embedded in the factory floor, which provide a reference path and appropriate control systems are utilised to correct measured deviations. Floor mounted wires can also be used to relay other information such as vehicle status, congestion or a change of the destination.

However, the flexibility offered by the current AGV guidance systems is limited and the cost of changing the layout of the predefined paths could have considerable impact on production overheads. Subsequently, reflective strips have been used in place of the inductive wires, but these have proved to be less durable and more failure prone.

In an attempt to reduce system failures, Senoo et al. (1992) investigated the application of fuzzy logic for the steering control of an industrial mobile robot in terms of response time and energy saving in case of a step change in the guide tape. The required steering energy is used as the performance index and is assumed proportional to the angular steering velocity. However, difficulties were encountered in terms of stability because of the effects of the centrifugal forces as the steering angle increased.

Another guidance technique accepted by industry utilises the signal of a rotating laser reflected from distributed sensors. The angular position between sensors in conjunction with a predefined map of the factory layout is used to calculate the location of a mobile platform. At any time, the signals from at least three sensors are required to triangulate the distance from the distributed sensors for safe operation. Moreover, sensors obstruction or interferences from diffracted light rays can cause the operation of the vehicle to be suspended until the problem has been rectified through manual intervention. Free ranging or at least partially independent AGVs would need to employ other means (e.g. odometry) of determining their locations within their operational environment.

A similar method for collision-free navigation of a mobile robot in a cluttered environment has been considered by Seneviratne et al. (1997). The free working space is divided into cells within which the robot is allowed to move. The cell boundaries are triangulated utilising ultrasonic sensors. This enables a triangulation graph to be constructed so that it represents the topological connectivity between the cells with free channels bounded by the obstacles and the environmental boundaries. This process selects segments of the path parallel to the environmental boundaries. This is favourable when relatively simple sensing devices such as ultrasonic sensors are used to measure the distances to objects for correcting navigation errors.

Martinez et al. (1993) addressed the uncertainty inherent in the collision avoidance problem by utilising a fuzzy logic based intelligent control strategy. The controller has been adopted to implement approximate reasoning necessary for handling varying degrees of uncertainty in the navigation and collision avoidance. Fuzzy logic was chosen because it allowed the intuitive nature of obstacle avoidance to be easily modelled using a linguistic terminology. Sonar and infrared proximity sensor data was used to avoid obstacles. While the fuzzy logic control strategy proved successful, difficulties were experienced when reflective characteristics of the obstacles led to deteriorating quality of the range and proximity data. In addition, the motion control actions dictated by the fuzzy controller were not always carried out successfully due to intermittent stalling of both the drive and steering motors. Song and Sheen (1995) applied real-time fuzzy-neuro based obstacle avoidance or local navigation to a mobile robot. The inputs for their system are ultrasonic sensory data from 8 of the 16 ultrasonic sensors (forward motion only) and the outputs generated are the velocity commands for the motor controller of the differentially driven experimental mobile robot. Another attempt to cope with uncertain, incomplete or approximate information was investigated by Surmann et al. (1995) who also proposed a fuzzy rule based system. Here eight ultrasonic sensors have been used to capture the environment of the robot. It was reported that difficult navigation tasks could be achieved by combining the local navigation tasks with a global strategy for example 'take a right when ever you can' within the fuzzy controller.

The use of neuronal networks for the extraction of a meaningful relationship between the robot and its environment from the sensed data has been proposed by Sethi and Yu (1994). In instances where the relationship between the sensed data and the desired output is so complex that it becomes very difficult to specify algorithms for the desired relationship, a non-algorithmic approach through the procedural learning capability of feedforward neural networks becomes very attractive. An ultrasonic ring sensor is utilised to capture the environment. Simulation and real sensor data have been used to show that the proposed scheme is useful in a static environment. Prabhu and Garg (1996) also utilise Artificial Neural Networks (ANN) for environment mapping because of their capability to acquire knowledge from training examples rather than by explicit programming. They concluded that ANN are appropriate tools for robot control.

## 2.2 Path Tracking

For the evaluation of the position tracking performance of a mobile robot, Duckett and Nehmzow (1997) compared three different localisation algorithms namely Dead Reckoning (DR), Self-Organising Feature Map (SOFM) and Evidence-Based Localisation (EBL). The raw odometry data of the robot was considered unsuitable for location tracking because of the familiar problems of accumulated rotational and translational drift errors due to wheel slippage. The rotational errors were corrected by combining the track data with the compass data and the translational errors were corrected by adjusting the tracked distance proportional to the distance travelled. The latter method utilised the robot's recorded sonar and infrared sensor data, which was then analysed with a self-organising feature map selecting one of 169 possible responses for the robot's location. The EBL algorithm associates the feature vector of the SOFM with the corrected dead reckoning data, assigns a confidence level to the hypotheses of 163 possible map locations and selects the most likely location. The results showed that the dead reckoning algorithm worsens over time while the self-organising feature map performs at a roughly constant level and the evidence-based system improves over time.

As a further development of their work (Ducket and Nehmzow, 1999) presented an approach that combines global localisation and position tracking using a topological map augmented with metric information. The method combines cross-correlation techniques for matching detailed sonar scans with a probabilistic algorithm for accumulating sensory evidence. They concluded that in order to achieve reliable self-localisation in a large, complex environment a mobile robot must be able to recognise places using landmarks rather than relying on dead reckoning methods. Even so, the robot may become disorientated for example when navigation begins in a long featureless corridor. Furthermore, it may be 'trapped' in a 'local minimum'; for example moving forward and backward between two incorrectly identified places in a phenomenon known as perceptual aliasing. As a possible solution, they suggested that in such situations the robot should revert to wall following until a higher confidence level of the robot's location has been reached.

A more systematic approach has been used by DeSantis (1995) for the development of path tracking controllers, which requires the convergence of the vehicle's state to a desired state rather than a pre-assigned function over time, a notion that is traditionally used in automotive applications. The advantage is that this procedure is applicable to a larger class of vehicles with single or double steering and in particular circular paths. With the assumption of slippage free motion at a constant tracking velocity, a controller is able to follow a straight line or a circular path may be implemented using classical PID (proportional, integral and derivative) techniques.

Sarkar et al. (1994) presented a state space realisation of a constraint system and discussed the input-output linearization and zero dynamics of the system. The approach is applied to the dynamic control of the system and two control algorithms, namely trajectory tracking and path following of mobile robots, are investigated. In each case, smooth non-linear feedback is obtained to achieve asymptotic input-output stability and Lagrange stability of the overall system. Simulation results are presented to demonstrate the effectiveness of the control algorithms and to compare the performance of trajectory tracking and path following algorithms. Experiments have been carried out on a differentially driven platform with four passive caster wheels. When the position and orientation of the vehicle are estimated from the integration of the velocities, small errors in the estimates due to either slippage and scuffing or errors in the wheel sensors can result in large position errors. This is because small errors in the angular velocities integrated over a large time interval result in large position errors and this presents a serious problem in control.

In order to minimise an accumulative position error Katz and Bright (1992) proposed a guidance system that uses fluorescent lights suspended from the ceiling as an external reference. While the method doesn't solve the problem of vehicle localisation, it can aid an appropriate navigation system to limit deviations from the proposed path. Improvements are however required for a more refined system, especially when lighting junctions are encountered or smooth turning around the corners of the gangways is required. An extension to this method has been described by Dulimatra and Jain (1997), whereby the indoor mobile robot self-localisation system uses ceiling lights and door number plates as landmark features to recover from failures during navigation. Usually, a system starts from a known state and updates the current state of the vehicle according to the actions implemented. However, sensor inaccuracies and signal noise may result in a navigation failure from which the robot may not be able to recover without additional external reference sources. Although the experiments exhibited occasional failure of individual components, the overall performance of the system has been consistent and it improved the localisation accuracy of the experimental vehicle.

Lee et al. (1995) implemented a real-time vision-based tracking system (ViTra) for controlling unmanned autonomous vehicles. ViTra is a DSP-based integrated vision system, which is characterised by low cost, computationally efficient and flexible in terms of control. In contrast to Dulimatra and Jain (1997), they used uniquely designed fiducial patterns suspended from the ceiling. These are used as landmarks to determine the location of the vehicle. A corresponding pattern recognition algorithm deduces the location, which is compared to the reference or desired location and the planned path is then amended accordingly. The system performance is evaluated with the characteristic trajectory to be a circle with 0.5 m radius. The average deviation between the known and measured position is reported at 4.86% and 3.65% in x and y coordinates respectively. The discrepancy is thought to be mainly attributable to the lighting conditions and possible misalignment of the vision sensor. For their method it is important that the CCD lens is perfectly levelled (pointing vertically to the ceiling pattern board) during operation to ensure precise measurement.

## 2.3 Tasks for a Motion Control System

An autonomous mobile robot is a system which is expected to be capable of interpreting, perceiving, executing and realising a task without any outside help (Beaufrere and Zeghloul, 1995). This can only be achieved if the robot is capable of sensing and perceiving its environment by means of appropriate sensors; the measurements are analysed and the environment represented in the form of a model. Using this information, a navigation algorithm could be developed which would allow the robot to determine freely a suitable trajectory from the available information. Additionally, the control system must be designed so to ensure that the robot moves correctly within its operating environment. Thus, a free ranging autonomous mobile robot needs to be equipped with the necessary resources to perform the required functions.

To handle the complex interaction between a number of different tasks, Roth and Schilling (1995) described a motion control system where the information about the path is stored in the vehicle's onboard computer. Fuzzy logic provides a robust method to derive reasonable control actions from limited ultrasound and infrared range measurements. Due to the overlap of the different fuzzy classes the resulting trajectories are smooth, as appropriate for payload safety. On detection of unforeseen obstacles on the planned path an efficient detour manoeuvre without major deviations from the target direction is activated.

Supported by four passive caster wheels the experimental robot's propulsion is achieved by a two-wheel differential driving system. The flexibility of the robot could be increased by reconfiguring the vehicle's parameters with respect to effective path planning and in particular to collision avoidance and docking target stations. Different control strategies are studied which support global path planning by local methods for collision avoidance and target docking. The local methods are based on the relative distance and velocity measurements to other objects using distance measurements and map feature comparisons. For docking, infrared beacons are used to assist measurements in order to correct the relative position towards the target. Advanced data processing algorithms are employed to compensate the deficiencies of the low cost, low precision sensors. The four level hierarchical control system incorporates:

- Mission Planning
- Path Planning
- Reactive Level
- Servo Level

Mission and path planning are typical applications similar to the travelling salesman problem and have been the subject of several publications (e.g. Katz and Bright, 1992, Bostel et al., 1994, Surmann et al., 1995, Seneviratne et al., 1997, Rus, 1997, Taghaboni-Dutta, 1997). An effective task assignment policy should include a strategy that is demand driven, which, for example, adds a value to a payload and increases this value the closer it resembles the finished product. Computer simulation of such a value added task assignment policy algorithm was carried out by

Taghaboni-Dutta (1997). His results showed that higher throughput and a reduction in 'work-in-progress' could be achieved. Benchmark test results for several performance factors such as throughput and transfer time showed that the algorithm that prioritises AGV payloads outperforms some of the best dispatching rules reported for manufacturing facilities where the transportation system is a critical resource.

The commands generated by the mission planning are passed on to the path planning and then to the low-level motion control system. At the path planning level, information exchange may already take place with the reactive level in terms of obstacle avoidance and the initiation of corrective actions (Song and Sheen, 1995). In this case the reactive level is responsible for devising appropriate actions to avoid collisions with unforeseen obstacles and to compensate for the accumulated position deviations. The servo level ensures that the demand remains within the performance envelope of the actuators and executes the desired traction and steering commands accordingly.

Weiczer (1995) developed an interactive CAD-simulation and visualisation tool to analyse different types of path planning systems for different vehicle configurations which include tricycle-type, differentially driven and multi-wheel steered vehicles. Methods have been proposed that allow the analysis of different localisation systems, which are applicable to mobile robots independent of the vehicle configuration. The path planning and the path following methods are implemented using virtual wheels to simplify the tracking of the trajectory, especially for curved trajectories. The reference trajectories are generated taking into account the dynamic and kinematics constraints of the vehicle and the steering mechanism. When a collision free trajectory has been generated, odometric methods are used to estimate the vehicle location. The different methods employed enabled the robot to follow the trajectory with minimum tracking error as long as the velocity did not exceed the value for which the trajectory was generated.

Based on a forward-reverse search algorithm and a neuronal network, Bostel et al. (1994) presented a navigation technique, which takes into account the current status of both the AGV fleet and the waiting job requests. A cost criterion was estimated for each node between the start and the goal node and the minimal cost path selected as the optimal path. It was noted that the proposed extension to the Branch-and-Bound method can perform better than many conventional methods in terms of flexibility and efficiency. In contrast, Kezong (1994) presented a path planning method which used the force field concept to direct the mobile robot towards the goal position and away from any obstacles. However, the method operates well only for a short range and hence renders itself as an obstacle avoidance system.

Cha and Gweon (1996) considered a local path planning algorithm which uses a directional weighting method using a laser range finder. The weighting method is based on both the configuration space method and the potential approach. The weighting value decreases the further the robot is required to deviate from the direct heading to the goal position in order to avoid obstacles. The directional weighting method enabled the mobile robot to approach the goal through the shortest path without colliding with nearby obstacles. A dynamical local path planning algorithm is considered by Choi and Lee (1996) for avoiding moving and stationary obstacles. If the dynamics of the robot is neglected the path planning on its own becomes an optimisation problem, a global search problem or a heuristic path finding problem. On the other hand, the dynamics of a robot has only limited impact on the path planning. The effects caused by neglecting the dynamics can be overcome by a path planning algorithm which allows only gradual changes of the velocity and acceleration. Hence, their proposed path planner incorporates a continuous time differential equation, which is based on Newton's second law.

The path-planning problem in the presence of kinematics constraints that cannot be integrated is considered by Barraquand and Latombe (1989). Such constraints, also known as nonholonomic constraints, are generally caused by one or several rolling contacts between rigid bodies and express that the relative velocity of two points in contact is zero. When the motion of the two contacts includes both rolling and sliding, then the expression, which depends on the friction coefficients between the two rigid bodies is non-linear, whilst without sliding the nonholonomic constraint is linear. This case is much simpler and widespread in practice. In a car-like robot, this corresponds to assuming no slipping of the wheels on the ground. Nonholonomic constraints make the dimension of the space of achievable velocities smaller than the dimension of the robot's configuration space. They reported that the implemented path planner is able to generate complex collision free paths for nonholonomic mobile robots among obstacles with a minimal number of manoeuvres.

From a pure theoretical view, the stabilisation problem for non-linear systems has been a challenging task ever since Brockett (1983) proved that a nonholonomic control systems with more degrees of freedom than controls cannot be generally stabilised by a smooth state feedback control law. However, it has been shown that the polar representation allows overcoming the obstruction to stabilisation and at the same time allows treating the issue of local and global stability in a straightforward way. The control law obtained is smooth in polar co-ordinates, while it is discontinuous in cartesian co-ordinates. The approach with polar co-ordinates enables devising a smooth, state feedback control law which yield in exponential stability of a closed system (Astolfi[(1)], 1995). The linear, globally stabilising, state feedback control law presented enables the robot to reach the final position without inverting the direction of its motion, thus avoiding discontinuous paths when initialised in certain configurations.

The properties of the closed loop system and its performance in the presence of errors and noisy measurements have been evaluated via simulations and real measurements. The results differ from those yielded by other approaches as the proposed control law exhibits the ability to counteract measurement noise and model mismatch. Astolfi[(2)] (1995) extends his approach to a discontinuous state feedback control law for nonholonomic chained systems, for example a vehicle with a trailer. Here the control law is derived as a linear control law with state dependent gain scheduling for a car-like vehicle. Even though the closed loop system equations are discontinuous, they admit a well-defined unique solution, provided the initial state satisfies a nonzero condition. Applying a similar control law to a two-degree of freedom nonholonomic mobile robot requires the use of a pure non-linear analysis. It was reported that every linear approximation approach turns out to be inadequate. Astolfi[(3)] (1995) noted that one of the most challenging topics in this area is the design of local or global stabilising control laws for nonholonomic systems with more degrees of freedom than controls.

Gartner and Astolfi (1996) addressed the problem of asymptotic stabilisation by means of a fuzzy logic controller for the kinematics of a mobile robot with two differentially driven wheels. A performance analysis of the closed loop system in the presence of a simple obstacle has been investigated. Using distance measurements from the obstacles and the goal position, the simulation attempts to position the mobile robot in a cluttered environment. They noted that the classic problem of reactive navigation, the so-called local minima trapping problem remains. Nevertheless, with additional control strategies, which guide the robot away from detected local minima such as corridors or cage like obstacles trapping can be avoided. Following from this work, Astolfi (1999) offered a solution to the kinematics and dynamic problem for a simple two degree-of-freedom wheeled mobile robot using a discontinuous, bounded, time invariant, state feedback control law and achieved exponential stabilisation for a driftless mobile robot.

Lizarralde et al. (1996) proposed a method, which uses an iterative algorithm to identify the singular controls of a driftless nonholonomic system. A singular control occurs when the centre of curvature coincides with a vertical centre of a wheel. This should also be avoided for practical reasons, since this usually implies high angular accelerations. Once the set of singular controls have been found, the algorithm is modified so as to avoid paths, which include singularities. A strategy has been implemented in a simulation similar to nonlinear predictive control. Popa (1998) examined a similar strategy for a class of iterative algorithm to solve the motion planning problems for nonholonomic systems. Computer simulations have been performed to validate his approach to the motion planning for various systems including tractor-trailer mobile robots. An exponential convergence rate is reported for the path-space iterative method when considering systems without drift. The algorithm is augmented with an exterior penalty function as an effective method to avoid obstacles. Then the method is applied to a system with drift. However, in systems with drift a problem arises since the drift term could cause the system at the last step to move away from the desired location more than the iteration at the next step could compensate for. Burke (1994) pointed out that all practical wheeled robots display slip to a certain extent, usually as a result of steering and drive controls failing to comply with the rolling criteria. The lack of compatibility can arise at any speed but it may not present a problem as long as the kinematics is not dominated by this shortcoming.

Historically, most of the research carried out on motion control systems has concentrated on road vehicles, but some of the basic information governing motion control and system integration is useful to the development of autonomous vehicles for shop floor applications. Potter et al., (1996) presented an approach to achieve totally integrated motion control. With an ever-increasing demand towards incorporating sophisticated control systems into vehicle design, they described three main systems which can influence the dynamic behaviour of a saloon car: vertical control (suspension), lateral control (steering), and longitudinal control (propulsion and braking). A significant degree of interaction between these systems is generally accepted. Indeed, previous research has shown that improvements in vehicle handling and stability can be achieved by co-ordinating these systems using an integrated control strategy. This is to be expected because of forces generated at the tyre/road interface by vertical, longitudinal, and lateral motions, interact with each other. For example, a large lateral force will limit the maximum obtainable longitudinal force. Thus, by means of an integrated control strategy, the available horizontal tyre force can be deployed more effectively for steering control. It should be noted that all vehicle motion control models rely greatly on an accurate description of road/tyre friction characteristics if realistic predictions of dynamic tyre forces were to be made.

## 2.4 Modelling of Tyres

A detailed history of the efforts undertaken to determine the dynamic properties of tyres and of automobiles, dating back to about 1933, can be found in Sakai (1981). He also comments on the significant early theoretical work and the experimental validation carried out by Fiala (1954) who considers the side force and self-aligning torque generated in a tyre rolling at constant slip angles. Fiala's theory retains its value even today as a fundamental theory of cornering properties of pneumatic tyres. Sakai's review of the overall dynamic properties of tyres in terms of cornering, braking and the tractive properties has been complemented by theoretical computations. The computations are carried out with respect to the six force and moment components generated in a tyre, and their dependence on slip angle, slip ratio, camber angle, load, internal pressure, velocity, temperature and drum curvature.

Characteristics of the tyre mechanics and the behaviour of an automobile undergoing combined longitudinal and lateral tyre forces have been analysed by Dugoff et al. (1970). They expressed the relationships of the longitudinal and lateral components of the tyre shear forces as an analytical function of the tyre normal load, the sideslip and inclination angles and the longitudinal slip. Simulations have been carried out to examine the influence of three parameters: the lateral tyre stiffness, longitudinal tyre stiffness and the coefficient of friction between the tyre and road surface. The characteristics curves generated from the simulation have been compared with the corresponding experimental data. They concluded that although the results obtained agreed qualitatively with the limited experimental data, a considerably base of accurately measured tyre shear force data is required to quantitatively assess the accuracy of the proposed model.

In an attempt to simplify the calculations required for Dugoff's tyre friction model Guntur and Sankar (1980) presented a method that implicitly utilises the traditional friction cycle concept. In general, the friction cycle says that in instances in which there is a combination of tractive and cornering forces, there is a trade-off between the amount of force that can be produced in each direction.

To deduce experimentally the wheel slip characteristics of a passenger car Germann et al. (1994) used the sensors designed for the Antilock Braking System (ABS) to measure the wheel rotations. For their friction monitoring system, the longitudinal and normal tyre forces were respectively deduced from a dynamics model for the vehicle's longitudinal motion. The angular velocity measurements of all four wheels have been used to deduce the longitudinal vehicle dynamics. The tyre forces at the front and rear axle have been reconstructed using a simplified model for the pitch dynamic. However, the proposed method is viable mainly during the acceleration manoeuvres because otherwise all tyres show the same slip and the differential velocity of the vehicle cannot be computed. A recursive least square algorithm is proposed to estimate one typical friction coefficient of the road/tyre system in real-time. They concluded that the friction characteristics could be approximated by a polynomial, providing that the excitation in the form of acceleration exceeded a certain threshold value.

Building on the work of Germann et al., Bachmann et al. (1995) performed driving tests in real road traffic to evaluate to which extent drivers utilise the available friction. The friction potential determines possible accelerations and the total vehicle dynamics. A standard passenger car had been equipped with a number of sensors including longitudinal acceleration, lateral acceleration at the front and lateral acceleration at the rear, yaw rate and angular wheel rotation. The results showed similar values for the mean acceleration potential used by standard and sporty drivers usually in the range between 30% and 43%. However, the maximum accelerations used were significantly higher for sporty drivers.

Filtered histories of forces acting on the vehicle can be used to construct through off-line analysis both tyre force estimates and state estimates. Szostac et al. (1988) analysed the effectiveness of an extended Kalman filter by means of a simulation for a simple control braking system slip and slip angle. An analytic pneumatic tyre force model has been developed from combinations of machine tyre testing and vehicle testing. With 47 input parameters they provide a comprehensive analytical tool for calculating saturation boundaries of pneumatic tyres when combined effects of longitudinal and lateral forces are to be taken into account.

A computer simulation of advanced ground vehicles carried out by Ray (1995) employs an extended Kalman filter to deduce state estimates for vehicle motion and tyre forces. A nine-degree-of-freedom non-linear bicycle model and an analytical tyre force model are used to simulate true vehicle motion. In the simulation, an equivalent single front wheel and a rear wheel has been used to replace the two front wheels and the two rear wheels respectively (double bicycle model). Subsequent state and force estimations use a five-degree-of-freedom vehicle model and append the tyre forces in combination with a simple PI (proportional integration) Slip Control Braking System as the state components to be estimated. The knowledge of the vehicle state and external tyre forces is essential to determine the dynamic behaviour of a vehicle and to design automotive control systems for enhanced vehicle safety and handling characteristics. Introducing a 'mismatch' between the true vehicle model and the estimated data enabled incorporating data from the real vehicle into the simulation. This allowed devising an adjusting algorithm for a yaw rate compensator. The analysis neglected the roll motion, but included the vertical (suspension) dynamics in order to retain the variation of normal forces due to the pitch dynamics of the vehicle. A suitable adaptation parameter was chosen so that the yaw response could be kept stable even for high-speed applications. However, the slip angle response was weak and the control signal had higher maximal amplitudes compared with those of a simple compensating controller. Only little performance degradation is reported from the simulation of stopping distances that were achievable if slip could be measured without error.

## 2.5 Control Architectures for Mobile Robots

A significant amount of attention has been paid to the development of different control architectures for mobile robots. Two principle architectures have been suggested:

- Horizontal decomposition
- Vertical decomposition

Horizontal (sometimes referred to as functional) decomposition is a classic top-down approach to building control systems. In this approach, the entire control task of the mobile robot is divided into a series of tasks, which are implemented by separate modules. These functional modules form a chain (Figure 3), through which information flows sequentially from the robots environment, via the sensors to the robot and back to the environment.

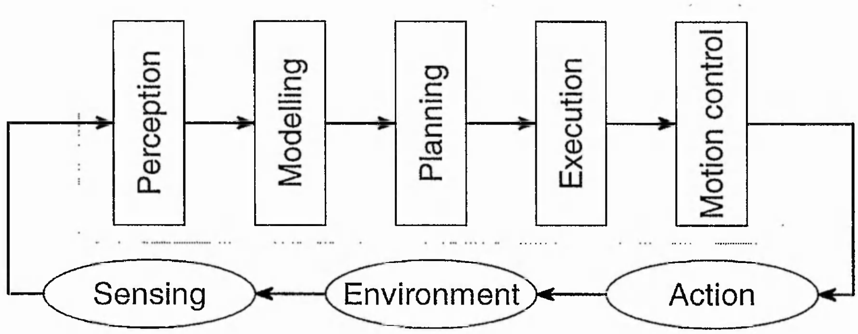


**Figure 3** Horizontal decomposition of a control system

In contrast, the vertical (sometimes referred to as behavioural) decomposition is a bottom-up approach. The term 'behavioural' encapsulates the following capabilities: perception, explore, avoidance, planning and task execution; these are essential attributes to achieve specific aspects of robot control. Individual components are capable of producing meaningful actions, which can be composed to form different levels of competence (Figure 4). In other words, a component can be implemented so to provide the connection between sensed data and actuation. The complexity of the system can be built up gradually starting from a very low level, say locomotion, to obstacle avoidance, then wandering and so on. Successive levels can be added incrementally in order to enhance the functionality of the robot.

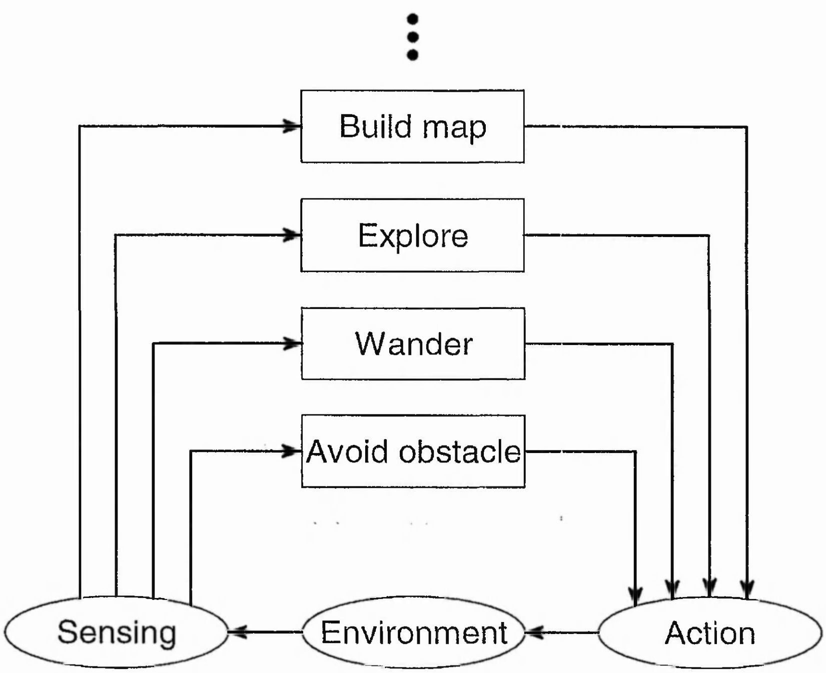


**Figure 4** Vertical decomposition of a control system

Cameron and Probert (1993) stated that the control task associated with a mobile robot is so complicated that one cannot simply follow a single decomposition scheme, while completely ignoring the other. A hybrid approach may be a better choice to build more robust, flexible, and high bandwidth architectures for mobile robots. In this context, it is worth noting that the Oxford AGV project has adopted a hybrid control structure, which combined the positive features of both the horizontal and vertical decomposition approaches a so-called distributed-real time architecture.

Furukawa et al. (1989) tried to identify the essential elements of the four-wheel steering technology in terms of vehicle dynamics and control techniques. Taking a broader view at the motor vehicles one could perceive that only the front tyres are involved in controlling the sideslip angle needed for cornering. The rear tyres generate cornering forces only by the sideslip angle resulting from the vehicle motion. The idea of steering the rear wheels simultaneous with the front ones means improving the vehicle performance in lateral motion. Steering the rear wheels can help to reduce the delay in the generation of the cornering forces and permits the vehicle's path and attitude (a body sideslip angle or float angle) to be controlled independently of each other. This characteristic would therefore reduce the required motion of the vehicle body around the vertical axis and offers a better response during a change in the vehicle's path. Cheng et al. (1994) developed a set of non-linear equations to represent the dynamics of a vehicle moving in a horizontal plane. Their analytical and experimental results from the dynamic model of automated wheeled vehicles showed a good approximation of a linear controller for small angles. To this end a vehicle with one steered front and one steered rear wheel and two non-steered caster wheels at the left and right side arranged in a diamond shaped is proposed. The optimal controller utilised a linearised model and takes into account the stiffness properties, rolling resistance and tyre properties. The results obtained from the experiments illustrated that the vehicle's motion can be represented quite accurately.

Cherry et al. (1995) investigated the use of Multi-Body Systems (MBS) modelling, non-linear simulation and linear analysis techniques for integrated vehicle control. The study provided support for the development of total motion control systems. It was concluded that the availability of a realistic vehicle model is essential for any analytical study of the automotive vehicle dynamics. To facilitate the evaluation of the overall dynamic behaviour of the vehicle with a totally integrated motion control system, a comprehensive vehicle model should contain all the primary subsystems including propulsion, steering, and suspension. Because realistic models for vehicle suspension systems can easily become very large and complex, Cherry and Jones (1995) applied fuzzy logic techniques for controlling a continuously variable damping automotive suspension system. The rule-based controller is independent of a representative model. They reported improvements in performance of the vehicle under both road and driver steering inputs.

In contrast, Hoyong (1996) favours a neural network approach. He noted that the operating condition of a vehicle may be under constant changes due to changing loads, tyre condition and a changing driving environment. It is therefore necessary to apply non-linear and robust control to stabilise the vehicle handling dynamics. He employed a conventional reference model and used a neural network compensator to minimise the errors, which may be present between the two components.

A detailed description of commercially available Vehicle Dynamic Controller (VDC) can be found in Bosch (1996). The controlled variables for passenger cars are the sideslip angle and the yaw rate. The feedback gains of the state controller are determined using a linearized four-wheel vehicle model. The wheel slip values, the resultant forces at the road/tyre interface and the slip angles are used in the linearized vehicle model to estimate the nominal slip at the appropriate wheels and correct deviations by adjusting the braking forces on the individual wheels accordingly.

Feng et al. (1998) extended the commonly applied two degree-of-freedom bicycle model for vehicle control. Their results showed lowered gain characteristics in the lateral acceleration response to the steering input that is generally attributed to the suspension dynamics in a three degree-of-freedom model. The linear model incorporated the suspension roll mode and the performance has been verified with experiments. The simulation results highlighted the importance of the effects of the roll dynamics in the steering controller design especially for vehicles with soft suspension systems. The results also showed generally reduced oscillations and that the controller was able to produce smaller gains around the suspension mode.

Halanay et al. (1994) presented a road vehicle rear wheel steering controller using a simple constant gain yaw rate feedback compensator and an adaptive yaw rate controller. The two uncertain parameters in the proposed controllers, namely the front and rear cornering stiffness, are dependent on road, tyre and motion characteristics and have been assumed constant for the simulations. Under these circumstances the adaptive controller appears to be more suitable for high speeds because the maximal amplitude decreases for the steering angles for the rear wheels which are typically limited to 9-10 degrees for passenger cars. The operation of the adaptation loop and its design relies on the fundamental hypotheses: "For any possible values of the plant parameters there is a controller with a fixed structure and complexity such that the specified performances can be achieved with appropriate values of the controller parameters" (Landau et al., 1998). Usually the plant models and the controller are assumed linear. Hence, the task of the adaptation loop is solely to search for the good values of the controller parameters.

Thuilot et al. (1996) derived two control laws, one incorporating dynamic and time-varying feedback, and one for trajectory tracking and stabilising for a class of nonholonomic systems, which exists in mobile robots equipped with two or more steering wheels. They highlighted an additional problem presented by these systems, which is that their configuration space presents intrinsically some singular points. The control laws have to be designed in such a way that it will be guaranteed that the singularities are not met during operation of the robot. The proposed scheme for singularity avoidance can also be used in a similar manner for obstacle avoidance. The robot state vector is identified assuming pure rolling and no slip. The proposed hybrid scheme has been divided into three tasks: variable initialisation via an open-loop control law; a dynamic feedback law which is used during robot motion; a time-varying control law which is used when the robot approaches the goal position in order to stop the robot.

Hemami (1994) derived the kinematics relationship for path following for a vehicle with front and rear steering. In the study four different feedback control schemes for correcting position and orientation errors in tracking a desired path are simulated. Because the closed-loop poles do not have significant variations for the different controllers, different behaviours were not noticeable. Moreover, the study revealed that for low speed vehicles for which the dynamic effects can be neglected and the relationship of the kinematics would suffice, such ad-hoc controllers can perform as good as an optimal controller.

The modelling of a wheeled mobile robot with an arbitrary number of wheels moving as a planar rigid body has been analysed by Alexander and Maddocks (1989). The rigid robot body was used only to carry a moving co-ordinate system in a horizontal reference plane and the three wheels are presented as axle-vectors in the reference frame. For their investigation, they use a tricycle in order to limit the number of parameters and simplify the calculation of the redundant parameter in order to satisfy the rolling compatibility condition. Their first application of the developed theory assumes that all wheels undergo ideal rolling. The analysis did not involve a force balance and did not explicitly solve Newton's laws of motion. Instead, it is implicitly assumed that the wheels and the supporting plane, are sufficiently large to balance the inertial forces that arise from the accelerations. This means that the pure rolling model ceases to be valid whenever the inertial forces dominate. Since the analysis did not consider inertial forces and acceleration, the model qualifies only for low speeds. However, the low speed model may still be valid for relatively fast motions, provided either the turns are not too tight or the friction between the wheels and the surface is sufficiently large.

In an attempt to control the lateral and yaw motion of a four-wheel steered four-wheel drive vehicle an investigation was carried out by Matsumoto and Tomizuka (1990). The use of differential torque (force) between two front wheels or two rear wheels and front and rear steering angles as a second control input has been explored. Their analysis utilised a linear and a non-linear vehicle model for lateral and yaw motions and a control algorithm featuring a gain scheduling mechanism has been presented. The performance of four different input types has been compared; conventional Front two Wheel Steering (FWS), Front and Rear Independent Steering (FIRS), Front Steering angle with the Front Differential torque (FSFD), and Front Steering angle with Rear Differential torque (FSRD). The control algorithms have been implemented using the state space representation. In addition to the front steering angle $\delta_f$, and rear steering angle $\delta_r$, the differential forces between the two front wheels and the two rear wheels are used. A cornering stiffness factor is introduced, which is estimated using the recursive least square method in order to reflect the road/tyre condition.

In order to validate the proposed control methodology a laboratory vehicle was built and the algorithms were implemented using the state space representation. In their experiments, the yaw rate was measured by an angular velocity sensor while the lateral velocity was calculated from an integration of the lateral acceleration, which was detected by another accelerometer. Since accurate lateral velocity was difficult to determine, because the measurement of the acceleration could be strongly influenced by the undulation of a road surface, a high pass filter was used to reject the resulting unwanted signals. Good control performance was obtained when the lateral velocity was kept constant, but it deteriorated in cases where the lateral velocity was not constant. They concluded that other methods should be explored, when attempting to obtain more accurate measurements of the lateral velocity and lateral acceleration.

Brown and Hung (1994) slightly modified the non-linear three-degree-of-freedom mathematical vehicle model presented by Matsumoto and Tomizuka (1990) in order to develop a controller which responds to wheel slip ratios and yaw rate errors. A lookup-table was used to relate wheel slip ratios to road/tyre friction characteristics in order to calculate the vehicle's lateral and longitudinal tyre forces. The model was simulated with disturbances, which mimic icy road conditions while the vehicle's dynamics was observed. Steering angels, wheel slip angles and vehicle slip angle variances are expected to significantly affect the vehicle's dynamics, and hence does not support the use of a linearised model. This model is intended to provide the basis for the design of a fuzzy logic controller, which does not necessitate the use of a linearised model. The performance of a fuzzy logic controller was integrated with a 3-DOF vehicle model to assist in minimising a vehicle's yaw rate error, or spin, when subjected to hazardous road conditions and was evaluated by Brown and Hung (1995). The controller observed the front and rear wheel slip ratios and the yaw rate error, the difference between the commanded and the observed yaw rate. The controller output implements steering compensation commands for front and rear independent steering and minimises undesired spin caused by icy road conditions. The simulations of the steering assistance provided by monitoring the vehicle's yaw rate and road-tyre slip ratios was compared with the drivers steering wheel angle and showed that vehicle spin is considerable reduced when compared to an uncompensated vehicle.

Tso and Fung (1994) presented a more systematic approach to capturing knowledge and experience for designing linear controllers for autonomous vehicles with differentially driven wheels. First a linear state feedback controller was developed. Then retaining the basic control structure the linear controller was replaced by a fuzzy logic controller. Refinement of the rule base yielded in a non-linear controller, which enables smooth control transitions and robust properties not easily matched by a linear controller. In a similar fashion, Hessburg and Tomizuka (1995) simulate a fuzzy logic controller for lateral vehicle guidance tuned by a model reference adaptive method. The independent feedback rule base and the preview rule base used a detailed non-linear vehicle model. The goal of this control method was to adjust the parameters of the closed loop system under fuzzy control to follow a reference output such that the output of the vehicle follows a desired output. The desired output was generated by another fuzzy system. This system had as its inputs the past states of the vehicle and as the system outputs the desired output of the vehicle. The results from the simulations showed good tracking capability of the controller.

Kim (1996) assumed that all states are measurable and attempted to compensate the model uncertainties of a vehicle system. The control system was composed of a conventional model reference term and a compensator term. An unsupervised neural network, whose teaching signal is the error between the actual plant and the reference model, generated the compensator term. Computer simulations have been carried out which verified the effectiveness of the proposed control scheme.

Saraf and Tomizuka (1995) developed a real-time model of the lateral road/tyre interaction for the development of a vehicle lateral controller. A least square technique was used to update the values of changing parameters. They gave three reasons why current models may not be convenient to use in the context of automated highway systems. (i) The models are derived for nominal conditions, and will often not hold for real conditions on the road when rain or ice may reduce the friction coefficient. (ii) The models require slip angle and slip ratio as inputs, which are not easily measured or estimated. (iii) Empirical models usually require a large number of parameters, to be tuned for each vehicle individually, which complicates their application. For this reason they circumvent the conventional tire models and measure or estimate the overall lateral force and moment exerted on the entire vehicle and develop relationships between these forces, the steering angle of the front tyres and tracking errors. In their model the tracking error is known as it is measured to an external reference. By lumping the forces generated on the four tyres, it has been possible to derive a simple relationship between the overall forces on the vehicle and measurable quantities. The derived equations did not provide the tyre forces, but could be seen as an estimator of the road tyre interaction.

Taheri and Law (1990) introduced a new slip control braking system that maintained pre-specified longitudinal slip values for the front and rear axle based on a sliding mode controller. The desired slip values were gain scheduled as a function of the front wheel steering angle, whereby the controller utilised the longitudinal velocity components, longitudinal acceleration, wheel angular velocity and yaw rate. The gain scheduled four wheel steering system exhibited significantly improved stability and directional control compared to a conventional ABS systems and four wheel steering in combined severe steering and braking manoeuvres.

By means of computer simulations Boyden and Velinsky (1994) attempted to show the importance and significance of dynamic modelling of wheeled mobile robots for differentially and conventionally steered wheeled mobile robots by comparing different models. A common assumption of simple kinematics models is that no tyre slippage occurs so the motion of the wheeled mobile robot can be described using simple rigid body kinematics. This simple kinematics model was compared with a three degree-of-freedom dynamic model including a simple tyre model and a dynamic model including an improved tyre model. It is necessary for dynamic models to find the longitudinal and lateral forces acting on each wheel using appropriate tyre representation. The simplified dynamic model used a linear non-dynamic tyre model, which assumes zero longitudinal friction and the lateral tyre force was thought to be proportional to the side slip angle. For the improved dynamic model a more comprehensive tyre model utilising the friction cycle concept was employed. The comparison of the dynamic models unveiled that simple kinematic models may have only limited use in high load applications. Reasonable results could be obtained from the simple dynamic model. However, despite the use of the friction cycle concept, the description of the vehicle motion was not significantly different when a safety margin from the saturation of the friction forces was maintained.

Feng et al. (1993) introduced the concept of reducing the most significant error as the design objective for their controller implementation. They noted that cross-coupling the motion controllers can directly minimise the errors resulting from unsymmetrical loading and high friction in the drive mechanism of a differentially driven mobile robot. The errors are predominant when the robot is to follow curved paths. The evaluation of the controller performance showed a substantially smaller position and orientation error than conventional methods, which is mainly attributable to minimising the differential error rather than minimising separately the error in each drive loop. Watanabe et al. (1996) applied a Fuzzy-Gaussian Neural Network (FGNN) in an control application to track the speed and the azimuth angle of a mobile robot with two independently driven wheels. The effectiveness of the proposed two-input two-output control method was illustrated by performing simulation of a circular and square trajectory tracking control.

Performance evaluation of new control methods often use differentially driven mobile robots mainly to limit the number of controllable parameters and hence the complexity of the evaluation. For a four wheel steered vehicle there are eight controllable parameters (i.e. four wheel velocities and four wheel steering angles), but only three of the eight parameters can be treated as independent. The remaining five have to be determined so to comply with the rolling compatibility condition. Thus by using two imaginary wheels the number of equations required to describe a four wheeled system can be reduced Burke (1994). This approach was used to describe the kinematics of a vehicle and the inverse kinematics was calculated based on the motion of the body and the wheel positions using the state space representation. A precondition for the inverse kinematics model was that the vehicle must be in motion otherwise the equations would become unstable.

Huh et al. (1999) proposed an active steering control method for driving vehicles on slippery road conditions. In the proposed control method, the lateral forces acting on the steering wheels are estimated and compared with the reference values and the differences compensated by the active steering system. A fuzzy logic controller was designed and its performance evaluated via simulations. The method can be realised for a steer-by-wire concept and exhibits scope for an active safety technology. The lateral tyre force, an essential variable in the concept, was estimated based on the vehicles dynamics model. The new approach assumed that the lateral tyre force during steering is closely related to the vehicle's roll angle and can be calculated from the vertical load variation and the average lateral force at the front and rear tyres. A simulation tool was employed to generate the reference tyre forces present when the vehicle is steered on a normal road. A fuzzy logic controller was used to cope with the non-linear characteristics of the lateral tyre forces. An estimator block obtains the estimation of the lateral tyre forces as well as the vehicular motion states, which represents an indirect adaptive control scheme.

Adachi et al. presented a four-wheel-steering system featuring front-end path memorising, whereby the rear wheel steering angle is controlled such that the rear end does not swing outside the memorised path of the front end. The control target was that a vehicle with a long wheel-base demonstrates the same degree of manoeuvrability as a compact passenger car. The results demonstrated that the method enables four-wheel steered vehicle to achieve rear-end swing out comparable to that of a front-wheel steered vehicle.

Gartner and Astolfi (1998) stated that the main merit of fuzzy control is its effectiveness since it is possible to solve difficult control problems with limited effort, either in terms of time or money. However, the main drawback of fuzzy logic is that it is in general not possible to obtain a-priori knowledge on the performance of a fuzzy controller although fuzzy control has shown to outperform conventional control in several applications.

Brown and Harris (1991) addressed the problem of adaptively controlling an autonomous vehicle. The vehicle was governed by complex, non-linear functions of many parameters, some of which are time varying e.g. vehicle mass and operating in a dynamic environment e.g. varying tyre/road friction coefficients. An associative memory neuronal network can be implemented in real-time and showed fast initial convergence and long term global convergence and ensures stable learning. However, for the on-line learning knowledge about the forward inverse plant Jacobian is necessary. Gaudiano et al. (1996) presented a neural network mobile robot controller which autonomously learns the forward and inverse odometry of a differential drive robot through an unsupervised learning cycle. They stated that the relative success of classical control theory has made it possible for many of these research endeavours to focus on higher-level tasks, while ignoring the details of the low-level control. However, the process of actually implementing a model in hardware has been found challenging by those who have tried. They identify three fundamental problems in the control of mobile robots: incorporating information from qualitatively different sensors; coping gracefully with noise; and address uncertainty arising from imprecise knowledge of the robot's location over time.

Wyatt (1996) investigated reinforcement learning methods and the applicability to robots. This trial and error approach attempted to adjust the decisions based on positive and negative feedback. Different to supervised learning procedures the error signal from reinforcement learning does not indicate which behaviour is correct, but allows comparing the behaviour relative to others. While in theory an appropriate function exists for all tasks, it is typically hard to find such a function. If guaranteed convergence is required the system has to try all actions in all states at least once, which is one of the main reasons for the slow convergence.

In practice, additional difficulties may be encountered if accurate positioning of the vehicle has to be achieved. A threshold moment overcoming the initial friction in bearings and drive motors has to be attained before any movement of the vehicle could take place. Once the required initial resisting torque has been overcome, the excess momentum might accelerate the vehicle and the resulting overshoot could introduce some jigging (cog) motion. The application of neural networks for micro-manoeuvring of DC motors was investigated by Tzes et al. (1995). A neural network was used to model the combined effects of the following contributing factors: friction, resistance and temperature in the motor. The neural network was trained using a sign gradient descent algorithm and the input vector consisted of the time history of the motor shaft angular velocity. The results showed that the proposed algorithm can provide an effective means of controlling a DC motor. This may provide a useful approach for the position control of an autonomous vehicle as the parameters (e.g. winding resistance, friction of the DC-motor bearings, friction of transmission gearbox and wheel bearings) would need to be obtained experimentally and updated as operating conditions change. Operating conditions may change due to variations of the operating temperature and wear and tear of vehicle components.

## 2.6 Summary Comments

The literature review has highlighted a number of aspects, which need to be taken into account when attempting to develop and implement a motion control system for an autonomous vehicle. The insight thus gained will enable a better understanding of the governing parameters and help to avoid common pitfalls. Whilst the literature offers techniques for guiding vehicles using external references, for improved flexibility the autonomous vehicles should be able to navigate between despatch locations without the need to rely on external guidance systems.

Most experimental vehicles utilise two differentially driven wheels so to limit the number of parameters. For exploratory vehicles, ultrasonic and infrared sensors have often been utilised to map the environment. In order to cope with uncertainty in these systems fuzzy logic and neuronal networks have been employed. However, the complexity of these techniques may render this approach inefficient for shop floor applications where any uncertainty must be avoided. For docking operations, infrared beacons may be used to assist a free ranging vehicle in order to correct cumulated position errors when moving towards the target position. It has been reported that accelerometer based motion measurements deteriorate at low speeds and varying velocities. During vehicle operation, errors and the implications of the dynamic effects may be kept at a minimum if the velocity corresponds to the value for which the trajectory is generated. It may also be permissible to combine the forces of all four wheels when estimating the road tyre interaction. Nonetheless, the majority of controllers described in the literature assume a driftless system. Furthermore, as long as the desired path is being planned well for a given velocity, it may be permissible to neglect the inertial forces and acceleration in the motion controller.

# 3 Adaptive Motion Control for an Autonomous Vehicle

A good understanding of the underlying principles governing the kinematics of an autonomous vehicle is essential for the design of a stable and robust motion control system. Several attempts have been made to develop mathematical models for road vehicles. While a number of models provide an adequate representation of the vehicle kinematics, they lack in their current form a proper representation of the road/tyre interface characteristics in the equations of motion. This is attributable to two factors: the lack of a good understanding of the road/tyre interaction and insufficient experimental investigation of the subject concerned. Furthermore, the existing models do not take full account of the advanced manoeuvrability expected from autonomous vehicles designed to operate on the shop floor. In addition, a motion control system will need to have access to the electronic subsystem which controls the traction and steering actuators so that the generated motion commands can be implement in an effective manner. Signals from sensors will need to be converted, so that they can readily be used by the controller for calculating and requesting corrective actions should any deviation from the intended path occur. In addition, the mechanical subsystem needs to include a power supply, the drive and steering actuators and the load carrying structure to support the payload.

## 3.1 Kinematics Models for Motion Control

Attempts have been made by researchers to improve driving stability, particularly during cornering, which have led to the development and implementation of Four Wheel Steering (4WS), Four Wheel Drive (4WD), Anti-lock Brake Systems (ABS) and Vehicle Dynamics Control (VDC) for passenger cars. There are a number of published models for motion control covering both three-wheel and four-wheel vehicles (Alexander and Maddocks, 1989, Halanay et al., 1994), but the model presented by Matsumoto and Tomizuka (1990) is considered to be one of the most comprehensives. In deriving the equations of motion for the vehicle, the model considers the forces acting on a vehicle resulting from the longitudinal, lateral and yaw motions. Because the lateral motion and the yaw rate of the vehicle cannot be controlled using only one steering wheel input, they used the differential torque between the wheels as an additional input to the measured lateral, longitudinal and yaw acceleration. The controlled outputs from the model are the wheel steering angles and driving torque. The mathematical model has been evaluated on an experimental four wheel drive model vehicle.

Practical consideration of Matsumoto and Tomizuka's approach suggests three main problems. First, the torque control of the individual dc drive motors was achieved by adjusting the supply current. It might be a simple choice to regulate the motor torque in this way, but deducing the torque from current measurements has potential for errors. Energy losses attributable to the internal motor resistance, heating of the motor windings and transformer actions in the wiring are difficult to quantify. Second, it is difficult to obtain the lateral velocity accurately by integrating the lateral acceleration measured by an accelerometer, as the measurements can be strongly influenced by the undulations of the road surface. Third, it is difficult to measure acceleration reliably when the vehicle is cruising at low speeds, as signal/noise separation becomes small.

Road vehicles have performance characteristics significantly different to those of intelligent autonomous vehicles designed primarily for shop floor applications. Passenger cars are designed to run on highways with relatively high cruising speeds, small variations of yaw rate and lateral velocity and a large radius of curvature. In contrast, autonomous vehicles designed for the transfer of materials on the shop floor are required to operate under vastly different conditions: low cruising speeds, frequent changes of the yaw rate and lateral velocity, and small radius of curvature. In particular, space constraints can mean that tight cornering and on-the-spot turn around are a common occurrence. Tight cornering often requires that one of the wheels on the vehicle is allowed to come to a standstill while the others follow their destine radii at specific speeds. On-the-spot turn around requires the vehicle to turn around about its centre, which means that each of the four wheels must be steered at a different angle. The steering angles at individual wheels may also vary depending on the magnitude of the lateral forces to be compensated, for example to prevent the vehicle drifting out of a turn during cornering.

Literature survey shows that there is little published information available on friction characteristics in a general and the lateral and longitudinal friction coefficients in particular. It is a common practice to assume a constant friction coefficient when determining the longitudinal tyre forces, while the lateral tyre forces are often assumed negligible. Furthermore, no account is taken of any changes in the surface characteristics (e.g. uneven floor surface) encountered during operation.

In view of the experience gained from reviewing the published work, an innovative approach is called for when determining the road/tyre interaction characteristics. Unlike previous attempts that involved the use of either accelerometers or measurements of forces and torque, the new approach seeks to deduce the motion of a vehicle from a set of geometric relations. Two assumptions need to be made: Firstly, the wheels fixed on the vehicle can be brought into known angular positions relative to a reference, the longitudinal vehicle axis (see also section 6.1). Secondly, the angular displacement of each wheel rotation can be monitored with sufficient accuracy. The proposed geometry based approach also requires that all four wheels are in permanent contact with the surface in order to enable the deduction of the wheel displacements.

The motion control system utilises an internal reference model essentially the ideal rolling model and correlates the captured wheel parameters in order to deduce the road/tyre interface characteristics. Satisfying the compatibility condition during operation at all times requires a smooth transition of traction and steering commands between the different steering modes. The fully integrated motion control system also needs to accommodate practical design constraints such as maximum steering lock and maximum speeds.

For accurate path tracking, the data acquisition system will be required to reliably deduce not only the longitudinal but also the lateral motion of the vehicle with a high degree of accuracy, in particular during cornering. The compatibility of the individual components for the motion control system is paramount for the successful implementation of the fully integrated system.

### 3.2 Overview of the control methodology

The proposed control methodology presented in the following section is based on a hybrid control structure (Cameron and Probert, 1993, see also section 2.5) combining the positive features of both, the behavioural and functional decomposition schemes of a generally hierarchical controller (Figure 5).

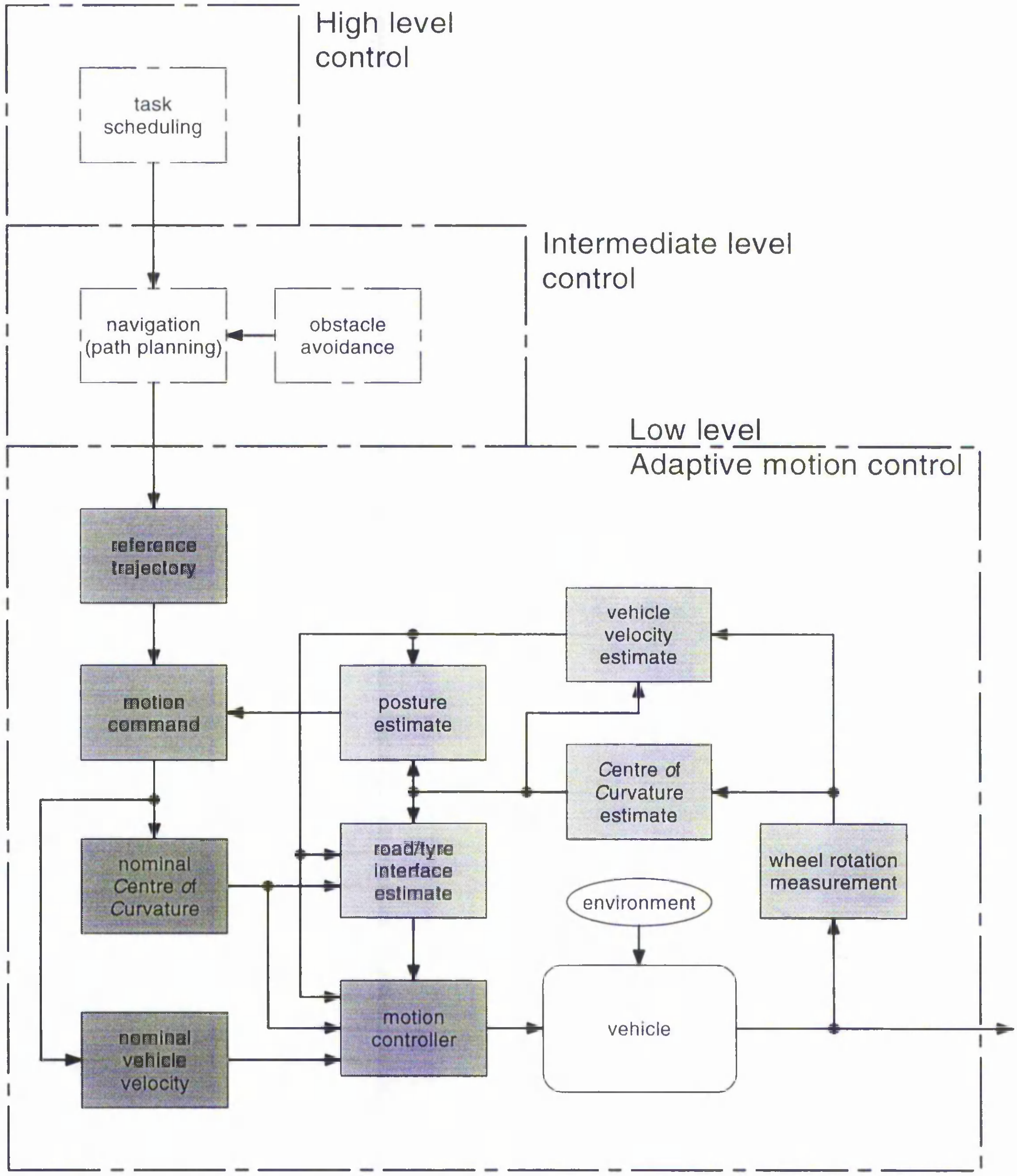


**Figure 5** Overview of the hierarchical adaptive motion control strategy

The current investigation focuses on the low-level aspect of the motion controller since task scheduling and navigation, that is path planning and obstacle avoidance, is considered as part of the higher control levels. This is because navigation can only commence when a task has been scheduled for a particular vehicle. On the other hand, task scheduling may require information about the current vehicle position from the navigation system so to assign a task with an optimised route to the vehicle. On the navigation and obstacle avoidance level, a close interaction between the two components is required so that diversions and path changes can be incorporated when generating or modifying the desired path for example to avoid an obstacle. Only when an obstacle free path has been successfully planned traction and steering commands can be determined and issued to the motion controller and the respective actuators.

For ease of implementation of the controller and to simplify the data analysis, only circular paths are considered in this investigation. Nonetheless, a very large radius may adequately approximate the straight-line movement of a vehicle operating on the shop floor. When generating a reference trajectory two main parameters need to be taken into account: the expected velocity of the vehicle and the location of the centre of curvature of the vehicle. The velocity is limited mainly by the anticipated adhesion potential particularly during tight corners, allowing sufficient margin to accommodate for variations in the available friction. The second factor is the dynamic envelope of the steering actuators. It is crucial that during manoeuvring all steering angles are adjusted without causing any mismatch, otherwise wheel shimmy may occur and the friction available for turning the vehicle may be reduced unnecessarily. It is therefore important to allow sufficient margin for the steering dynamics so that further adjustment can be made should it become necessary to compensate undesired lateral motion of the vehicle for example during cornering.

It is anticipated to control primarily the lateral position of the vehicle i.e. the steering angles. This is because moderate deviations from the nominal velocity may be acceptable since the time for transferring the material is not considered critical, however maintaining sufficient clearance between the vehicle and the environmental boundaries is crucial.

During material transfer, corrective motion commands are generated from the difference between the posture estimate and the reference trajectory, in essence a comparison of the estimated vehicle position with the desired position. Then the current nominal *C*entre *o*f *C*urvature (*CoC*) and the nominal vehicle velocity $v_{cn}$ are commanded to the motion controller. The controller adjusts the steering angles according to the current road/tyre interface estimate and sets the traction commands according to the vehicle velocity estimate.

Once the vehicle moves the angular displacement of each wheel can be measured and hence allows the deduction of information about the vehicle's motion. Since the vehicle moves in a horizontal plane the displacement of three wheels with known distances between the wheels can be used to estimate the *CoC* of the vehicle. Combining the wheel rotation measurement with the known *CoC* allows the velocity of the vehicle centre or more specifically the yaw rate $\psi$ to be estimated. Both the vehicle velocity estimate and the *CoC* estimated can be tracked to allow estimation of the vehicle's posture from a known reference position, for example the last docking station of a machining centre. The estimated posture is then compared with the nominal posture in relation to the reference trajectory and the next motion command generated accordingly.

Comparing the estimated *CoC* with the nominal *CoC* at a given velocity enables information of the road/tyre interface characteristics to be deduced. The motion controller can then incorporate the information about the road characteristics when calculating the next steering command. Hence knowledge of the road/tyre interface characteristics enables corrections of the steering commands prior to implementation and hence helps to minimise deviations from the reference trajectory. Because the road/tyre interface characteristic is estimated every time the control loop is executed, localised changes in the interface characteristics can be included in the motion command generation. As Landau et al. (1998) pointed out, an adaptation loop searches for possible values of the plant parameters so that the specified performance can be achieved. To this extent, the proposed approach forms an adaptive motion controller for autonomous vehicles

## 3.3 Components for the motion control system

When designing a motion control system for an autonomous vehicle there are two possible strategies:

i) Selection of simple hardware components that require the use of sophisticated mathematical analysis and software solutions or

ii) Development of proprietary hardware that includes advanced functionality for motion control implementation and hence does not necessitate complex post-processing solutions.

When carefully designed, it is possible to implement some of the functionality in hardware and hence limits the amount of post-processing of the captured data, which in turn allows the selection of a less powerful processor. For a satisfactory autonomous vehicle implementation, the following three building blocks need to be integrated so as to form a complete system (Figure 6). The motion control system block incorporates position tracking utilising only the rotation and angular displacement of the wheels.

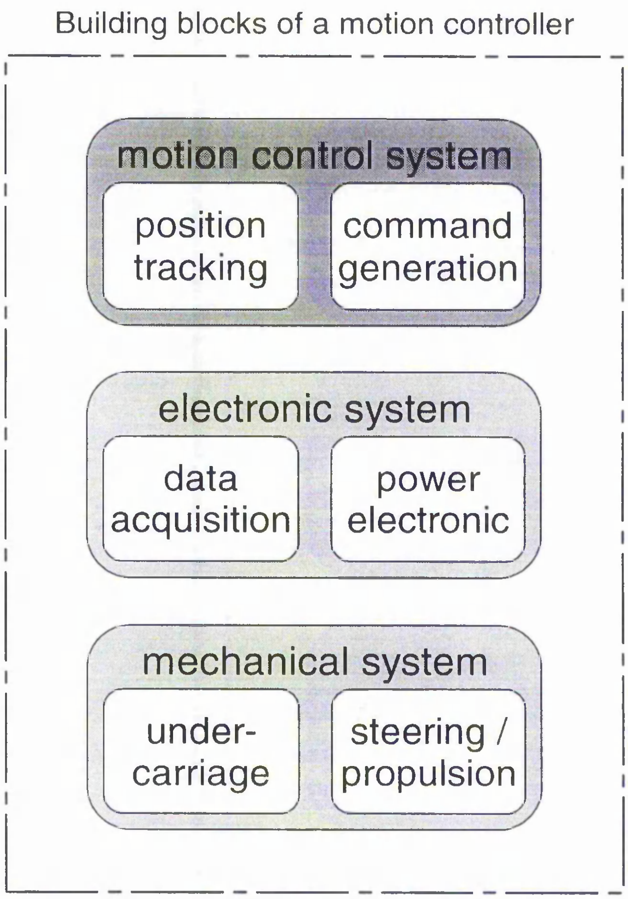


**Figure 6** Building blocks for an autonomous vehicle motion controller

The *command generation* module is responsible for generating appropriate steering and traction commands by considering the nominal and the estimated vehicle states to minimise any deviation from the reference trajectory. The second building block is an electronic system, which incorporates a data acquisition system using dedicated hardware to efficiently deal with tedious and time-consuming data processing tasks. This approach helps to ensure a smooth operation of the autonomous vehicle particularly if only limited processing power is available. In order to minimise inaccuracies likely to occur from timing delays between the samples, the data should be acquired in a parallel manner. Similarly, the steering commands need to be implemented in a co-ordinated manner so as to eliminate any synchronism errors. A third building block is the mechanical system comprising an undercarriage, the steering and propulsion motors and gear and transmission systems.

The adaptive motion control strategy relies on accurate measurements of the displacements of the wheels. If the vehicle displacement is deduced from the wheel rotation, the governing factor is the effective wheel radius, which may vary depending on the payload. This effect is more dominant when pneumatic tyres are used. However, correction factors may be introduced to minimise the adverse effects. Moderate undulations of the floor are expected to have a minimal impact on the overall accuracy of the motion control system, provided the wheels maintain ground contact at all times.

In order to measure the angular displacements of the wheels with reasonable accuracy, optical shaft encoders will be used to sample instantaneous angular positions of all four wheels. Knowledge of the number of completed wheel rotations allows tracking of the vehicle's path over a relatively long distance. However, the expected angular displacements of the wheels are expected to be small and hence encoders with high resolution and high accuracy are required. Any time delay between the samples taken may contribute to further inaccuracies.

The electronic interfaces to be developed for data acquisition and for driving the power electronic modules of the steering and drive motors should be based on a parallel sampling - parallel actuating approach. With this approach, any errors attributable to taking samples at different times would be negligibly small.

Furthermore, measurements are to be taken after the vehicle has reached steady-state condition and just before new actions are implemented in the new control cycle.

For the proposed adaptive motion control strategy to work properly, a seamless integration of the individual components is needed so that they are optimised for their dedicated tasks as well as providing full compatibility with the other components. The proposed strategy also combines the road/tyre interaction of the individual wheels in order to determine the overall vehicle motion. This approach is similar to that used by Saraf and Tomizuka (1995) where the individual forces acting on the wheels are lumped together to derive only the overall forces acting on the vehicle. This was necessary because it was not possible to measure the individual force components or their interaction.

### 3.4 Summary Comments

The proposed adaptive motion control strategy utilises the angular displacements and the rotations of the wheels to estimate the longitudinal and lateral motions of the vehicle. The motion controller consists of three building blocks: the motion control system comprising the position tracking algorithm and the motion command generation; the electronic system comprising a high precision high resolution data acquisition system and proprietary power electronics; the mechanical system which includes the steering and traction actuators and an undercarriage enabling permanent contact of the wheels with the floor. The components have to be designed not only to perform optimally in their specific functions but also to ensure full compatibility within the integrated system. The synergy of all the individual components is essential for the adaptive motion control system to function properly.

# 4 Mechanical Design of the Experimental Vehicle

In the laboratory, vehicles developed to test vehicle navigation systems often utilise two differentially driven wheels and one, two or more castor wheels to provide stability. For a differentially driven vehicle turning the vehicle around a corner is achieved by adjusting the differential speed between the two driven wheels. Consequently, only the two rotations of the wheels need to be controlled and hence the analysis can be simplified significantly. Industrial applications may alternatively use one driven steering wheel and two passive wheels mounted on a fixed axle. Combining traction and steering in one wheel enables a simple control algorithm to be employed to correct any deviations from the intended guidance path. In this case, only one steering angle and one wheel velocity need to be controlled. The steering angles of the other wheels are fixed and the velocities follow an explicit geometric relationship. However, the generation of appropriate paths, particularly for docking operation, may prove more demanding. Additional wheels can increase the static stability of the vehicle for both operational consideration and payload safety. For the proposed method of controlling the motion of the vehicle, it is essential that all the wheels must remain in permanent contact with the floor and appropriate means have to be provided for. Furthermore, the undercarriage of an experimental vehicle needs to accommodate all the auxiliary components required for the operation of the vehicle, so as to provide a complete test platform for deducing the road/tyre interface characteristics and to experimentally validate the proposed adaptive motion control system.

## 4.1 Stability of the autonomous vehicle

The static stability is an important factor, which needs to be taken into account when designing a mobile robot. This is because the mass of the payload may at times exceed that of the vehicle by a factor of four or five. Thus the area within which the vertical projection of the centre of gravity is allowed to fall must be carefully considered if the risk of the vehicle tipping over is to be minimised. This is particularly important during cornering with elevated payloads because the centrifugal forces may displace the effective centre of gravity and may cause the vehicle to tip over. One factor governing the tipping resistance of the vehicle is the area of stability determined by the number and position of the wheels. For example, a two wheeled vehicle is only stable as long as the centre of gravity is kept below the wheel axle (see Figure 7a) although this may exhibit undesired effects such as swinging of the payload while the vehicle is moving.

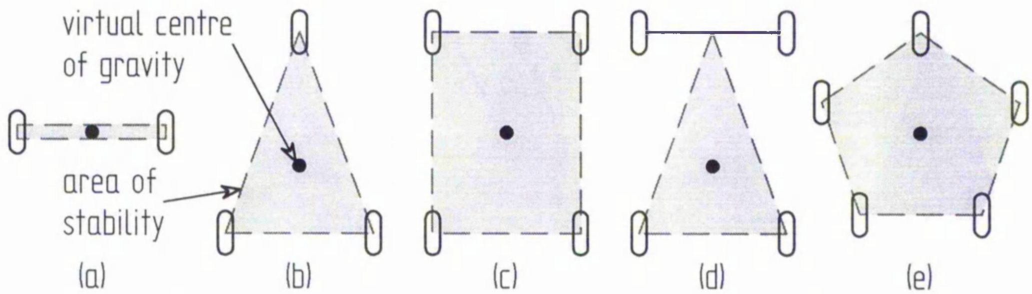


**Figure 7** Comparison of the area of stability within which the virtual centre of gravity can be located for different wheel positions and different vehicle designs

A number of experimental and medium sized industrial vehicles employ three wheels as shown in Figure 7b. Such a vehicle can always be placed so that all three wheels are in contact with the floor. However, the vertical projection from the effective centre of gravity must be kept inside the triangle formed by the position of the three wheels in order to avoid overturning of the vehicle. The stability of the vehicle can be greatly enhanced by arranging four wheels in a rectangular shape (Figure 7c). However, unless means to accommodate the undulations of an uneven surface are provided, a rigid vehicle will have a tendency to rock about a virtual line connecting the contact patches of the two supported diagonally opposite wheels. Any path tracking method relying on the rotational measurements of the wheels would be adversely affected by a loss of ground contact. Any uncertainty should be avoided for both safety and control stability reasons.

Traditionally spring suspension systems are often used to accommodate vertical wheel displacements when traversing uneven surfaces. Appropriate damping mechanisms need to be provided to reduce swaying of the vehicle brought about by the floor undulations. The wheel friction force available for traction and cornering is proportional to the normal force acting on that wheel. Ideally, the load is distributed evenly on all wheels so that the centre of gravity is located at the centre of the vehicle. Even with a suspension system, any unevenness will result in an increase of the normal forces on the elevated wheels and a reduction of the normal forces on the extended wheels. In general, passive suspension systems are susceptible to roll and pitch motions during manoeuvring, which increase the risk of vehicles with a high centre of gravity to tip over. For minimal roll and pitch movement the suspension system would need to be very stiff. However, this in turn limits the amount of available vertical wheel displacement. Active suspension systems can overcome some of the limitations highlighted but the complexity involved could not be justified for vehicles designed to operate on the shop floor.

The ideal undercarriage for automated load carriers should maintain a set loading height independent of the size of the payload. With a passive suspension system, the empty vehicle is more elevated and a payload needs to be descended from a higher elevation onto the vehicle platform during loading operations. Once the weight of the payload is fully transferred to the vehicle, the loading height will have decreased due to the compression of the springs. To unload the payload needs to be lifted until the springs are extended to match the new condition and the payload completely clears the vehicle. To overcome the drawbacks of a suspension system and to allow the wheels to follow the undulations of the floor, an arrangement comprising one fixed axle and one oscillating axle is often used. The arrangement shown in Figure 7d distributes the load to the laterally spaced wheels. Nonetheless, in terms of mechanical stability and assuming a free revolving joint, this arrangement is comparable to a vehicle with three wheels. The triangular area formed by the two fixed wheels and the pivot point of the oscillating axle marks the area of stability. One should note that the location of the centre of gravity might be displaced towards the fixed axle for equal stability margins. This, however, increases the load on the two wheels attached to the fixed axle.

An arrangement with five wheels (Figure 7e) allows a further increase of the stability area. However, this also increases the likelihood of ground contact loss when traversing uneven floors, should a rigid arrangement be used. One means to reduce the likelihood of ground contact loss is to deploy suspension systems but they have similar drawbacks as described above. Furthermore, the non-standard footprint may not be favourable for docking of the vehicle to machining centres or manufacturing cells.

Based on stability and safety considerations a four wheel vehicle design is favoured. The area of stability can be maximised and the likelihood of the vehicle tipping over during operation is reduced. However, a suspension system should be avoided, to eliminate any adverse effects of pitch and roll motion of the vehicle on the experimental results and to allow a more accurate evaluation of the motion control system. During operation of the vehicle on the shop floor, the stability margins are mainly utilised along the longitudinal and the lateral vehicle axes. This is attributable to the additional forces acting on the virtual centre of gravity of the vehicle when accelerating, decelerating or during cornering, particularly at either relatively high speeds or relatively low speeds and tight turns. Ideally, maximum safety margins should be provided in these two directions. On a vehicle with an oscillating axle, stability in the lateral direction is greatly reduced, therefore a new design is needed that does not require the use of springs for the suspension.

Imagine two axles with a wheel attached at each end of the axle (Figure 8a). One axle rotably attached to the front and one at the rear of the vehicle body. In the absence of any limiting means, the vehicle body can rotate freely along the longitudinal vehicle axis. With this arrangement the vehicle is statically stable as long as the centre of gravity is kept between the two front and two rear wheels along and below the longitudinal axis. Now consider a lever rotably attached at its midpoint to one side of the vehicle body (Figure 8b) perpendicular to the two axles, so that one end of the lever is attached to the front axle and the other to the rear axle. This lever prevents the vehicle body from rotating along the longitudinal axis and constrains the motion of the axles to a generally opposite rotation. Consequently, oscillation of the axles allows vertical displacement of the wheels so as to accommodate any unevenness of the floor. Furthermore, the vehicle is statically stable as long as the centre of gravity is kept between the two wheels on the left and the two wheels on the right and along the lateral vehicle axis.

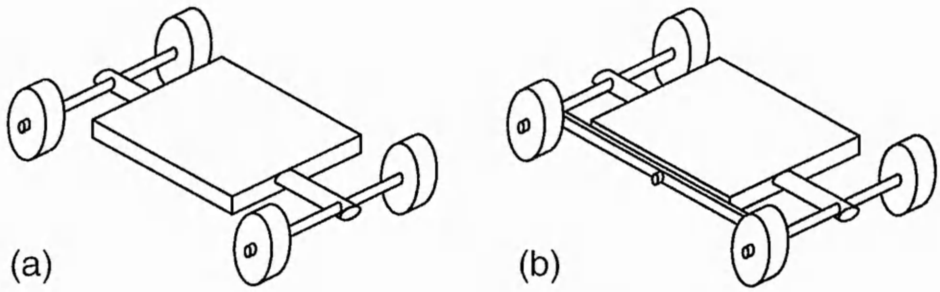


**Figure 8** a) A schematic view of the vehicle body with only front and rear axle attached b) the vehicle body with the two axles and the lever attached.

A preliminary investigation has been carried out to demonstrate the function of the mechanism. A simple implementation with two support members has been designed for a stationary platform 1 (Figure 9). Figure 9a shows the exploded view of the arrangement of the parts and their assembly with the cylindrical joints 7, 8, and the spherical joints 6. For a vehicle undercarriage, the support members 2 and 3 would be replaced by axles and would include the wheel assemblies instead of the vertically extended legs 9. On the model one support member is attached to the front of the platform 1 and one at the rear via the cylindrical joints 8 which enable the platform to rotate along the longitudinal axis. Additionally, two levers 4 and 5 are attached on both sides of the platform via the cylindrical joints 7 and via the spherical joints 6 to the two support members. This arrangement constrains the movement of the support members to generally opposite rotation with respect to the platform.

The function of the self-stabilising mechanism can be described as follows. Imagine the assembly is suspended above a surface and is then gradually lowered vertically. Assuming one leg, say leg 9 on the right hand side, touches the surface first which causes this leg to move upwards relative to the support surface. This in turn pivots the support member around the cylindrical joint 8 which causes the opposite leg to be lowered. Pivotal movement of one support member is transmitted via the levers 4 and 5, which impose opposite rotary oscillation on the other support member. This rotation results in the lowering of the adjacent leg and a simultaneous upward movement of the diagonally opposite leg. Assuming that both legs on one support member are now floor engaging, further lowering of the assembly causes a rotating motion of the whole assembly about a virtual axis joining the contacts on the legs already touching the surface. When a third leg becomes floor engaging, further lowering of the assembly causes simultaneous rotation of both support members 2 and 3 in opposite direction until the remaining leg is in floor engagement.

When all four legs are in contact with the floor, the gravitational force exerted on the platform is transmitted through the cylindrical joints to the two support members, to the legs and then to the supporting surface. While only one of the two levers 4 or 5 is required to prevent the platform from rotating freely, for symmetrical movement of both support members and a balanced load distribution, an arrangement with two levers may be beneficial.

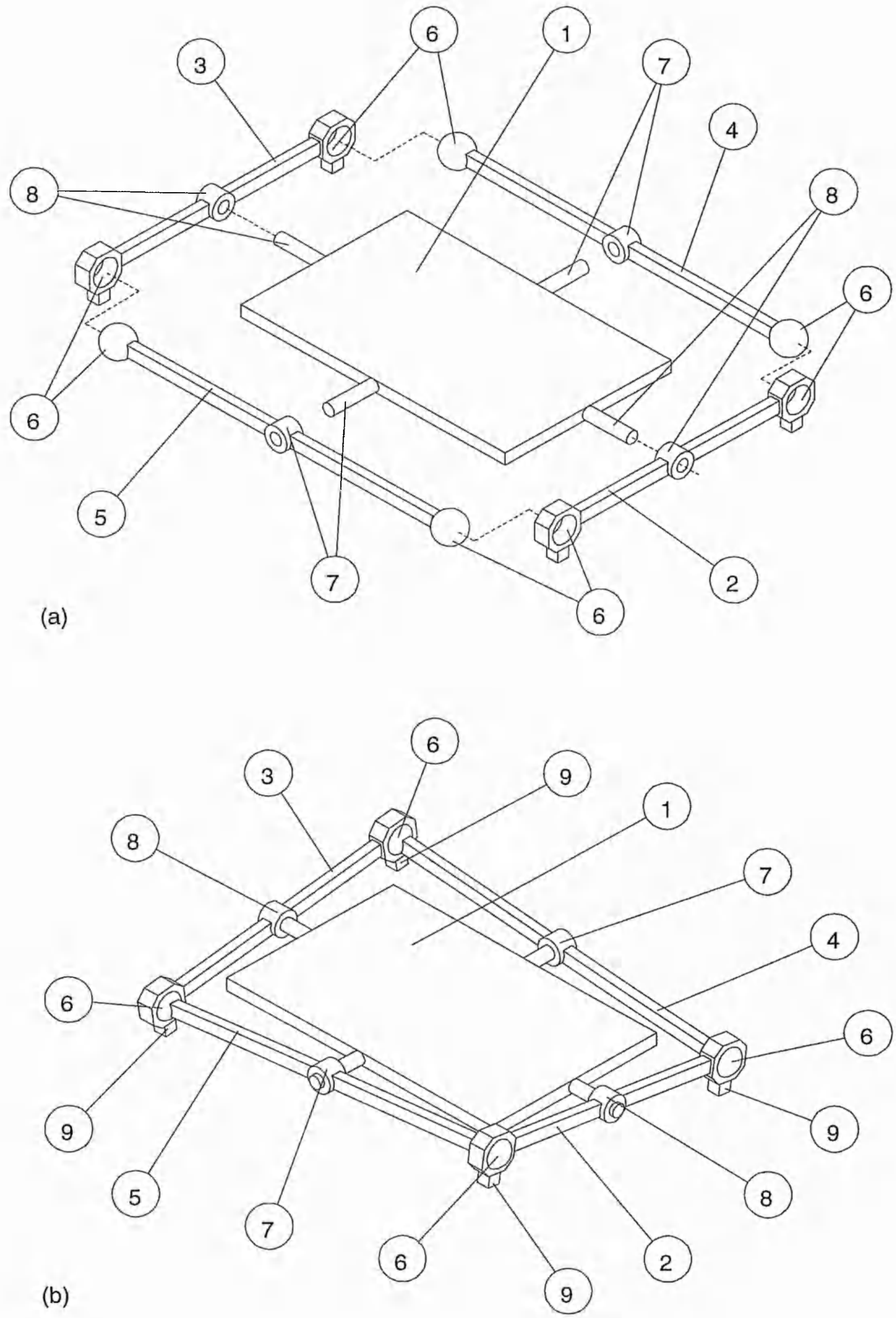


**Figure 9** a) An exploded view of the components for a model self-stabilising support assembly b) the assembled self-stabilising mechanism when placed on an uneven surface.

When in use on a flat and level surface each leg 9 will carry equal weight, assuming that the centre of gravity is located at the geometric centre of the support assembly. Unevenness of a generally level surface has only marginal influence on the distribution of the weight carried by each leg. When the assembly is subjected to unequal loading, each leg will carry its proper share of the weight. Unequal loading arises if the centre of gravity is not coincident with the geometric centre or when the assembly is positioned on a slope sufficiently steep to cause the vertical projection of the centre of gravity to shift. A comparison of the stability envelopes within which there would be no danger of overturning of the assembly for two vehicle designs are indicated by the dashed lines in Figure 10.

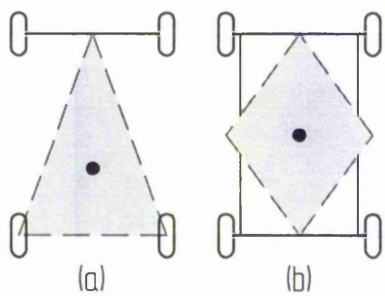


**Figure 10** Comparison of area of stability for two vehicle designs. a) Area of stability for a vehicle with an oscillating axle. b) Area of stability for the vehicle fitted with the self-stabilising mechanism.

Assuming all the joints are free rotating, it can be seen that for the vehicle fitted with the self-stabilising mechanism the stability margin is maintained in the longitudinal direction and has increased in the lateral direction. When the mass of the vehicle is taken into consideration, the stability areas expand in both the longitudinal and lateral directions. A patent application has been filed for this self-stabilising support assembly and a copy of the description can be found in Appendix A.

In an arrangement with four spherical joints, the cylindrical joints need to allow for small variations in the distances between the two levers and the two axles. The distances vary slightly when the axles and levers are rotated to compensate for surface undulations. Telescopic means in the levers may avoid such movements.

However, it may be more difficult to ensure that the two axles remain parallel, which in turn would complicate data analysis.

## 4.2 The Prototype Vehicle

### 4.2.1 Vehicle Dimensions

The dimensions of the prototype vehicle have been chosen based on the consideration of the following factors. First, the vehicle must be capable of operating within an environment where safe and easy access by workers is of prime importance. Second, the width of the vehicle needs to be such that it must pass easily through standard doors, with reasonable clearance between itself and the doorframe. This implies that the overall width of the mobile platform should not exceed 0.6 m, and the overall length of the vehicle has been chosen to give a ratio of $lenght/width = \sqrt{2}$, which is 0.85 m. However, the distances between the centres of the steering axes are specified as 0.42 m wide × 0.596 m long. This is to cater for the turning of the wheels and the clearance required for mounting the motor-gearbox combination for the drive system.

### 4.2.2 Drive Systems

The vehicle, with an unladen weight of 73 kg, has been designed to carry up to 100 kg payload, providing it is located close to the geometric centre of the vehicle. In cases where the loading condition is asymmetric or when the load is to be pushed onto the vehicle from one side, the maximum allowable load should be reduced so that the permissible wheel load is not exceeded. To minimise potential adverse effects of small obstructions or surface undulations on the driving characteristics, the diameter of the wheel should be as large as practicable in order to increase the contact patch between the tyre and the road surface. Solid rubber wheels with Ø125 mm and a nominal load capacity of 85 kg have been selected.

For shop floor applications, the speed of the vehicle should be restricted to the limit of about 5 Km/hour (3 miles/hour in the UK), corresponding to an angular velocity of 214 rev/min at a wheel. For this type of application, permanent magnet DC motors are considered the most appropriate choice for the drive system, owing to their ease of implementation, minimal maintenance, robustness and energy efficiency.

To enable the fully loaded vehicle to accelerate from stationary to the maximum speed within 3 seconds, the required power output from the drive motor is 56 W. Allowing for some losses of the motor and the gearbox a 12 V-DC motor with a rating of 60W has been selected. The attached gearbox provides an unloaded speed of 260 rev/min. To improve the compactness of the motor-gearbox combination and to minimise transmission losses, a gearbox with two drive shafts has been selected. A wheel is mounted directly onto one side of the gearbox shaft and an optical shaft encoder onto the other side so that the wheel rotation can be measured directly.

### 4.2.3 Steering System

As discussed above, the proposed new approach for determining the longitudinal and lateral motions of the vehicle requires that all four wheels can be brought into accurate angular positions in a co-ordinated manner. Servomotors are relatively expensive and the characteristics of the integrated servo controller would need to be investigated beforehand and taken into account when adjusting the steering angles. Furthermore, the relatively high currents which are required for holding the motor shaft at a constant position or moving it at very low speeds renders the servomotors inefficient. In contrast, a stepper motor exhibits a high holding and starting torque at a more efficient current to torque ratio. By applying the correct number of steps in exciting the windings in the appropriate sequence, stepper motors can be accurately positioned and easily held at one position. To double the accuracy in positioning the motor shaft and to improve the starting and stopping dynamics of a stepper motor, the excitation sequence can be altered so as to move the motor only half a step at a time. It is important to ensure that the load applied to the motor does not exceed the specified torque. Similarly, the maximum ramp-up and ramp-down speeds need to be obeyed, otherwise the motor may loose steps and a mismatch of the steering angles occurs. One way to avoid the need for implementing a feedback loop to detect such step losses is to ensure adequate torque is provided. For this reason, it has been decided to use stepper motors in combination with a gearbox. The steering action is transmitted via timing belts connected to the gearbox. This enables a further reduction of the gear ratio. De-coupling of the gearbox unit from the steering wheel also helps to ensure a smoother transition between steering directions by cancelling out peaks of high momentum, thus reducing the stress on the shaft of the stepper motor and gearbox.

Preliminary experiments have been carried out to determine the torque required for steering the wheels. It was concluded that about 5 Nm is sufficient to turn the wheels of a stationary vehicle. Once the vehicle is moving, the required torque is substantially reduced. A gearbox with a reduction ratio of 5:1 and two timing pulleys with a reduction ratio of 40:15 enable a 1.8 degree 12 volt stepper motor with 0.5 Nm output torque to produce approximately 6 Nm torque for steering the wheel. This value of torque is sufficient to turn the wheel at low speeds and to gradually increase the angular steering velocity when the vehicle is stationary. When the vehicle is moving, a higher torque margin allows faster adjustment of the steering angles. The dynamic envelope of a stepper motor limits the achievable angular acceleration and this needs to be taken into account when generating the reference trajectory. As long as there is sufficient torque margin, the steering angle can be adjusted in a co-ordinated manner by synchronously applying the appropriate step sequence to each stepper motor. There exists an explicit relationship between the steering angle and the number of steps; once the steering system has been initialised to a known state, a feedback loop for the steering controller may not be necessary.

### 4.3 Energy Storage

As autonomous vehicles are likely to be used mainly indoors, environmental issues including those related to the Health and Safety at Work need to be taken into account when considering how such a vehicle should be powered. Rechargeable batteries are commonly used as the principal power source for AGVs, in spite of their relatively low efficiency in energy conversion during charging and discharging and their relatively low energy stored to weight ratio. The capacity of the batteries needed for a vehicle to operate effectively depends on a number of factors including the type of operation, time interval between charging cycles, the power consumption of the drive and steering systems. Additionally, most controller hardware requires a minimum supply voltage of 18 V in order to provide a stable supply of +5V, −5 V and +12 V for the controller system. The response of the stepper motors can be improved by using a higher supply voltage to energise their windings initially and limiting the supply current to the nominal value by using chopper circuits. For the experimental vehicle two 12 volt batteries designed for frequent load and discharge cycles are connected in series to provide 24 volt onboard power.

## 4.4 Self-Stabilising Vehicle Undercarriage

The undercarriage of the experimental vehicle is designed to incorporate the lever mechanism described above. For this implementation, the parallel movement of a vertically extended axle is important. Therefore, a lever has been added radial to the longitudinal axis which connects the cylindrical joints of the two vehicle axles. This additional lever counteracts the excessive bending moments expected on the cylindrical joints by distributing the force to the third lever. The third lever has no impact on the kinematics of the lever assembly.

In order to understand the displacement of the undercarriage when traversing an uneven surface a mathematical model needs to be developed. This will allow the motion of any point on the vehicle to be described when the wheels are vertically displaced and the evaluation of the proposed design.

For ease of evaluation, a reference plane is defined through the centres of the spherical joints *W1*, *W2*, *W3* and *W4* (Figure 11) when the mechanism is placed on a level surface. Five points on the mechanism have been considered for the evaluation: one at each of the centres of the spherical joints and one at the geometric centre elevated above the reference plane. Assume the origin of an orthogonal co-ordinate system is fixed on the reference plane at the centre between the spherical joints. Then the vector describing the position of the centre *C* in the x, y and z direction when the heights of the spherical joints vary is given by Equation(1). The length between the spherical joints is denoted by *le* and the width by *wi*. The height of the four spherical joints with respect to the reference plane is denoted by $h_1$ to $h_4$ and the elevation *el* is the height of point C from the reference plane.

$$C_{dis} = \begin{bmatrix} \frac{1}{2}\frac{el}{le}\left(h_2 + h_3 - h_1 - h_4\right) \\ \frac{1}{2}\frac{el}{wi}\left(h_3 + h_4 - h_1 - h_2\right) \\ \frac{(h_1+h_2+h_3+h_4)}{4} + el\sqrt{1 - \frac{(h_2+h_3-h_1-h_4)^2}{4\,le^2} - \frac{(h_3+h_4-h_1-h_2)^2}{4\,wi^2}} \end{bmatrix} \tag{1}$$

Since the cylindrical joints are mounted midway, the lever halves the x and y components and considers the elevation to length ratio and the elevation to width ratio respectively together with the height differences. The z component is mainly governed by the quarter of the height difference and small vertical variations due to the tilting of the platform.

The vectors describing the x, y and z position for the four spherical joints is given by Equation(2), where the height equals the z component. The variation of the x and y components is due to the rotation of the axles and the levers which is governed by the height variation.

$$W2_{dis} = \begin{bmatrix} -\sqrt{\left(\frac{le}{2}\right)^2 - \left(\frac{h_2-h_1}{2}\right)^2} \\ \sqrt{\left(\frac{wi}{2}\right)^2 - \left(\frac{h_2-h_3}{2}\right)^2} \\ h_2 \end{bmatrix} \qquad W1_{dis} = \begin{bmatrix} \sqrt{\left(\frac{le}{2}\right)^2 - \left(\frac{h_2-h_1}{2}\right)^2} \\ \sqrt{\left(\frac{wi}{2}\right)^2 - \left(\frac{h_4-h_1}{2}\right)^2} \\ h_1 \end{bmatrix}$$

$$W3_{dis} = \begin{bmatrix} -\sqrt{\left(\frac{le}{2}\right)^2 - \left(\frac{h_4-h_3}{2}\right)^2} \\ -\sqrt{\left(\frac{wi}{2}\right)^2 - \left(\frac{h_3-h_2}{2}\right)^2} \\ h_3 \end{bmatrix} \qquad W4_{dis} = \begin{bmatrix} \sqrt{\left(\frac{le}{2}\right)^2 - \left(\frac{h_4-h_3}{2}\right)^2} \\ -\sqrt{\left(\frac{wi}{2}\right)^2 - \left(\frac{h_4-h_1}{2}\right)^2} \\ h_4 \end{bmatrix} \tag{2}$$

An example of the motion of the five points on the mechanism is shown in Figure 11 using dimensions taken from the experimental vehicle. The points indicate the 3-D positions in response to a lifting of the spherical joint *W1* in discrete steps. It can be seen that the vertical displacement of the centre from position *cog* to *D* is approximately a quarter of the displacement of the spherical joint *W1*. The other joints *W2* and *W3* are slightly displaced towards the joint *W1*.

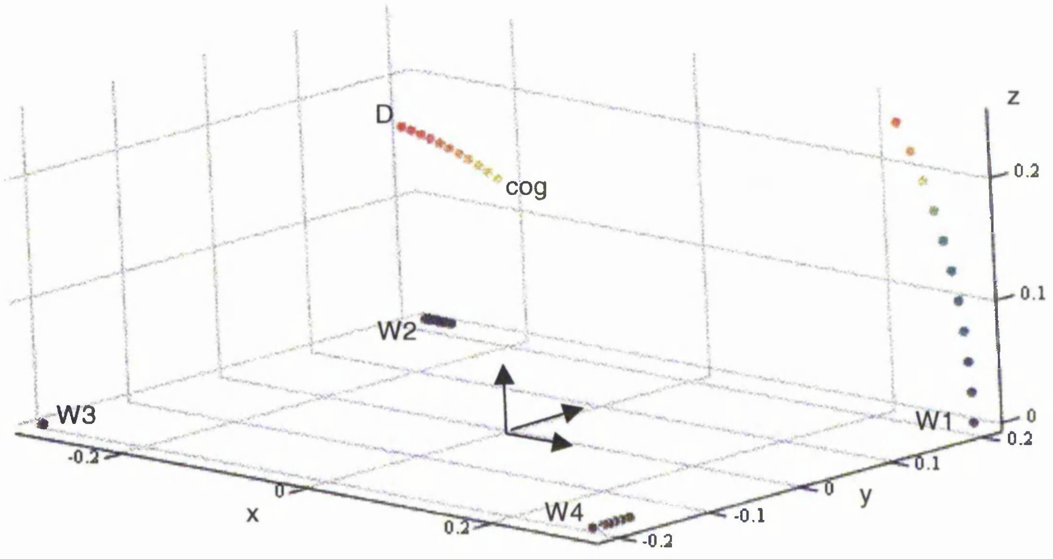


**Figure 11** Motion of the cylindrical joints and the elevated centre of the self-stabilising mechanism when one wheel is elevated.

Figure 12 shows a comparison of the displacements of the points on the mechanism as a response to lifting point *W1*. Is can be seen that the displacement in z and y direction is negligible up to a vertical displacement of *W1* to approximately 25 mm. This is an important feature, because the variation in the y direction could have adverse implications when utilising the angular displacement of the wheels to position tracking. Variations in the y direction would cause additional angular wheel displacements resulting from the vertical wheel displacement.

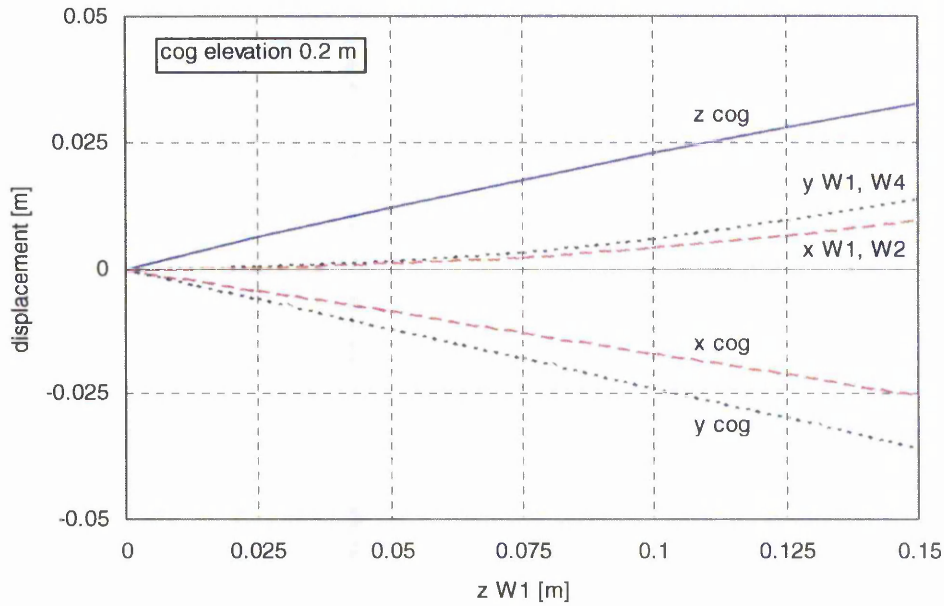


**Figure 12** A comparison of the displacement of the elevated centre of gravity and the four wheels.

The experimental vehicle has been designed using the CAD software package ProEngineer to produce parts and assembly drawings. Figure 13 shows the complete assembly drawing of the experimental vehicle with the controller housing mounted on top of the undercarriage.

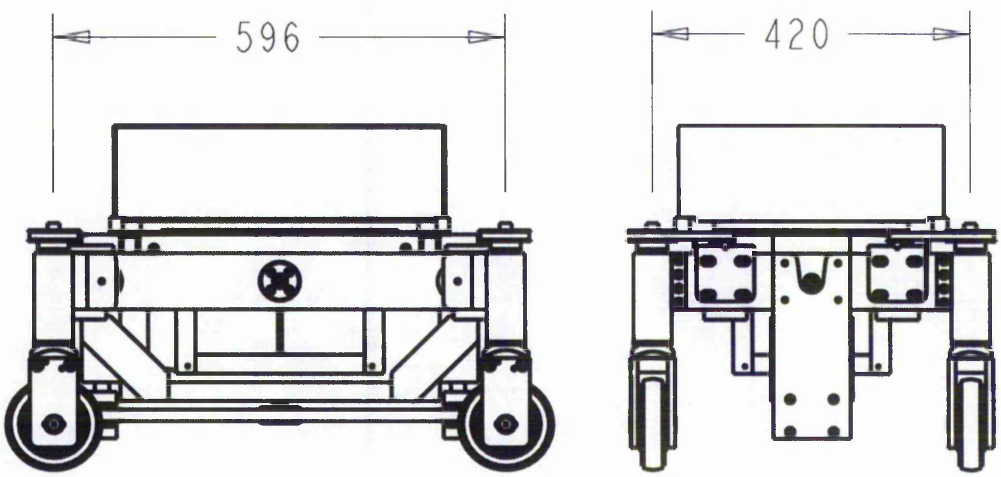


**Figure 13** Assembly drawing of the experimental vehicle with three levers and the computer system on top of the undercarriage.

The parts for the experimental vehicle have been manufactured from aluminium. Machining has been carried out mainly on a CNC milling machine using the data extracted from the CAD model. The parts of the vehicle have been assembled and Figure 15 shows the photo-rendered view of the experimental vehicle in its nominal and Figure 15 in its tilted position.

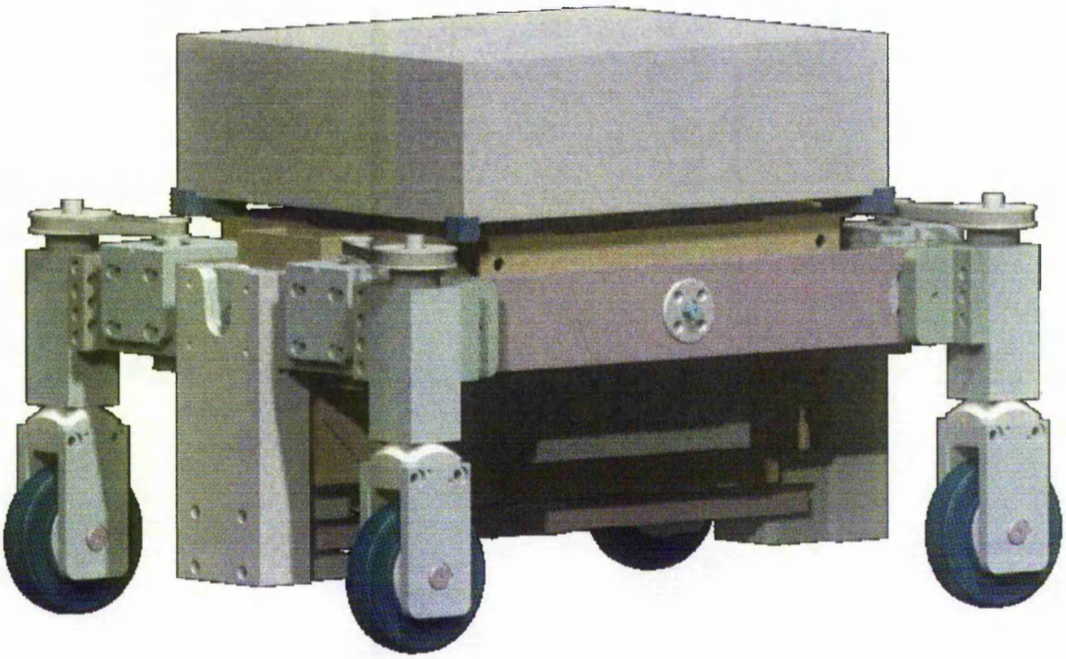

**Figure 14** Photo rendered view of the full CAD model in its nominal configuration.

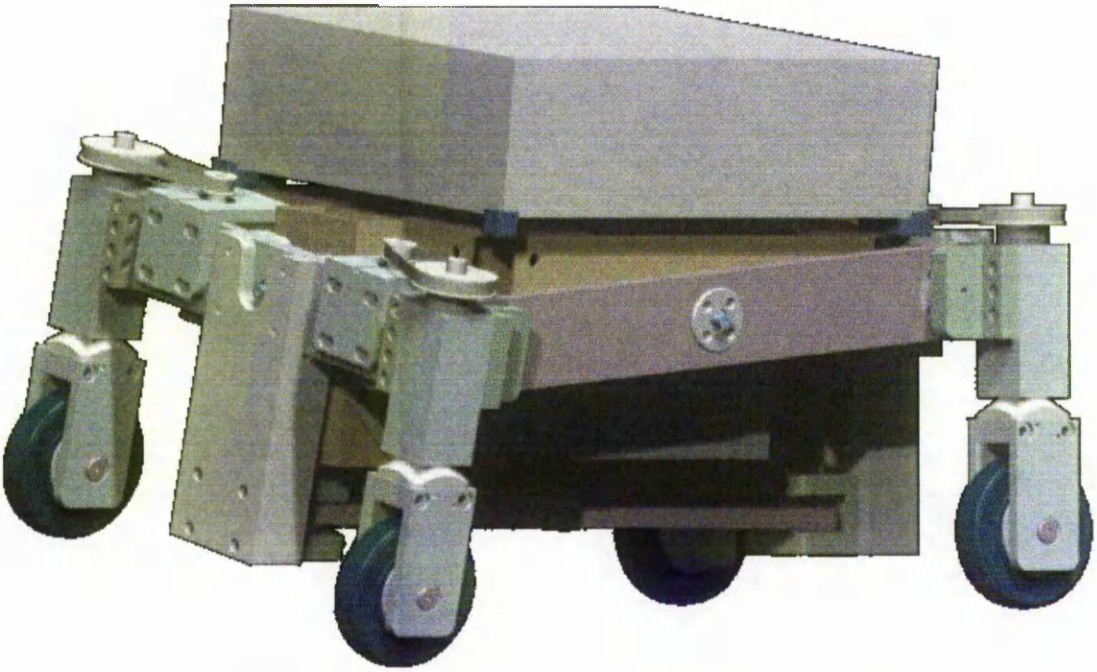

**Figure 15** Photo rendered view of the full CAD model when one wheel is lifted up.

### 4.5 Validation of the Self-Stabilising Undercarriage

In order to assess the performance of the undercarriage when traversing a hump it, was considered appropriate to simulate the motion of the vehicle using a dynamic modelling software. LS-Dyna is an explicit dynamic simulation package capable of iterating the rigid body motion and forces exerted on the vehicle. A model using simple beams for the axles and levers has been constructed. In order to evaluate the wheel loads it is preferred to construct the tyres and the fully constrained road surface with shell elements. Shell elements are better suited for calculating the road/tyre contact forces.

In the simulation, the vehicle was set to traverse a hump with a height of 55 mm in a straight line at a constant velocity. Only the vertical motion was considered important hence a straight line motion would suffice. The simulation uses a constant friction coefficient of 0.8 for the road/tyre interface so that the wheel rotation can be incorporated in the analysis. The motion is prescribed by a constant velocity applied to the rear axle of the vehicle. For the simulation, the dimensions of the experimental vehicle have been used.

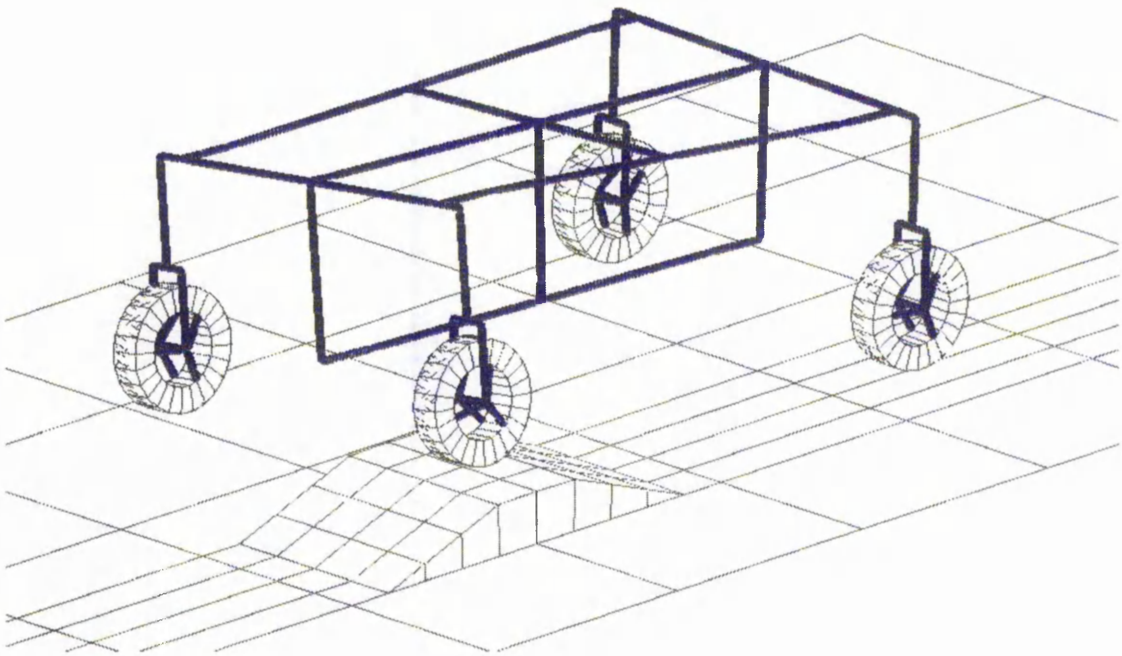

**Figure 16** Snapshot of the finite element simulation of the self-stabilising undercarriage while traversing a hump.

Several simulation runs have been carried out to compare the self-stabilising undercarriage with a rigid undercarriage. The rigid set-up is achieved by simply disabling the spherical and cylindrical joints of the lever mechanism. Two parameters are of particular interest: the vertical displacement of the vehicle centre and the forces acting on the wheels while traversing a hump.

Figure 17 shows the vertical displacement of the centre of gravity of the vehicle when the front wheel traverses the hump and then followed by the rear wheel. The broken line shows the position of the hump. It can be seen that with the rigid undercarriage the centre has been displaced by approximately half of the hump's height, while with the self-stabilising undercarriage, the displacement has been reduced to a quarter of the hump's height. When traversing the hump at velocities above 0.25 m/s the wheel bounced and lost surface contact.

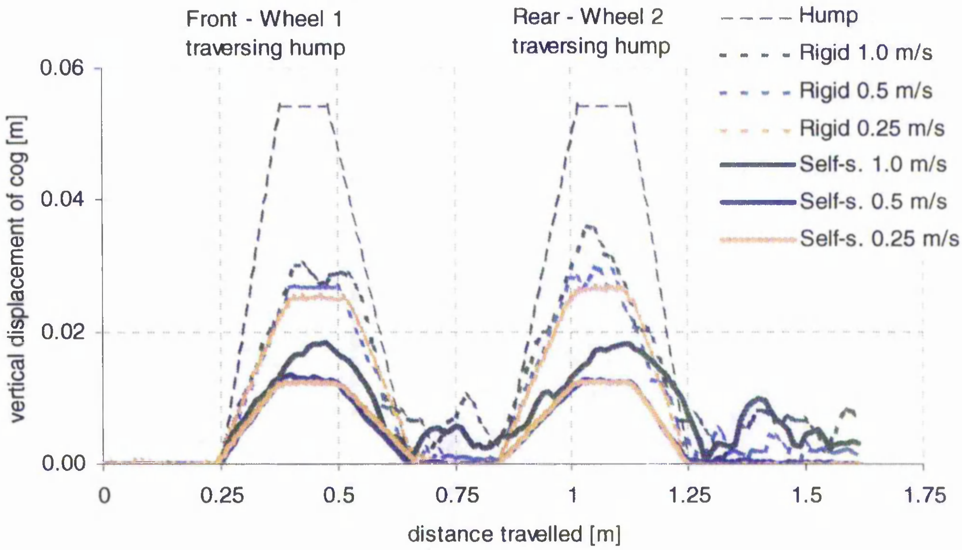


**Figure 17** Comparison of the simulated vertical displacement of the centre of gravity (cog) for the self-stabilising assembly and the rigid platform when traversing a hump.

A comparison of the vertical contact forces at the road/tyre interface for all four wheels is shown in Figure 18 when the vehicle traverses the hump with a velocity of 0.25 m/s. The results show that the forces associated with the rigid undercarriage vary significantly more. In cases where they reach zero value, this indicates that contact has been lost. In contrast, the contact forces associated with the self-stabilising undercarriage vary only moderately and contact with the ground is maintained at all times. The increase the amplitude of the force towards the end of the simulation is probably attributable to the absence of energy dissipating damping in the vehicle which causes the undercarriage to virtually sway from one side to the other after the hump has been traversed. The ramping up of the forces at the start of the simulation is due to the dynamic relaxation when the weight of the vehicle is gradually transferred to the tyres.

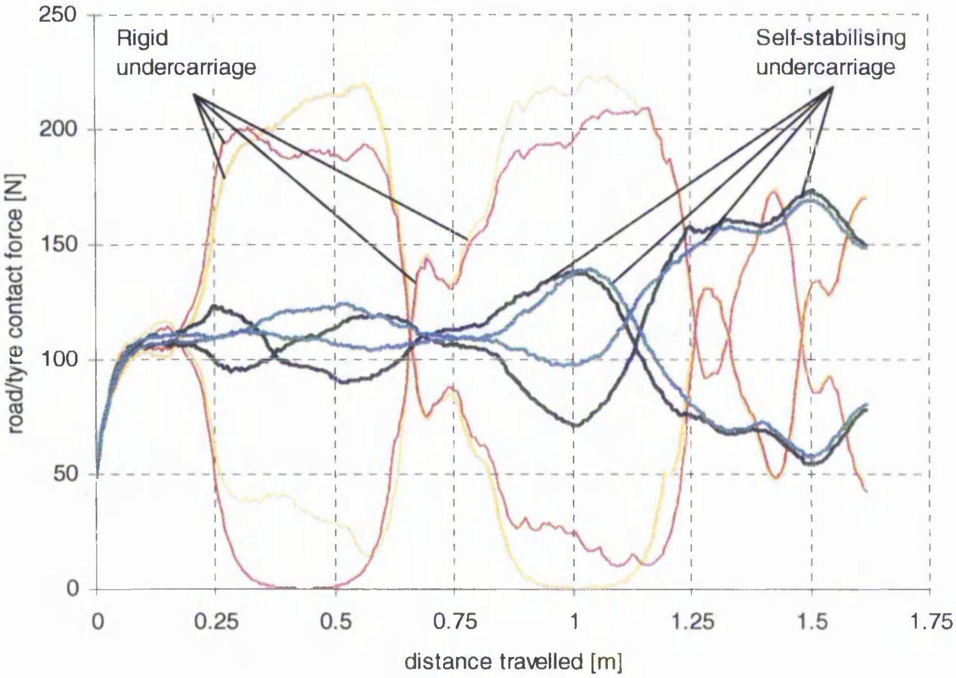


**Figure 18** Comparison of the simulated vertical contact forces between the road and tyre for the self-stabilising and the rigid undercarriage.

## 4.6 Experimental Assessment

Experiments have been conducted to assess the performance of the vehicle. Because of the expected high impact loads expected when the vehicle traverses a hump, the experiments were carried out without the onboard computer system and optical shaft encoders mounted on the wheels. This was to protect the electronic interfaces and sensors from damage. The steering columns were locked at their nominal positions and the drive motor was powered by a 12 volt battery. By means of two links connected to the front and rear axles, the vehicle was forced during the experiment to describe a circular path but it could move freely in the vertical direction. One side of the vehicle was driven over the hump while the other side remained on the level surface. The hump had been covered with white paper and the dust collected by a rubber wheel whilst driving towards the hump enabled a footprint to be 'marked' onto the paper. In other words, when the footprint was interrupted the wheel had completely lost surface contact. The average velocity of the vehicle was approximately 1.2 m/s. This was determined by measuring the time taken by the vehicle to complete a full circle. Visual inspection of the footprint showed that the results agreed qualitatively with the dynamic simulations.

## 4.7 Summary Comments

A four-wheel steered vehicle has been designed and built as the experimental platform for stability and performance considerations. Stepper motors are used to enable independent steering of the wheels. In order to maintain a set loading height and to eliminate adverse effects resulting from the roll and pitch motion, a lever mechanism has been fitted to the experimental vehicle. A geometrical model has been developed which describes the vehicle motion of selected points on the mechanism. Computer simulations have been carried out to evaluate the performance of the undercarriage when traversing a hump. Simulation results showed that the mechanism enables permanent ground contact for moderately undulated surfaces at reasonable speeds. Consequently, the variations in the road/tyre contact forces have been significantly reduced. This helps to simplify the evaluation of the proposed motion control method. The experimental vehicle has been manufactured and tested.

# 5 Development of the Electronic Interfaces

As described in chapter 3 an effective adaptive motion control system enables a vehicle to autonomously navigate itself from a known reference position to a given dispatch location on the shop floor. The proposed controller uses the knowledge of the rotations and the angular displacements of the wheels to deduce the current position and to initiate appropriate control actions for directing the vehicle to a target position. An electronic system (see also section 3.3) is required to convert the signals from the sensors mounted on the wheels to useful data for the motion control system. Moreover, the generated commands need to be translated into appropriate control signals for the drive and steering systems. Dedicated electronic interfaces needed to be developed to enable the data acquisition and the execution of control actions.

## 5.1 The Controller Hardware

The vehicle controller is based on the facilities provided by a standardised bus system – the STEbus IEEE 1000 standard. This 8-bit backplane bus provides extensive I/O capability with asynchronous, non-multiplexed data transfer at over 5 Mbyte/s and multiprocessor capability. As the present study forms part of an ongoing research programme, the backplane has enough capability to cater for the current implementation of low-level motion control and future expansion of related tasks such as obstacle avoidance and navigation, when required. The STEbus system has provisions to host up to four master processors. The present investigation only required the use of one host processor, which is responsible for the distribution of the data to several slave interfaces. The flexibility offered by the STEbus provides a platform for hosting several Digital Signal Processor boards, if necessary, in order to provide higher computing performance for future developments.

## 5.2 Sensors for measuring the wheel rotation

A functional relationship exists between the motion of the vehicle and the angular wheel rotation. It is therefore necessary, for the proposed method to work properly that the rotation and the angular position of a wheel can be measured with acceptable accuracy. A small lateral motion of the wheel, which is generally expected during cornering, can result in a small angular displacement of the wheel. Hence, it is necessary to deduce the angular wheel position with a relatively high degree of accuracy so as to be able to deduce such small angular displacements. The accuracy of the sensors have to be selected accordingly. A data acquisition system has to be designed to allow reliable deduction of the displacements and determine the motion of the vehicle.

For operations involving measurements of continuous wheel revolutions with a high degree of accuracy, optical encoders mounted on wheel axles have proven a reliable means. Optical shaft encoders produce outputs in form of two square waves Figure 19. The two outputs, Channel $A$ and Channel $B$ are produced with a 90 degree phase shift. A synchronising pulse $M$, which only occurs once per full revolution, is set in phase with Channel $A$ to enable the determination of the shaft rotation, direction and angular position. The synchronising pulse $M$ is generally provided to reset a counter or to trigger another counter which cumulates completed wheel revolutions for example to measure longer distances.

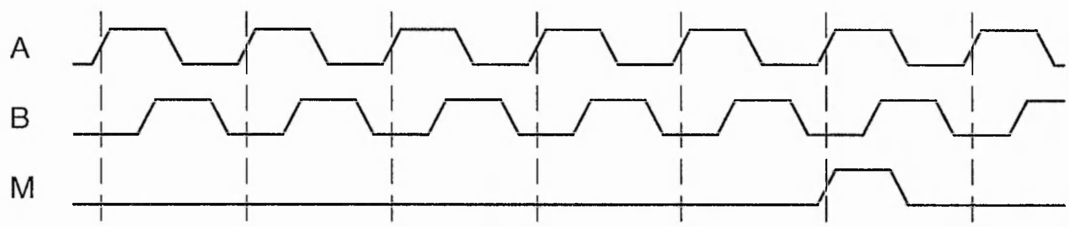


**Figure 19** Typical waveform output of optical shaft encoder

An approximation of the resolution that is required for the encoder can be obtained as follows. Assume the point the vehicle is to turn about is located midway between the front and the rear wheels at a lateral distance $CoC$. This allows the arc's length covering the distance between the inner front and the rear wheels to be calculated. When the radius increases by $\delta_{CoC}$ for example because of a lateral displacement of the vehicle, then the length of the arc between the two wheels increases. Assuming no longitudinal slip occurs between the road and the tyres, the lateral displacement of the vehicle causes a small angular displacement of the wheels. To calculate the angular wheel displacement from the radius variation, consider the following relationship. The distance from the centre of curvature to the line connecting the two inner wheels is $CoC$ minus half of the vehicles width $wi$. This distance in relation to half the vehicle's length is proportional to the distance plus the lateral displacement $\delta_{CoC}$ to half the vehicles length plus the wheel displacement (3).

$$\frac{CoC - \frac{wi}{2}}{\frac{le}{2}} = \frac{CoC - \frac{wi}{2} + \delta_{CoC}}{\frac{le}{2} + \delta_W} \tag{3}$$

Solving for $\delta_W$ yields in (4)

$$\delta_W = le \frac{\delta_{CoC}}{2\,CoC - wi} \tag{4}$$

The motion controller is developed to detect a lateral displacement $\delta_{CoC}$ of 2 mm at a radius $COC$ of 3 m. For this, the expected wheel displacement is approximately 0.186 mm. Using wheels with a diameter of 125 mm the resolution of the encoder can be calculated to 2121 pulses per revolution to provide the required angular displacement. The resolution of the shaft encoder has been selected at 2500 pulses per revolution, which gives a resolution at the circumference of the wheel of 0.157 mm.

The speed of the vehicle can vary between 0 and 1.4 m/s, which gives a maximum of 214 wheel revolutions per minute when the vehicle is driving in a straight line. Therefore, the pulses from the encoder are transmitted at a rate of 0 to 9 KHz.

### 5.3 Data Acquisition Interface

Commercially available data acquisition interfaces provide typically up to three channels and allows measuring the position, velocity and direction of up to three optical encoders. These interfaces are generally multiplexed which means the encoders are sampled one after another in a sequential manner. The delays between data captures can introduce deviations since the wheel position may have changed, before the next wheel can be measured. It is therefore difficult to deduce the state of the four wheels at one particular instance in time accurately. To improve the accuracy the measurements must be correlated by interpolating the data points to account for any time jitter. This may be acceptable when a generally constant or slowly changing velocity is measured. When the signals vary significantly, for example during entering and exiting tight turns, time delays can introduce significant inaccuracies.

In addition, it is typically not possible to synchronise sampling across several data acquisition interfaces. This introduces further uncertainty when four data channels, one for each wheel, have to be sampled. To reduce the effects of timing jitter, data should be sampled at a much higher rate, which necessitates significantly higher specifications for the data acquisition interfaces and increased processing power for correlating the angular positions of the four wheels. Over-sampling presents redundant data to the controller and only helps to improve the accuracy if the sampling sequence and the time between samples can be guaranteed. Otherwise, unplanned processor interruptions during the execution of the sampling sequence may invalidate such efforts and causes further deterioration in accuracy.

However, if a recurring synchronising signal is used to trigger a snapshot capturing all four angular wheel positions, the state of the vehicle can be deduced accurately. The data may be stored in temporary buffers, which can then be read by the processor in a sequential manner before the buffers are updated in parallel with the next set of state data. Because a snapshot represents a 'true' state of the vehicle at a given time, correlation or interpolation of the data to eliminate the effect of time delays resulting from sequential sampling is generally not necessary. Hence, with a parallel sampling data acquisition interface the required processing power can be reduced significantly while at the same time increasing the reliability of the data analysis.

In order to further improve the reliably in deducing the data, maximum settling time should be allowed after any new motion command has been implemented. This implies that measurements need to be delayed until just before the new motion commands are implemented. This can easily be achieved by synchronising the command implementation to the same synchronising signal as the data acquisition.

### 5.4 Strategy for the data acquisition interface

For high data integrity, the design of the data acquisition interface SQD4 includes the capability of acquiring the angular wheel rotation without any intervention from the host processor. This in turn enables the interface to be tolerant to faults. Faults can occur when the processor is servicing other components or interrupts during a data acquisition cycle. The data from the buffer may not be transferred in time and hence cannot be calculated for a few acquisition cycles. Nonetheless, as long as the angular wheel positions are kept accurately in the decoder buffers, the processor can fetch the state of the vehicle a few samples later. When evaluating the data the processor would see the motion of the vehicle as an average between the last successful and the current data acquisition cycle. Therefore, the design of the interface includes a mechanism to update the buffers automatically in predefined time intervals so that the controller can recover from such faults.

Additionally, dedicated hardware is used to relieve the processor from time consuming decoding functions. Hence, the signals from the encoder must be decoded with a dedicated hardware solution where the two square-wave signals are translated into binary numbers representing the angular wheel positions. The number held in the decoder is typically a 16-bit value. However, since the bus width in the motion control system is limited to 8-bit, the synchronising signal triggers stores the values into two halves (Figure 20). The host processor can then at a convenient time sequentially address a buffer, fetch the data, and restore the 16-bit number in the internal processor registers for data processing.

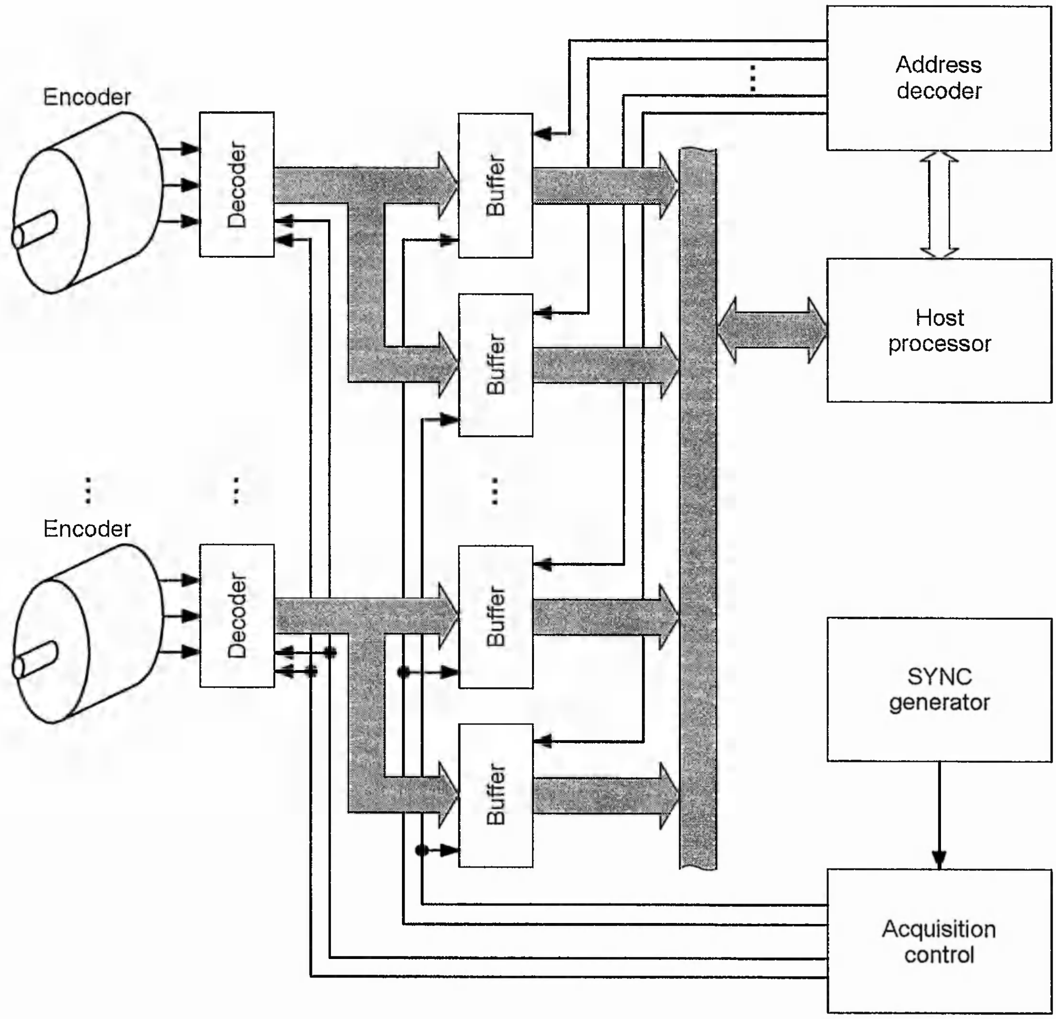


**Figure 20** Block diagram of the data acquisition interface SQD4 for parallel sampling of the encoder signals

Acquisition control is implemented so that the data is transferred to the buffer without any processor intervention. A decoder requires two signals to read a 16-bit data word, a read signal and a selection signal, the latter selects whether the high or low byte is being output to the data bus. The read signal suspends updating of the internal count register of the decoder and puts the low byte onto the local data bus if 'low' is selected. Then the select signal is brought high and the high byte is presented to the local data bus. At the same time the control signals are presented to the respective buffers which stores the data. The synchronising signal *SYNC* is generated by a timer located in the system and distributed across the backplane so that several interfaces can be synchronised, for example in the implementation of required control actions.

## 5.5 Components for the Data Acquisition interface

The HCTL-2016 from Hewlett Packard is a CMOS integrated circuit that performs quadrature decode, 16-bit binary up/down state count, and an 8-bit bus interface function. Quadrature decoding of an encoded signal means that both the rising and falling edges of the two square-wave signals are counted and a count value that is four times the nominal number of pulses per revolution is achieved. The HCTL-2016 is designed to improve the performance in digital closed loop motion control systems and digital data input systems by shifting time intensive quadrature decoding functions to a cost-effective hardware implementation. The use of Schmitt-triggered CMOS input and noise filters allows a reliable operation in noisy environments at clock frequencies up to 14 MHz.

When the quad decoders are used with optical encoders with less than $2^{16} = 65536$ pulses per revolution, provisions have to be provided to zero the count value every time an encoder completes a full revolution. Otherwise, the counter would count up to the maximum value and then continue from zero, or when counting down the counter would reach zero and then decrement from 65535. For example, an encoder produces 2500 pulses per revolution so that the angular position using a quadrature decoder is expressed by a value between 0 and four times the resolution of the encoder to a maximum count of 9999.

The acquisition control needs to generate the two signals *ENABLE** and *SEL**, whereby the * denotes a signal which is active if a logic 0 is selected. The *SEL** signal determines whether the high byte or the low byte is output by the decoder. The required sequence is generated using J-K Flip-Flops (FF) which are synchronised to the system clock signal *CLK* of 16 MHz. The timing of the two signals triggered by an active *SYNC* input is shown in Figure 21. The *SYNC* signal may be active for a variable duration and hence a means is required to return *ENABLE** and *SEL** to the inactive state after two *CLK* cycles.

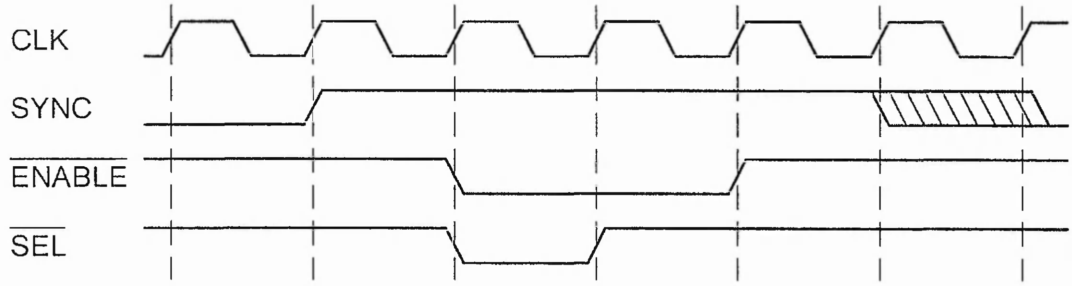


**Figure 21** Timing diagram for the acquisition control signals

The combinational logic and the two J-K Flip-Flops are implemented according to the block diagram in Figure 22 to produce the signal sequence shown in Figure 21. The clock signal *CLK* for the Flip-Flops is not shown in the diagram.

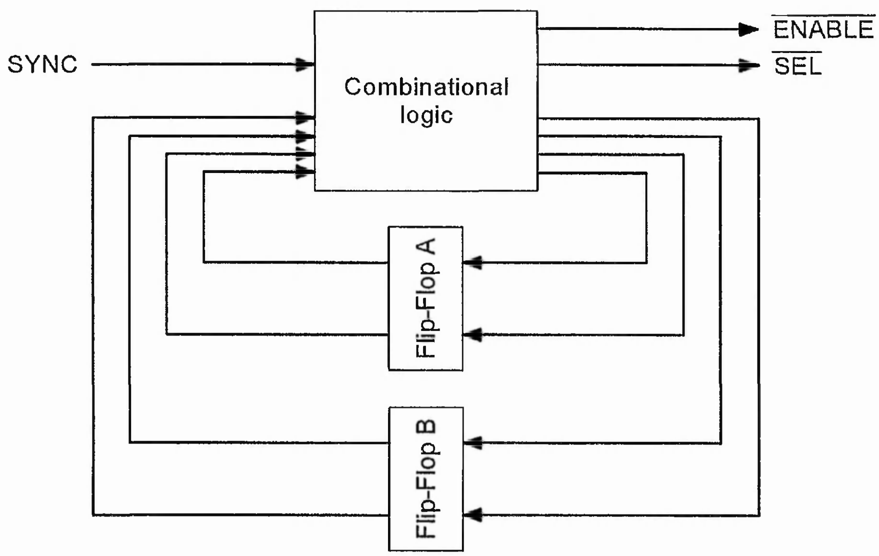


**Figure 22** Block diagram of the acquisition control

The combinational logic required for the sequence is given in equation (5). The Flip-Flops are of J-K* type because this allows a more efficient implementation from the signals provided. They have both direct and inverted outputs which can be used directly as input for the other FF. For example, the output from the FF *B* is directly connected to the *J* input of FF *A*.

$$\left.\begin{aligned}
\overline{SEL} &= \overline{SYNC \wedge \overline{A} \wedge \overline{B}} \\
\overline{ENABLE} &= \overline{\overline{\overline{SYNC} \wedge \overline{B}} \wedge \overline{A}} \\
J_A &= B \\
\overline{K}_A &= \overline{\overline{SYNC} \wedge \overline{B}} \\
J_B &= \overline{\overline{SYNC} \vee A} \\
\overline{K}_B &= \overline{A}
\end{aligned}\right\} \qquad (5)$$

Resetting the decoder after a full revolution has been completed is triggered by the signal from Channel *M* and this resets the count registers in the HCTL-2016 to 0. The reset signal to each decoder is gated so that the register of the decoder cannot be cleared while data is being transferred to the intermediate buffers. This is easily achieved by combining the *ENABLE** signal with Channel *M* from the optical encoder, effectively preventing a reset of the decoder while *ENABLE** is active. Two implications result from this. First, the *ENABLE** signal must be kept as short as possible independent of the duration of the *SYNC* signal. Second, the reset pulse from the encoder must be longer than the duration of *ENABLE** plus a minimum time so that the decoder recognises the reset. The time for the *ENABLE** is fixed to two clock cycles, which equates for a 16 MHz system clock to 125 ns. Allowing a further 125 ns for the reset, the maximum number of encoder revolution per minute is limited to approximately 96000 rev/min which is well above the required range and most likely above the specification of most encoders.

The capability of the data acquisition interface was investigated (Plantenberg et al., 1997, see Appendix B) using a purpose built condition monitoring test bed (Figure 23). A series of experiments were carried out to evaluate the capabilities of the interface with particular emphasis on the following aspects:

- Resolution of encoders
- Accuracy of the sampled data
- Obtainable sampling frequencies
- Data and positional integrity on failure of the host processor

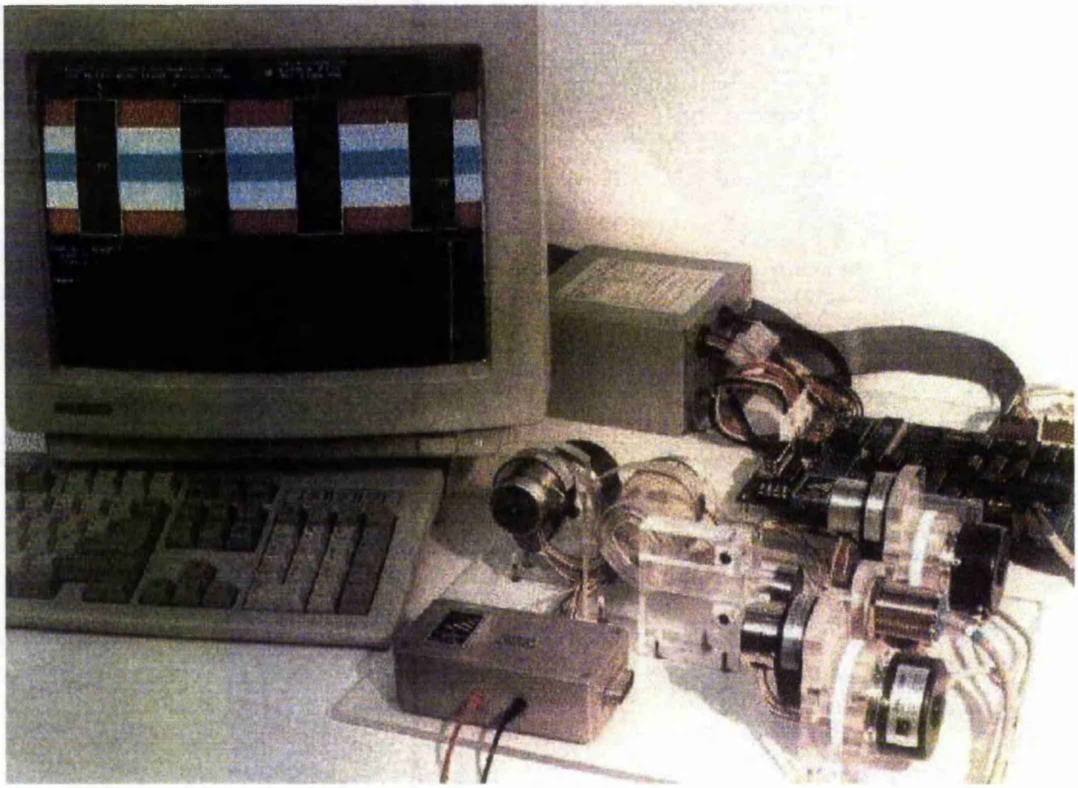

**Figure 23** Test bed for evaluating the data acquisition interface

This was achieved by varying the following parameters in isolation and/or in combination:

- The operational speeds of the stepper motors driving the encoders
- Angular displacements of encoders
- Variation of the sampling frequency (i.e. how often the data is stored in the buffers)
- Variation of the scan frequency (i.e. how often the computer reads the data)

The test results (see also Appendix B) have shown that a high degree of accuracy can be achieved consistently. Using an encoder with an output of 2500 pulses per revolution, an angular displacement resolution of 0.036 degrees was obtained. Based on a wheel with a diameter of 125 mm, this is equivalent to a longitudinal resolution of 0.039 mm. Therefore, the new interface is expected to have the resolution to facilitate the determination of the angular displacement of a wheel.

## 5.6 Stepper Motor Interface

The stepper motors are to be controlled directly by the localised controller. In order to minimise the implications of the control actions on the controller's functionality dedicated hardware has been provided. Improved response time and torque can be achieved by utilising a power source with a higher voltage rating so as to accelerate energising the motor windings. This is achieved by implementing a 24 V battery for the 12 V stepper motors. Chopper drivers are used to control the supply voltage and current to the motors. However, care must be taken that the maximum rating of the motor current is not exceeded. The L297 integrated circuit (IC) provides the necessary functions required for generating the step sequence, energising the individual windings of the four phase stepper motor and circuit protection on one IC. Augmented with the power driver IC, L298, the current through a winding of a stepper motor is sensed and the supply switched off once a threshold has been reached. The current is only switched on again after the current falls below a certain threshold level. This switching takes place at 20 KHz and is synchronised for all four stepper motor drivers to prevent inter-modulation effects likely to occur from the switching of the power outputs. The L297 requires only the direction and a pulse signal for each step to be implemented from the controller. In order to prevent spikes and other disturbances to influence the controller, optocoppler are used to isolate the high power electronic components from the low power controller.

Figure 24 shows the block diagram of the stepper motor interface. The strategy is similar to that used in the data acquisition interface but in reverse. The step and direction sequence is calculated by the host processor and sequentially stored in the intermediate buffers. In this case, double buffering is required to separate the updates by the host processor from the updates of the drivers. On an active *SYNC* signal the next set of data is presented to the drivers and held until the next *SYNC* signal. For safety reasons the relays controlling the power supply to the steering and traction systems are independent of the *SYNC* signal. Otherwise, should the *SYNC* signal fail, for whatever reason, there would be, apart from a manual emergency stop, no means of switching the power off. With the current implementation, a power failure on the host processor would automatically switch off the power to the high current systems.

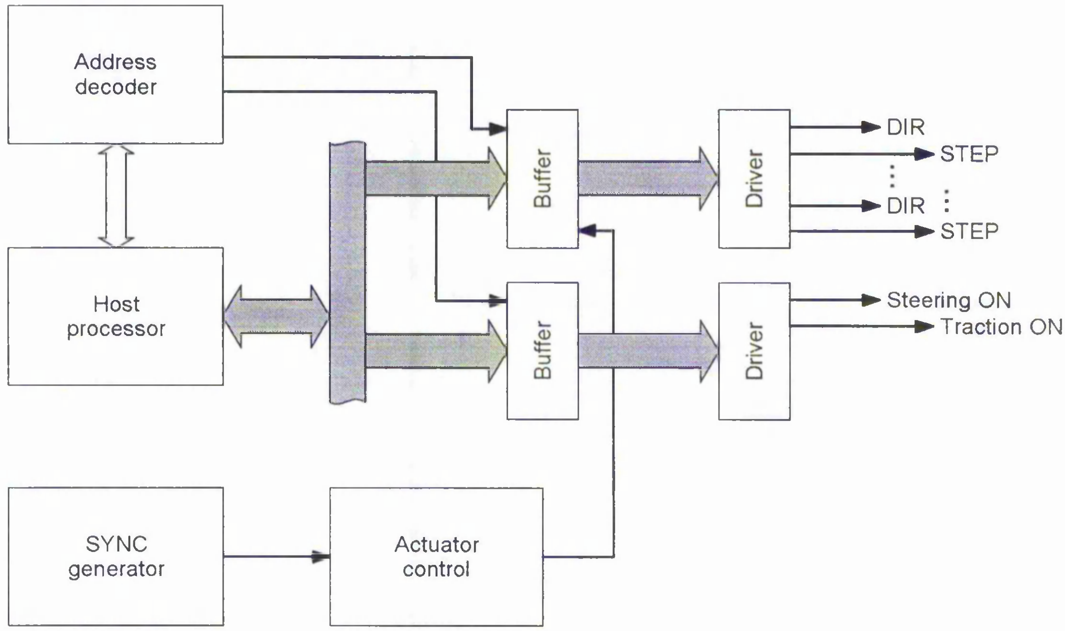


**Figure 24** Block diagram of the actuator interface SPC1

Figure 24 shows the timing of the write signal *SPWR** signal in respect to the *SYNC* signal. The data is clocked to the drivers on the rising edge of *SPWR**. To remind the reader, updating of the internal counter registers of the decoder is suspended on the falling edge of *SYNC*, which ensures that the data is captured before new actions are implemented.

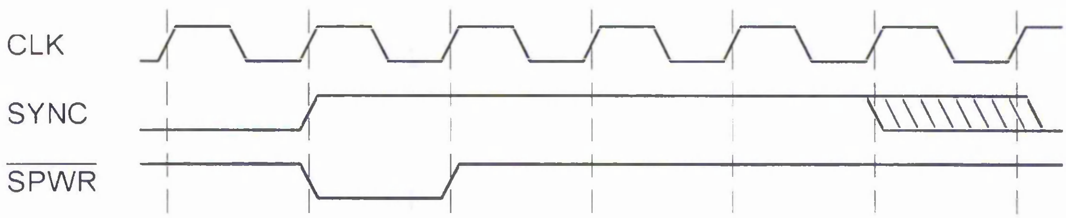


**Figure 25** Timing of the actuator write signal

The actuator control follows the same method as for the acquisition control. The *SYNC* signal is presented to the combinational logic (see Figure 22 as a reference) and two Flip-Flops are used to produce in this instance only the one signal *SPWR** for writing the data word to the stepper motor driver. The required combinational logic is given in equation (6).

$$\left.\begin{aligned}\overline{SPWR} &= \overline{SYNC \wedge \overline{A} \wedge \overline{B}} \\ J_A &= B \\ \overline{K}_A &= SYNC \vee \overline{B} \\ J_B &= SYNC \vee A \\ \overline{K}_B &= SYNC \vee \overline{A}\end{aligned}\right\} \tag{6}$$

This interface has been implemented on a prototyping board SPC1 for the STEbus system. The interface for the power electronic is isolated using reed relays for the power paths and opto coppler for the data paths to avoid any spurious glitches upsetting the host processor.

## 5.7 Summary Comments

In order to reliably deduce the wheel rotation and wheel displacement with high accuracy a dedicated data acquisition interface has been developed, which enables data to be captured from four optical encoders mounted directly on the wheel axles. In order to eliminate time jitters between samples, the data is captured in a parallel manner readily available for the host processor to sequentially process the measured angular wheel positions. In addition, control actions are implemented synchronously to capturing the data, which allows maximum settling time between actuation and capture and in turn helps to obtain more stable measurements. The data acquisition interface has been evaluated on a test bed and has shown accurate and reliable results.

# 6 Method of Determining the Vehicle Position

When the vehicle is moving from a known reference towards a target, the position of the vehicle is estimated by tracking its motion from the measurements of the wheel rotations. Unlike published mathematical models, the proposed method does not determine the forces acting on the vehicle or its acceleration for motion control. This follows the notion that, on a reasonably level surface, it is more important to know that the vehicle moves clear of the environmental boundaries (kinematics) rather than knowing the forces causing the vehicle to move (dynamics) along a given path. Using Descard's principle that an object is either purely translating or rotating about an axis, this investigation considers only circular motion. When the radius is chosen sufficiently large, the difference between a pure translation and a rotation may be considered negligibly small. It follows that relatively complex paths can be generated by combining sections of a circular path and by varying the location of the centre of rotation.

The motion controller needs to know the current position of the vehicle and then determines the new steering commands to compensate for any undesired lateral motion of the vehicle and hence minimises its deviations from a reference path. The displacement of any three wheels can be used to deduce the centre of rotation and the vehicle yaw angle, which effectively describes the longitudinal and lateral motions of the vehicle. By cumulating the circular segments travelled, the current vehicle position can be estimated with respect to a known reference and the vehicle's path mapped out. The new steering angles are calculated so that the compatibility condition is satisfied and the vehicle moves with a given radius.

## 6.1 Determination of the Vehicle Steering Angles

The literature offers a number of methods for calculating the steering angles of four-wheel steered vehicles. Most mathematical models for this type of vehicles are based on the double bicycle principle, combining the two front wheels and the two rear wheels to one wheel at the front and one at the rear. To control the yaw of the vehicle the front and rear steering angles must be known. The motion controller then calculates the required actions based on the vehicle yaw rate and the radius of curvature. Different to the published steering models the present investigation only calculates one vehicle steering angle, often referred to as the vehicle float angle. The float angle is governed by the location of the centre of rotation. From this location, the individual steering angles are calculated so that they satisfy one transcendental rolling compatibility condition. The steering commands are implemented in a co-ordinated manner to satisfy compatibility of all steering angles. Should the vehicle float angle need to be changed, the centre of rotation is relocated to accommodate the new requirement. Mathematically sense the whole vehicle can be represented by just one virtual wheel that has the capability to explicitly control the driving direction.

This approach offers a simple method to estimate the location and the direction of travel of the vehicle in respect to a know reference. Following from the investigation carried out by Astolfi[1], (1995), the vehicle trajectory and position can be described in the complex notation to overcome the limitations associated with the Cartesian co-ordinate system. Figure 26 shows a vehicle moving in a horizontal plane carrying a co-ordinate system with the origin attached at the geometric centre.

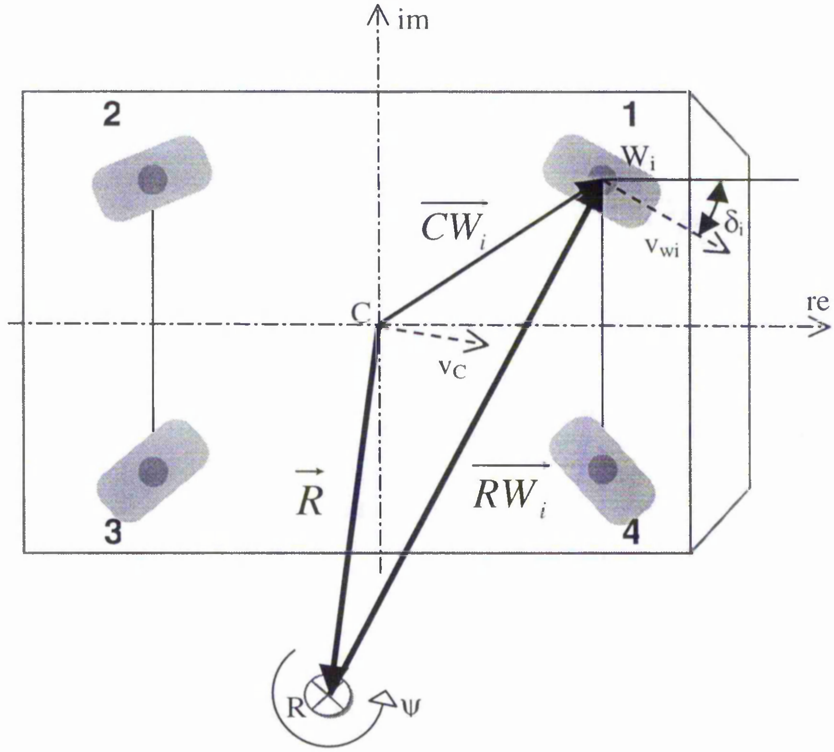


**Figure 26** Vehicle nomenclature

For convenience, the reference point has been chosen to coincide with the geometric centre of the vehicle, *C*, midway between the left and right wheels (track), and midway between the front and rear wheels (wheelbase). The real axis is parallel to the longitudinal vehicle axis and the imaginary axis parallel to the lateral vehicle axis. The position vector of the centre of curvature $\vec{R}$, about which the vehicle is to turn around, can be written as:

$$\vec{R} = \rho \cdot e^{j\Theta} \tag{7}$$

Where $\rho$ is the amplitude and $e^{j\Theta}$ the argument.

The wheels are numbered corresponding to the numbering of the four quadrants. The position vector $W_1$ for wheel 1, relative to the geometric centre *C*, is given by (8), where *le* is the length of the wheelbase and *wi* the track.

$$\vec{W}_1 = \frac{le}{2} + j\frac{wi}{2} \tag{8}$$

Arranging the wheels in a matrix form according to their locations, we have

$$\boldsymbol{CW} = \begin{bmatrix} -\overline{\left(\frac{le}{2} + j\,\frac{wi}{2}\right)} & \left(\frac{le}{2} + j\,\frac{wi}{2}\right) \\ -\left(\frac{le}{2} + j\,\frac{wi}{2}\right) & \overline{\left(\frac{le}{2} + j\,\frac{wi}{2}\right)} \end{bmatrix} \tag{9}$$

where the overhead bar implies the complex conjugate. The position vectors for all wheels with respect to the centre of rotation can be calculated By subtracting $\boldsymbol{CW}$ from $\vec{R}$

$$\boldsymbol{RW} = \vec{R} - \boldsymbol{CW} \tag{10}$$

Consider the vehicle outline marked 1 in Figure 27 the position vectors of the wheels with respect to the centre of rotation are represented by the dotted lines. The vehicle outline marked 2 is a scaled version of outline 1 rotated 90 degree anticlockwise so that the vehicle geometric centre coincides with the centre of rotation of the vehicle outline 1. Note that the directions of the position vectors correspond to the direction of the wheels direction. Furthermore, the vector between the two geometric vehicle centres points in the travelling direction of the vehicle outline 2 and the angle

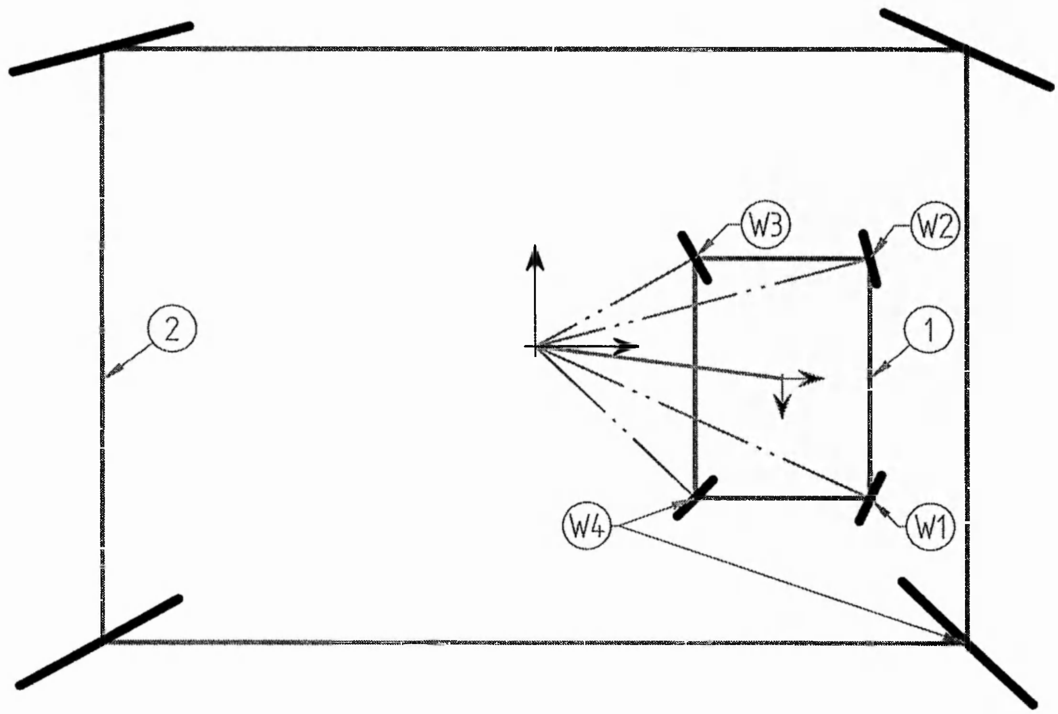


**Figure 27** A Schematic for visualising the relationship between the position and velocity vectors

Consequently, the arguments of the position vectors correspond to the wheel steering angles $\delta$ as shown in (11).

$$\delta = \arg(\boldsymbol{RW}) \tag{11}$$

## 6.2 Determining the motion of the vehicle

For a given vehicle velocity $v_c$ at the geometric centre $C$, all wheel velocities have to satisfy one transcendental rolling compatibility condition. The driving rate of an individual wheels is given by the ratios of the vector magnitudes. Therefore, the rolling velocity at the circumference of wheel $i$, $v_{Wi}$, is determined by

$$v_W = \vec{v_c} \cdot \left| \frac{RW}{\vec{R}} \right| \tag{12}$$

Furthermore, the yaw angle $\psi$ of the vehicle is calculated from the distance it travelled with the velocity $v_c$ for the duration $t$ at radius $\vec{R}$.

$$\psi = -\frac{\vec{v_c} \cdot t}{\vec{R}} \tag{13}$$

The minus sign in equation (13) indicates that the vehicle is rotating in a anticlockwise rotation which is taken as positive.

When the vehicle is driving towards a target position, the angular vehicle orientation, $\psi_c$, with respect to a known starting orientation can be obtained by cumulating the instantaneous vehicle yaw angle as follows

$$\psi_c = \sum \psi \tag{14}$$

The vehicle orientation can be altered by suitably locating the centre of rotation while the vehicle follows the reference path. Since the motion controller samples the motion of the vehicle in discrete time steps, it is preferred to use a displacement vector $\vec{D}_V$ (15) to describe the vehicles advancement. The displacement is governed by the radius vector $\vec{R}$ rotated by $\psi$ and the cumulated yaw angle $\psi_c$ contributes to the vehicle orientation at the last sample.

$$\vec{D}_V = \overrightarrow{\vec{R} \cdot \left(e^{j(\psi_c - \psi)} - e^{j\psi_c}\right)} \tag{15}$$

Similarly, the motion of the vehicle centre, $T_V$, can be mapped out by cumulating the distance vector $\vec{D}_V$

$$\vec{T}_V = \sum \vec{D}_V \tag{16}$$

And the location of the centre of rotation $T_{CoC}$ can be mapped by

$$\vec{T}_{CoC} = \overrightarrow{\vec{T}_V + \vec{R} \cdot e^{j\psi_c}} \tag{17}$$

Figure 28 shows a schematic representation of tracking four displacement vectors with two different initial settings of the centre of rotation as the vehicle begins a turn. Each circle on the dotted line marks the rotation centre during one sample. In a practical controller implementation the time steps are significantly smaller and hence the actual vehicle motion should be better represented. Furthermore, while the displacement vectors shown here may suggest a linear approximation, in practice the vehicle follows a circular path. The yaw rate determines the angular advancement so that the start and endpoint of the vector, shown as small crosses, only mark the start and the end points on a smooth circular path covered in the duration of one sample.

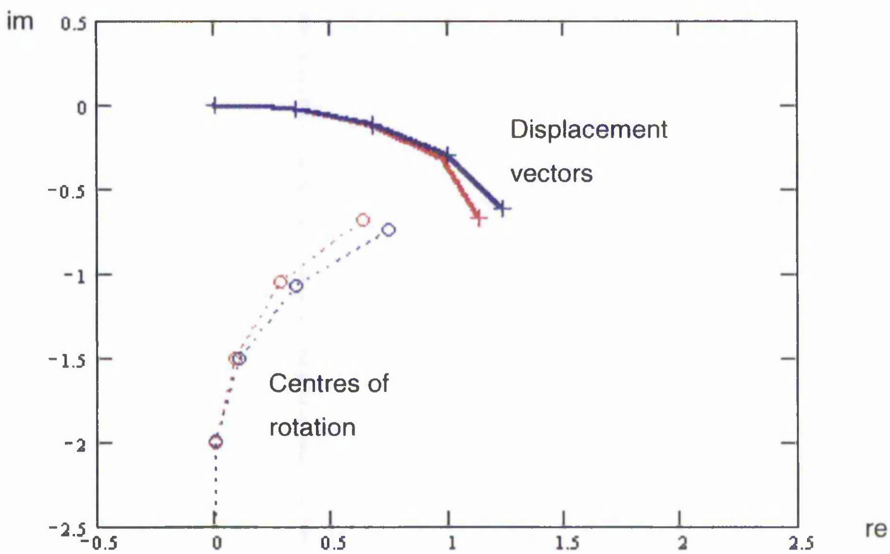


**Figure 28** A schematic representation of tracking the displacement vectors and the location of the centre of rotation when entering a turn

A simulation of an idealised reference trajectory $T_V$ and a trajectory in the presence of lateral drift which is added as a function of the radius is shown in Figure 29. In both cases the vehicle moves with a constant velocity of 1 m/s and is to describe a

180 degree turn starting at (0, 0) with the centre of rotation located midway between the front and rear wheel. The required centre of rotation is determined as a function of the distance travelled. As the radius is reduced gradually until the minimum value has been reached. Thereafter the radius is kept constant at the minimum value to follow through the turn until the vehicle exits the turn. On exit from the turn the radius is progressively increased again. The ideal trace of the centre of curvature $T_{CoC}$ shows a symmetrical path.

In comparison to the ideal trace for the vehicle trajectory, the trace with lateral drift is shortened due to the trade-off between the lateral and longitudinal motions, consequently the vehicle doesn't reach its target position. Furthermore, since the radius is increased too early because of the reduced effective longitudinal advance, the direction with which the vehicle exits the turn is adversely affected and this causes a large position error.

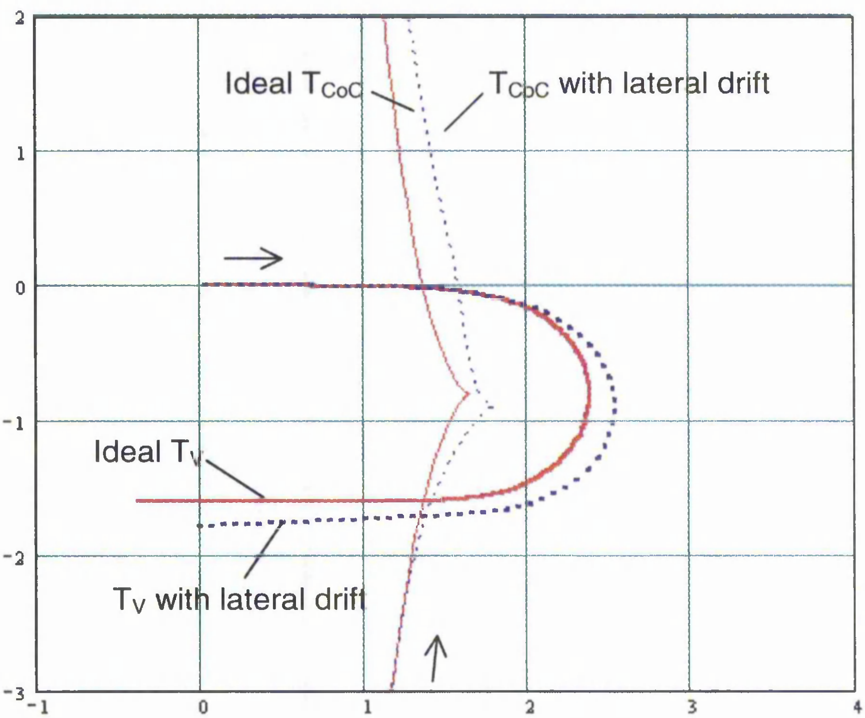


**Figure 29** A schematic diagram of the reference vehicle trajectory and CoC trajectory without (ideal) and with lateral drift

### 6.3 Deducing the Centre of Rotation

Once a workable reference trajectory has been generated motion commands can be devised and implemented. The lateral and longitudinal vehicle motions need to be determined and the signals are feed back to allow the motion controller. This allows remedial actions to be devised should any deviations be detected. Since only circular motion is considered, there exists in the absence of any wheel slippage an explicit relationship between the distance travelled, the vehicle yaw rate and the radius of curvature.

Equation (12) shows that the position vectors of the four wheels are proportional to the wheel velocities and wheel position vectors. It follows from Figure 27 that apart from a scaling factor the distances between the centre of rotation and a point on the vehicle provide a measure of the velocity of that point and the wheel tangential travelling direction. Reversing this approach, the location of the centre of rotation $R$ can be calculated from the distances travelled by any three of the wheels.

The distances covered by the wheels between two consecutive samples can be expressed as increments of the optical encoder. Therefore, the velocity of a wheel is equivalent to the increments multiplied by a factor $k$. Additionally, there is a fixed relationship between the wheel positions.

Figure 30 shows a graphic representation of the geometric relationships which exist between the wheel increments and the length and width of the vehicle. These relationships allow the centre of rotation to be determined. The parameters $a$, $b$ and $c$ represent distances to form the following set of equations:

$$\left.\begin{aligned}
(k \cdot inc_2)^2 &= (a + wi)^2 + b^2 \\
(k \cdot inc_3)^2 &= a^2 + b^2 \\
(k \cdot inc_4)^2 &= a^2 + c^2 \\
le &= b + c
\end{aligned}\right\} \tag{18}$$

Solving for $a$ gives:

$$a = \pm\sqrt{k^2 \cdot {inc_3}^2 - b^2} \tag{19}$$

and for $b$:

$$b = \frac{1}{2}\frac{k^2 \cdot {inc_3}^2 - k^2 \cdot {inc_4}^2 + le^2}{le} \tag{20}$$

The factor $k$ is calculated by:

$$k = \sqrt{\frac{le \cdot wi}{T1}\left(le \cdot wi \cdot \left({inc_2}^2 + {inc_4}^2\right) \pm \sqrt{T2}\right)} \tag{21}$$

Where $T1$ is

$$\begin{aligned} T1 = &\left({inc_2}^4 - 2 \cdot {inc_2}^2 \cdot {inc_3}^2 + {inc_3}^4\right) \cdot le^2 + \\ &\left({inc_3}^4 - 2 \cdot {inc_3}^2 \cdot {inc_4}^2 + {inc_4}^4\right) \cdot wi^2 \end{aligned} \tag{22}$$

and $T2$

$$\begin{aligned} T2 = &\left(2 \cdot {inc_2}^2 \cdot {inc_3}^2 - {inc_3}^4 - {inc_2}^4\right) \cdot le^4 + \\ &\left(2 \cdot {inc_2}^2 \cdot {inc_3}^2 + 2 \cdot {inc_2}^2 \cdot {inc_4}^2 - \right. \\ &\left. 2 \cdot {inc_3}^4 + 2 \cdot {inc_3}^2 \cdot {inc_4}^2\right) \cdot le^2 \cdot wi^2 + \\ &\left(2 \cdot {inc_3}^2 \cdot {inc_4}^2 - {inc_3}^4 - {inc_4}^4\right) \cdot wi^4 \end{aligned} \tag{23}$$

The system of equations (18) has 16 solutions, but only four are different. From (21) the two solutions are $k_1$ (24) and $k_2$ (25) .

$${k_1}^2 = \frac{le \cdot wi}{T1}\left(le \cdot wi \cdot \left({inc_2}^2 + {inc_4}^2\right) + \sqrt{T2}\right) \tag{24}$$

$${k_2}^2 = \frac{le \cdot wi}{T1}\left(le \cdot wi \cdot \left({inc_2}^2 + {inc_4}^2\right) - \sqrt{T2}\right) \tag{25}$$

The sign in the solution of (19) determine whether the result is above or below the line connecting Wheel 3 and Wheel 4 as shown in Figure 31. The value of $k$ from

(21) determines whether the solution is outside the circle (24) described by the Pitch Centre Diameter (PCD) of the four wheels or inside the circle (25).

The PCD is calculated as

$$PCD = \sqrt{\left(\frac{le}{2}\right)^2 + \left(\frac{wi}{2}\right)^2} \tag{26}$$

Hence, the location of the centre of rotation $\vec{R}$ can be deduced from the position vectors as follows

$$\vec{R}_{ded} = \begin{cases} \left(b_{k1} - \frac{le}{2}\right) + j\left(a_{k1} - \frac{wi}{2}\right) \text{ if } \mathrm{Im}\left(\vec{R}\right) > -\frac{wi}{2} \wedge |\vec{R}| > PCD \\ \left(b_{k2} - \frac{le}{2}\right) + j\left(a_{k2} - \frac{wi}{2}\right) \text{ if } \mathrm{Im}\left(\vec{R}\right) > -\frac{wi}{2} \wedge |\vec{R}| \leq PCD \\ \left(b_{k1} - \frac{le}{2}\right) - j\left(a_{k1} + \frac{wi}{2}\right) \text{ if } \mathrm{Im}\left(\vec{R}\right) \leq -\frac{wi}{2} \wedge |\vec{R}| > PCD \\ \left(b_{k2} - \frac{le}{2}\right) - j\left(a_{k2} + \frac{wi}{2}\right) \text{ if } \mathrm{Im}\left(\vec{R}\right) \leq -\frac{wi}{2} \wedge |\vec{R}| \leq PCD \end{cases} \tag{27}$$

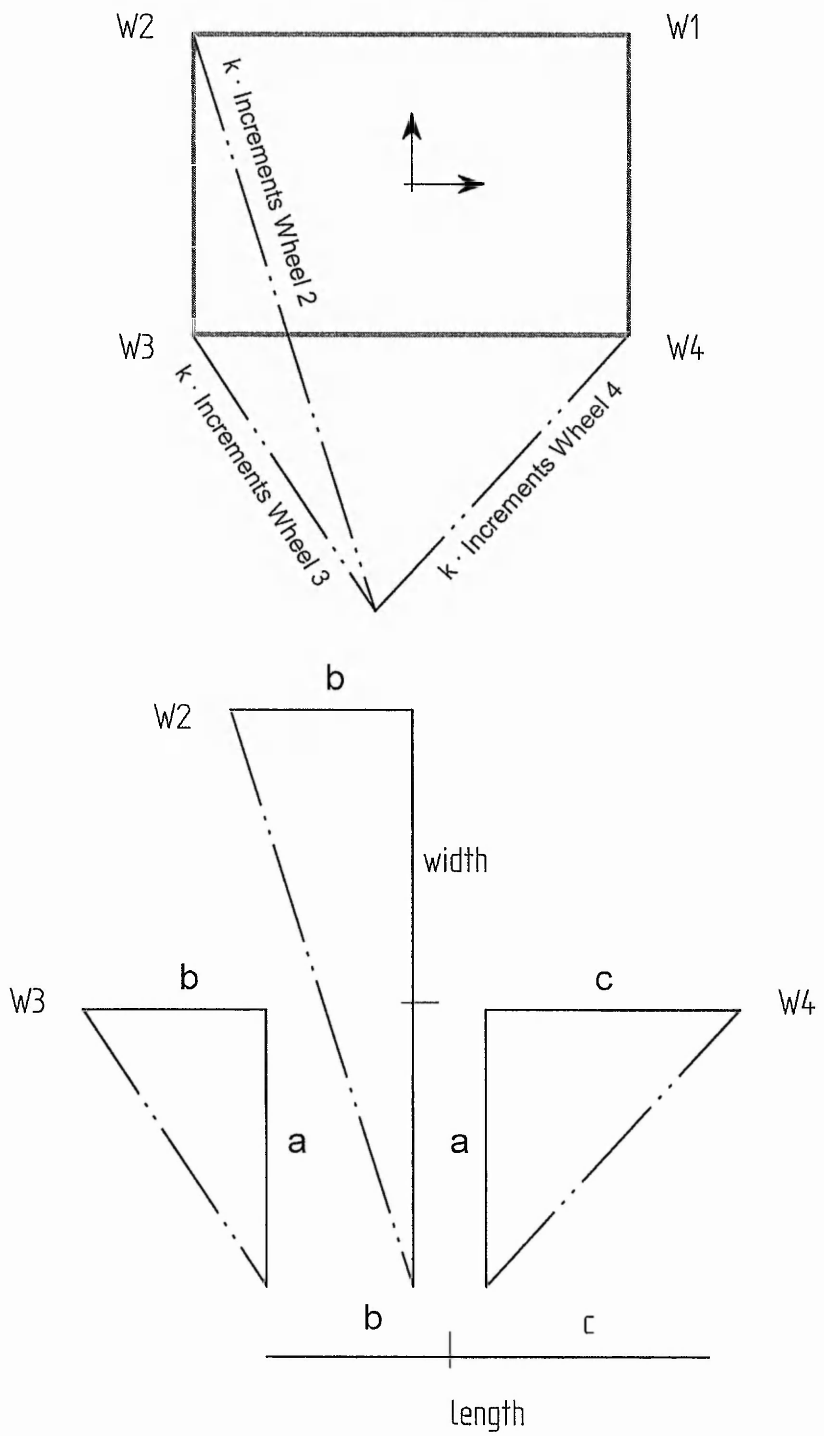


**Figure 30** Graphic representation of the geometric relationships between the wheel displacement and the location of the centre of rotation

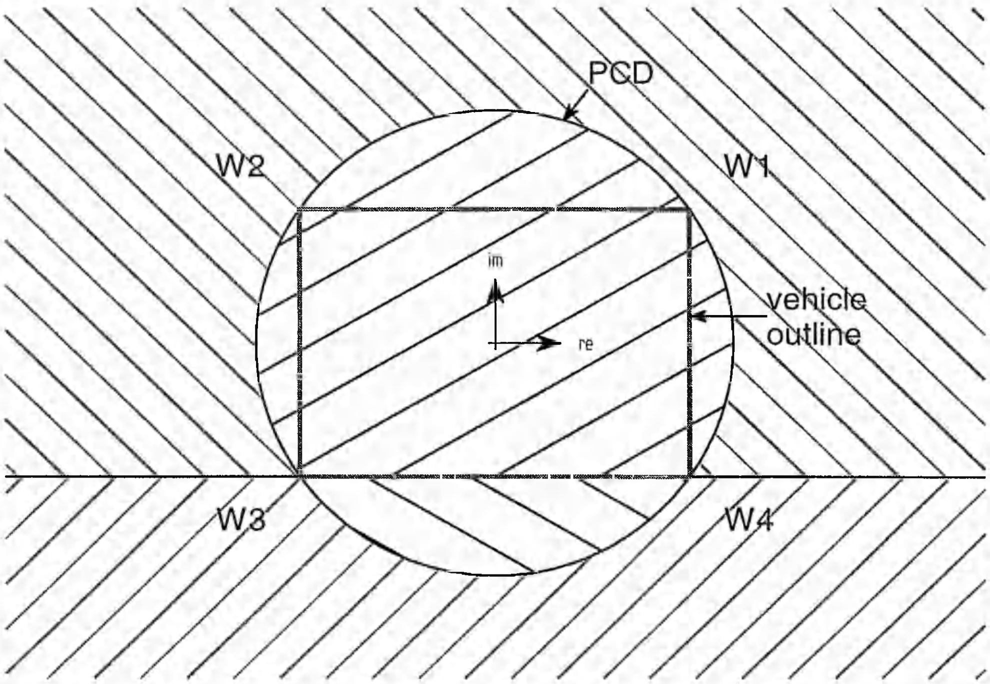


**Figure 31** Areas showing the range of valid solutions when determining the centre of rotation

## 6.4 Controlling Lateral Vehicle Displacement

In practical applications, all vehicles experience some lateral drift when turning around a corner. Consider the vehicle cornering at a constant velocity about a constant centre of rotation. Then superposition the pure circular motion with a velocity component which moderately displaces the vehicle in the lateral direction. This lateral vehicle displacement causes the wheels to experience small angular displacement. The angular wheel displacement increases or decreases the wheel velocity vector and since the wheel velocity and the centre of rotation are related, the angular displacement causes the centre of rotation to vary accordingly.

Figure 32 shows the effects of an increased lateral displacement on the location of the centre of rotation for different settings of the radius vector $\vec{R}$ when the vehicle is turning clockwise. The dot depicts the nominal centre of rotation without lateral displacement. Then a lateral component is introduced by adding a motion along the line connecting the idealised centre of rotation and the vehicle's centre amounting to

25% of the tangential velocity. This in turn causes the location of the centre of rotation to veer anticlockwise by an angle $\kappa$.

This change in the location of the centre of rotation is estimated by employing the method described above. The motion controller incorporates the new estimated location to calculate the next steering angels, so that the estimated location coincides with the nominal location of the centre of rotation and hence minimises deviations from the reference path.

The difference between the location of the nominal centre of rotation *nCR* and the deduced centre of rotation *dCR* is a measure of the road/tyre interface characteristics. For a constant road/tyre characteristics and a constant velocity one would expect that the angle $\kappa$ increases significantly as *nCR* approaches the proximity of the vehicle envelope. When the *nCR* is placed on the line connecting two wheels, these wheels do not have a significant lateral contribution and the friction forces have to be generated entirely by the two remaining wheels.

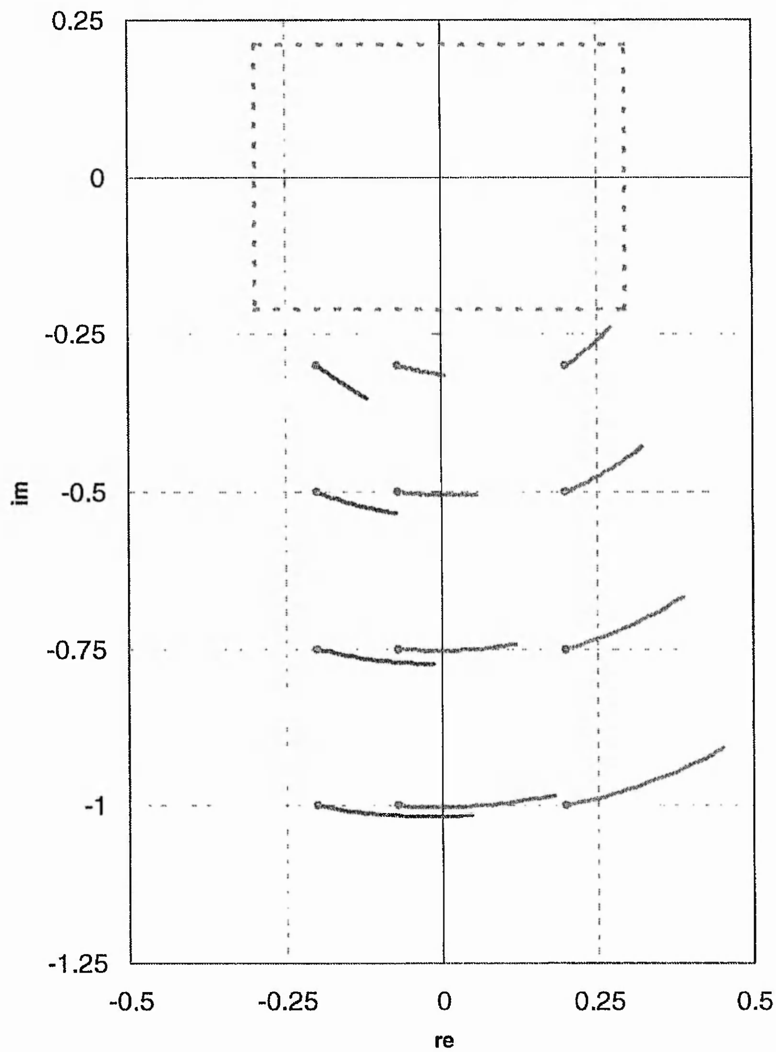


**Figure 32** Simulated deviation of the CoC as the lateral velocity increases to 25%

The angle $\kappa$ is significantly influenced by the location of *nCR*. Therefore, the function (28) is proposed to approximate a 3D control surface $\xi$ for a constant velocity and a constant road/tyre interface as shown in Figure 33. The function takes into account the location of *nCR* in the complex form and the length and width ratios of the vehicle.

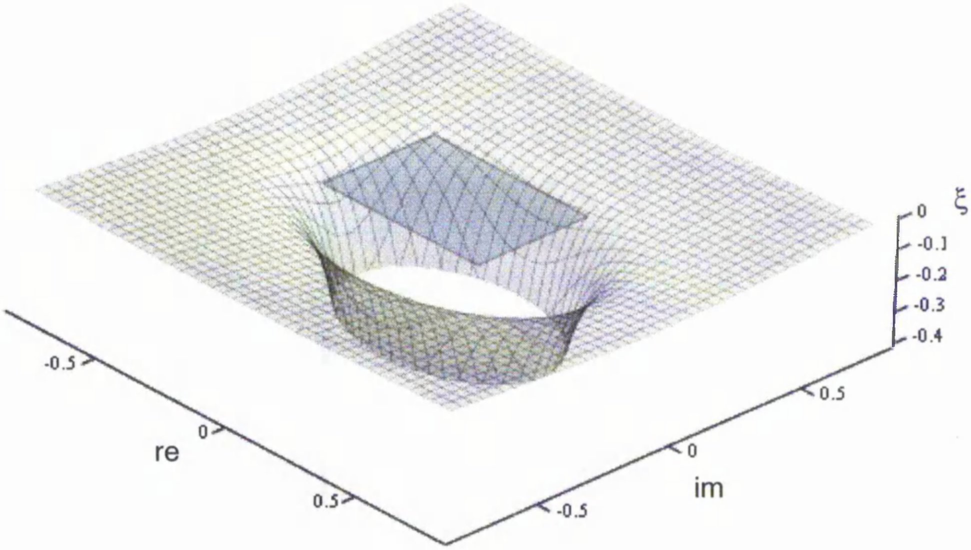


**Figure 33** 3D Control surface of the motion controller for constant velocity and road/tyre interface characteristics

$$\xi = -a\,e^{-b\left(p^2\left(R+\overline{R}\right)^2 - q^2\left(R-\overline{R}\right)^2\right)} - c\,e^{-d\left(p^2\left(R+\overline{R}\right)^2 - q^2\left(R-\overline{R}\right)^2\right)} \tag{28}$$

where

$$q = \frac{2\,le}{le + wi} \tag{29}$$

and

$$p = \frac{2\,wi}{le + wi} \tag{30}$$

The parameters *a*, *b*, *c* and *d* are optimised off-line so to closely approximate the required shape of the control surface.

The motion controller determines the required control actions from the nominal vehicle velocity and the nominal centre of curvature taking into account the estimated vehicle velocity and the current road/tyre interface characteristics. The steering angles are calculated from the set centre of rotation vector, $s\vec{CR}$

$$s\vec{CR} = \vec{R}\, e^{j\kappa} \tag{31}$$

which is determined by rotating the nominal centre of rotation vector by the angle $\kappa$.

The angle $\kappa$ is calculated as

$$\kappa = \xi\, \eta\, v_{cn}^2 \tag{32}$$

whereby $v_{cn}$ represents the nominal vehicle centre velocity and $\eta$ a factor describing the road/tyre interface characteristics. The factor $\eta$ can be estimated from the difference of the nominal $nCR$ and the deduced $dCR$ at the estimated velocity. Feeding this factor back into the calculation of the next control action allows the motion controller to adapt to changes in the road/tyre interface characteristics.

## 6.5 Summary Comments

Considering only circular motion of the vehicle, a method for determining the steering angles and wheel speeds based on the complex notation is presented. In order to control the yaw and the lateral motion of the vehicle, an approach has been described which considers the vehicle as one virtual wheel and is capable of controlling explicitly its steering angle. By cumulating the displacement vectors of the wheels of the vehicle and the location of the centre of rotation between consecutive samples of the controller, the path of the vehicle is estimated. The difference between the nominal and the deduced centre of rotation can be used as a measure of the vehicle deviation from a reference trajectory. This allows the controller to adapt to changes in the road/tyre interface characteristics.

# 7 Integration of the Components for the Motion Control System

A new approach to determine the road/tyre interface characteristics has been proposed in Chapter 3. For controlling the motion of the vehicle, the position in relation to a reference is deduced principally from the measurements of wheel displacements and velocity vectors of individual wheels in conjunction with the geometric analysis. In order to evaluate the performance of the control methodology an experimental vehicle and proprietary electronic interfaces have been designed and manufactured. The careful design of the individual components enables the full integration of the constituent parts so to form the synergy of the mechanical, electronic and software components. Individual components have been tested in isolation to verify the specified requirements. In isolation, the individual components have been tested to satisfaction so that the remaining task is to integrate the various constituent parts so that the proposed motion control methodology can be evaluated experimentally.

## 7.1 Integration of the Components

Figure 34 shows the assembled vehicle with the controller hardware mounted on top of the undercarriage and the drive motor mounted on one of the wheels. Thereafter this wheel will be referred to as wheel 1 or *W1*. The figure shows wheel 1 on the hump during the preliminary experimental evaluation of the self-stabilising undercarriage. The height and slope of the hump corresponds to the dimensions chosen for the computer simulation of the mechanism. All electronic interfaces are placed inside the vehicle envelope mainly as a precautionary measure.

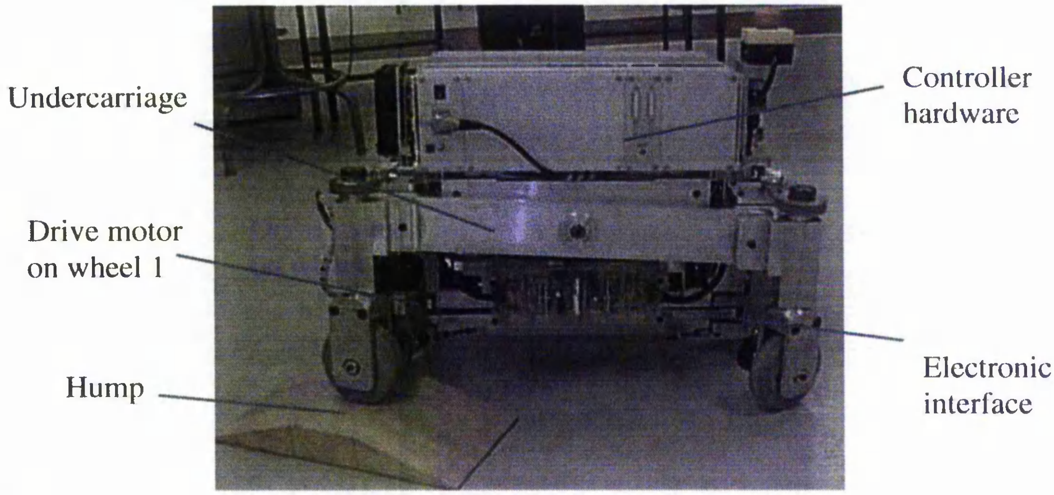


**Figure 34** Image of the fully assembled experimental vehicle

### 7.1.1 Drive and Steering System

A close-up view of the drive motor-gearbox assembly is shown in Figure 35. Mounted on one side of the gearbox is the wheel assembly and on the other side the optical shaft encoder. The cables for the power supply to the motor and for the signals from the encoder constrain the permissible range of the steering angle of the wheel. In the current implementation, the steering angles are limited in the software, when counting the steps from the initialised zero position to allow a maximum of 100 degrees of rotation in either clockwise or anticlockwise direction. This 10 degrees margin over the theoretically required 90 degrees have been chosen to enable improved manoeuvrability of the vehicle when traversing sideways to allow adjusting the longitudinal vehicle position by varying the steering angles accordingly. If only ±90 degrees were allowed and for example 91 degrees were necessary to drive the vehicle to the target position, the wheel would have to turn to -179 degrees and reverse its direction.

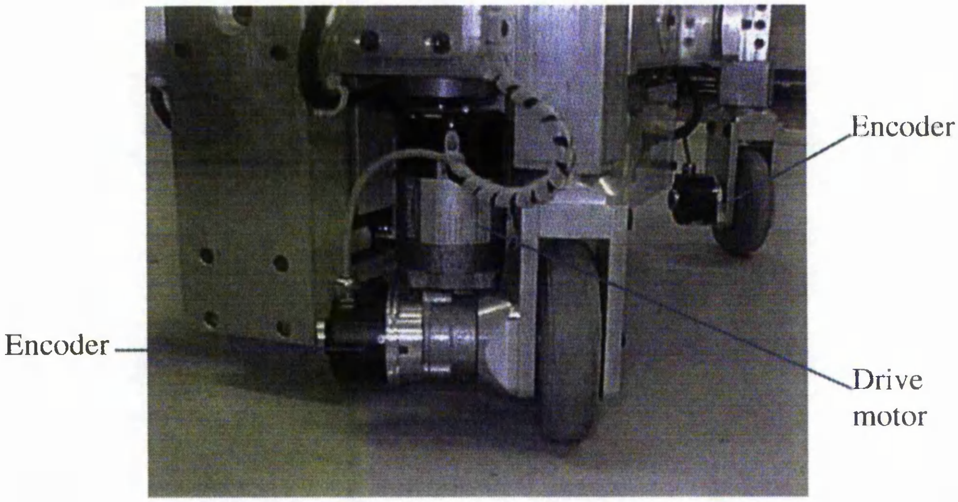


**Figure 35** Image of the motor-gearbox combination with the wheel mounted directly onto one side of the drive shaft and the encoder on the other side

The experiments are carried out by commanding the vehicle to turn about a nominal centre of rotation at maximum velocity so as to investigate the performance of the controller where substantial lateral deviation is anticipated. By allowing the vehicle to circle continuously for several cycles the vehicle motion settles to a steady state and therefore enables a more reliable evaluation of the measurements. The drive motor is switched on by the motion controller for the duration of the experiment and is limited to 12 volts to protect the motor.

### 7.1.2 Self-Stabilising Undercarriage

To demonstrate the capability of the self-stabilising undercarriage and to ensure that the stipulated clearance between the components is sufficient, one wheel has been placed on a block 100 mm high for evaluation of the stationary system (Figure 36). It can be seen that the inclination angle of the main vehicle body (top surface) is approximately half of the angle between the floor surface and the front axle of the vehicle. A similar observation can be seen when the inclined vehicle is viewed from the side. Nonetheless, the experiments for evaluating the motion controller are carried out on a virtually flat surface with only small surface variation.

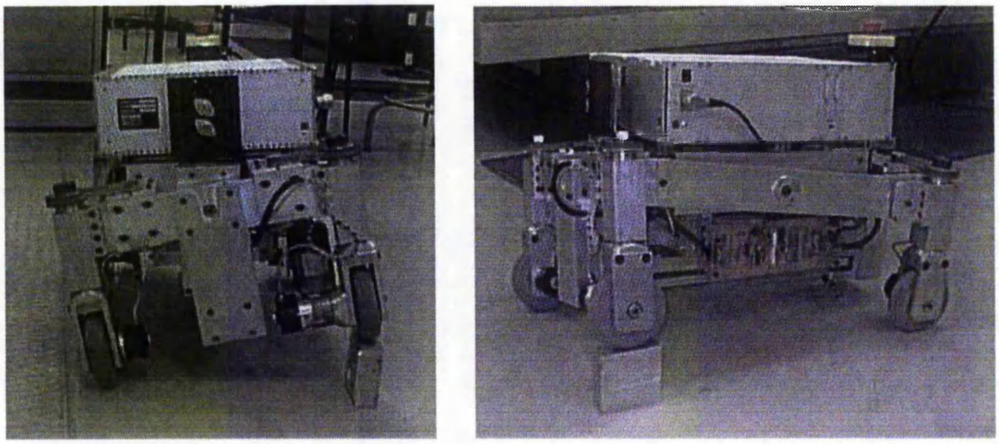

**Figure 36** Images of the self-stabilising undercarriage seen from the front (left) and from the side (right) with wheel 1 vertically fully displaced.

### 7.1.3 Data Acquisition Interface

The data acquisition interface for deducing the wheel rotations is located in the controller hardware crate with the cable connections to the encoder routed through the hollow sections of vehicle frame. The performance of the interface has been evaluated on a test-bed (see Appendix B and 5.5). However, for implementation on the vehicle the software procedure for reading the data need to be adapted by optimising the required calculations so that a sampling rate of 5 ms can be achieved.

Figure 37 shows an image of the fully populated *S*ynchronous *Q*uadruple *D*ata acquisition interface for *4* encoders (SQD4). The two top rows of the integrated circuits are the buffers for the low and the high byte to the corresponding HCTL2016 quadruple decoder circuit from Hewlett Packard. The acquisition control circuit translates the synchronisation signal *SYNC* to the write signals *ENABLE** and *SEL** for updating the buffers with the new angular wheel positions. The *SYNC* signal is generated by a timer on the 80188 host processor and distributed on the backplane of the STEbus so that all interfaces can be triggered simultaneously. Voltage dividers are provided on the SQD4 interface for conditioning the output signals of the encoders to Transistor-Transistor Logic (TTL) levels for the HCTL2016 decoder inputs.

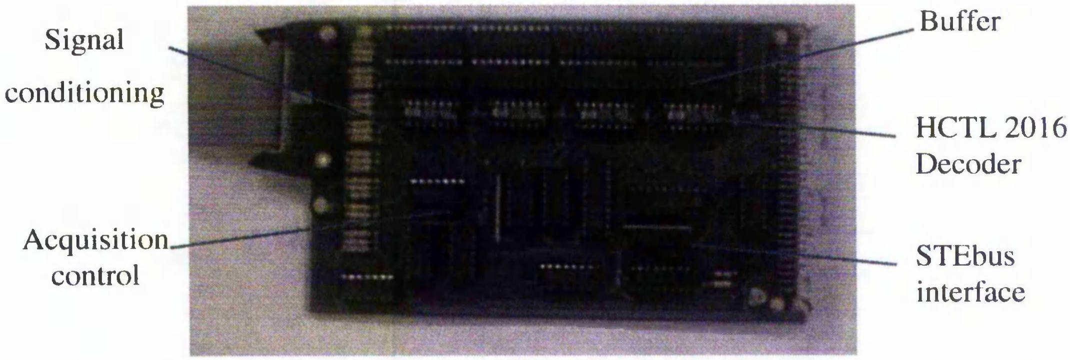


**Figure 37** Image of the parallel sampling data acquisition interface card SQD4

### 7.1.4 Actuator Control Interface

The interface for the vehicle power and the stepper motor control is split into several printed circuit boards (PCBs). Figure 38 shows the *S*ynchronous *P*arallel *C*ontrol interface *1* (SPC1) which resides in the controller hardware crate. The direct buffers can be accessed by the host processor and implement settings without waiting for the *SYNC* signal. The direct buffer drives two TTL level reed relays that switch two high current relays one for the power supply to the drive system and one for the steering system.

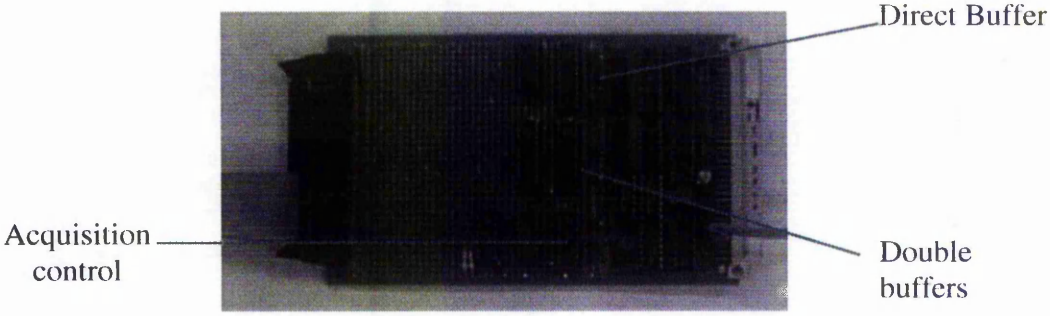


**Figure 38** Image of the parallel actuator

The acquisition control initiates the transfer of the data written by the host processor to the first buffer. On receiving the *SYNC* signal the data is then transferred to the second buffer and is immediately presented to two stepper motor power control interface cards. One stepper motor power control interface controls the two stepper motors on the left side of the vehicle and the other interface the stepper motors on the right side. The location of the electronic interfaces has been chosen for easy access while protecting the on each side the front and rear motor.

The stepper motor power drivers L297 switch the voltage to the individual stepper motor windings to the positive Voltage or Ground depending on the energising direction. The switching sequence is controlled by the L298 controller circuit, which also senses the current and when necessary chops the voltage to limit the maximum current. The chopper rate is synchronised to a 20 KHz oscillator to minimise inter-modulation effects. The inductance of a motor winding determines the time it takes to fully energise the coil and hence limits the maximum permissible chopping frequency. Whilst the frequency should be above the audio range to minimise the noise levels of the shop floor environment, higher frequencies also increase the switching losses of the power drivers.


**Figure 39** Image of a stepper motor power control interface cards for controlling two stepper motors

Since the stepper motor power control cards are built from breadboard type printed circuit boards, insufficient ground connections and spurious signals can cause occasionally the chopping frequency to lock to a sub-harmonic frequency. When chopping the motor current at a frequency in the audio range, a noticeable noise is emitted from the stepper motors. The addition of smoothing capacitors and better ground connections has helped to minimise the inter-modulation effects.

### 7.1.5 The Controller Hardware

The controller hardware is based on the STEbus IEEE 1000 standard 8-bit backplane with a 16 MHz bus clock frequency. The master processor for the motion control system of the vehicle is based on an Intel 80188 industrial processor located on the SCIM88 processor board from Arcom. The SCIM88 card has 256 KByte on board volatile and 64 KByte non-volatile memory and a serial interface. The serial interface is temporarily connected to an external desktop Personal Computer (PC) to download the controller program into the onboard volatile memory and then disconnected for the duration of the experiment. A serial interface card is plugged into the STEbus and has been programmed to send data to a second external PC for downloading the captured data after completion of the experiment. The connections between the onboard controller and the two external PCs are removed during the experiment so that the cables do not interfere with the motion of the experimental vehicle.

## 7.2 Motion Control Program

The software program for the motion control system (Appendix C) consists of several set-up procedures and a continuous loop. Within the loop, the data is read from the interface, new control actions are calculated and the next set of data is then transmitted to the interfaces. The time it takes the program to complete all the commands in the loop must be kept to a minimum to enable relatively high sampling rates. Therefore, the calculations of values, which remain unchanged for the entire program, have been placed in the set-up procedures of the program. Furthermore, due to the limited data processing performance available in the current implementation, variations of the steering angle of the wheels to compensate for any changes in the lateral vehicle motion have been stored in a memory array within the set-up procedure. It is envisaged that a slave processor may be added in future development to support the host processor in the calculation of the array elements in the proximity of the current nominal Centre of Rotation *nCR* amongst other supporting functions, for example data logging. Each element in this array stores the four steering angles corresponding to a set Centre of Rotation (*sCR*). In the program loop, only the pointer to an array element is changed according to the required steering action. This serves two purposes. First, the computational effort within the loop is minimised since the required stepper motor positions in terms of steering steps have been pre-calculated. Secondly, moving the pointer only one step at a time ensures that the stepping sequence of the stepper motors is obeyed and steps are incremented sequentially.

### 7.2.1 Synchronisation of the Program Flow

It is important to synchronise the start of the control program loop with the capturing of the data. This is achieved by triggering the loop execution by the *SYNC* signal (see also 5.4). Since the *SYNC* signal is generated by the timer in the host processor this is easily implemented with the internal flag registers of the processor. The program flow starts by testing that no time overrun has occurred in the previous loop before reading the data from the buffers for processing. All calculations such as saving of the calculated data for post-processing and updating of the intermediate buffers for the control action data must be completed within the loop duration. The time it takes to complete all the commands within the loop effectively determines the minimum sampling time. If the program ran asynchronously, the results would be unreliable since the program might have only captured half of the data needed to determine the next control instruction. A mechanism to ensure that the program completes the execution of the entire loop is implemented. At the start of the program loop, a flag called *TM_C* is cleared by a software command. The timer of the system sets this flag at the start of the *SYNC* pulse. If on entry of the program loop the *TM_C* flag is set, a *SYNC* signal must have occurred prematurely. This indicates that the loop has not been completed and hence an emergency stop of the vehicle is initiated to avoid uncontrolled driving conditions. On the other hand, if the flag is still cleared at the start of the program loop, the software waits for the *SYNC* signal to occur before the first instruction in the loop is executed.

The *SYNC* signal also controls the maximum step rate of the stepper motors. The sampling time for this implementation is set to 5 ms which equates to a maximum step rate of 200 steps per second for the stepper motors. The acquired data is stored in the non-volatile memory for off-line data analysis by the host processor, which results in a considerable amount of overhead and prevents a further increase in the sampling rate.

### 7.2.2 Determination of the Angular Wheel Displacement

The angular wheel displacement is calculated from the difference of two angular wheel positions sampled consecutively. Particular attention must be given when a full revolution of the wheel has been completed and the encoder is reset to 0. The output from the encoder is translated in a saw-tooth like signal increasing or decreasing the number of encoder increments for clockwise or anticlockwise rotation respectively. Assuming the encoder rotates in a clockwise direction at a constant velocity, the values of the deduced encoder increment increases in a manner similar to the graph shown in Figure 40 and the incremental value is then reset to 0. When calculating the angular displacement $a$ the second sample value of 4999 increments is simply subtracted from the first sample value of 1999 increments which gives an angular displacement equivalent to 3000 increments. However, when this procedure is repeated for the sample sequence $a'$ (Figure 40) and the value of 7999 increments is subtracted from the value of 0999 increments this equates to –7000 which is obviously not correct. This problem can easily be cured by adding the number of maximum encoder increments (i.e. 0999 - 7000 + 10000 = 3000) when the difference is less than half the negative value of encoder increments.

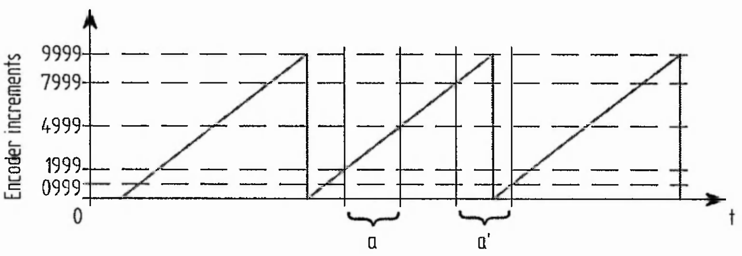


**Figure 40** Example of angular wheel rotation measured from an encoder with and without reset of the decoder

Since this is a sampled system, the Nyquist criterion applies here as well. This effectively means that the wheel must not be allowed to rotate more than half a revolution between two consecutive samples. The maximum angular velocity of the drive motor is 260 revolutions per minute (see 4.2.2) on the drive wheel hence the sampling time should not exceed 115 ms.

For a typical experiment, the vehicle should complete a minimum of four circles. Therefore, the motion controller switches the drive motor on and accumulates 70 complete wheel revolutions before the power is switched off. This effectively determines the length of the vehicle trajectory.

### 7.2.3 Procedures in the Motion Control Program

For a particular test the motion control program executes the following procedures. After initialising the program variables, the nominal Centre of Rotation, $nCR$, is chosen. The possible locations for the set Centre of Rotation, $sCR$, with the corresponding step positions of the stepper motors are calculated. Then the program enters the loop and waits for the *SYNC* pulse. On this pulse the program reads the data from the encoder and calculates the wheel increments. Depending on the wheel increments since the last sample the vehicle centre velocity is calculated and the new location of *sCR* determined. A predetermined parameter for the proportional controller determines by how much the location of the *sCR* deviates from the *nCR* for the current vehicle velocity. Then, the travelled distance is evaluated and if the maximum path length has not been reached the loop executes again until the target distance has been covered and the power to the drive system is switched off. In the current implementation, every third sample is stored into the non-volatile memory for post-processing purposes.

## 7.3 Summary Comments

The individual mechanical and electronic components have been assembled and the fully integrated system tested. During the integration phase shortcomings of the system have been identified and rectified. It has been decided to limit the voltage supplied to the drive motor so as to allow maximum vehicle velocity, which in turn demands significant variations in the steering commands.

The self-stabilising undercarriage has been evaluated on the fully integrated system in terms of stationary performance and conformity to the design specifications. The data acquisition interface and the actuator control interfaces have been linked to the motion controller hardware. Serial connections allow downloading of the software program to the motion controller and retrieving the captured data. A proportional controller has been implemented synchronising the program flow to the timing of the hardware interfaces.

# 8 Experimental Evaluation of the Methodology

The methodology for controlling the motion of an autonomous vehicle operating on the shop floor has been described in the previous chapters. Based on the measurements of the wheel rotations and angular wheel displacements the approach, estimates the current position and motion of the vehicle relative to a known reference. Any deviations from a given reference trajectory will result in new motion commands to be generated by the controller. These commands cause the steering angles of the wheels to be adjusted in a co-ordinated manner. The experimental evaluation of the proposed methodology focuses on the vehicle performance during steady state circular motion. In order to allow a linear correlation of the deduced data a proportional controller has been implemented.

## 8.1 Experimental Set-up

The evaluation of the motion controller's performance has been carried out in a laboratory environment. An area of approximately 4.5 m by 5 m has been prepared to minimise the variation of the surface condition. Experiments have been conducted with several settings of the nominal centre of rotation (*nCR*) and the set Centre of Rotation (*sCR*) has been varied as a function of the vehicle velocity. In each case, the angular wheel displacements have been assessed to determine the deduced Centre of Rotation (*dCR*) and the location of the Centre of Curvature (*CoC*) has been observed. From the results of the experiments where a minimal deviation between the *nCR* and the *CoC* was achieved, the angle $\kappa$ and the vehicle velocity $v_c$ has been determined to construct the control surface $\xi$ for the motion controller (see section 6.4). A prerequisite for adequately deducing the *dCR* is that the effective wheel radius is known.

### 8.1.1 Determination of the Effective Wheel Radius

The determination of the lateral and longitudinal velocities of the vehicle depends on the accuracy with which the angular displacement of the wheels can be translated into travelled distance. Knowledge of the effective wheel radius is paramount for an accurate experimental evaluation. In order to deduce the effective radius experimentally and to test the accuracy of the data acquisition system, the vehicle is forced by means of two linkages to rotate about a point with a known centre of rotation. The two linkages are rotably attached to the floor on one end and to the vehicle axles on the other end. One link is attached to the front axle and the other to the rear axle of the vehicle. Refer to Figure 42 and imagine the free ends of the two linkages are attached to a revolving joints, which is fixed to the floor. This helps to ensure a virtually perfect circular motion of the vehicle. The steering angles have been set to comply with the rolling compatibility condition and the wheel increments are measured for all wheels while the vehicle is pushed slowly to follow a circular path. Since the ratios between the wheel increments must be proportional to the ratios of their distances to the *CoC*, any deviations in the ratios highlight deficiencies in terms of the wheel radii and the data acquisition system. Table 1 summarises the results and the deviations from the mean radius of wheels W2, W3 and W4 (refer to Figure 41) where $|RW_i|$ denotes the radius from the centre of rotation $R$ to the wheel $W_i$ for $i = 1..4$.

Centre of Rotation: (-0.07 – j1.00) m

| Wheel | $\lvert RW_i \rvert$ [m] | Wheel increments | Wheel radius [m] | Deviation of radius to mean radius of W2..W4 |
|---|---|---|---|---|
| W1 | 1.264 | 203922 | 0.06252 | 0.800% |
| W2 | 1.231 | 196856 | 0.06225 | -0.044% |
| W3 | 0.822 | 131419 | 0.06257 | -0.074% |
| W4 | 0.872 | 139561 | 0.06245 | +0.118% |
| mean radius of W2, W3 and W4 | | | 0.06252 | |

**Table 1** Comparison of the effective wheel radius

Note that wheels W2 to W4 are the free running wheels while wheel 1 has the drive motor and a gearbox attached. The wheel radius is calculated from the incremental values and the measured radius relative to a nominal centre of rotation located at (-0.70 -j1.00) m.

## 8.2 Mapping of the Centre of Rotation

In order to test the performance of the controller it is necessary to observe the motion of the vehicle while it is rotating freely about a virtual centre of rotation. Of particular interest is the lateral motion of the vehicle while it is circling steadily.

This allows extended data capturing periods and in turn improves the accuracy of the measurements. In order to observe how well the controller can maintain a given centre of rotation the vehicle's centre of curvature needs to be tracked.

Tracking of the centre of curvature (*CoC*) can be achieved by mounting a pen at the nominal centre of rotation (*nCR*) onto the two linkages attached to the vehicle axles. Figure 42 shows the vehicle with the pen mounted at the (*nCR*) ready for an experiment. When the vehicle moves in a circular path the location of the pen and the line plotted in relation to the actual *COC* can be observed. In an ideal situation, the predetermined *nCR* should coincide with the *CoC* so that the pen rotates at a fixed point. Hence, the smaller the radius described the better the performance of the motion controller. However, if the vehicle describes a radius slightly larger than the |*nCR*| the resulting plot would be a circle (Figure 41) whose centre represents the actual centre of curvature. In this case, the steering commands are insufficient to compensate for the lateral motion and hence the controller is under-compensating. If in a different situation the vehicle describes a radius smaller than the expected |*nCR*|, then the pen would plot a circle with the centre (i.e. the *CoC*) towards the vehicle and the controller is over-compensating.

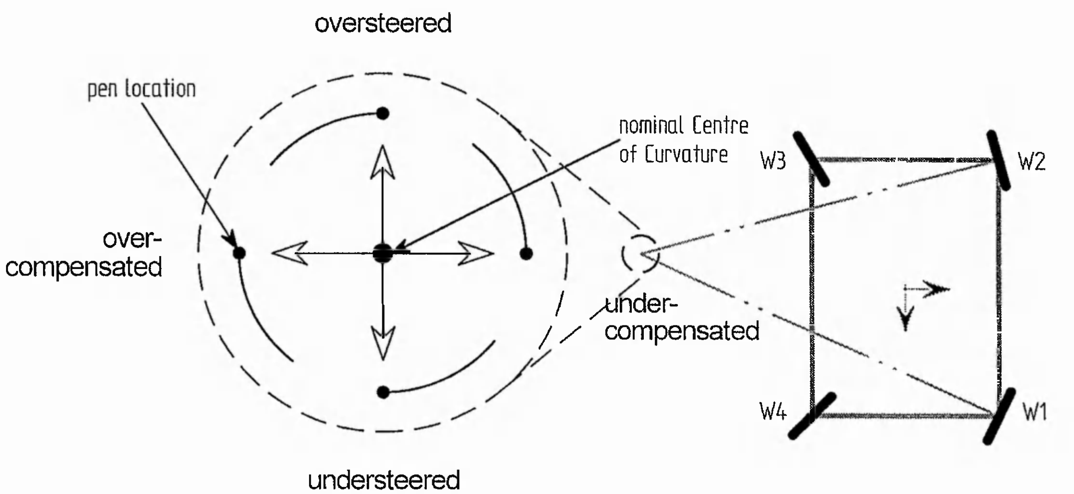


**Figure 41** Method for tracing the nominal centre of rotation with the pen

If the *CoC* is located towards the back of the vehicle it is deemed to be under-steered, that is only the wheels *W1* and *W4* on the front axle are describing a radius larger than required for the given *nCR*. The opposite is true when the *CoC* is located towards the front of the vehicle.

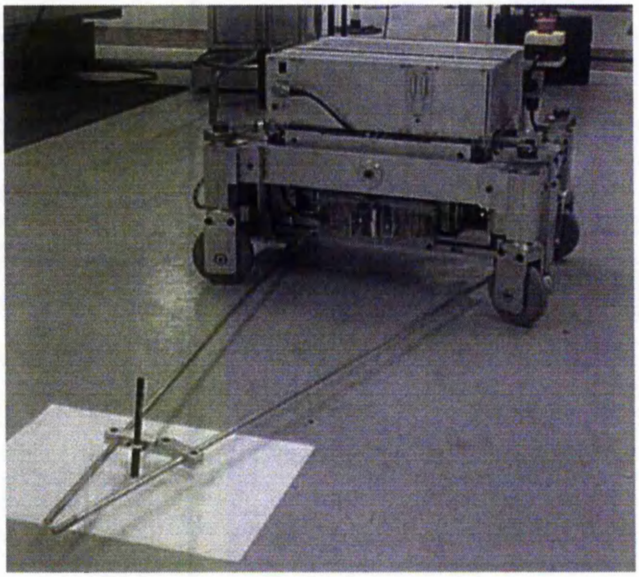

**Figure 42** Experimental set-up for tracing the nominal centre of rotation with the pen mounted between the two rods

### 8.3 Experimental Procedure

The experimental procedure is as follows. First, the pen is mounted at the nominal centre of rotation for an experiment and a sheet of paper fixed on the floor. Then the PCs are connected via the serial cable to the motion controller and the control program downloaded. Upon completion, the cables are disconnected and after a pre-programmed delay the motion controller adjusts the steering angles to the nominal settings and then switches the power supplied to the drive motor on. After the vehicle has travelled the pre-programmed distance the power to the drive motor is switched off and the steering angles are set to adjust to their nominal positions.

After the vehicle has stopped the two external PCs are connected to the motion controller and a program is downloaded which transfers the data in the non-volatile memory to the second PC for further analysis. Typically, there are approximately 1400 to 1700 data sets stored in the memory depending on the vehicle velocity chosen for the experiment. The distance the vehicle travelled is determined by the pre-programmed 70 wheel rotations of *W1*, which equates to approximately 27.5 m for all experiments. This is to allow at least four circles to be completed for the experiments with a larger *nCR*. Each sample data set consists of the four steering positions and the four angular wheel positions.

Figure 43 shows the wheel velocities deduced from the measured encoder increments and the steering actions for a typical experiment. The cyclic variation of the wheel velocities, once the vehicle has reached the predetermined velocity is mainly attributable to the unevenness of the floor. The floor level increases gradually by about 20 mm over three quarters of the circular path and falls back to its original level in the last quarter. The reduction of the velocity over the last quarter has resulted in a small variation to the set steering angle. A number of experiments with different settings of the nominal Centre of Rotation (*nCR*) have been carried out (see also Appendix D). For each case, a factor which determines by how much the angular velocity of wheel *W1* must alter before the set Centre of Rotation (*sCR*) is to advance to the next possible compatibility condition has been varied. The factor has been manually adjusted to yield a minimum deviation between the *nCR* and the CoC. The results from the experiments are used to construct the control surface for the motion controller (see also section 6.4). As an example the wheel velocities and the difference between the theoretical and the deduced steering angle settings for *nCR* = (-0.2 - j0.75) m is shown in Figure 43, which comprises 1499 deduced data sample sets.

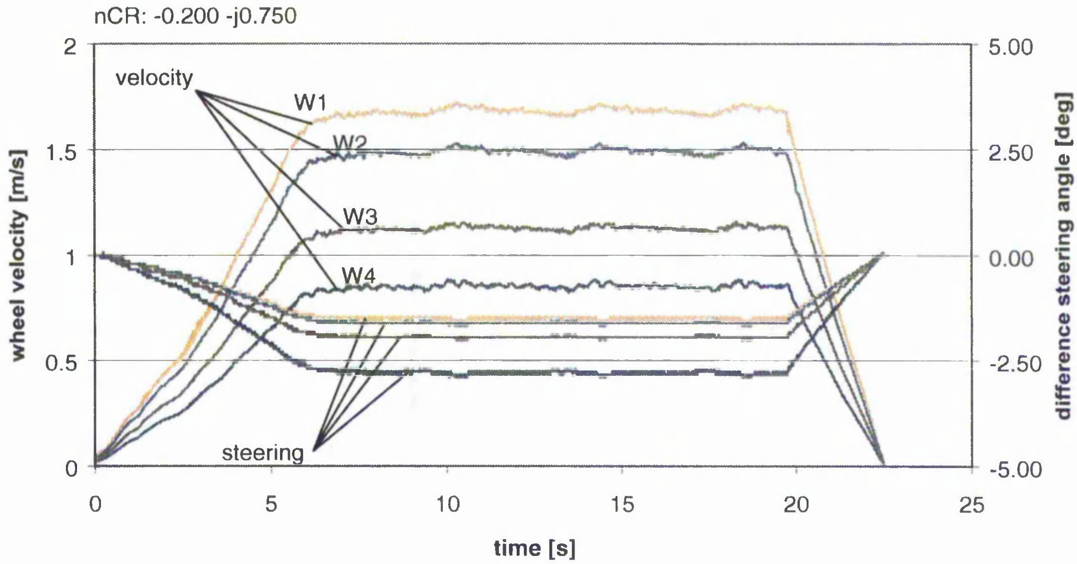


**Figure 43** Deduced wheel velocities and the difference in steering angle settings

During a typical experiment, the nominal centre of rotation, *nCR*, is prescribed and the vehicle is allowed to drive in a circular motion. While turning, the Centre of Curvature, *CoC*, is traced and the angular wheel displacements are deduced from the wheel rotation measurements. The performance of the controller is evaluated by calculating the difference between the *nCR* and the *CoC*. An experiment is conducted initially with the proportional controller disabled and the steering angles held at their theoretical values. This is done to establish a benchmark for the controller. Then the controller is enabled and the gain for the proportional controller varied until a reasonably small deviation between the *CoC* and the *nCR* has been achieved. The results to be presented in the following sections are for the vehicle circling steadily.

Figure 44 shows the trace produced by the pen mounted at the nominal centre of rotation when the controller was disabled. In this case the *nCR* is located on a line midway between wheel 3 and wheel 4 of Figure 41. The velocity of the vehicle centre during steady state circular motion averaged at approximately 0.6 m/s. It can be seen that with a controller the trace of the *nCR* spirals outwards while the vehicle accelerates and continues at an almost constant radius with the *CoC* drifting slightly to one side.

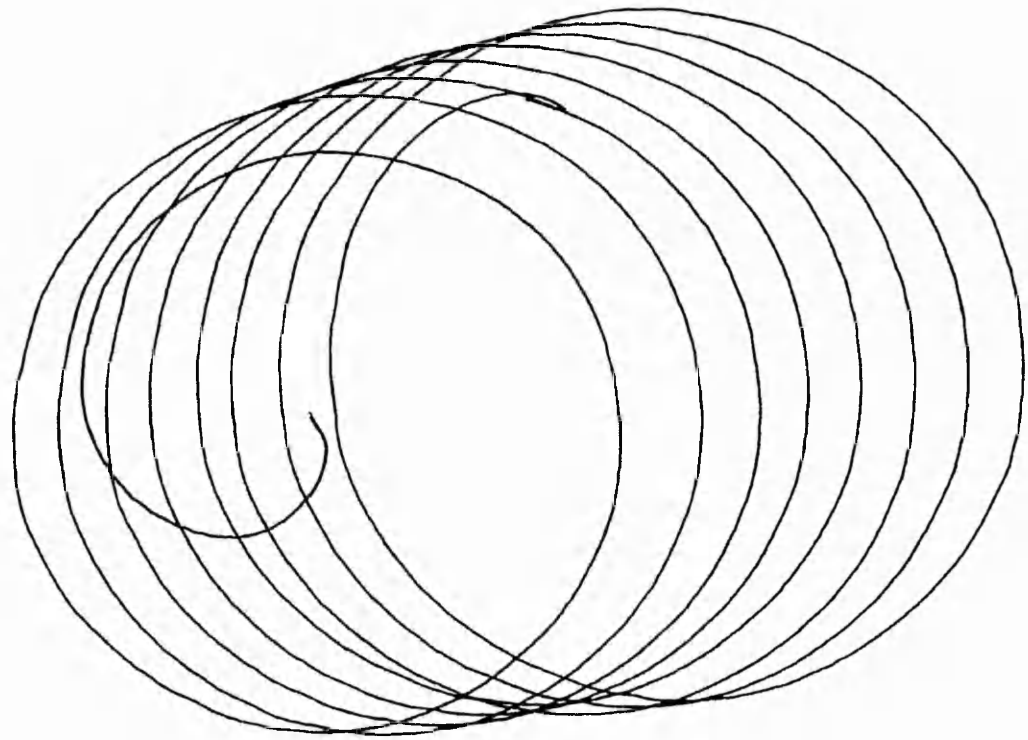

**Figure 44** Trace of the nominal centre of rotation with nCR = -j0.21 m and the controller disabled

In comparison, Figure 45 shows the trace of the *nCR* when the controller was enabled and a gain for the proportional controller has been programmed. The function of the proportional gain is to adapt the angle $\kappa$ depending on the increments per sample from the driven wheel 1. For the trace shown in Figure 45 the controller adjusted the angle $\kappa$ between the *nCR* and *sCR* by $-18.1$ degrees. The average velocity of the vehicle centre was 0.64 m/s for a complete circle.

**Figure 45** Trace of the nominal centre of rotation with nCR = -j0.21 m and the controller enabled

With the *nCR* chosen at -j0.21 m, the two wheels on the right hand side of the vehicle (wheel 1 and 2, Figure 40) were parallel to the vehicle axles and remained largely unaffected by the variation of the *sCR*. Only the two wheels on the left-hand side of the vehicle (wheel 3 and 4) needed to be significantly adjusted to compensate for the lateral vehicle motion.

Figure 46 shows the trace of the *nCR* which was set to (-0.2 - j0.75) m when the steering controller was disabled. It can be seen that the drift of the *CoC* has been significantly reduced. Indeed, similar results were obtained for experiments where the *nCR* was located further away from the vehicle centre (see Appendix D).

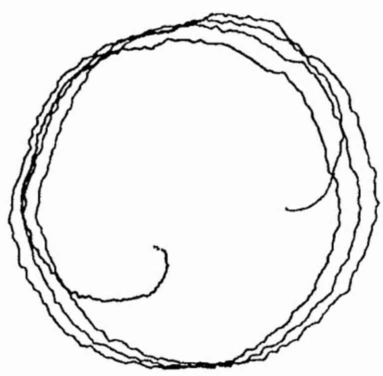

**Figure 46** Trace of the nominal centre of rotation with nCR = (-0.2 - j0.75) m and the controller disabled

Figure 47 shows the trace at identical setting for the nCR = (-0.2 - 0.75) m but with the controller enabled. A total of 1423 data sets containing four angular wheel positions and four steering angles have been downloaded for the off-line analysis. The figure also shows that during the first quarter where the vehicle increases its velocity, the controller adjusts the steering angles to compensate for any change in the lateral motion and hence the circular path.

**Figure 47** Trace of the nominal centre of rotation with nCR = (-0.2 - j0.75) m and the controller enabled

From the wheel increments the location of the deduced centre of rotation, *dCR*, has been estimated with equation (27) and consequently the velocity of the vehicle centre deduced with equation (12). Figure 48 shows the real part and the imaginary part of the complex position vector *dCR* and the vehicle velocity the duration of the experiment. Noticeable increased noise levels can be seen during the start-up and slow-down periods. A marginal cyclic variation of the maximum vehicle velocity is also visible. Figure 47 also shows the increased vehicle velocity which also corresponds to a distortion of the circle.

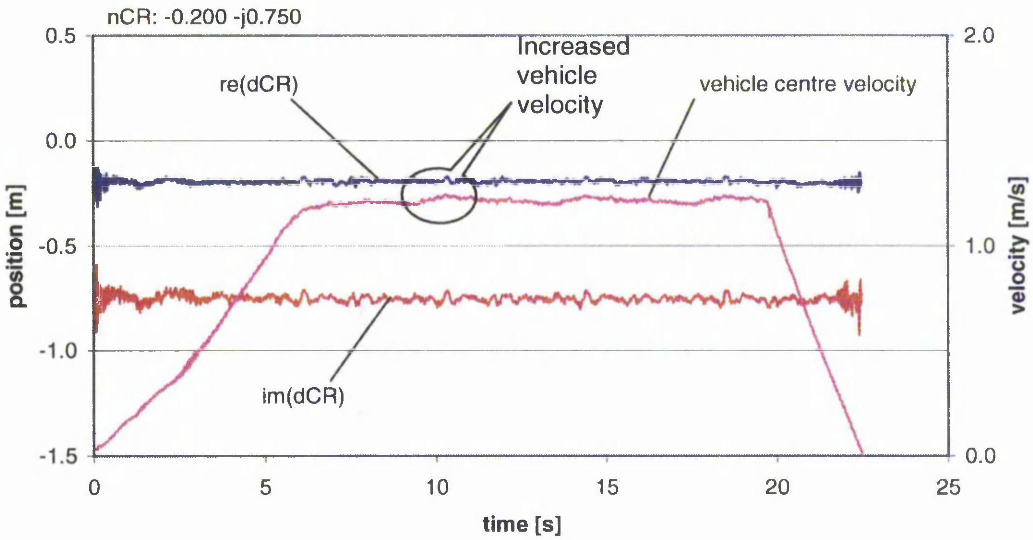


**Figure 48** Deduced centre of rotation and vehicle velocity from the measurements of the wheel rotations

The deviation between the *dCR* and the *nCR* over the duration of the experiment is shown in Figure 49. Once the maximum vehicle velocity has been reached, the noise of the signal remains generally within ± 25 mm (3.2%) of the *nCR* position. In order to evaluate the trend, the signal is averaged over 100 samples, equivalent to 1.5 seconds.

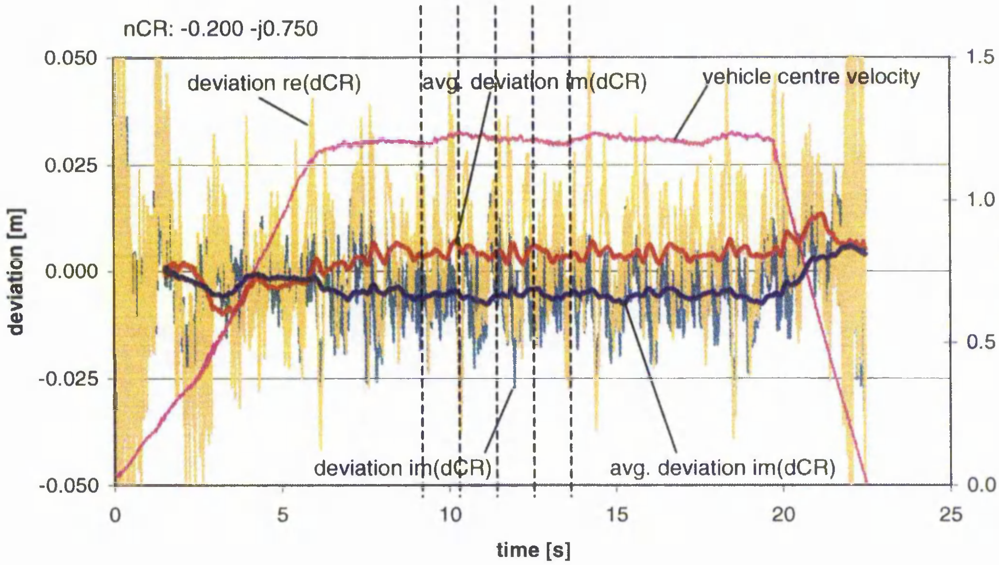


**Figure 49** Deviation between the deduced centre of rotation dCR and nominal centre of rotation nCR averaged over 1.5 seconds

From the deduced wheel displacements a trace of the deduced centre of rotation can be reconstructed by employing equations(15) – (19) as shown in

Figure 50.

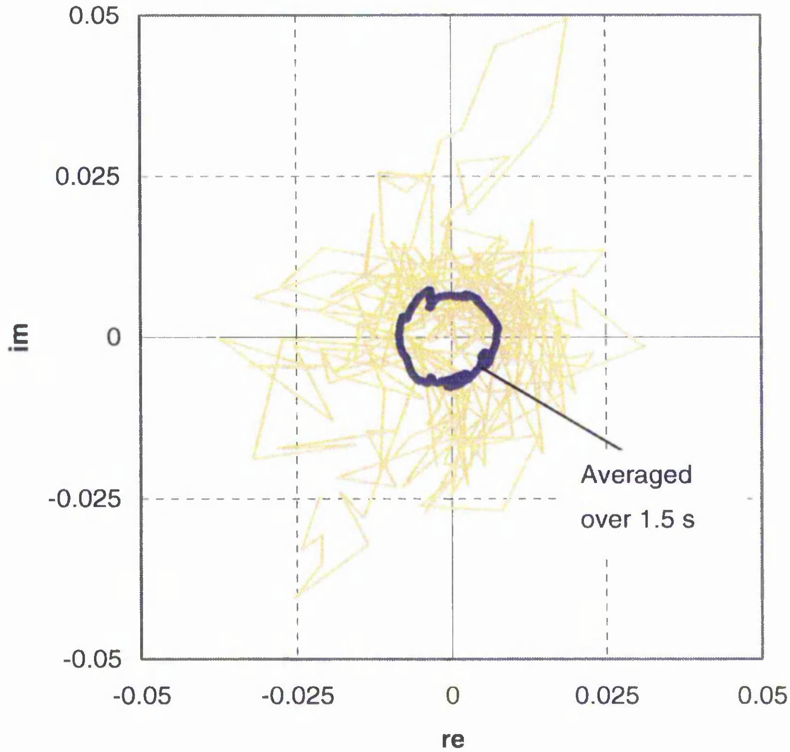


**Figure 50** A trace of the deduced centre of rotation reconstructed from the unfiltered experimental data and averaged over 1.5 seconds

The trace is plotted using unfiltered experimental data averaged over 100 samples. It can be seen that the trace from the averaged data describes a circle with reasonable accuracy. The capability of the proposed method to deduce the vehicle centre of rotation becomes apparent when the trace of *dCR* in Figure 50 is compared with that of the *nCR* shown in Figure 47. The good agreement suggests that the proposed motion control strategy is acceptable.

Figure 51 provides a summary of the results from the experiments with *nCR* at (0.2 m - j0.75) m. A set of the Centre of Rotation *sCR* and deduced Centre of Rotation *dCR* is shown for different settings which reflect different gain values for the proportional controller to minimise the deviations between the *nCR* and *dCR*. The location of the *dCR* for the experiment with the disabled controller is also shown.

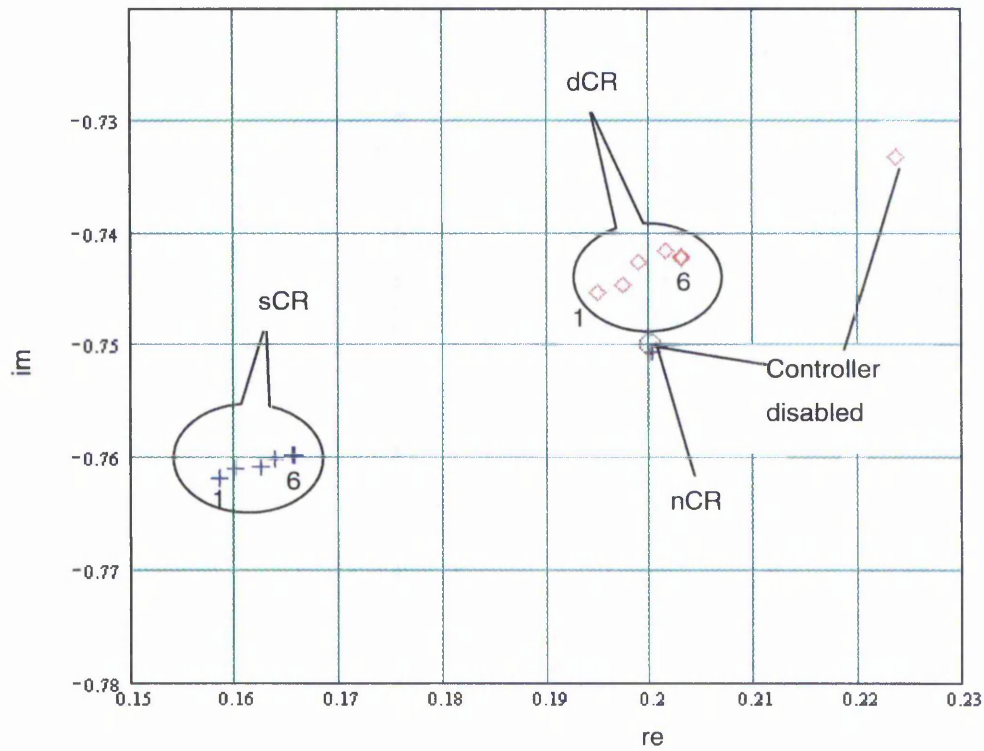


**Figure 51** Locations of nominal, set and deduced centre of rotation from the measurements averaged over one 360 degree turn

Figure 52 shows a 3-D view of the reconstructed control surface from the deduced angle $\kappa$ (Equation (32)) where $\eta = 1.0$ has been chosen. The value of $\eta$ is to be altered by the motion controller to adapt to changes in surface conditions. The dots represent the locations of the nominal Centre of Rotation and the height indicates the angle $\kappa$ required to maintain the *nCR* for a vehicle centre velocity of 1 m/s. The rectangle represents the vehicle outline.

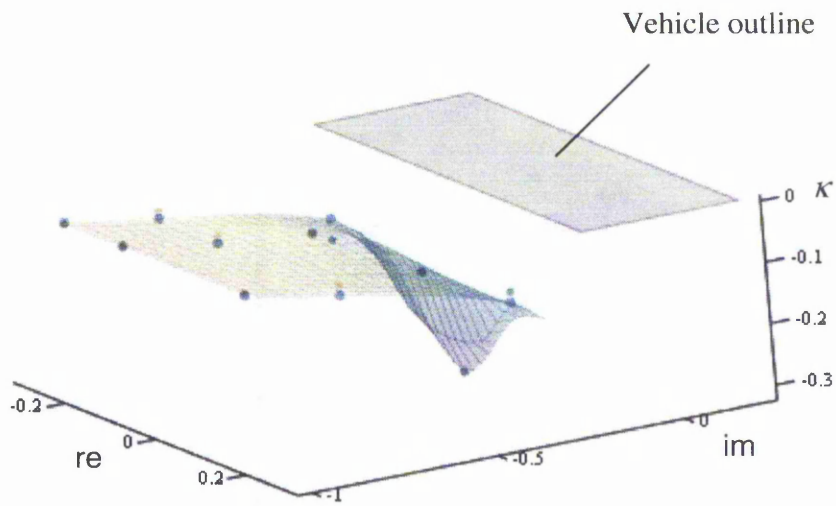


**Figure 52** 3D view of the steering angles for constructing the control surface

Table 2 shows a collection of the experimental data of the experiments where minimum deviation between the deduced and the nominal CR has been achieved by the controller. The percentage of the deviation between the magnitude |nCR| and |dCR| is given. Therefore, the deviation has been subtracted from the magnitude of |nCR| and the percentage difference is given in the last column of the table.

| nCR | \|nCR\| | DCR | \|dCR\| | \|nCR\| - \|dCR\| | Measured deviation | \|CR\| corrected | $\|CR\|_{corr.}$ - \|dCR\| |
|---|---|---|---|---|---|---|---|
| **0.00 - j0.21** | 0.210 | -0.001 – j0.212 | 0.212 | -0.86% | 0.003 | 0.213 | 0.56% |
| **0.00 - j0.30** | 0.300 | 0.001 – j0.298 | 0.298 | 0.57% | 0.002 | 0.298 | -0.10% |
| **-0.20 - j0.30** | 0.361 | -0.211 – j0.297 | 0.365 | -1.12% | 0.005 | 0.366 | 0.26% |
| **0.20 - j0.30** | 0.361 | 0.188 – j0.293 | 0.348 | 3.45% | 0.007 | 0.354 | 1.54% |
| **-0.07 - j0.50** | 0.505 | -0.067 – j0.499 | 0.504 | 0.25% | 0.006 | 0.499 | -0.95% |
| **-0.20 - j0.75** | 0.776 | -0.194 – j0.751 | 0.776 | 0.04% | 0.008 | 0.768 | -1.00% |
| **-0.07 - j0.75** | 0.753 | -0.065 – j0.751 | 0.754 | -0.03% | 0.005 | 0.758 | 0.63% |
| **0.20 - j0.75** | 0.776 | 0.195 – j0.745 | 0.770 | 0.84% | 0.002 | 0.774 | 0.58% |
| **-0.20 - j1.00** | 1.022 | -0.198 – j1.013 | 1.032 | -1.21% | 0.016 | 1.036 | 0.35% |
| **-0.07 - j1.00** | 1.002 | -0.069 – j0.993 | 0.995 | 0.70% | 0.005 | 0.997 | 0.20% |
| **0.20 - j1.00** | 1.020 | 0.191 – j1.012 | 1.030 | -0.98% | 0.017 | 1.037 | 0.68% |

**Table 2** Deviation of the deduced CR from the nominal CR and the corrected CR

## 8.4 Summary Comments

The constituent parts of the experimental test platform have been integrated and the functions of the motion control system have been verified. The effective wheel radii have been determined experimentally, which in turn enabled an evaluation of the overall system functionality and accuracy. For the experimental set-up a simple method of tracing the centre of curvature with a pen mounted at the nominal centre of rotation of the vehicle has been proposed in order to evaluate the performance of the integrated motion control system.

The experimental data is temporarily stored in the onboard memory and can be downloaded after completion of the experiment for further data analysis. Experiments have been carried out with the steering angles set to their theoretical values for the nominal centre of rotation with, and without the proportional steering controller being enabled. The results showed a significantly reduced deviation from the nominal centre of rotation can be achieved with the controller enabled.

The data of the angular wheel positions and steering angle settings has been downloaded for detailed off-line analysis. From the angular wheel displacements the location of the deduced centre of rotation of the vehicle has been estimated. Good agreement is achieved between the deduced and actual centre of rotation for the averaged measurements over 1.5 seconds. A number of different *nCR* locations have been investigated and the corresponding required steering angles have been plotted to form a control surface for the motion controller. The deviation between the estimated and the corrected magnitude of the centre of curvature was less than 1.6% and adequate results could be obtained with the proportional steering controller.

# 9 Discussion

Attempts to automate material handling systems began in the early 1950's when trucks were used to tow rolling loads following wires embedded in the factory floor. Many motion control systems for Autonomous Guided Vehicles (AGVs) still rely on external guidance systems to deduce the deviations from the intended path. The industry has favoured two guidance methods: following an embedded wire or a reflective tape; triangulation of the current AGV position utilising ultrasonic, infrared or microwave sensors for motion control purposes.

Although fixed guidance paths provide an accurate lateral reference for vehicles operating on the shop floor, such external guidance systems may attract high costs and delays are likely should the material transfer system be altered periodically. Triangulation methods employed to map the AGV position using distributed beacons or sensors have similar disadvantages since changes to the layout may require relocations or additions of sensors. The literature reports that a navigation system which is based on triangulation method requires that there should be at least three beacons 'visible' to the vehicle at any time to ensure safe operation. As an alternative, tracking the position utilising accelerometers has been suggested. However, for vehicles driving at low speeds and moderate accelerations typical for a shop floor environment, the measurements can be adversely affected by the undulations of the floor and hence deducing the position of the vehicle accurately becomes very challenging. Upon investigation, a method for deducing the position of an autonomous vehicle in a reasonably even shop floor environment in relation to a known reference is proposed. The method utilises the wheel rotation and more specifically the angular displacement of the wheels to track the longitudinal and lateral motions of the vehicle. The knowledge thus gained is used to devise appropriate remedial actions in order to minimise any deviations from a given reference trajectory.

In the present investigation, the vehicle is required to follow a circular path without the use of external guidance systems. Assuming the vehicle moves in a horizontal plane and that the displacement of the wheels can be deduced with reasonable accuracy, then the centre of rotation and the yaw angle can be estimated analytically.

The current position of the vehicle can be ascertained in relation to a known reference by tracking its centre of rotation and the angle swept. A principal requirement of the method is that the steering angles of all four wheels must be capable of being adjusted continuously and in a parallel manner to satisfy the rolling compatibility condition. It has also been proposed to utilise the angular wheel displacements as an approximate measure of the lateral motion of the vehicle during cornering. The difference between the nominal centre of rotation and the estimated centre of rotation can be used to deduce the vehicle's float angle. Knowledge of the float angle, the vehicle velocity and the nominal centre of rotation allows an estimate to be made of the road/tyre characteristics. This in turn permits the motion controller to determine the steering angles of the wheels so that the deduced centre of rotation coincides with the nominal centre of rotation. Should the surface characteristics vary the motion controller is able to adaptively update the parameters according to the current vehicle's float angle.

To test the proposed methodology it was necessary to design and manufacture an experimental vehicle. The impact of roll and pitch motions during tight cornering and acceleration/deceleration respectively has to be avoided to minimise any adverse effects on the experimental results. Preliminary experiments have been carried out with a rigid vehicle undercarriage. The results showed that vertical wheel displacements could cause individual wheels to lose ground contact and render the results invalid. Hence, it was considered important for the main investigation that the vehicle must maintain perennial road/tyre contact. To this end a self-stabilising undercarriage has been designed and implemented. A set of equations has been developed to describe the displacement of the mechanism, which was found to be very useful in the design of the undercarriage and its subsequent validation. The performance of the undercarriage was evaluated by means of computer simulation of the vehicle traversing a hump at different speeds. The results showed that the displacement of the vehicle body is halved in relation to the height of the hump. When the vehicle operates on a reasonably level floor the effects are negligible small and have consequently been neglected in the analysis.

The proposed methodology also requires that all four wheels must be capable of being steered independently. The desired adjustment of the steering angles of the wheels is achieved by means of stepper motors because they can provide a higher torque for a given size. It is imperative that the implementation of the steering commands must be synchronised to ensure rolling compatibility for all four wheels and to facilitate the determination of the centre of rotation.

The velocity of the vehicle has not been explicitly controlled but taken as an input parameter for the motion control system. This approach is considered acceptable since the velocity at which the vehicle is travelling is not critical provided there is sufficient adhesion potential. From the experiments conducted by Bachmann et al. (1995) with a passenger car, it was suggested that under normal driving conditions sufficient adhesion potential can be assumed. Furthermore, as long as a sufficient safety margin is maintained Boyden and Velinsky (1994) suggested that the performance of the motion control system is not likely to be significantly affected.

The emphasis of this investigation has been placed on minimising any deviations from a given reference trajectory based on the assumption that sufficient adhesion potential is available. The amount of available friction between the road and the tyre interface needs to be taken in consideration when generating the reference trajectory and hence no effort has been made to optimise the available friction. Furthermore, while the friction is proportional to the normal forces acting on a wheel, these forces have been considered as equal for all wheels and hence only a single parameter describing the road/tyre characteristics is deduced. It is generally accepted that during tight cornering the weight transfers from the wheels closer to the centre of rotation towards the outer wheels. However, this weight shift accounts for less than 5% of the nominal force, which is significantly lower than the margin of the available friction and has consequently been neglected.

For the present investigation, it is considered that the use of a simple proportional controller should suffice, as the principal requirement is to keep the autonomous vehicle moving along a given trajectory. However, there exits only little knowledge on the difficult task of compensating the lateral vehicle motion.

The experimental results demonstrated that individual wheels stall intermittently when the self-stabilising undercarriage was disabled so that it behaved like a rigid vehicle. The diagonally opposite wheels also showed irregularities in the wheel rotation measurement and it was not certain which of the wheels lost contact with the floor. The consequently, the uncertainty in deducing the wheel rotation and hence the centre of rotation renders the tracking of the vehicle position invalid. When the self-stabilising undercarriage was enabled irregularities in the wheel rotation measurements were not noted.

All experiments have been carried out on an industrial PVC floor, which provided a realistic surface/tyre combination. Although the vehicle experienced minor surface undulations of the floor, analysis of the signals suggested that the impact was sufficiently small to have any adverse effects on the experimental data. Such surface undulations are commonly encountered in a practical environment.

Evaluation of the controller by using a pen for tracing the nominal centre of rotation is found to be well suited for steady state operation of the vehicle. Though simple, the method adequately shows that the motion controller has achieved a close match between the nominal centre of rotation and the centre of curvature. By observing the position of the pen with respect to the centre of the traced circle, it is possible to obtain an estimate of the offset between the actual centre of rotation and the nominal centre of rotation.

Strong emphasis has been placed on the development of the functionality of the individual constituent subsystems and their compatibility with the mechatronic system. Consequently, a proportional control algorithm has been shown to be adequate in determining the steering commands for compensating the lateral motion.

The current implementation of the controller is unique to this vehicle. This is because the inertia of the vehicle and the steering dynamic has not been explicitly isolated in the equations for the controller, although the length and the width of the vehicle (i.e. wheelbase and track) have been taken into account. However, further work is necessary to determine the maximum velocity, when changing the location of the centre of rotation without the danger of saturating the friction potential. This velocity must be determined in advance to ensure that the dynamic of the steering system is not saturated. Furthermore, the available friction potential should not be utilised beyond the 43%, equivalent to the potential for a standard driver in a passenger car. This leaves sufficient scope for the controller to adaptively set the appropriate steering commands. Therefore, the results of this investigation should provide a basis for designing similar autonomous vehicles.

The outcome of this investigation is twofold. Firstly, the proposed method is intended to minimise any deviations of the vehicle a given reference path. This is accomplished by taking into account a priori an estimated of the road/tyre interface characteristics when determining the new steering commands required to compensate for anticipated lateral motion likely to occur as a result of the centrifugal acceleration during cornering. Secondly, knowledge of the road/tyre interface characteristics, once estimated, can be taken into account when generating a new reference path. This enables a more accurate determination of the steering and propulsion commands required and more importantly avoids the saturation of the steering dynamics and hence excessive use of the adhesion potential. The results have shown that the developed methodology offers potentially an alternative method for controlling the position and motion of an autonomous vehicle operating on the shop floor.

# 10 Conclusion and Further Work

The aim of this investigation is to develop a simple but reliable method for quantifying road/tyre interaction to facilitate adaptive motion control for an autonomous vehicle operating on the shop floor. The proposed methodology fulfils this aim by integrating three main building blocks, namely, a method for deducing the vehicle motion and determining the control actions, a self-stabilising undercarriage to ensure permanent ground contact and a data acquisition interface to facilitate parallel data sampling and parallel actuation of steering commands.

A new approach has been presented for deducing the position and motion of an autonomous vehicle with respect to a known reference position. The mathematical model is based on the kinematics of the vehicle and does not require measuring deviations from external guidance paths or the integration of vehicle accelerations to determine the vehicle position. Deriving the governing equations with the assumption that the vehicle is rotating about a point has helped to derive efficient mathematical equations for determining the steering angles and to satisfy rolling compatibility for a real-time motion controller.

For the performance evaluation of the proposed methodology a four wheel independently steered model autonomous vehicle was designed and manufactured, covering both mechanical and electronic aspects. The experiments were conducted during steady state circling of the vehicle. The results have shown that the developed methodology offers potentially a viable alternative to conventional guidance techniques.

The methodology seeks to deduce the vehicle position and motion in both longitudinal and lateral directions based on the measurements of angular wheel displacements and revolution. The level of accuracy with which the position can be deduced is considered to be adequate to enable the vehicle to operate autonomously between despatch locations.

The ability of the motion control system to adapt to variations of the road/tyre interface characteristics is another important feature. The difference between the nominal and the deduced centre of rotation allows estimates to be made of the road condition. These estimates can be used in the determination of the new steering commands to minimise any deviations. Furthermore, knowledge of the surface characteristic can aid the generation of appropriate reference paths.

The self-stabilising vehicle undercarriage enables all four wheels of the vehicle to remain floor engaging at all times. In situations where the vehicle has to traverse a surface undulation, the vertical displacement of the vehicle body has been deduced to approximately a quarter of the vertical wheel displacement. The results from computer simulation of the vehicle traversing a hump have shown that the performance objectives of the undercarriage have been met.

Additionally, the work has led to the development of a data acquisition interface capable of sampling the angular wheel positions simultaneously. Parallel sampling of the angular wheel position and parallel actuation of all steering motors improves the accuracy of the system state and gives a higher degree of certainty.

### 10.1 Suggestion for Future Work

At present, the experiments have focussed on steady state circling. More work needs to be done to improve the controller for transient performance when the vehicle enters or exits a turn. Because of the relatively complex road/tyre interaction when the vehicle enters a turn a more advanced controller implementation may be required. The controller should then also include the steering system dynamics such as stepper motor inertia and the maximum steering motor accelerations.

The mathematical model should be expanded so that the inertia of the vehicle can be taken into account to further improvements of the vehicle response when the surface characteristics change. The estimation of the friction characteristics should be improved so as to cater for utilising the friction potential in excess of the adhesion potential utilised under standard driving conditions.

The control methods may also be an area worth pursuing where the set Centre of Rotation is not only varied to compensate for the lateral vehicle motion, but also to compensate for the yaw inertia when changing the travelling direction of the vehicle.

An investigation of the current hardware limitations may enable to understand the implications if the proposed control methodology were to be employed at higher speeds and with much more dominant vehicle inertia.

Further work should also include the effects of varying effective wheel radii on the accuracy with which the position of the vehicle can be determined. Fuzzy logic may be able to cope with the non-linearity expected from the relationship between the wheel load and the effective wheel radius.

Furthermore, during docking operations the vehicle's traction needs to be controlled with a high degree of accuracy so as to exactly position the vehicle for example in front of a loading bay of a material handler. To achieve this, short range guidance systems may be incorporated which can correct any accumulated error and hence helps to avoid collision between the vehicle and the docking station. The velocities encountered during such docking operations are expected to be relatively low and special procedures may be needed for such docking manoeuvres.

# 11 References


Adachi, K., Ito, K. and Fujishiro, T., 'Study on a new four-wheel-steering control method at low speeds - front-end path memorising', Nissan Motor Co. Ltd., Japan, p. 214-218 ........ 34

Albrecht, T., 1995, SIMATIC PLC Monitors and Steers Automatic Guided Vehicles at South-East Asian Container Port, engineering & automation, XVII, No. 6, p. 24-25 ........ 9

Alexander, J. C. and Maddocks, J. H., 1989, 'On the Kinematics of Wheeled Mobile Robots', The International Journal of Robotics Research, 1989 Massachusetts Institute of Technology, Vol. 8, No 5, p. 15-27 ........ 5, 28, 37

Astolfi, A.(1), 1995, 'Exponential stabilization of a car-like vehicle', Proceedings of the Conference on Robotics and Automation, Nagoya, Japan ........ 19, 81

Astolfi, A.(2), 1995, 'Exponential stabilization of a mobile robot', 3rd European Control Conference, Rome, Italy ........ 19

Astolfi, A.(3), 1995, 'Exponential stabilization of nonholonomic systems via discontinuous control, IFAC Symposium on Nonlinear Control System Design, Lake Tahoe, Canada ........ 19

Astolfi, A., 1999, 'Exponential stabilization of a wheeled mobile robot via discontinuous control' ASME, Journal of Dynamic Systems Measurement and Control ........ 20

Bachmann, T., Bielaczek C. and Breuer, B., 1995, 'Der Reibwert zwischen Reifen und Fahrbahn und dessen Inanspruchnahme durch den Fahrer', ATZ Automobiltechnische Zeitschrift 97, No. 10, p. 658-666 ........ 23, 122

Barraquand, J. and Latombe, J-C., 1989, 'On Nonholonomic Mobile Robots and Optimal Manoeuvering', Proceedings of the Intelligent Control Workshop, IEEE 1989, p. 340-347 ........ 18

Beaufrere, B. and Zeghloul, S., 1995, 'A mobile robot navigation method using a fuzzy logic approach', Robotica, Vol. 13, Cambridge University Press, p. 437-448 ........ 15

Bosch, 1996, 'Automotive Handbook', Rober Bosch GmbH, ISBN -8379-0333-1 ........ 27

Bostel, A. J., Gan, W. W., Sagar, V. K. and See, C. H., 1994, 'Generation of Optimal Routes in a Neural Network Based AGV Controller', Intelligent System Engineering, Conference Publication No. 395, IEE, 1994, p. 165-170 ........ 1, 16, 17

Boyden, F. D. and Velinsky, S. A., 1994, 'Dynamic Modeling of Wheeled Mobile Robots for High Load Applications', IEEE 1994, p. 3071-3078 .... 32, 122

Brockett, R. W., 1983, 'Asymptotic Stability and Feedback Stabilization', In Differential geometry control theory, Vol. 27, p. 181-191 ........ 19

Brown, G. L. and Hung, J. C., 1994, 'A Mathematical Model for Vehicle Steering Control', IEEE 1994, p. 2027-2031 ........ 30

Brown, G. L. and Hung, J. C., 1995, 'Vehicle Steering in Extreme Situations Using a Fuzzy Logic Controller', IEEE 1995, p. 225-229 ........ 30

Brown, M. and Harris, C. J., 1991, 'Intelligent Control for Autonomous Vehicles using Real-Time Adaptive Associative Memory Neural Networks', Proceedings of the 2nd International Conference on Artificial Neural Networks, p. 144-149. ........ 34

Burke, T., 1994, 'Design of a Modular Mobile Robot', PhD Thesis, University of Oxford ........ 20, 33

Cameron, S. and Probert, P., 1993, Advanced Guided Vehicles: Aspects of the Oxford AGV Project, World Scientific Publication Company, Singapore.... 25, 39

Cha, Y. Y. and Gweon, D. G., 1996, 'Local path planning of a free ranging mobile robot using the directional weighting method', Mechatronics Vol. 6, No. 1, p. 53-80 ........ 18

Cheng, R.M.H., Mehrabi, M.G. and Hemami, A., 1994, 'Linear and nonlinear models of automated vehicles analysis and experiments', IFAC Emerging Intelligent Control Technologies, Hong Kong, p. 207-212 ........ 26

Cherry, A. S. and Jones, R. P., 1995, 'Fuzzy logic control of an automotive suspension system', IEE Proceedings, Part D, p. 149-160 ........ 26

Cherry, A. S., Costas, A. and Jones, R. P, 1995, 'MBS Moddeling, Non-Linear Simulation And Linear Analysis Techniques For Integrated Vehicle Control', Proceedings of IFAC Workshop on Advances in Automotive Control, 76-81 5, 26

Choi, C. and Lee, J-J, 1996, 'Dynamical Path-Planning Algorithm of a Mobile Robot: Local Minima Problem and Nonstationary Environments', Mechatronics, Vol. 6, No. 1, p. 81-100 ........ 18

DeSantis R. M., 1995, 'Path-Tracking for Car-Like Robots with Single and Double Steering', IEEE Transactions on Vehicular Technology, Vol. 44, No. 2, p. 366-377 ........ 13

Duckett, T. and Nehmzow, U., 1997, Quantitative Analysis of Mobile Robot Localisation Systems', British Conference on Autonomous Mobile Robotics and Scientific Methods in Mobile Robotics', ISSN 1361-6161 ........ 13

Duckett, T. and Nehmzow, U., 1999, 'Knowing Your Place in Real World Environments', Proceedings of EUROBOT`99, 3rd European Workshop on Advanced Mobile Robots ........ 2, 13

Dugoff, H., Fancher, P. S. and Segel, L., 1970, 'An Analysis of Tire Traction Properties and Their Influence on Vehicle Dynamic Performance' SAE Paper No700377, p. 1219-1243 ........ 22

Dulimatra, H. S. and Jain, A. K., 1997, 'Mobile Robot Localization in Indoor Environment', Pattern Recognition, Vol. 30, No. 1, p. 99-111 ........ 14, 15

Durrant-Whyte, H. F., 1996, An Autonomous Guided Vehicle for Cargo Handling Applications, The International Journal of Robotics Research, 1996 Massachusets Institute of Technology, Vol. 15, No. 5, p. 407-440 ........ 10

Entao, Z., Jiancheng, Z., Chengrui, G., Xinming, C. and Shichang Z., 1996, 'Four Wheel Steering System Controlled by Single Piece Processor', Chinese Journal of Mechanical Engineering, Vol. 9, No. 2, p. 175-178 ........ 10

Feng, K-T., Tan, H-S. and Tomizuka, M., 1998, 'Automatic Steering Control of Vehicle Lateral Motion with the Effect of Roll Dynamics', Proceedings of the American Control Conference Philadelphia, Pennsylvania, p. 2248-2252 ........ 27

Feng, L., Koren, Y. and Borenstein, J., 1993, 'Cross Coupling Motion Controller for Mobile Robots', IEEE 1993, p. 35-43 ........ 33

Fiala, E., 1954, 'Seitenkräfte am rollenden Luftreifen', Z. VDI, Vol. 96, No. 29 .... 21

Furukawa, Y., Yuhara, N., Sano, S., Takeda, H. and Matsushita, Y., 1989, 'A Review of Four-Wheel Steering Studies from the Viewpoint of Vehicle Dynamics And Control', Vehicle Systems Dynamics, Vol. 18, p. 151-186 ........ 26

Gartner, H. and Astolfi, A., 1996, 'Stability study of a fuzzy controlled mobile robot', Journal of Intelligent Control and Systems ........ 20

Gartner, H. and Astolfi, A., 1998, 'Fuzzy control of a mobile robot with guaranteed stability', Special Issues in Fuzzy Control, J. Aracil Ed., Springer Verlag ........ 34

Gaudiano, P., Zalama E. and Coronado, J. L., 1996, 'An Unsupervised Neural Network for Low-Level Control of a Wheeled Mobile Robot: Noise Resistance, Stability, and Hardware Implementation', IEEE Transaction on Systems, Man, and Cybernetics, Part B, Vol. 26, No 3, IEEE 1996, p. 485-496 34

Germann, S., Würtenberger, W. and Daiß, A., 1994, 'Monitoring of the friction coefficient between tyre and road surface', IEEE 1994, p. 613-618 ............ 22, 23

Guntur, R. and Sankar, S., 1980, 'A Friction Cycle concept for Dugoff's tyre friction model', International Journal of Vehicle Design, Vol. 1, No. 4, p. 373-337 ........ 22

Halanay, A., Ionita, A. and Rasvan, V., 1994, 'Stability and Manoeuvrability Analysis of Vehicle with Four Wheel Steering System', IEEE 1994, p. 385-390 ........ 27, 37

Hemami, A., 1994, 'A control scheme for low speed automated vehicles with double steering', Proceedings of the 33rd Conference on Decision and Control, IEEE 1994, p. 2452-2454 ........ 28

Hessburg, T. and Tomizuka, M., 1995, 'An Adaptation Method for Fuzzy Logic Controllers in Lateral Vehicle Guidance', Mechatronics, Vol. 5, No. 8, p. 873-898 ........ 31

Hoyong, K., 1996, 'An On-line Learning Control of Unsupervised Neural Network for a Vehicle Four Wheel Steering System', Society of Automotive Engineers, p. 1191-1197 ........ 27

Huh, K., Seo, C., Kim, J. and Hong, D., 1999, 'Active steering based on the estimated tire forces', Proceedings of the American Control Conference, San Diego, California, p. 729-733 ........ 33

Katz, Z. and Bright, G., 1992, 'A Guidance Technique for an Automated Guided Vehicle', The International Journal of Advanced Manufacturing Technologies, Vol. 7, p. 198-202 ........ 14, 16

Kezhong, H., 1994, 'Intelligent control of mobile robot', IFAC Emerging Intelligent Control Technologies, Hong Kong, p. 207-212 ........ 18

Kim, H., 1996, 'An On-line Learning Control of Unsupervised Neural Network for a Vehicle Four Wheel Steering System', SAE Transaction, Part 6, p. 1191-1197 ........ 31

Landau, I. D., Lozano, R. and M'Saad, M., 1998, 'Adaptive Control', Springer Verlag London, ISBN 3-540-76187-X ........ 28, 42

Lee, K-M., Zhou, Z., Blenis, R. and Blasch, E., 1995, 'Real-Time Vision Based Tracking Control of an Unmanned Vehicle', Mechatronics, Vol. 5, No. 8, p. 973-991 ........ 15

Lizarralde, F., Wen, J. T. and Popa, D., 1996, 'Feedback stabilisation of nonholonomic systems', Proceedings of the 1996 Conference of Information Sciences and Systems, Princeton, New Jersey ........ 20

Martinez, A., Tunstel, E. and Jamshidi, M., 1994, 'Fuzzy Logic Based Collision Avoidance for a Mobile Robot', Robotica, Vol. 12, No. 6, p. 512-527 ........ 12

Matsumoto, N. and Tomizuka, M., 1990, 'Vehicle lateral velocity and yaw rate control with two independent control inputs', Proceedings of the 1990 American Control Conference, San Diego, p. 1868-1875 ........ 5, 29, 30, 37

Popa, D-O, 1998, 'Path-Planning and Feedback stabilization of nonholonomic control systems', PhD Thesis, Rensselaer Polytechnic Institute, Center for Advanced Technology of New York State. ........ 20

Potter, T.E.C., Cherry, A. S. and Jones, R. P., 1996, 'Modelling And Simulation Of An Automotive Vehicle For Integrated Motion Control', Proceedings of the International Symposium on Advanced Vehicle Control, Aachen, 471-4845, 21

Prabhu, S.M., and Garg, D.P., 1996, Artificial Neural Network Based Robot Control: An Overview, Journal of Intelligent and Robotic Systems, 15, p. 333-365 ........ 12

Ray, L. R., 1995, 'Nonlinear State Tire Force Estimation for Advanced Vehicle Control', IEEE Transactions on Control Systems Technology, Vol. 3, No. 1, p. 117-124, IEEE 1995 ........ 23

Roth, H. and Schilling, K., 1995, 'Control Strategies for Mobile Robots Based on Fuzzy Logic', Proceedings of 1995 IEEE International Conference on Fuzzy Systems, Vol. 1, IEEE 1995, p. 89-96 ........ 16

Rus, D., 1997, On-Line Algorithms for Manupulation with Autonomous Robots, Proceedings of the International Conference of Information Sciences, Vol. 2, p. 14-18 ........ 16

Sakai, H., 1981, 'Theoretical and experimental studies on the dynamic properties of tyres Part 1: Review of theories of rubber friction', International Journal of Vehicle Design, Vol. 2, No. 1, p. 78-110 ........ 21

Saraf, S. and Tomizuka, M., 1995, 'Road/tire interactions in lateral control of passenger vehicles', Proceedings ASME 1995, p. 85-90 ............................ 31, 45

Sarkar, N., Yun, X. and Kumar, V., 1994, 'Control of Mechanical Systems With Rolling Constraints: Application to Dynamic Control of Mobile Robots', The International Journal of Robotics Research, Vol. 13, No. 1, 1994 Massachusetts Institute of Technology, p. 55-69 ............................................. 14

Seneviratne, L. D., Ko, W-S. and Earles, W. E., 1997, 'Triangulation-based path planning for a mobile robot', Proceedings of the Institution of Mechanical Engineers, Vol. 211, Part C, p. 365-371 .................................................. 11, 16

Senoo, S., Mino, M. and Funabiki, S., 1992, 'Steering Control of Automated Guided Vehicle for steering energy saving by fuzzy reasoning, IEEE, p. 1712-1716 ................................................................................................ 11

Sethi, I. K. and Yu, G., 1994, 'Making a mobile robot learn to determine its location using neural networks', Pattern Recognition Letters, Vol. 15, p. 393-402 .................................................................................................. 12

Shladover, S. E., Desoer, C. A., Hedrick, J. K., Tomizuka, M., Walrand, J., Zhang, W-B., McMahon, D. H., Peng, H., Sheikholeslam, S. and McKeown, N., 1991, 'Automatic Vehicle Control Developments in the PATH Program', IEEE Transactions on Vehicular Technology, Vol. 40, No. 1, p. 114-130 ........... 9

Song, K-T. and Sheen, L-H., 1995, 'Fuzzy-neuro Control Design for Obstacle Avoidance of a Mobile Robot', p. 71-76 ................................................ 12, 17

Surmann, H., Huser, J. and Peters, L., 1995, 'A Fuzzy System for Indoor Mobile Robot Navigation', IEEE 1995, p. 83-88 ...................................................... 16

Szostac, H. T., Allen, R. W. and Rosenthal, T. J., 1988, 'Analytical Modeling of a Driver Response in Crash Avoidance Maneuvering', Vol. II: An interactive tire model for driver/vehicle simulation, US Department of Transportation, report no. DOT HS 807-271 .................................................................... 23

Taghaboni-Dutta, F., 1997, 'A Value-Added Approach for Automated Guided Vehicle Task Assignment', Journal of Manufacturing Systems, Vol. 16, No. 1, p. 24-34 ............................................................................................ 17

Taheri, S. and Law, E. H., 1990, 'Investigation of a Combined Slip Control Braking and Closed Loop Four Wheel Steering System for an Automobile During Combined Hard Braking and Severe Steering', Proceedings of the 1990 American Control Conference San Diego, p. 1862-1867 ......................... 32

Thanjavur, K. and Rajagopalan, R., 1997, 'Ease of Dynamic Modelling of Wheeled Mobile Robots (WMRs) using Kane's Approach', Proc. 1997 Int. Conf. on Robotis and Automation, IEEE 1997, p. 2926-2931 .............................. 5

Thuilot, B., d' Andréa-Novel, B. and Micaelli, A., 1996, Modeling and Feedback Control of Mobile Robots Equipped with Several Steering Wheels, IEEE 1996, Vol. 12, No. 3, p. 375-390 .............................. 28

Tso, S. K. and Fung, Y. H., 1994, 'Effective development of fuzzy-logic rules for real-time control of autonomous vehicles', IFAC Emerging Intelligent Control Technologies, Hong Kong, p. 133-138 .............................. 30

Tzes, A., Peng, P-Y. and Houng, C-C., 1995, Neural Network Control for DC Motor Micromanoeuvring, IEEE Transaction on Industrial Electronics, Vol. 42, No. 5, IEEE 1995 .............................. 35

Watanabe, K., Tang, J., Nakamura, M., Koga, S. and Fukuda, T., 1996, 'A Fuzzy-Gaussian Neural Network And Its Application to Mobile Robot Control', IEEE Transactions on Control Systems Technology, Vol. 4, No. 2, p. 193-199, IEEE 1996 .............................. 33

Weiczer, F., 1995, 'On Path Planning and Path Following for Wheeled Mobile Robots', PhD Thesis, School of Mechanical and Vehicular Engineering, Chalmers University of Technology, Sweden .............................. 17

Wyatt, J., 1996, 'Issues in Putting Reinforcement Learning Onto Robots', DAI Research Paper No. 784 .............................. 35

# Appendix A

### Patent application

Plantenberg, D.H., and Lai, E., Self-Stabilising Support Assembly, Application No: GB9910919.1, filed for Patent application 12 May 1999

# SELF-STABILISING SUPPORT ASSEMBLY

## TECHNICAL FIELD OF THE INVENTION

The present invention relates to supports for various objects. More specifically but not exclusively the invention relates to self stabilising, supports for articles such as bases, load bearing structures, pedestals, rolling chassis for furniture, appliances, rolling chassis for vehicles and other similar and dissimilar objects.

## BACKGROUND TO THE INVENTION

Problems relating to flooring and other support surfaces include the fact that they may not always be even, level or flat. Even when laid to stringent specifications they may, in time, become uneven from unequal loading, subsidence and wear, for example. Objects with four rigidly assembled legs, for example a table when placed on such an uneven floor may tend to wobble. Three legs of the object will rest on the floor, while the insufficient support of the fourth leg can cause the object to rock, pivoting on the two supported diagonally opposite legs and hence lifting the remaining leg.

Previous proposals for addressing these problems include placing shims underneath individual legs to compensate for the shortfall, but has frequently proved to be only a temporary solution. The object may be moved to a new position or the floor may settle under the subjected load and hence the object may become unstable again.

Adjustment screws attached to some or all legs are an alternative for shims and provide a more convenient means of adjustment, but customised settings are required for every situation. Depending on the application areas, frequent readjustment of the screws may become necessary. Stabilising an object may require specialist knowledge and skilful adjustment if proper weight distribution is desired for example for heavy duty workbenches or machine supports.

A design for a floor-engaging leg assembly has been disclosed in U.S. Patent No. 3,844,517. This prior patent discloses disengaging a leg assembly connected to the main structure and thereby permitting a rotary oscillation about a generally horizontal axis and accomplishes self-adjusting floor-engagement before the assembly is locked in its current position with a series of teeth or friction pads. When all four legs are in contact with the support surface the table is firmly positioned even on an uneven floor. This design, however, requires manual intervention whenever the profile of the supporting surface changes.

Prior U.S. Patent No. 5,690,303 discloses another design using a leg assembly with two opposing faces which rotate around a generally horizontal axis to enable floor-engaging positioning of all four legs. Once a suitable configuration has been found, the leg assembly is locked into position by means of friction pads and a locking screw to prevent rotary motion of the leg assembly.
Consequently, wobbling of the table when a load is applied

or shifted is prevented. Again, this mechanism requires manual readjustment whenever the table is subjected to changes of the floor due to relocation or settlement of the soil or deformation of the support surface.

Prior U.S. Patent No. 3,954,241 provides a levelling assembly adapted to position and prevent 'walking' of appliances such as washing machines on an uneven floor. Two mounting brackets with interconnected vertically movable floor-engaging leg members are attached to the appliance and ensure that each leg member is supporting its proportional share of the weight. The brackets can be frictionally engaged to prevent shifting movement of such appliances once all the legs are in normal contact with the floor. However, manual input is still required should the profile of the support surface change.

For structures requiring regular relocation, for example a mobile workbench, these prior proposals appear impractical. Frequent readjustment of the supporting legs can be time consuming and consequently is often neglected leading to a wobbling bench and therefore may pose unsuitable working conditions and safety hazards. In the case where the legs are replaced with wheels so that the bench is movable, continuous adjustment would be required which may not be practical.

This is particular the case for trolleys or carts used for transport of materials on the shop floor. A course of action to prevent wobbling is to utilise vehicles with only three wheels. Although such a vehicle will not wobble, the triangular shaped envelope within which the centre of gravity can be located before the vehicle overturns is restricted.

Four-wheeled carts often make use of two axles connected to the vehicle, with one wheel attached at each end of the axle. One of the axles is rigidly fixed, while the other is allowed to pivot around a horizontal axis to enable floor-engaging positioning of all four wheels. This arrangement however, constitutes in a mechanical sense, only a three-point support. Two points can be assumed between each of the wheels on the fixed axle and the support surface. The third point is located at the intersection of the virtual line connecting the contacts between the support surface and each of the wheels on the oscillating axle and the vertical plane through the pivot axis.

Prior German Patent No. 28 36 494 discloses a two part rolling chassis whereby the front axle and the driver cabin is permitted to rotary oscillate about a generally horizontal longitudinal vehicle axis with respect to a fixed rear axle or axles. This patent discloses means to keep the wheels in contact with the road surface, by allowing limited rotational movement. As no provision to counteract such rotation is provided, it resembles a construction similar to the above four-wheeled cart, which provides non-overturning stability only within the triangular shaped envelopes.

Prior U.S. Patent No. 4,071,259 discloses a pair of unsprung pivotally connected axle structures connected to a rigid frame on a pair of laterally spaced longitudinally extending axes. This suspension system establishes an effective pivot or roll centre, which is at a higher elevation than the centre of gravity of the vehicle, hence mimicking a suspended mass. When the wheels traverse uneven ground the suspension system permits the axle assemblies to shift laterally, around the elevated pivot point, so that all wheels remain in ground contact, thus ensuring proper support and traction. This arrangement resembles in a mechanical sense an oscillating axle. The vehicle remains stable (i.e. not overturning), so long as the centre of gravity remains in lateral direction within the virtual vertical triangle formed by the elevated virtual pivot axis and the two laterally spaced wheel contacts. In order to prevent the axle structure to swivel outwards excessively, end stops are required to allow only a small lateral displacement and therefore limits its effective range. Using a similar approach an axle suspension system for rigid axles on utility vehicles has been disclosed in Patent No. WO97/00176 and WO97/00177 with variations on the implemented levers to accomplish lateral shift motion to enable compensation for uneven support surfaces.

A vehicle with variable length banking links is disclosed in U.S. Patent No. 2,689,747. This invention employs springs in order to increase the banking range and the maximum permissible unevenness of the support surface before individual wheels loose road contact.

Spring suspension of axles and individual wheels are commonly used in today's vehicles, mobile and fixed structures to ensure floor engagement. The main shortcomings of passive suspension systems are that the height of the system varies dependent on the payload. Furthermore, roll and bank movements are inevitable when the load shifts and above all the load distribution on the contact points is directly dependent on the unevenness of the support surface. Active suspension systems can overcome some of the limitations of their passive counterparts, but often introduce unnecessary complexity and add to the costs.

## SUMMARY OF THE INVENTION

According to the present invention there is provided a self-stabilising article having a plurality of support members together providing four contact points for contacting a reference surface, the support members being constrained by a mechanical linkage such that rotation of the line joining any two adjacent contact points relative to the article is accompanied by rotation of the line joining the other two adjacent contact points relative to the article in the opposite sense.

In a practical embodiment of the invention, the are two support members pivotally attached to the article about substantially parallel axes, each support member providing two contact points for contacting a reference surface and being constrained by a mechanical linkage such that rotation of one support member about its axis is accompanied by corresponding rotation of the other support member in the opposite sense.

Likewise, the contact points are preferably so constrained by a mechanical linkage that comprises the plurality of support members. The mechanical linkage may include a first linking member pivotally attached to the body about an axis substantially normal to the axes about which the support members are pivotally attached to the article, each of the two ends of the first linking member being articulated to a respective support member. A second linking member may also be provided, pivotally attached to the body about an axis substantially parallel to the axis about which the first linking member is pivotally attached to the article, each of the two ends of the second linking member being articulated to a respective support member.

Each support member may have legs, wheels, chain crawlers or pontoons taking real support from a fixed or varying uneven support surface or floor for stationary, mobile or moving support of the support surface.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention will now be described by way of example with reference to the accompanying drawings in which:

Figure 1 is an exploded view of components of the support assembly according to embodiment of the invention;

Figure 2 is a perspective view of the assembled components of Figure 1 as they would appear when placed on an uneven support surface;

Figure 3 is the top view of the support assembly according to one embodiment of the invention;

Figure 4 is a perspective view of a support assembly according to a further embodiment of the invention, which details an implementation of the stabilising support assembly incorporating three radially positioned levers;

Figure 5 is a perspective view of an embodiment of the invention comprising a staggered multi-lever assembly;

Figure 6 is a perspective view of an embodiment of the invention comprising a trapezoidal arrangement of the levers;

Figure 7 is a perspective view of an embodiment of the invention comprising a chain crawler assembly with a single lever arrangement;

Figure 8 is a perspective view of an embodiment of the invention incorporated within a vehicle chassis;

Figure 9 is a perspective view of a further embodiment of the present invention as applied to a domestic appliance or white good, not showing the connection of the support members and linkage to the body of the appliance;

Figure 10 is a cross-section of the coupling between the support members and the linkage;

Figures 11 and 12 show an adjustable coupling for use in connecting the support members and linkage to the body of the appliance; and

Figures 13 and 14 show an alternative adjustable coupling for light loads, for use in connecting the support members and linkage to the body of the appliance.

DETAILED DESCRIPTION

The components of a self-stabilising support assembly 1 according to one embodiment of the invention are illustrated in an exploded view in Figure 1 which details how the parts are assembled together. Two longitudinally spaced support members 11 and 13 are connected via cylindrical joints 15 to a support surface 9. This allows rotary oscillation about a generally horizontal axis. In this embodiment, two laterally spaced levers or linking means 3 and 5 are connected via cylindrical joints 7 to the support surface 9 permitting generally rotary oscillation about a generally horizontal axis. Each of the levers 3 and 5 are connected at their longitudinal opposite ends to the support members 11 and 13 by means of generally spherical joints 17. This arrangement constrains the two support members to opposite rotary oscillation about a generally horizontal axis with respect to the superstructure.

When implemented according to the preferred embodiments, all legs of the assembly can be brought into floor-engaging contact and support their proportional amount of weight. Hence, objects can be positioned on a support surface without wobbling when loaded, the load shifts or the profile of the support surface changed.

The embodiment of the invention is suitable, but not limited to provide supports for, appliances such as refrigerators, washing machines, work benches, trolleys, carts, mobile platforms, agricultural vehicles and devices, earth moving appliances, utility vehicles, floating structures and the like. When the contacts on the support surface are not level, the assembly stabilises objects by bringing the support members into floor-engaging contact without any manual intervention and hence minimises problems associated with the unevenness of the support surface. Means may be included to adjust the elevation of some or all individual legs or wheel assemblies for levelling the superstructure when placed on an uneven or sloped support surface.

Although the embodiments illustrated in Figure 1 to 7 show a support surface 9 mounted internally, in the support assembly, it is also envisaged that the support surface could be mounted externally.

Figure 2 illustrates the components as assembled according to an embodiment of the invention. The assembly is suspended above a surface and is then gradually lowered vertically. Each end of the support members 11 and 13 comprises vertically extending legs 19, 21, 23, 25. Further lowering of the assembly causes leg 19 to move upwards relative to the support surface and simultaneously, pivotal action of the support member 11 around the cylindrical joint 15 causes the opposite leg 21 to be lowered. Rotary oscillation of the support member 11 is transmitted via levers 3 and 5 and imposes opposite rotary oscillation on the support member 13. This rotation results in the lowering of the adjacent leg 25 and a simultaneous upward movement of the diagonally opposite leg 23. Assuming both legs 19 and 21 are now floor engaging, the further lowering of the assembly a rotating motion of the whole assembly about a virtual axis joining the contacts on the leg 19 and leg 21. When leg 23 becomes a third floor engaging leg, further lowering of the assembly causes simultaneous rotation of both support members 11 and 13 in opposite direction until the remaining leg 25 is in floor engagement.

When all four legs are in contact with the floor, the gravitational force exerted on the support surface 9 is transmitted through the cylindrical joints 15 to the two support members 11 and 13, to the legs and then to the support surface. Only one of the two levers 3 or 5 is required to prevent the otherwise free rotation of the support surface 9 around the longitudinal axis. However, for symmetrical movement of both support members 11 and 13 and a balanced load distribution, an arrangement with two movable levers may be beneficial.

In use assuming the centre of gravity is located at the geometric centre of the support assembly, each leg 19, 21, 23, 25 will, on a flat and level surface carry equal weight. Unevenness of a generally level surface has within limit only marginal influence on the distribution of the weight carried by each leg. When the assembly is subjected to unequal loading e.g. the centre of gravity is not coincident with the geometric centre or when positioned on a slope sufficiently steep to cause the vertical projection of the centre of gravity to shift, each leg will carry its proper share of the weight. The envelope within which no danger of overturning of the assembly is to be expected is indicated by the dotted line 10 shown in Figure 3 which essentially connects the cylindrical joints 7 and 15.

When placed on a horizontal and even surface an ideal implementation would provide all joints located on a generally horizontal plane with the cylindrical joints 7 and 15 positioned midway between the spherical joints 17.

In an embodiment of the invention where parallel movement of the vertically extended support members is important, Figure 4 shows a further embodiment of the self-stabilising assembly incorporating three levers 3, 5 and 14. In the embodiment illustrated in Figure 4, it is preferred to locate the third lever 14 such that the distances between the cylindrical joints 15 and adjacent spherical joints 17 are matched on the support members 3 and 5 for all levers 3, 5 and 14. More levers may be distributed radially to the longitudinal axis which connects the cylindrical joints 15 of the support members 16.

In all the embodiments, provision for marginal longitudinal and lateral movement of the support members 11, 13 and 16 and levers 3, 5 and 14 respectively is provided by the cylindrical joints 7, and 15. Other embodiments of the invention may feature a revolving joint for the support members, which would prevent such longitudinal contraction. In this case means would be required for the levers to facilitate such contraction when the support members 11 and 13 rotate. This may be implemented by allowing a lever to slide through the spherical joints 17 or by telescopic, folding or other means of the lever to adjust for the variation of the length as the lever 11, 13 and support members 3, 5 rotate.

Further embodiments of the invention are indicated in Figures 5 and 6. Figure 5 illustrates a vertically extended superstructure 20 while Figure 6 illustrates a trapezoidal shaped support surface 24 for different lengths of the support members 26, and 28 and unequal distances between the spherical joint 17 and the cylindrical joint 7 on the levers 25. The function of the embodiments of the invention shown in Figures 5 and 6 is similar to that for the previous embodiments.

Furthermore, although the embodiments disclosed 1-6 only feature stationary objects, the invention is not restricted to stationary implementations. The legs may easily be replaced with wheel assemblies to provide an unsprung running gear, which generally enables continuous road to tyre contact even on an uneven support surface. Additionally, each support member is not limited to distinct contact points such as legs or wheels, but one element such as a chain crawler for example for earth moving equipment. Such an embodiment of the invention is illustrated in Figure 7.

In Figure 7 a surface 9 or platform forming the base of a vehicle body is connected to two laterally spaced tracks 27 via revolving joints 37. Lever 11 is attached to the platform 29 via cylindrical joint 15 and attached at each of its ends to tracks 27 via generally spherical joints 38 with sliding means to allow for the variation in the effective length of the lever. The support surface or platform 29 self-stabilises any object or vehicle body, for example placed, thereon.

Figure 8, illustrates the wheeled undercarriage of the vehicle. The base for the support surface in this embodiment is provided by support elements 32, each attached to support members 33 and 34 and levers or linking means 3 and 5 by the cylindrical joints 15 and 7 respectively. The support members in this embodiment form the front axle 33 and rear axle 34 attached to the levers 3 and 5 via the spherical joints 17.

It is envisaged that the present invention may incorporate a ratchet and pawl mechanism or other suitable device for levelling the support surface after stabilisation of the structure has been achieved.

It is also envisaged that one support assembly may be mounted on another so as to provide a tiered arrangement.

Figure 9 is a perspective view of a further embodiment of the present invention as applied to a domestic appliance or white good, not showing the connection of the support members and linkage to the body of the appliance. In this case, each end of the support members 11 and 13 comprises a moulded plastics foot 19′, 21′, 23′, 25′. One end of each support member 11, 13 is coupled to a respective end of a linkage or lever 3. The coupling is shown in detail in Figure 10. At the medial point of each of the support members and the linkage is a bracket 10, 12, 14. One end of each bracket includes a "T"- shaped extension that is retained in a slot in the support member or linkage; the other end is secured by a screw or bolt. The brackets provide a journal for the connecting pins 16, 18, 20, that are secured to the body of the appliance (not shown).

Figure 10 is a cross-section of the coupling between the support members and the linkage. Briefly, as can be seen, the lever 3 and support member 13 are formed into a dish or dome shape at corresponding points. The dish or dome includes a central aperture for passage of a connecting bolt 22, secured at its lower end by a nut 24. Part-spherical washer or collars 26, 28 are located on the bolt and/or the nut, or threaded onto the bolt, to co-operate with the dish- or dome-shaped regions. A spacer 30 maintains a proper spacing between the lever 3 and the support member 13. It will be appreciated that relative movement between the shaped washers or collars 26, 28 and the dished or domed regions of the lever 3 and the support member 13 allows limited relative rotation of the support member 13 and the lever 3 about any axis.

As explained extensively above, pivoting movement of the support member 11 is transmitted via the lever 3 and causes opposite pivoting movement of the support member 13. This stabilises the appliance. When placed on a horizontal and even surface the ideal implementation would provide all connecting pins 16, 18, 20 to be located in a horizontal plane. Whilst this cannot be achieved if the points at which the pins 16, 18, 20 attach to the body of the appliance are fixed relative to it, some degree of adjustment of those points can allow an appliance that has been stabilised to be levelled as well.

Figures 11 and 12 show an adjustable coupling for use in connecting the support members and linkage to the body of the appliance. The concept is a simple one. A plate 50 to which the pin 16 is perpendicularly attached is provided at each end with a threaded bolt hole. A bolt 52, 54 passes first through a washer 56, 58 then through a slot 60, 62 in the body 64 of the appliance, allowing the position of the plate 50 to be adjusted vertically relative to the body of the appliance. This adjustable coupling may be used for all three pins 16, 18, 20 or for just one, typically the centre pin 18.

In this case, for example, if the appliance were to lean forward, the bolts 52, 54 locating the pin 18 on the body of the appliance my be loosened and the appliance pushed into the desired position before tightening the bolts 52, 54 again. During adjustment, the weight of the appliance would rest mainly on the two centre mountings of the support members and would hence allow easy levelling of the appliance.

Figures 13 and 14 show an alternative adjustable coupling for light loads, for use in connecting the support members and linkage to the body of the appliance. In this case, the plate 50 is absent. Instead, a bolt 52 passes first through a washer 56, then through a slot 60 in the body 64 of the appliance and into the pin 16, allowing the position of the pin 16 to be adjusted vertically relative to the body of the appliance. Again, this adjustable coupling may be used for all three pins 16, 18, 20 or for just one, typically the centre pin 18.

CLAIMS

1. A self-stabilising article having a plurality of support members together providing four contact points for contacting a reference surface, the support members being constraint by a mechanical linkage such that rotation of the line joining any two adjacent contact points relative to the article is accompanied by rotation of the line joining the other two adjacent contact points relative to the article in the opposite sense.

2. An article according to claim 1 in which there are two support members pivotally attached to the article about substantially parallel axes, each support member providing two contact points for contacting a reference surface and being constrained by the mechanical linkage such that rotation of one support member about its axis is accompanied by corresponding rotation of the other support member in the opposite sense.

3. An article according to claim 2 in which the support members are elongate and pivotally attached to the article at a medial point.

4. An article according to claim 2 or claim 3 in which the two contact points of each support member are provided on opposite sides of the axis about which it is pivotally attached to the article.

5. An article according to any one of claims 1-4 in which

the contact points are so constrained by a mechanical linkage that comprises the plurality of support members.

6. An article according to any one of claims 2-4 in which the mechanical linkage includes a first linking member pivotally attached to the body about an axis substantially normal to the axes about which the support members are pivotally attached to the article, each of the two ends of the first linking member being articulated to a respective support member.

7. An article according to claim 6 in which the mechanical linkage further includes a second linking member pivotally attached to the body about an axis substantially parallel to the axis about which the first linking member is pivotally attached to the article, each of the two ends of the second linking member being articulated to a respective support member.

8. An article according to claim 6 or claim 7 in which the support members are of fixed length and the linking member or members are of variable length.

9. An article according to any preceding claim comprising a support surface to which the support members and any such linkages are pivotally attached.

10. An article according to claim 9 in which the support surface is internal or external to the support members and any such linkage or linkages.

11. An article according to claim 9 or claim 10 in which the support members are attached to a medial point of opposite sides of the support surface.

12. An article according to any preceding claim in which the support members comprise railed elements of a vehicle.

13. A self-stabilising article substantially as described herein and/or as illustrated in the accompanying drawings.

ABSTRACT

SELF-STABILISING SUPPORT ASSEMBLY

A self-stabilising support assembly 1 for stabilising structures mounted on a support surface 9 of the assembly 1, comprises two support members 11, 13 linked to each other via linking bars 3. Legs 21 are provided at each corner of the support surface 9 rotary oscillation of the support member 11 is transmitted to linking bars 3 and 5 and imposes opposite rotary oscillation on support member 13. This rotation results in the lowering of leg 25 and hence stabilising of the assembly on the floor.

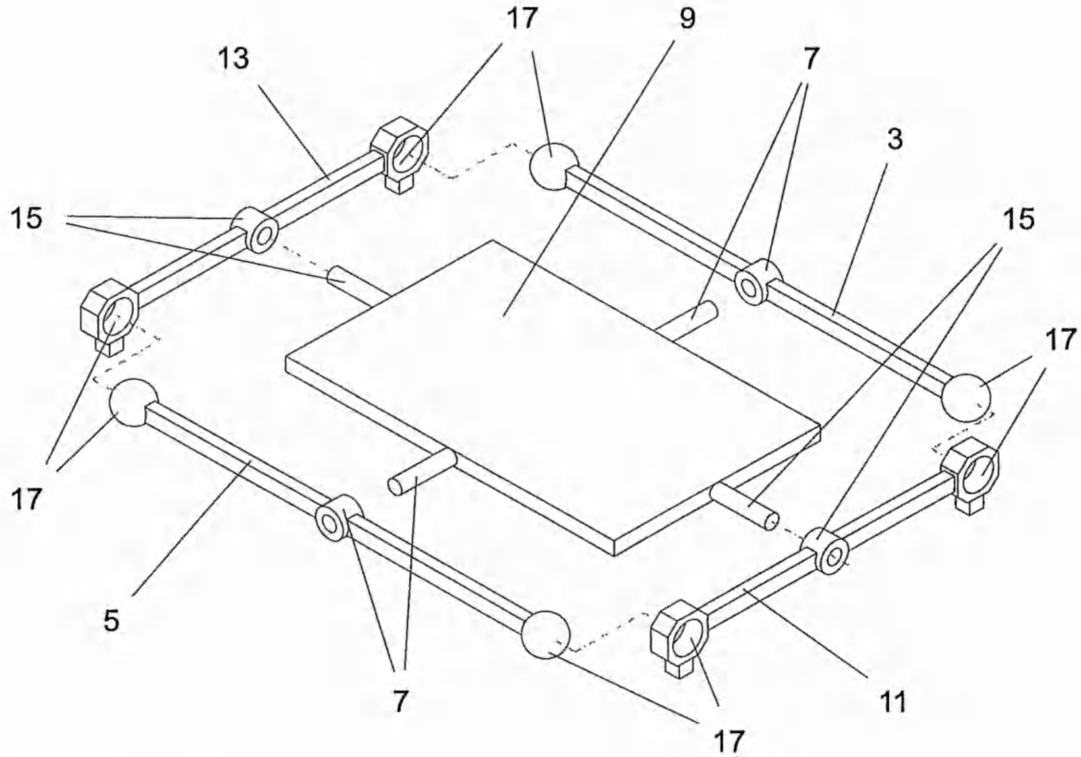


**FIGURE 1**

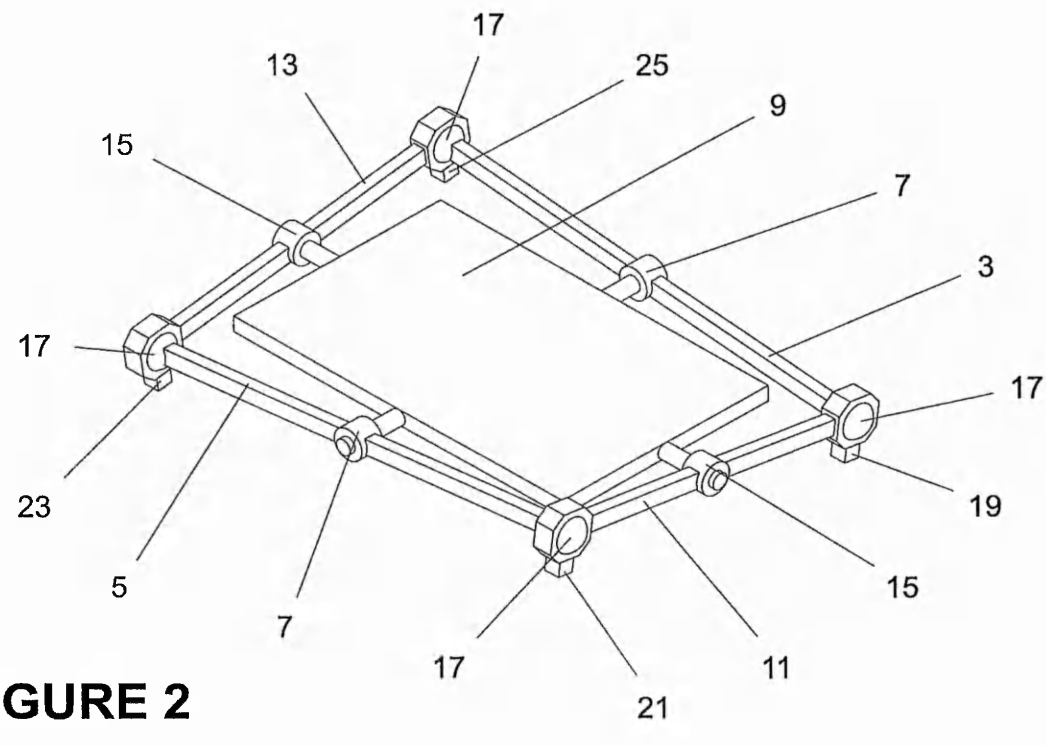


**FIGURE 2**

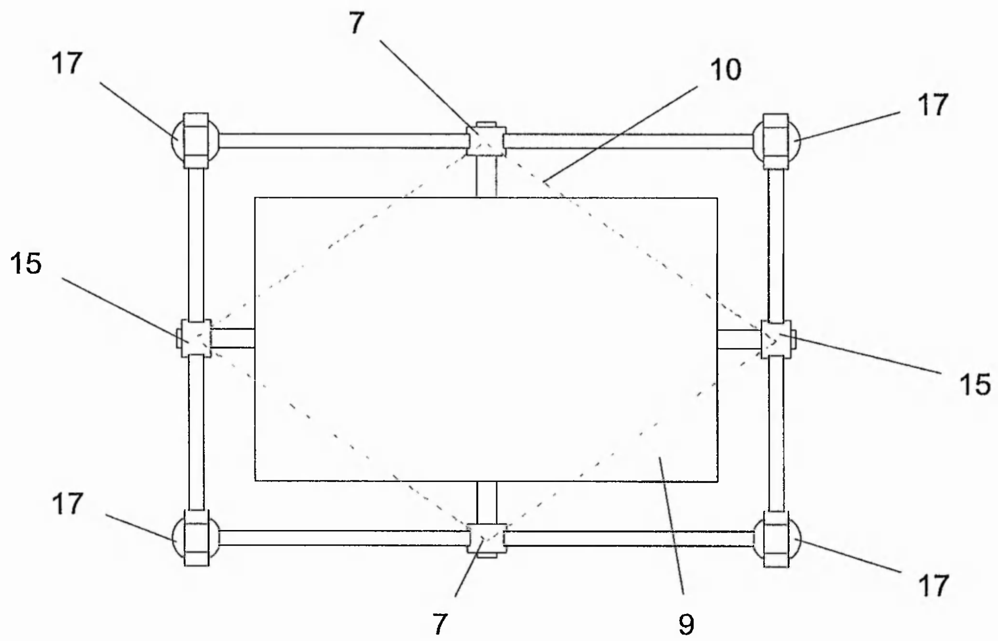


FIGURE 3

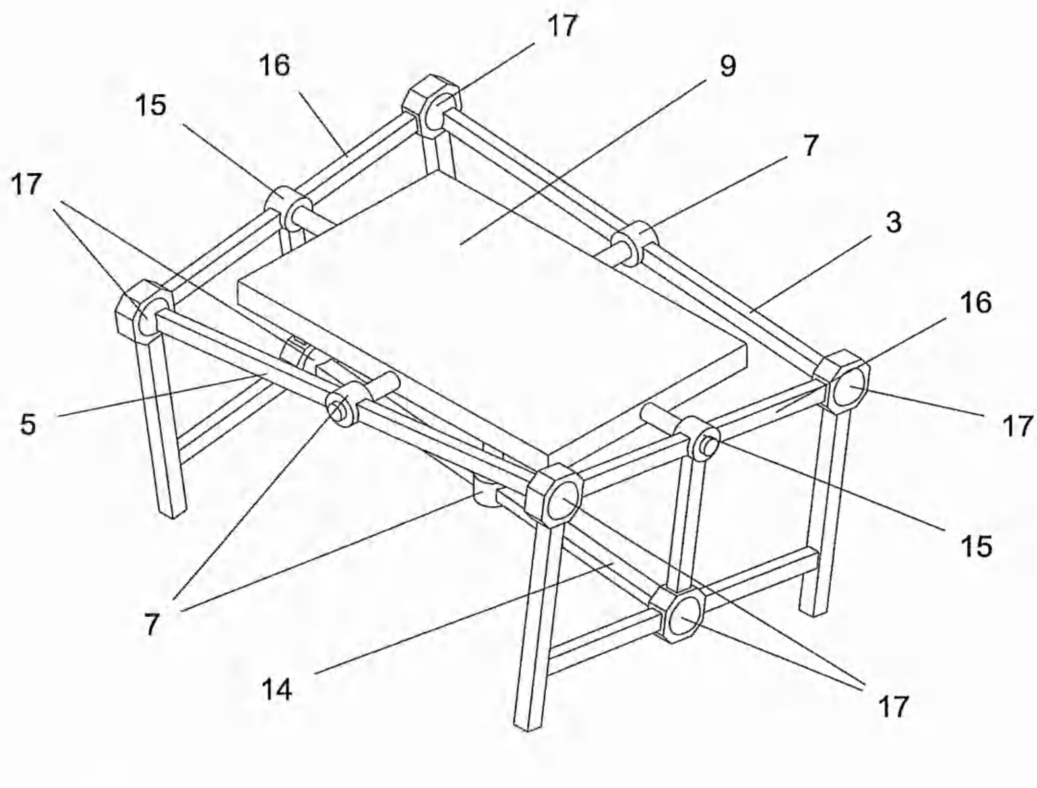


FIGURE 4

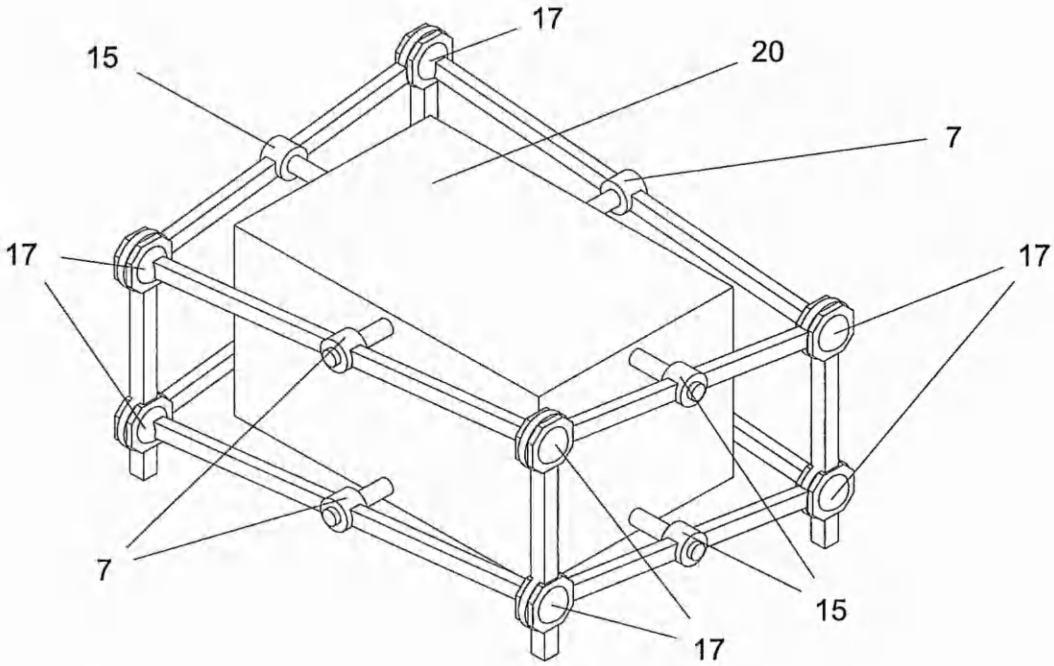


**FIGURE 5**

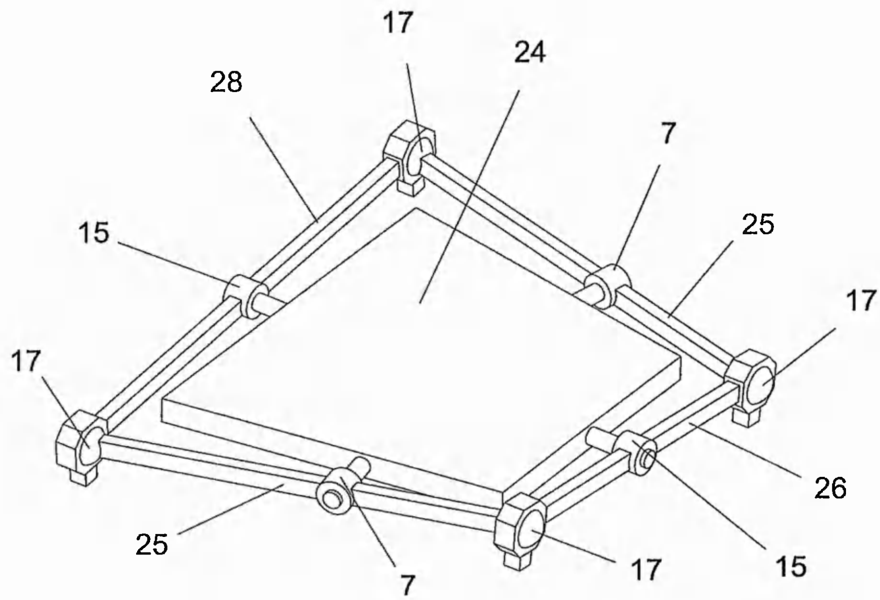


**FIGURE 6**

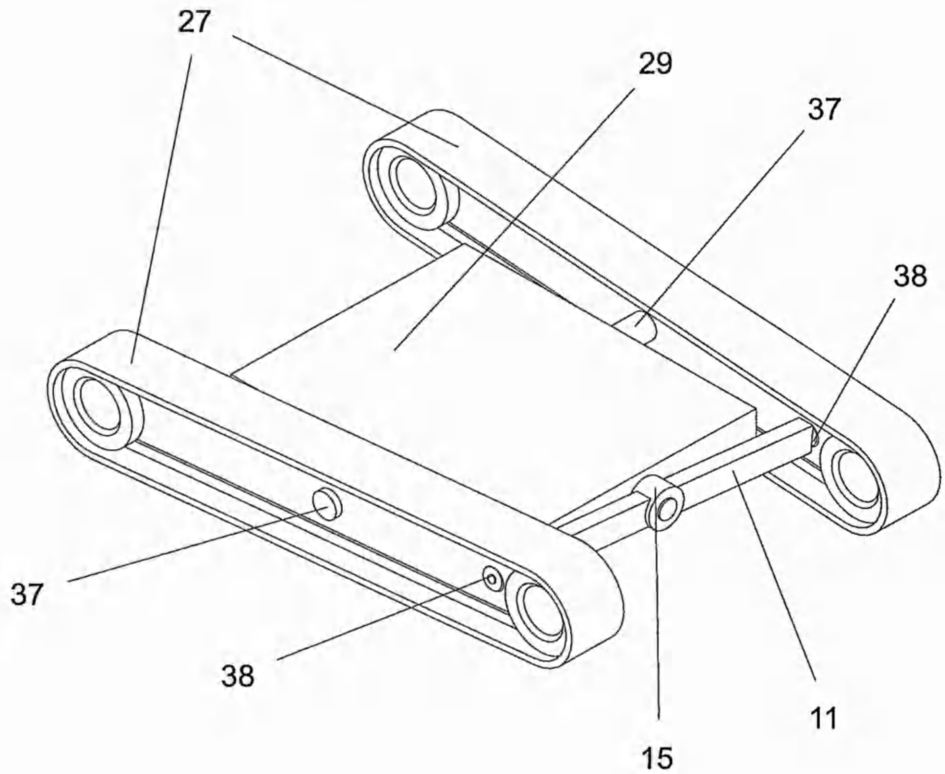


FIGURE 7

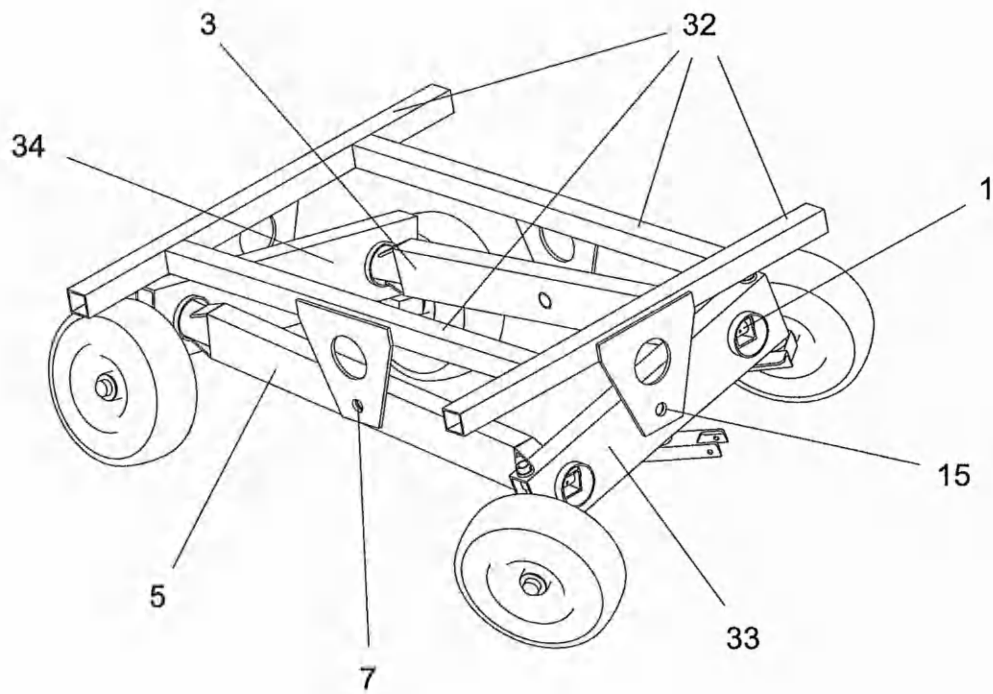


FIGURE 8

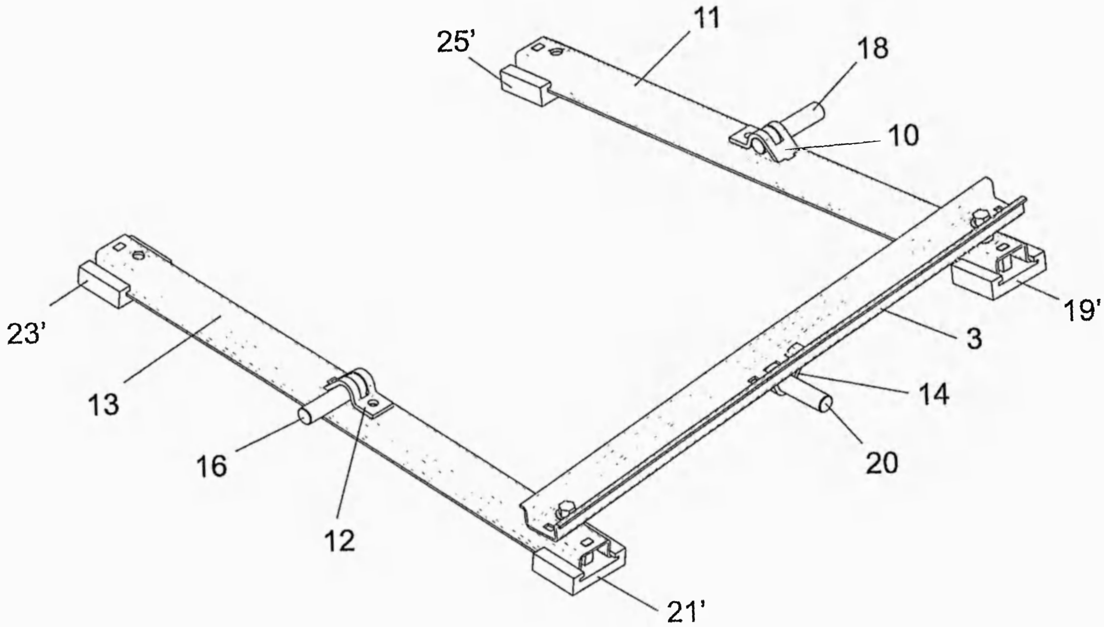


FIGURE 9

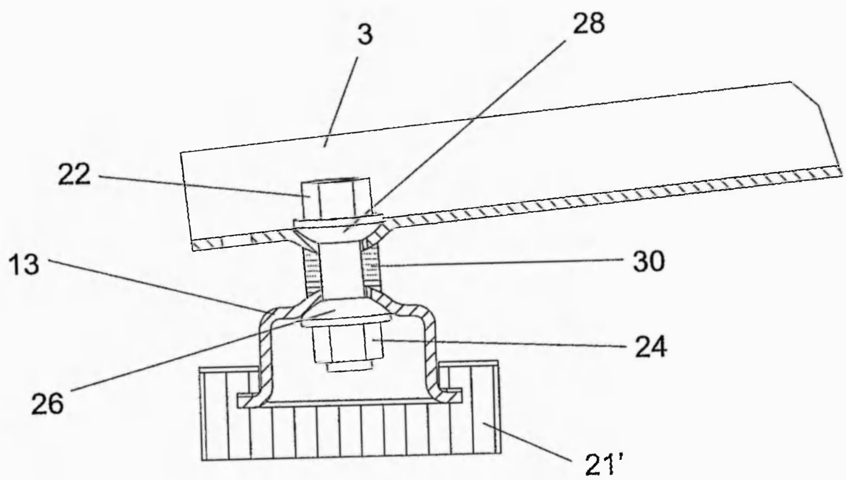


FIGURE 10

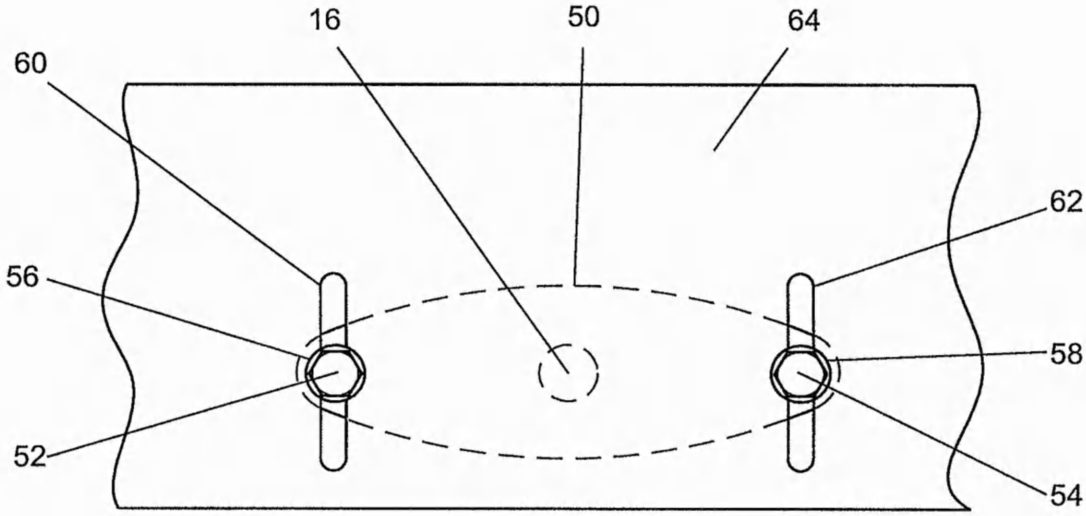


**FIGURE 11**

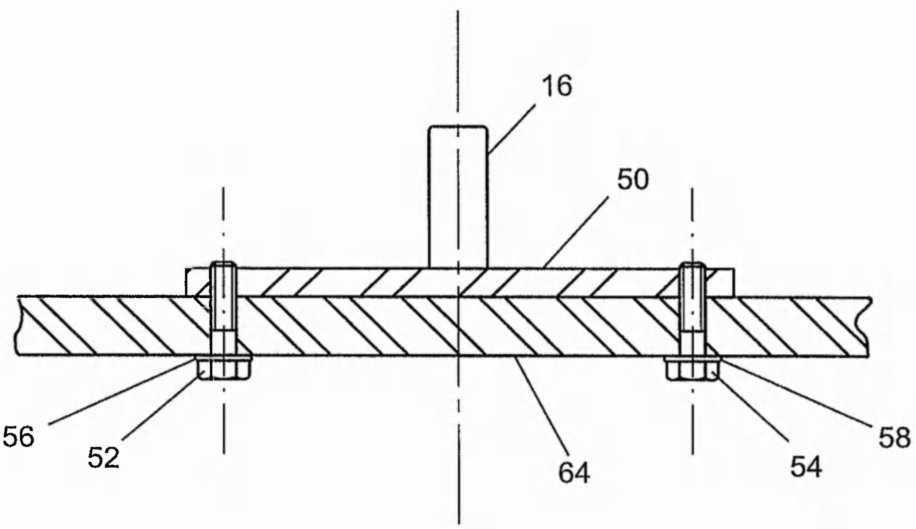


**FIGURE 12**

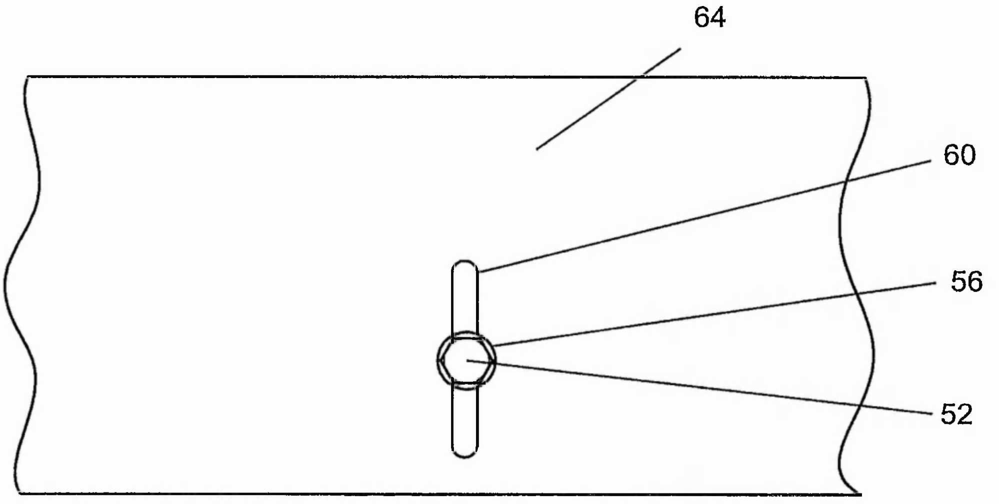


**FIGURE 13**

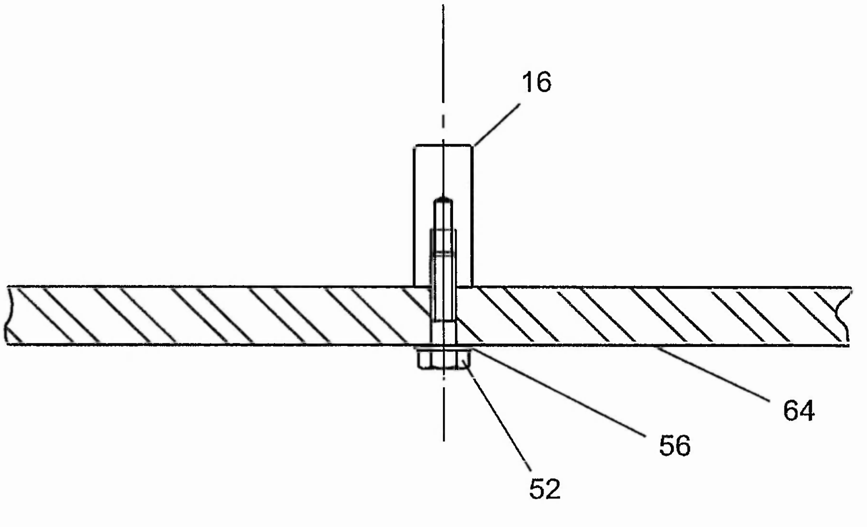


**FIGURE 14**

# Appendix B

**List of publications**

The results derived from the measurements using the proposed method could not be published. Consequently it has been decided to assess the performance of the design of the individual components using a condition monitoring rig previously developed for the packaging industry. The performance of the electronic interface and the proposed data acquisition method has been published in the following papers.


Lai, E., Plantenberg, D.H., Grant, Z.R., and Hull, J.B., 1996, 'Condition Monitoring of Plants and Equipment with the Aid of Neural Networks', Proc. International Conference of Mechanics in Design MID'96, University of Toronto, Canada, Vol. 2, p. 515-522.

Plantenberg, D.H., Lai, E., and Jeffries, M., 1997, 'A Novel Approach to Reduce Packaging Process Down-time', Proc. International Conference of Advances in Materials and Processing Technologies AMPT'97, Universidade De Minho, Guimaraes, Portugal, 22-26 July, Vol. 2, p. 630-635.

Jeffries, M., Lai, E., Plantenberg, D.H., and Hull, J.B., 1997, 'A Fuzzy Approach to the Condition Monitoring of a Packaging Plant', Proc. 6th International Scientific Conference on Achievements in Mechanical & Materials Engineering AMME'97, Gliwice, Poland, 28-30 November, p. 95-98. (Extended Abstract)

Jeffries, M., Lai, E., Plantenberg, D.H., and Hull, J.B., 1999, 'A Fuzzy Approach to the Condition Monitoring of a Packaging Plant', Special Edition of Journal of Materials Processing Technology.

Jeffries, M., Lai, E., Plantenberg, D.H., 1998, 'Real-time Implementation of a Fuzzy Condition Monitoring System for Predicting Process Breakdown', Proc. International ICSC Symposium on Engineering of Intelligent Systems EIS'98, University of La Laguna, Tenerife, Spain, 11-13 February, p. 87-92

# A NOVEL APPROACH TO REDUCE PACKAGING PROCESS DOWN-TIME

D.H. Plantenberg, E. Lai and M. Jeffries
The Nottingham Trent University, Department of Mechanical and Manufacturing Engineering,
Nottingham NG1 4BU, United Kingdom. Fax: +44 115 948 6506, email: e.lai@domme.ntu.ac.uk

**Abstract:** Semi-automated systems are widely used for packaging products, which require the support of experienced mechanics to perform tasks such as tool changes and machine adjustment. This arrangement can lead to rising production overheads owing to the need to allocate additional resources to cover emergencies. The present study seeks to develop a method by which the task of machine adjustment can be delegated to an unskilled operator in a "controlled" manner.

An experimental set-up, which consisted of three independently driven rotating shafts and an reciprocating device, has been designed to mimic the movement of the key components in a packaging machine. A novel data acquisition unit has been developed to capture simultaneously the signals produced by three rotary encoders and a proximity sensor. The assimilated signals enable the state of the system to be ascertained. Any non-compliance of pre-determined operating limits can be detected before the end of each cycle of operation. Furthermore, the condition monitoring system can also provide information concerning the performance history of the machine.

The experimental results have demonstrated a high degree of repeatability and accuracy. This enables an innovative method to be devised to reduce process down-time through a supervised delegation of tasks. The method offers three distinct advantages: (i) to alert an operator of an imminent machine failure due to misalignment; (ii) to authorise and guides the operator to carry out appropriate machine adjustment; (iii) to record all the events for managerial reference.

**Keywords:** condition monitoring, packaging, data acquisition, down-time.

## 1. Introduction

The UK manufacturing industry has, over the past two decades, undergone radical changes both in the use of production methods and in working practices. The implementation of methods such as 'Just-In-Time (JIT)' and 'Single-Minute-Exchange-Die (SMED)', has helped to raise the awareness of the need to improve all aspects of the manufacturing process. Substantial productivity gain can be achieved through the automation of processes and the rationalisation of staffing levels. Greater dependence on machinery has meant that any production strategy becomes less effective if the performance of machines in the production chain can not be relied upon. Consequently there has been a drive, particularly amongst larger companies, to develop and implement preventative maintenance programmes in order to reduce the possibility of unexpected equipment breakdowns, thereby optimising plant availability and reducing maintenance cost. This approach requires the continuous monitoring of the 'health' of machinery and a diagnostic capability to allow the prediction of the onset of failures, usually involving the processing and interpretation of a large amount of data generated by sensors [1,2]. To this end, investigations have been conducted to assess the feasibility of using artificial neural networks to assist with the development of condition monitoring systems for various applications [3-5]. Packaging of products, often conceived as a relatively simple operation and where open-loop control is a common practice, is one area that has attracted less investment in new technology. Automation in such an operation often refers to the automatic execution of tasks with little or no measurement of parameters for feedback control or parameter adjustment. A mistake can result in the loss of 'finished' products due to damage; this becomes especially relevant if the cost of additional quality checks, after problems have been identified, exceeds the value of the product. A typical packaging machine relies mainly on two types of motion to perform its task: rotation and translation; these manifest in the form of pushers, conveyor belts and other manipulators. These motions are transmitted to the required parts from the power sources through various mechanical parts: gear trains, rack and pinions, belts, cams and followers. It is important that all these parts perform in synchronisation in order to achieve the desirable outcome.

Problems arise when misalignment is present in the system, the cause of which can come from several sources. For example, the unintentional variations in the set-up of a machine and wear and tear on the mechanical components amongst others. There is also the tendency to try to achieve higher yields by operating the machine closer to its operational limits. This will put more stress on the synchronised systems, which in turn may start to show the strain by exaggerating any misalignment, leading to an eventual breakdown of the packaging process. To minimise the machine down-time and hence the loss of production, skilled mechanics are employed whose expertise is used to identify, track and eliminate faults. Though commonly used, this approach has a number of drawbacks: increased maintenance overheads, non-availability of 'experts' and 'expertise', human variance and production inflexibility. By focusing on the maintenance problems exhibited by a typical industrial packaging machine, this project seeks to develop a method for predicting potential causes of misalignment. This method will help to facilitate condition based maintenance, either to provide assistance to the mechanic and hence reduce set-up variance, or to delegate system adjustment tasks to unskilled operators in a controlled manner.

## 2. Experimental Set-up

The present study is a collaborative project between an international company and the Nottingham Trent University. The industrial packaging machine being investigated has three rotating shafts (the "cams") and a reciprocating mechanism (the "rake"). The movements of individual components must be synchronised through repeated adjustment in order to achieve an optimum operation. To minimise disruption to the packaging process during the development and testing of the Real-Time Condition Monitoring (RTCM) system, an experimental set-up has been designed to mimic the movements of the key components of the packaging machine (Figure 1). The rig consists of three main parts: Instrumentation; Hardware interface and data processing; Software analysis and display.

### 2.1 Instrumentation

The main problem of machine down-time is believed to be attributable to the lack of useful information governing the general machine set-up and operating window. Current remedial actions often involve the subjective adjustment of machine parameters until a workable solution is found, but this approach has a number of drawbacks: time consuming, inconsistent and expensive. For condition monitoring new instrumentation has to be introduced to the packaging machine; rotational movements of individual shafts are to be measured by shaft encoders, while translational motion of the rake is to be measured by an inductive proximity sensor which has a high degree of consistency and reliability. Shaft encoders are well suited for measuring repetitive rotational movements and are available in a variety of different resolutions and mounting types. For the experimental rig, rotational motions are provided by three stepper motors each of which is coupled to an encoder, and the reciprocating motion is generated by a cam and follower driven by a fourth stepper motor.

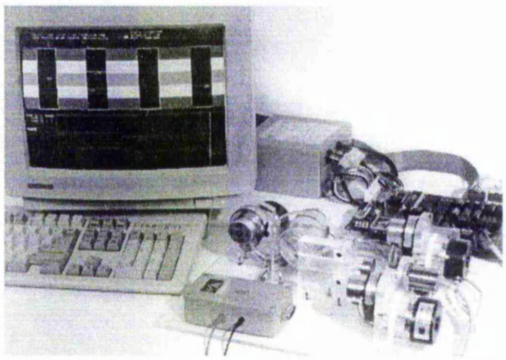

Figure 1 Experimental Set-up

### 2.2 Hardware Interface And Data Processing

The next stage involves the quantification of the offset and tolerance band for each shaft, as well as the exact position of the reciprocating device, but the shaft encoders and the proximity sensor produce two different types of output signal: digital and analogue. Figure 2 shows the design of the data acquisition interface. The encoders produce output in the form of two square waves with a 90° phase shift, allowing directional as well as positional information to be obtained. Conversion of the encoder output to a numerical form, representing the angular position, is performed by Quadruple decoders.

The proximity sensor provides a standard 4-20 mA output which is transformed into the numerical form by means of an analogue-to-digital (ADC) converter. For practical reasons, the following two constraints need to be overcome. Firstly, although the condition monitoring system is designed to be operated with a personal computer, the latter can not be relied upon to provide a steady timing between sampled data points. This is because a personal computer is not designed as a dedicated system for real-time monitoring of external devices; any "timing jitter" during the acquisition of data will distort the information contained therein. Secondly, individual rotating shafts in the industrial packaging machine may run at variable speeds and hence a fixed reference may not be available. Thus any varying time shift that exists between sampled data points will make extraction of synchronisation information extremely difficult. Parallel sampling offers the only workable solution if the above problems are to be eliminated. A novel technique has been developed which involves the use of the timer within the ADC to generate a synchronisation signal to enable parallel sampling to take place. The sampled data is then stored in buffers ready to be processed by the personal computer. The RTCM interface has the following key features;

- 8 Encoder inputs (square wave)
- 8 Analogue input channels
(expandable to 16)
- 24 Digital I/O lines (optional)
- Parallel sampling of analogue & digital data
- 8 ms scan-time (based on 486DX2/66)
- Up to 120 process cycles per minute

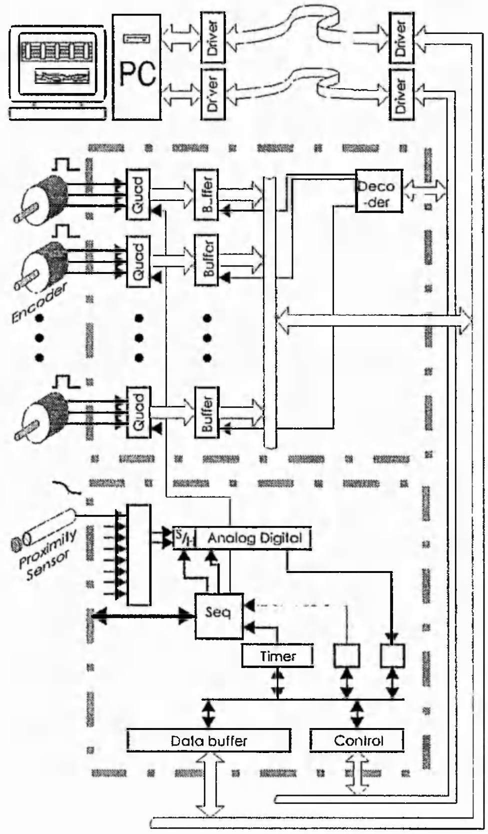


Figure 2 Data Acquisition Interface

**2.3 Software Analysis And Display**

The data acquired by the hardware must be processed and displayed within one machine cycle so that an operational decision can be made. Figure 3 shows how an offset between two transducers, $\Delta$ , is determined, with encoder 0 chosen as the reference. The rotational offset, $\Delta_{10}$ , between encoders 1 and 0 is the difference of their angular positions, while the translational offset, $\Delta_{p0}$ , is a measure of the position of the rake relative to the angular position of encoder 0.

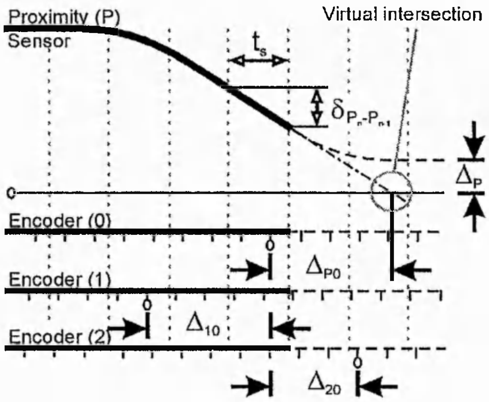


Figure 3 Determination of relative offsets

The software converts the signals stored in the buffer into relative phase shifts from which the minimum and maximum values are identified. It checks if the predetermined operational limits have been exceeded and displays the relevant information on a VDU. Since the interface can acquire data simultaneously from all transducers at the rate of 125 samples per second (i.e. at a sampling time of 8 milliseconds), a 'time slicing' technique has been used to overcome the limitations imposed by the graphic display. The software also has the capability to cater for different set-up conditions and operational windows. The RTCM software has the following key features.

- Cyclic representation of machine status
- Graphical display or tabular format
- Data rationalisation and interpretation algorithms
- Logging of raw or assimilated data
- Real-time system with sampling priority

**3. Results**

Predicting the onset of failures caused by the misalignment between moving parts is the principal goal of the present investigation, but the case study is also intended to provide design information on various aspects pertaining to the development of a low cost PC-based real-time condition monitoring system for packaging operation. These include:

- reviewing a range of packaging processes;
- identifying the key methods for power transmission;
- selecting low maintenance and reliable transducers;
- designing an electronic interface for parallel data sampling;
- developing efficient data analysis methods;
- designing an ergonomic information display.

In common with other manufacturing processes, the packaging operation has a set of limits (tolerance) within which efficient performance can be attained. The determination of these tolerances holds the key to predicting the onset of failures and the case study has shown that detailed and reliable information is needed in three areas: absolute displacement of individual transducers, relative offset of rotary elements (synchronisation) and correlation of rotational and translational movements. Figure 4 shows the image of a screen displaying the results of the immediate past cycle of machine operation. For convenience, the term "cam" refers to a shaft and the "rake" refers to the reciprocating device. The first three columns (from left to right) on the top half of the image provide information on the relative offsets between cam 0 and the rake, cams 0 and 1, cams 0 and 2, while the fourth column gives the proximity of the rake. Colour bands (or in this case varying grey levels) are used to signal the degree of misalignment. The positions of the markers signify the status of the operation: normal (region 1); machine adjustments are needed (region 2); shut down is recommended (region 3). The relative offsets of rotary elements vary between minimum and maximum values, as displayed in columns 2 and 3. By measuring the absolute displacement of the rake using the proximity sensor, it is possible to detect positional undershoot or overshoot of the rake during the filling function. The most important part of the data analysis is to develop a simple, but reliable, method of determining the relationship between the rotating shafts and the reciprocating rake. The synchronisation of these two mechanical systems ultimately determines the success of the packaging operation. By linking both systems to a datum (Figure 3), the relative offsets can be determined for each machine cycle. The monitoring system is capable of measuring angular displacement to a resolution of 0.018° and linear displacement to an accuracy of ±0.01 mm, thus allowing accurate determination of the synchronisation characteristics. It is not only important to show the current state of the machine operation, but to predict the likelihood of failure. This can be achieved by showing the trends of the relevant parameters. The bottom half of the screen (Figure 4) charts the trends of individual transducers over the past 500 cycles. A method analogous to the Statistics Process Control (SPC) could be used to help an operator to identify changes in performance over time.

## 4. Discussion

The aim of the project is to develop a method, through real-time condition monitoring, for improving the productivity of a packaging operation by reducing its down-time. A detailed investigation of the problem, taking into account the requirements specified by the industrial partner, has led to the conclusion that a deductive approach is needed to predict the onset of failures based on the gathered "intelligence". To this end, shaft encoders and an inductive proximity sensor have been chosen as the appropriate instruments for their accuracy and reliability. Although the resolution achievable with these devices may appear to be too good for tackling the misalignment problems associated with industrial packaging machines, the case study is also intended to provide a framework within which a generic solution can be developed for monitoring other processes and machines. Indeed tests carried out, but not presented here, have suggested that the current set-up may be used to identify the inherent errors present in gearboxes, from which the standard operating variations can be determined.

For practical reasons all basic functions of the RTCM system including data logging, analysis and information display must be completed within one cycle of machine operation. This requirement presented an interesting challenge to the research team because of the relative small timeframe within which operational decisions have to be made. A data acquisition interface capable of sampling analogue and digital data simultaneously has been developed to facilitate the determination of synchronisation and relative offsets at individual sampled data points.

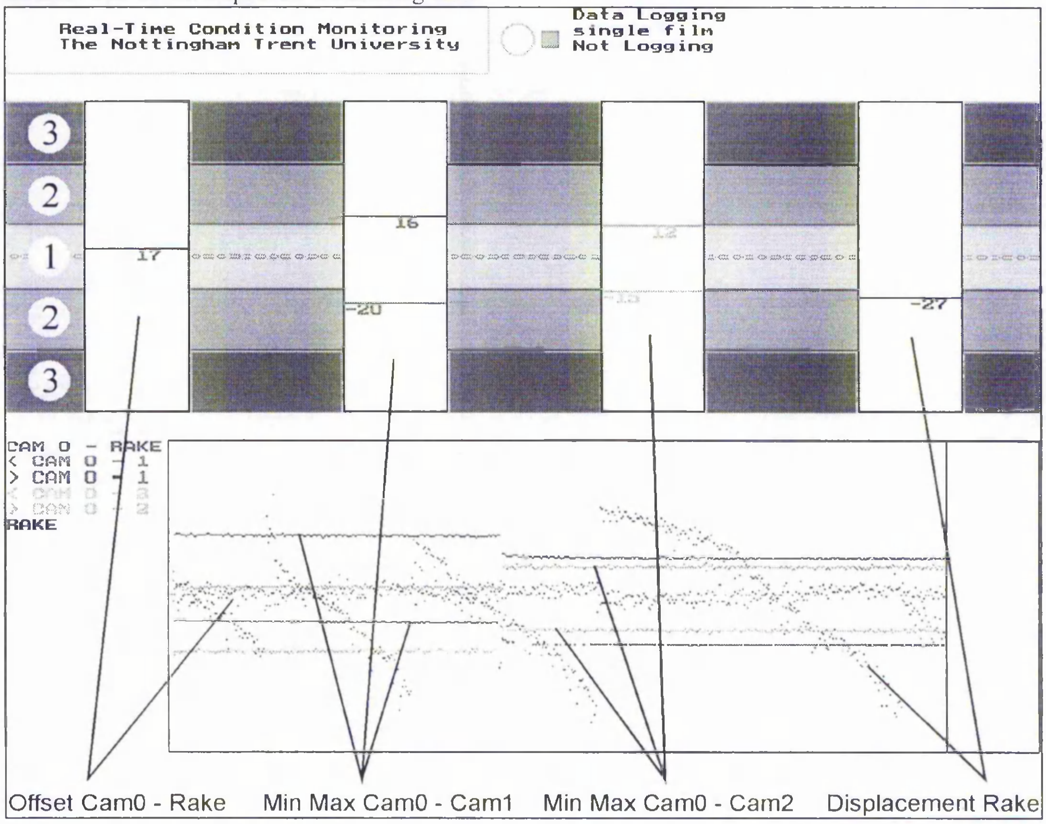


Figure 4 The RTCM display screen

The correlation between the rotating shafts and the reciprocating mechanism was obtained by means of a gradient technique which has shown to be independent of the sensitivity of the proximity sensor. By combining the historical information governing the mechanical behaviour of individual components given in the trend chart, with the knowledge of the degree of misalignment, if any, between them, the RTCM system is capable of discerning the state of a packaging machine.
The capability demonstrated by the monitoring system permits a cost effective maintenance strategy to be developed through the principle of ownership of responsibility. A comparison of the proposed strategy with the current practice is shown in figure 5. The iterative process of identifying faults, deciding on the corrective actions and carrying out parameters adjustment is to be replaced by the two stage process. Whenever a fault is identified or an initial machine set-up is needed, the RTCM system will decide on the corrective actions and this will be communicated to a person authorised to carry out the adjustment in the form of graphical instructions.

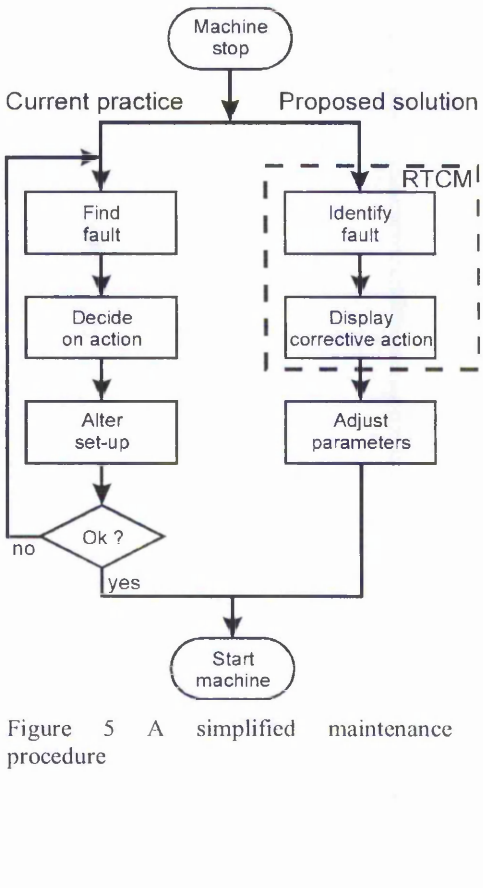


Figure 5 A simplified maintenance procedure

Figure 6 shows an example where the status of all three cams (i.e. shafts) are displayed along with suggested corrective actions. In this case cams 0 and 2 are misaligned but not cam 1. For each cam, the degree of offset is represented by the shaded segment of the outer circle, while the colour of the inner circle indicates whether correct adjustment has been completed. In the case of cam 0, corrective actions should be taken to gradually reduce the shaded segment by turning the shaft clockwise until it has completely disappeared and the inner circle changes its colour, as shown in cam 1. By the same analogy, corrective actions for cam 2 should be anticlockwise. As the RTCM system monitors the operation continuously, all events are logged automatically for operational control and production statistics. The proposed methodology was tested on the experimental set up and encouraging results were obtained.

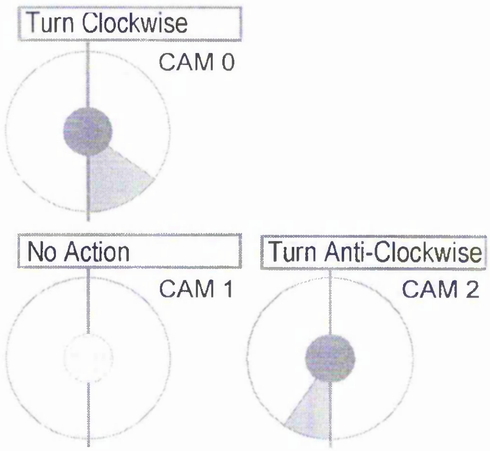


Figure 6 Computer-guided machine set-up

**5. Conclusions**

The success of a real-time condition monitoring system for an industrial application will be judged by a number factors including technical merits, ease of installation and implementation, cost effectiveness and reliability. The causes of breakdowns in packaging machines are reasonably well understood by skilled mechanics, but the interaction between moving parts within the machine has meant that "operating windows" can only be identified by a time consuming trial and error process. In terms of investment in new technology, packaging operation is likely to remain the poorer cousin of its manufacturing counter parts.

With this in mind, a low cost system has been developed to enable the prediction of the onset of failures through continuous monitoring of the "health" of a packaging machine. By means of an experimental set up, the capability of the system was ascertained and the methodology validated. Once installed and commissioned, the system can help to further reduce process down-time through a supervised delegation of machine adjustment tasks. The proposed method offers three distinct advantages: (i) to alert an operator of an imminent machine failure due to misalignment; (ii) to authorise and guide the operator to carry out appropriate machine adjustment; (iii) to record all the events for operational control and production statistics.

**References**

[1] Moon C. 1995. Condition Monitoring puts the squeeze on plant downtime *Noise and Vibration Worldwide*, ***Vol.26***, *9, Institute of Physics Publishing Ltd Bristol, 12-14.*

[2] Poon H L. 1991. A Knowledge-based Condition Monitoring System For Electrical Machines. *Computers In Industry, **Vol.16**, 2, 159-168.*

[3] Lai E, Plantenberg D, Grant Z R and Hull J B. 1996. Condition Monitoring of plants and equipment with the aid of neural networks *Proc. Symposium on Mechanics in Design, University of Toronto, Canada, 6-9 May, 515-522.*

[4] Aluindigue I E et al. 1993. *Monitoring and diagnosis of rolling element bearings using artificial neural networks. IEEE Trans. on Industrial Electronics, **Vol.40**, 2, 209-216.*

[5] Chow M, Magnum P M and Yee S O. 1991. A neural network approach to real time condition monitoring of induction motors. *IEEE Trans. on Industrial Electronics, **Vol.38**, 6, 448-453.*

# A FUZZY APPROACH TO THE CONDITION MONITORING OF A PACKAGING PLANT

M Jeffries, E Lai, D H Plantenberg, J B Hull

Department of Mechanical and Manufacturing Engineering, The Nottingham Trent University, Nottingham NG1 4BU, United Kingdom. Fax: +44 115 9486506; email: e.lai@domme.ntu.ac.uk

## ABSTRACT

The packaging of manufactured products attracts relatively little attention from research or benefits from the opportunities available through condition monitoring. The technical requirements of packaging can easily be underestimated in the overall production process. A typical packaging machine has translational, reciprocating and rotational motions guiding and manipulating the product. The contributing factors to packaging problems include machine set-up variations and synchronisation problems, due to the misalignment of key components such as cams or transmission systems. These misalignments can cause jamming within the machine leading to unnecessary damage or loss of product, resulting in the need for additional quality checks, the cost of which may exceed the value of mass-produced high premium products.

A widely used method of reducing waste relies upon the knowledge of an experienced mechanic to set up the machine before production runs and to correct misalignments should problems occur. Whilst this method offers a workable solution, there are a number of inherent problems. Relevant practical experience may take a long time to acquire, which can lead to a shortage of expertise. For economic reasons, there is also a tendency for manufacturers to reduce production overheads through rationalisation of maintenance staffing levels and the formation of multi-discipline teams. Instead of allocating specific responsibilities to individuals, members of a team are expected to provide emergency cover for a number of machines over the factory floor, in addition to their routine maintenance tasks. This can present problems in scheduling maintenance personnel should a machine require regular adjustments or a number of machines breakdown at the same time. Even with the mechanic present, the current practice of machine adjustment consists of a certain amount of hit-and-miss judgement and therefore the time taken to diagnose and remedy faults can significantly vary even for similar problems. Larger companies often implement Real Time Condition Monitoring (RTCM) systems as a means of tackling this problem. Although condition monitoring can produce the information on performance parameters, expertise is required to decode this intelligence. Furthermore, the cost of analysing the large volume of data logged from even the simplest machine can be prohibitively high. Production overheads are necessarily raised in both these scenarios, either by the employment of highly skilled labour or the resultant downtime from their lack of availability.

This study aims to develop an efficient, hybrid method for capturing the machine information, by means of fuzzy logic to decode the intelligence supplied from an RTCM system. It also intends to provide an insight into the operation of the machine from a management perspective and the operator's point of view. The fuzzy diagnostic system is required to consolidate a number of input variables into a single output value representing the current state of the machine. This is to be achieved by introducing a modification to a standard fuzzy logic correlation technique.

Input variables measured by the RTCM system, including synchronisation offsets between moving parts, are classified into five fuzzy sets: LARGE-NEGATIVE, SMALL-NEGATIVE, NOMINAL, SMALL-POSITIVE and LARGE-POSITIVE. The output domain has three sets: SAFE, CAUTION and CRITICAL, representing the possible states of a typical packaging machine. A rule base links the input variables to form a single defuzzified output value. The sensitivity of the individual rules is governed by the distribution of the input sets.

A typical rule base example is

*if* ENC0-ENC1 LOWER LIMIT *is* NOMINAL *then* MACHINE *is* SAFE

*if* (ENC0-ENC1 LOWER LIMIT *is* SMALL -VE)
*or* (ENC0-ENC1 LOWER LIMIT *is* SMALL +VE) *then* MACHINE *is* CAUTION

*if* (ENC0-ENC1 LOWER LIMIT *is* LARGE -VE)
*or* (ENC0-ENC1 LOWER LIMIT *is* LARGE +VE) *then* MACHINE *is* CRITICAL

where ENC0-ENC1 LOWER LIMIT is the minimum relative offset between two encoders during one machine cycle.

The structure of the rules is kept simple, to allow for a generic solution to be generated for all similar processes. The effectiveness of the fuzzy logic reasoning process will depend on the calibration of the input variables, where the input fuzzy sets are to be tuned to match intended operational tolerance windows.

In order to maintain low computational overheads in real-time condition monitoring systems, a variant of the commonly used min-max to centroid method has been developed to correlate the outcomes of all rules into a matrix of output sets. When using min-max inference, information from some rules is lost - only the maximum 'cut' is used. In this new 'matrix' variant all information is stored before defuzzification in the matrix, so that no outcomes of the rules are dismissed. A variation on a standard centroid method is then used to produce the defuzzified result. A key feature of the 'matrix' approach is to cause subtle changes to the shape of the fuzzy decision surface. This results in the profile becoming flattened around the areas associated with the centre of the output sets, while allowing the surface to still pass through the anchor points at the corners. Application of the new 'min-matrix' correlation technique in conjunction with the modified centroid method will help to reduce computational overheads.

To evaluate the proposed methodology, a purposely-built condition monitoring experimental test rig has been designed to mimic the operation of a film packaging machine, including the main rotational and reciprocating motions. The rig consists of three constituent parts.

- Instrumentation - three optical shaft encoders and a proximity sensor are used, each of which is driven by a stepper motor.

- Hardware interface and data processing - a novel data acquisition interface has been developed, capable of the parallel sampling of analogue and digital signals.

- Software analysis and display - a personal computer is used to log, assimilate and display the data. Data analysis is to be carried out by the fuzzy diagnostic software.

The test rig allows for the introduction of controlled errors into individual measured parameters. The parameters of the packaging machine have associated tolerances within which satisfactory performance of the machine is assured - often referred to as operational windows. Although the instrumentation can provide intelligence associated with individual parameters, the interaction between moving elements that determines the overall system state is more difficult to decipher. The use of operational windows does provide a means of quantifying performance but two main problems need to be addressed. Firstly, a mechanism must be devised to ascertain the position of individual windows. Secondly, the functional relationships between individual operating windows need to be established. As these effects cannot be easily described through mathematical models, fuzzy logic appears to offer a simple and robust approach enabling heuristic information to be represented in a structured manner. The developed fuzzy condition monitoring framework allows for the easy transfer from the current application to other similar problems, with minimal alteration to the software. The operational windows of the machine can be mapped onto the fuzzy domain allowing individual fuzzy sets to become the corresponding tolerance bands within each window. The fuzzy set is well suited to this role, showing degrees of membership rather than a simple true or false association. By describing an operating window in a fuzzy manner, even though the true value is discrete, a close approximation can be produced.

The min-matrix correlation method is capable of combining multiple, but not necessarily similar, variables into a single output value. This in turn will provide a useful means of tracking machine performance through the monitoring of a single value, by tuning fuzzy sets to match estimated operational windows. Preliminary results have shown that the min-matrix correlation technique introduced within a fuzzy RTCM system can produce useful information for diagnosing the state of a packaging machine and provide a methodical approach for close estimation of operational windows. The work has therefore lent support to the use of fuzzy condition monitoring as a reliable and inexpensive means of reducing wastage and maintenance overheads in the packaging industry.

# A FUZZY APPROACH TO THE CONDITION MONITORING OF A PACKAGING PLANT

M Jeffries, E Lai, D H Plantenberg, J B Hull

Department of Mechanical and Manufacturing Engineering, The Nottingham Trent University, Nottingham NG1 4BU, United Kingdom. Fax +44 115 9486506
email: martyn.jeffries@ntu.ac.uk, e.lai@domme.ntu.ac.uk

Abstract.

The packaging of manufactured products attracts relatively little attention from research or benefits from the opportunities available through condition monitoring. The technical requirements of packaging can easily be underestimated in the overall production process.
A widely used method of reducing waste relies upon the knowledge of an experienced mechanic to set up the machine before production runs and to correct misalignments should problems occur. This approach has the following drawbacks: non-availability of expertise, the relevant practical experience takes time to acquire and the reduced staffing levels present in the current economic climate.
This study aims to develop an efficient, hybrid method for capturing the machine information, by means of fuzzy logic to decode the intelligence supplied from an RTCM system. Developed from the min-max fuzzy inference method, the present approach replaces the maximum selection technique for output set truncation, by a matrix of truncation values derived from all rule outcomes. A suitably generic method has been employed allowing the developed techniques to be utilised for a wide range of manufacturing problems.
The work has lent support to the use of fuzzy condition monitoring as a reliable and inexpensive means of reducing wastage and maintenance overheads in the packaging industry.

*Keywords: Fuzzy Logic, Condition Monitoring, Packaging*

## 1. INTRODUCTION

The role of condition monitoring has been exploited in many aspects of manufacturing industry, but its use in the final stage packaging is only now being investigated [1]. The technical requirements of packaging can easily be underestimated in the overall production process as any problems occurring at this stage can lead to unnecessary damage or loss of product.
In industry, the availability of a skilled mechanic becomes a crucial element when attempting to achieve performance or efficiency targets. The set-up and maintenance of a machine can be a time-consuming operation, which requires iterative steps, including some trial and error, to achieve near-optimal set-up. The problems arise when condition monitoring is employed as a means to develop an understanding of the relationship of machine elements that leads to desirable performance. The tracking of individual variables is not problematic, but to deduce the overall machine condition is more difficult. The mechanic can perform this task, to some extent, by using a set of rules learnt through experience. A similar approach is available by means of fuzzy logic, which has a proven ability to convey heuristic information governing non-deterministic problems using a rule base. Fuzzy logic may even allow an insight into the performance of the machine from a management perspective or operator's point of view.

Fuzzy logic has enjoyed much attention from control researchers in their attempts to capture the heuristic and often non-discrete nature of real-world problems. It has proved invaluable in its ability to provide simple and robust solutions to problems considered difficult to tame with conventional control theory. However, there may be one drawback, although some may see it as a benefit, is the ease and simplicity with which a fledgling fuzzy logic 'toolbox' can be developed without thoroughly understanding the mechanisms contained therein. The ability to select methods within a fuzzy toolbox, for reasons of computational ease or apparent suitability, can create algorithms that do not necessarily have a solid theoretical basis linking the separate components - correlation, defuzzification etc. This has been recognised as a problem by Rondeau et al. [2] and is an important consideration in the choice of the fuzzy method used for a specific application. The criterion for selection of the method should not be based purely on the need to match the output to the expected result. But rather, to understand the algorithms involved, by ensuring correct manipulation of the data and 'respecting' the intention of the given application. This approach will provide a solid foundation from which an optimal fuzzy system can be developed, and hence the attainment of the desired output. This means that fuzzy logic does not become the quick fix for those people uninterested in control theory, thereby giving greater credibility to the field of research in the industrial sector.

Investigations into the use of Artificial Neural Networks (ANN) in condition monitoring [3-5] have shown the benefits of using artificial intelligence techniques in the diagnosis of machine condition. A hybrid approach utilising fuzzy logic in conjunction with ANNs has also been explored [6]. Within the packaging process, the monitoring takes on another aspect. The focus is not only on individual elements, such as tool wear [7], but the complex interaction between transmission systems and the various types of motion used to convey product through the machine. In this context, the application of fuzzy logic must be considered in its approach to solving the problems encountered. A thorough understanding of the processing involved must be a pre-requisite to achieve the goal of successful condition monitoring.
This paper outlines the theoretical aspects of a proposed condition monitoring system using fuzzy logic. The fuzzy system is to be discussed in its relevance to the problem in hand. Experimental validation of the proposed method has been carried out on a purposely-built condition monitoring rig and some results are presented here.

## 2. FUZZY OPERATIONAL WINDOWS

The determination of state for a packaging machine or any other type of machine can be a difficult process. A machine typically has several different types of motion (rotational, translational, reciprocating) performing a variety of tasks. The parameters governing these motions are susceptible to machine set-up variations and relational offsets causing synchronisation problems. A method for quantifying machine parameters is to use so called 'operational windows'. An operational window is a way of analysing how the actual value of a variable relates to a pre-determined datum. The term 'error' is used to describe a measure of the difference between the pre-determined datum and the actual value of the parameter. In theory, a machine running at absolute optimum would have no errors, and all values would be at the centre of their respective operational windows (Figure 1)
The width of the operational window is the tolerance within which a parameter can lie without causing the machine to fail. The window is subdivided to differentiate smaller errors, which tend to reduce machine performance, from larger errors risking process breakdown.

Although operational windows do provide a means of quantifying machine performance, their determination and use can suffer from two problems. Firstly, a mechanism must be devised to ascertain the positions of individual windows. Secondly, the functional relationship between these windows needs to be established. As these relationships cannot be easily described through mathematical models, fuzzy logic appears to offer a simple and robust approach enabling the heuristic information to be represented in a structured manner. The operational window of a parameter can be mapped onto the fuzzy domain, thus allowing individual fuzzy sets to become the tolerance bands within each window (Figure 2).

Input parameters can therefore be classified with five fuzzy sets: LARGE-NEGATIVE, SMALL-NEGATIVE, NOMINAL, SMALL-POSITIVE and LARGE-POSITIVE. The fuzzy set is well suited to this role, because it shows varying degrees of membership rather than a simple true or false association. By describing an operating window in a fuzzy manner, although the true value is discrete, a close approximation can be produced by utilising the flexibility and robustness of the fuzzy logic method.
In describing individual operational windows within the fuzzy domain, the opportunity to combine information using the fuzzy rule base becomes apparent. If operating windows are to be combined, the result must also be an operating window representing the overall machine state. A typical machine can be described by three basic states:

**Safe**, all parameters are close to nominal error and there is minimal risk of process failure.

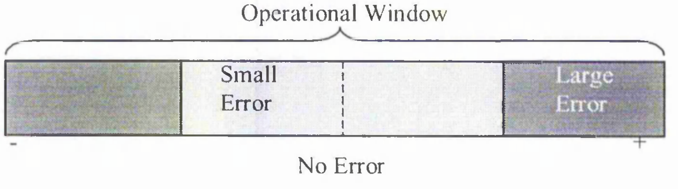


Figure 1. A schematic diagram of an operating window

**Caution**, some parameters are showing small errors and there is a possible risk of process failure.

**Critical**, one or more parameters have exceeded acceptable tolerance limits and process failure is imminent.

These states can be classified in the output fuzzy domain as three sets - SAFE, CAUTION, CRITICAL.

The rule base, to link input to output sets, is kept simple to allow a generic approach to be formulated (Figure 3). The sensitivity of the rules is governed by the distribution of the input sets. The effectiveness of the fuzzy logic process will depend on the calibration of the input variables, where input fuzzy sets are tuned to match intended operational tolerance windows.

The implementation of the rule base within the fuzzy logic framework must be properly considered. The choice of correlation/defuzzification method is an important aspect in 'respecting' the information present in the input parameters. To ensure this, a method has been developed to encapsulate the needs of this application.

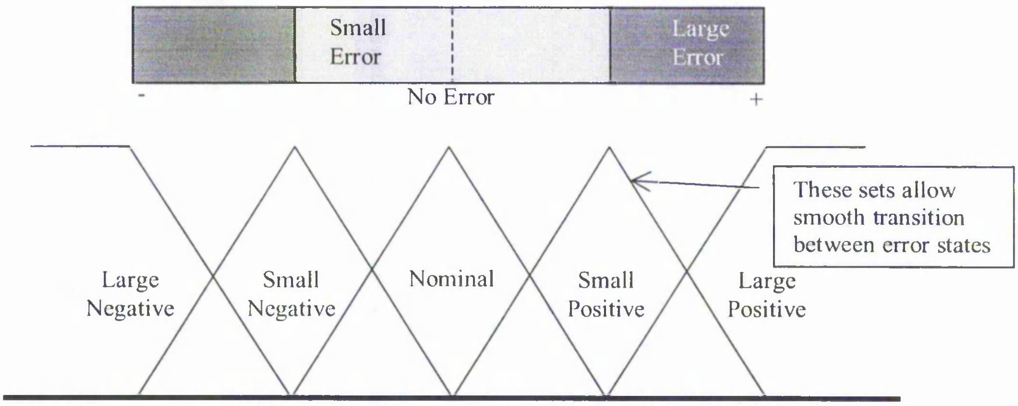


Figure 2. The mapping of an operating window onto the fuzzy domain.

***RULE BASE***

*if* PARAMETER_1 *is* NOMINAL *then* MACHINE *is* SAFE

*if* (PARAMETER_1 *is* SMALL -VE) *or* (PARAMETER_1 *is* SMALL +VE) *then* MACHINE *is* CAUTION

if (PARAMETER_1 is LARGE -VE) *or* (PARAMETER_1 *is* LARGE +VE) *then* MACHINE *is* CRITICAL

where PARAMETER_1 is, for example, the relative offset between two rotational elements during one machine cycle.

Figure 3. An example of the rule base structure.

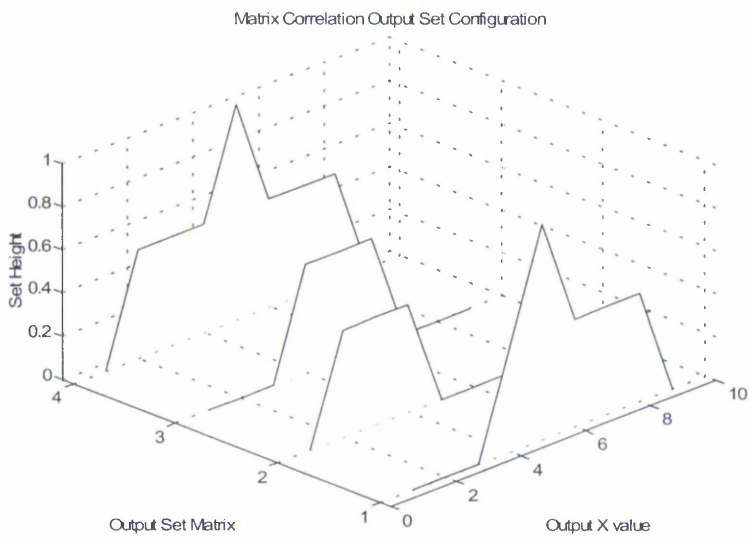


Figure 4. Three-dimensional representation of stored truncation values.

## 3. MATRIX INFERENCE

A commonly used method for fuzzy inference, is the min-max method. However, this simple but effective technique does have a drawback when used in problems related to risk assessment type problems, such as monitoring and diagnosis of the state of a machine. In this application, it is desired that all rules contribute towards the final defuzzified result to ensure the derivation of a good estimation of machine state. With the min-max method, only rules presenting a maximum truth-value for a given output set are contributing to the final outcome. A method for alleviating this problem is to use the fuzzy additive approach, where the results of the rules are summed to determine the output set truncation. However, this approach is deemed unsatisfactory, as rules are prevented from contributing further if the total calculated truncation value exceeds the height of the output set. Therefore, a new approach, entitled the 'matrix' method, has been devised to meet the requirements of this application. Developed from the min-max method, the new approach replaces the maximum selection technique for output set truncation, by a matrix of truncation values derived from all rule outcomes. By utilising the matrix values during defuzzification the fuzzy output domain, in effect, becomes three dimensional - as each possible cut is retained (Figure 4).
A centroid method (Eq. 1) can be used to defuzzify the matrix.

$$C_o = \frac{\int x\mu_o(x)dx}{\int \mu_o(x)dx} \quad (1)$$

*where,*
$x$ *is the position on the x-axis of an output set*
$\mu_o(x)$ *is the height of any given set at point x.*
$c_o$ *is the total centroid.*

This can be made discrete using Eq. 2, for computational ease.

$$C_o = \frac{\sum_{i=1}^{N} c_i A_i}{\sum_{i=1}^{N} A_i} \quad (2)$$

*where,*
$A_i$ *is the area of a set.*
$c_i$ *is the individual set centroid*
$N$ *refers to the total number of output sets.*

As can be seen from Eq. 2 the areas of the fuzzy sets are critical to the determination of the final defuzzified result. This is an important factor in shaping the way a fuzzy logic system links input and output parameters, by means of centroid defuzzification. This in turn leads to the creation of fuzzy decision surface. The flexibility of the centroid method proves useful in accommodating the new matrix approach, as two fuzzy sets can effectively occupy the same x-position and contribute equally to the defuzzified result.
A property of the present method is that the relationship between a given rule outcome and the consequent output set has been altered to provide a more suitable approach for this application, compared to that offered by other inference techniques. As mentioned earlier, the min-max method suffers from a disadvantage of not utilising all rule outcomes for determining output set truncation. As emphasis moves between rules, changes are only visible when the current maximum truncation value has been exceeded, even if the new rule outcome is almost equal in weighting. This leads to 'surprise' changes in the defuzzified output. In the fuzzy additive approach, truncation values are added but a non-linear result is formed, as the relationship between a rule outcome and the area created by truncation of the output set is not equal for all cases. For example, if two rules modify the same output set and both produce a result of 0.5, the consequent set would be of height 1.0. However, the second rule has only one-third the additive effect of a single rule, due to the triangular shape of the set defining the area. Where in fact the consequent set might be desired to have twice the area, which would double the effective weight of the set during the process of centroid defuzzification. The matrix method can overcome both of these problems. Firstly, all rules are considered however small their contribution. Secondly, rules producing equal results have equal influence on the final defuzzified value. The effect, therefore, is to subtly change the topology of the fuzzy decision surface linking input to output values.
A comparison between min-max and matrix methods is shown in Figure 5. Here the effects of two input variables, given an arbitrary scale of 0-50, were tested using the rule base developed specifically for the condition monitoring application. Unlike the min-max method, the matrix method shows a flattening of the areas associated with the centre of the output sets, creating smoother transitions through these points. It has also been found that computational overheads are not increased by this new method, as the computationally expensive maximum comparison is no longer required.

## 4. APPLICATION AND RESULTS

To evaluate the proposed methodology, a purposely-built condition monitoring experimental test rig has been designed to mimic the operation of a film packaging machine (Figure 6), including the main rotational and reciprocating motions.
The rig consists of three constituent parts.

- Instrumentation - three optical shaft encoders and a proximity sensor are used, each of which is driven by a stepper motor.
- Hardware interface and data processing - a novel data acquisition interface has been developed, capable of the parallel sampling of analogue and digital signals.

- Software analysis and display - a personal computer is used to log, assimilate and display the data. Data analysis is to be carried out by the fuzzy diagnostic software.

Instrumentation used on the test rig is identical to that intended for use on the actual packaging machine. The stepper motor drive components represent four of the main machine elements, the performance of which is critical in determining machine condition. Three of the motors are used to provide rotational motion, which simulate parts of the conveyor system that moves product through the machine. The fourth is used to create a reciprocating motion, by means of a cam and follower, to model the action of a 'rake' used to bring the carton boxes in-line with the product. This component is critical to the success of the packaging action, because misalignment here will guarantee failure. The rig permits the introduction of controlled errors into individual simulated components.

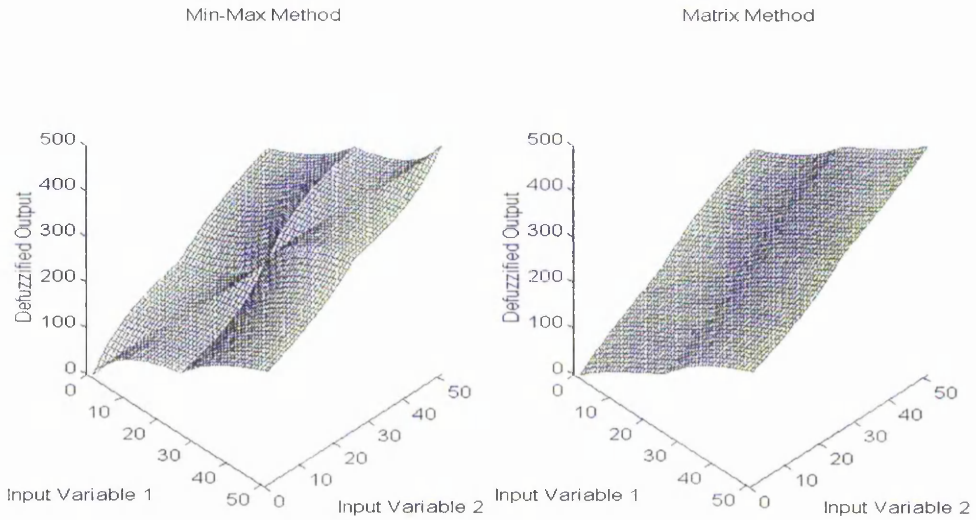


Figure 5. A comparison of the fuzzy decision surface generated by Min-max and Matrix methods.

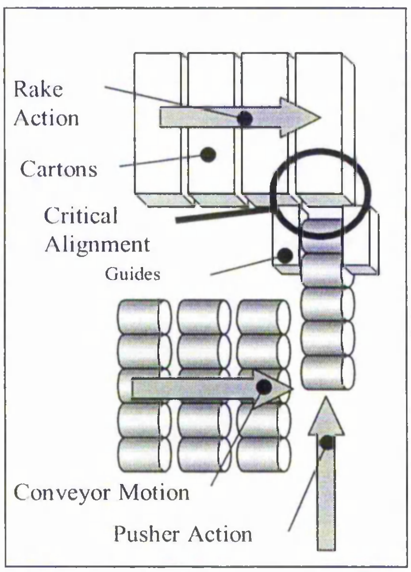


Figure 6. Representation of the packaging machine

The real-time condition monitoring system acquires signals from the optical encoders and the proximity sensor and, through processing, forms six observable parameters. All are considered over one machine cycle, which is the time taken to fill a single box.

Preliminary investigations have been carried out to test the general performance of the developed fuzzy logic approach, with respect to its ease of implementation and process applicability. Initial results were obtained by specifying arbitrary input sets to simulate operational windows. It should also be noted that by tuning the input sets, the fuzzy processing has been made more sensitive to changes in absolute position of the rake, measured by the proximity sensor, reflecting its importance in ensuring successful packaging operations.

Figures 7, 8 & 9 show three events over a 100-cycle section from a test run of the condition monitoring rig. The first event, covering the first ten cycles, is a large offset between two encoders (Figure 7). The proximity sensor is showing a small error at this stage (Figure 8). The output from the fuzzy logic (Figure 9) registers a high value of approximately 0.7, indicating an increased likelihood of process failure. Over the next 80 cycles, a minimal offset situation is recorded; while the proximity sensor indicates an increasing error on an upward trend. Over this period, the fuzzy result (Figure 9) can be seen to track the small variations manifested in the proximity sensor readings, whilst noting an offset error spike between the encoders just before machine cycle 70. Towards the end of the test run a sharp change in encoder offset is introduced, when the proximity sensor is already registering a large error. This results in a large increase in the output from the fuzzy logic diagnosis. The flatness of the profile is an indication that the tracked parameters have exceeded the limits of their operating windows. Hence, if this data represented a scenario of the operational conditions of an actual packaging machine, it could be concluded that a high probability of a critical process failure would occur at this time.

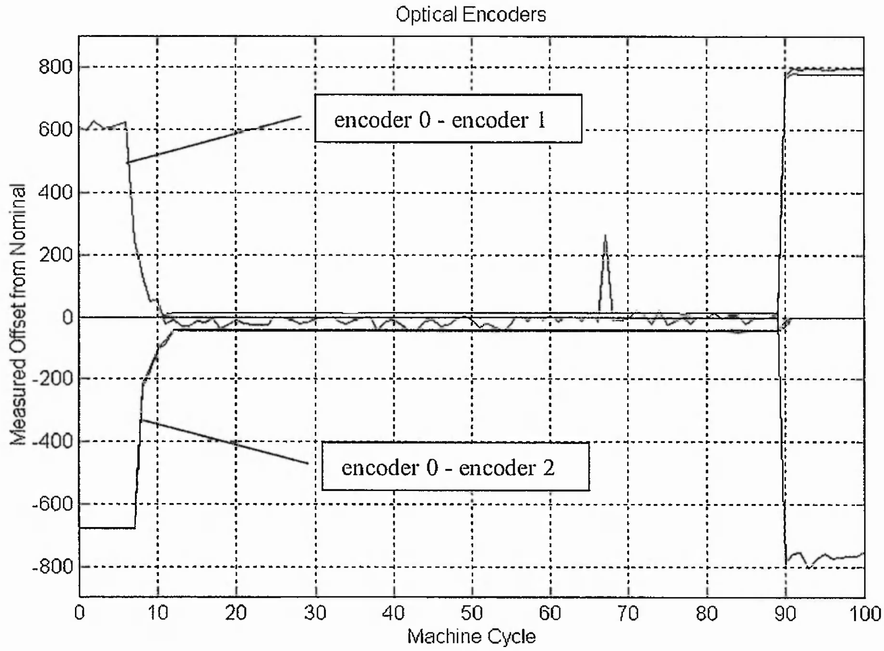


Figure 7. Optical Encoder Measurements.

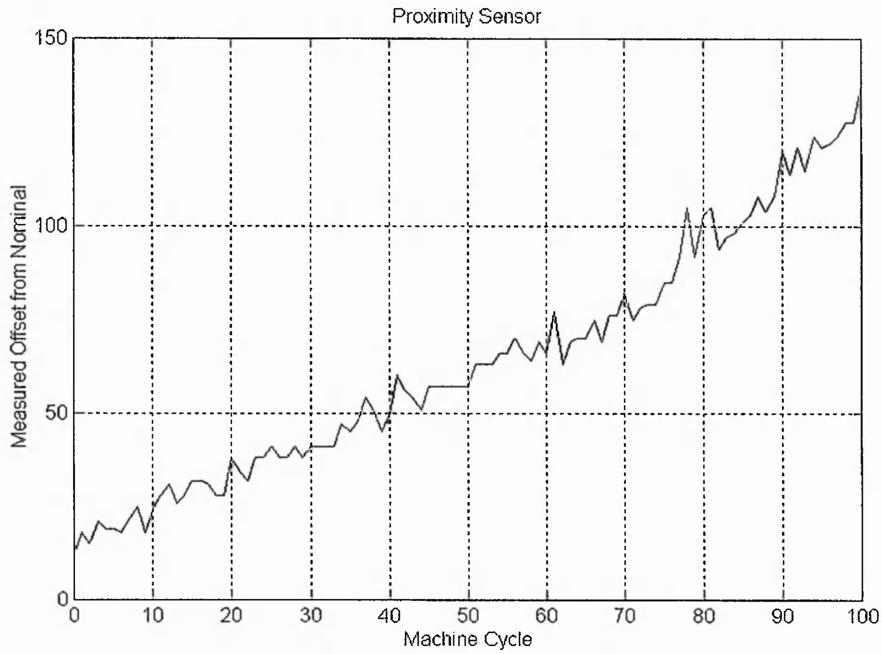

Figure 8. Proximity Sensor Measurements.

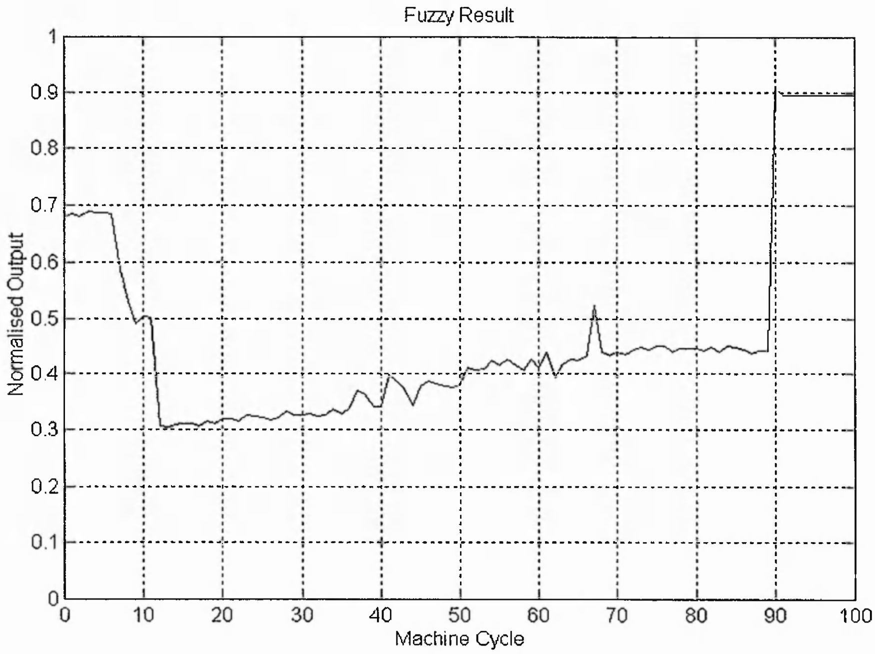


Figure 9. The calculated fuzzy result classifying machine state.

## 5. CONCLUSIONS

The matrix inference method is capable of combining multiple, but not necessarily similar, variables into a single output value. This in turn will provide a useful means of tracking machine performance through the monitoring of a single value, by tuning fuzzy sets to match estimated operational windows. A suitably generic method has been employed in the formation of the rule base and the linking of operational windows to the fuzzy domain. This will allow the developed techniques to be utilised for a wide range of manufacturing problems.
Preliminary results have shown that the matrix inference technique introduced within a fuzzy real time condition monitoring system can produce useful information for diagnosing the state of a packaging machine. Furthermore, fuzzy logic can provide a methodical approach for close estimation of operational windows. The work has therefore lent support to the use of fuzzy condition monitoring as a reliable and inexpensive means of reducing wastage and maintenance overheads in the packaging industry.

## 6. REFERENCES

[1] Plantenberg D H, Lai E and Jeffries M. 1997, A Novel Approach to Reduce Packaging Process Down-Time *Advances in Materials and Processing Technologies, AMPT '97* **Vol. 2**, 630-635.

[2] Rondeau L, Ruelas R, Levrat L, Lamotte M. 1997, A defuzzification method respecting the fuzzification *Fuzzy Sets and Systems* **86**, 311-320

[3] Chow M, Magnum P M and Yee S O. 1991, A neural network approach to real-time condition monitoring of induction motors. *IEEE Trans. On Industrial Electronics,* **Vol. 38**, 6, 448-453.

[4] Aluindigue I E et al. 1993, Monitoring and diagnosis of rolling element bearings using artificial neural networks. *IEEE Trans. On Industrial Electronics*, **Vol. 40**, 2, 209-216.

[5] Lai E, Plantenberg D H, Grant Z R, Hull J B. 1996, Condition Monitoring of plant and equipment with the aid of neural networks. *Proc. Symposium on Mechanics in Design, University Of Toronto, Canada*, **6-9 May**, 515-522.

[6] Li S and Elbestawi M A. 1996, Tool Condition Monitoring in Machining by Fuzzy Neural Networks,

*Journal of Dynamic Systems, Measurement, and Control, Transactions of the ASME,* **Vol. 118**, 665-672.

[7] Das S, Bandyopadhyay P P, Chattopadhyay A B, 1997, Neural-networks-based tool wear monitoring in turning medium carbon steel using a coated carbide tool, *Journal of Materials Processing Technology*, **Vol.63**, No.1-3, pp.187-192

# REAL-TIME IMPLEMENTATION OF A FUZZY CONDITION MONITORING SYSTEM FOR PREDICTING PROCESS BREAKDOWN

M. Jeffries, E. Lai, D.H. Plantenberg and J.B. Hull
The Nottingham Trent University, Department of Mechanical and Manufacturing Engineering, Nottingham NG1 4BU, United Kingdom. Fax +44 115 948 6506, email: martyn.jeffries@ntu.ac.uk or e.lai@domme.ntu.ac.uk

## Abstract

For a packaging process, machine set-up/adjustment and corrective maintenance are the two main contributors to recurrent production overheads. Whilst the diagnostic capability of condition monitoring may enable preventative maintenance work to be carried out when required, machine set-up/adjustment is a time consuming trial and error process normally carried out by experienced mechanics. Incorrect machine set up can lead to misalignment between moving parts and hence products jamming during packaging operation. The present study seeks to develop a fuzzy condition monitoring system to enable an expert decision to be made on the probability of an imminent process breakdown brought about by components misalignment.

An experimental rig has been designed to mimic the rotational and translational movements of a packaging machine. It consists of three shaft encoders, a proximity sensor, a novel data acquisition interface capable of parallel sampling, and a personal computer. The state of the machine is to be ascertained by means of fuzzy logic based on the analysis of the measurements of six input variables. The input sets represent the operational windows of the measured variables and expert knowledge is represented by rules. A new matrix inference technique has been developed to allow the determination of a single output value upon which an assessment can be made on the likelihood of an imminent breakdown.

By adjusting the weighting of the rules and the sensitivity of the input variables, the experimental results have shown that the fuzzy system is capable of modelling the dynamic behaviour of a packaging machine, which can have several set up variations. Furthermore, the novel data acquisition interface and software enables all performance calculations, decision making and information display to be completed within one cycle of machine operation. The fuzzy condition monitoring system thus developed is generic and can be applied to other manufacturing processes.

Keywords: condition monitoring, fuzzy logic, real-time, packaging, and downtime.

## 1 Introduction

The investigation of condition monitoring within the manufacturing industry has been largely confined to tool-wear monitoring [1] and direct machinery measurement such as gearboxes or bearings. The monitoring of overall machine condition, especially in the field of packaging has only recently begun [2]. This is despite the relatively high recurrent production overheads resulting from routine adjustment and corrective maintenance of packaging machines. To obtain satisfactory performance, current remedial actions utilise the knowledge and skills of an experienced mechanic. However, machine set-up/adjustment can still be a time consuming process involving a certain amount of trial-and-error. The types of machines being considered often contain rotational, reciprocating and translational motions. Two major factors need to be overcome when attempting to reduce maintenance-related overheads: (i) the lack of information describing the state of a process and (ii) the problem solving skills of an experienced mechanic usually take a long time to acquire. It may be considered that fuzzy logic, in combination with a real-time condition monitoring (RTCM), can offer a solution to helping resolve these problems, by encapsulating the heuristic qualities of the problem solving skills used by an experienced mechanic.

Artificial neural networks (ANNs) have proved valuable in the traditional areas of condition monitoring [3-9], and fuzzy logic in combination with ANNs has also been explored [10-11]. Nevertheless, the benefits of applying fuzzy logic alone in real-time condition monitoring have been under-exploited. While fuzzy logic has proved popular in the control field for solving 'difficult-to-quantify' problems, the danger of careless implementation has also been highlighted [12]. The need to observe good theoretical practice when implementing fuzzy logic systems can often be ignored with little adverse effect, due to their generally robust nature. However, long-term reliability and quality assurance is difficult to maintain should such a method be employed. When implementing a fuzzy condition monitoring within a real-time environment, there is an expectation that the software developed is more reliable than the machine being measured. It is therefore necessary to fully understand all processes involved in data acquisition, correlation, and display. Fuzzy logic can then be used for its abilities to manage difficult-to-quantify variables and its stability under unforeseen conditions.

Due consideration must also be given to ensure that any software within the fuzzy monitoring system is designed from the outset with real-time implementation in mind. The use of techniques such as multi-tasking or parallel processing have to be considered in ensuring data integrity, especially if synchronisation information is contained within the measured parameters [13].

This paper discusses the techniques and problems associated with implementation of a fuzzy logic within a real-time condition monitoring system. Particular emphasis is given to the real-time monitoring of packaging machines. The results of experimental evaluation of a prototype fuzzy condition monitoring system are also presented.

## 2 Application

The main aspects of a packaging machine are rotary motions of conveyor systems and reciprocating motions of actuating objects, such as a rake or a pusher. A pre-requisite to achieving good monitoring performance is the measurement of synchronous offsets, which necessitates the need for parallel sampling of analogue and digital signals from different sensors - a feature not readily available from commercial interface or monitoring products. It was therefore necessary to develop a complete package incorporating hardware and software aspects capable of meeting these requirements. A standard PC (486 class) has been used for software processing.

A purposely-designed test rig has been built to enable the validation of the prototype system prior to industrial implementation. Four main constituent motions of the packaging machine - three rotational, one reciprocating - are simulated using stepper motors (Figure 1), with the reciprocating motion achieved through the use of a cam-and-follower. Instrumentation includes three optical shaft encoders and an induction proximity sensor, identical to those intended for use on the packaging machine.

In addition to the developed instrumentation system, investigations into the benefits of fuzzy logic have been undertaken. As machine parameters often interact with each other in a non-linear manner, it is difficult to determine an overall machine state from several measured parameters of the monitoring using standard algorithmic techniques. However, fuzzy logic appears to offer a solution by encapsulating heuristic information on machine condition, in a similar manner to the method currently employed by a mechanic to diagnose problems. By representing operational tolerance windows of the machine parameters (Figure 2) in the form of fuzzy sets, the errors associated with individual parameters can be quantified and transformed into the fuzzy domain.

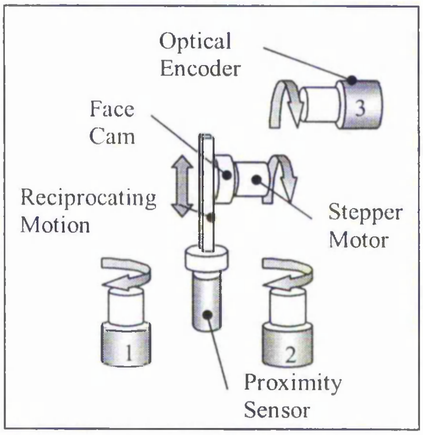


Figure 1. Test rig schematic

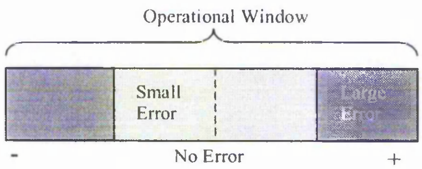


Figure 2. An operational window

This permits the use of the fuzzy rule base to consolidate the input variables into a single (defuzzified) value representing overall machine state (Figure 3). A new 'matrix' method [14] has been developed to overcome some of the shortcomings associated with other inference techniques (such as min-max, additive), when tackling this type of risk assessment problem. These shortcomings include the lack of ability to consider all rules when formulating the defuzzified result and to provide predictable influence of each rule. Nevertheless, implementation of the proposed fuzzy condition monitoring system requires due consideration for application within a real-time operating environment.

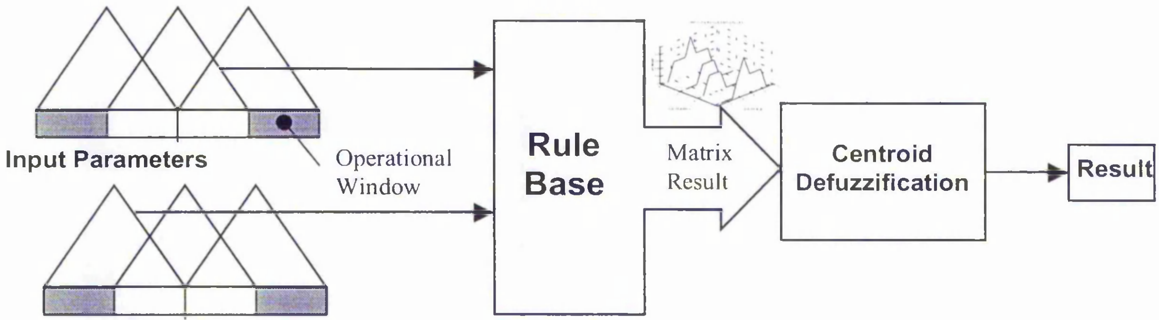


Figure 3. The Fuzzy Approach

## 3 Real Time Implementation

In software simulations, lengthy calculation time between computational iterations begets only inconvenience. However, in real-time systems, highest priority is given to sampled data and its integrity must be maintained at all costs. When data is sampled from the outside world, it is imperative that no data is lost at this sampling stage, as this may invalidate any calculations, assumptions or predictions made.

Three features are commonly present in condition monitoring systems: sampling, logging, and data display. By adding inference in the form of fuzzy logic to a system, computational requirements are necessarily increased. Using the programming 'FOR' loop to control software processes, the PC is restrained within tasks until completion. Consequently, data integrity is compromised by the increasing demands of these tasks. Parallel processing is one way of resolving this problem by allowing separate processors to deal with separate parts of the process. Each task can have it's own processor and memory, thus preventing data loss. One major drawback in this approach can be prohibitively expensive both in development time and in cost. A more realistic approach for producing a cost-effective solution is multi-tasking; this is where a single processor is shared amongst a number of concurrent processes, in such a way that no one process completely takes over. This is achieved by making each process aware of its current state and the next planned state so that control can be relinquished without loss of data.

In many multi-tasking systems, it is possible for a processor to move between tasks freely. Control between tasks is implemented by a technique known as time slicing/sharing. Time Slicing (TS) allocates a small amount of CPU time to each process as required so that all are allowed to progress; this is obviously much slower than each task having its own processor. Multi-tasking of a PC usually takes place at the Operating System (OS) level and is usually outside the grasp of an average C programmer. Nevertheless, through adaptation, the same techniques can be applied at a higher operational level, i.e. in a condition monitoring system software.

The packaging process being measured operates in a cyclic fashion; information is logged and displayed once per cycle after being collated for the whole of the previous cycle. Data sampling has highest priority in this system and must be performed to ensure that sampling rate is never compromised. The rest of the program must utilise the remaining unused processor time. Unfortunately, computationally intensive processes such as graphics or time-consuming data logging cannot be completed between samples. Time slicing splits these 'troublesome' processes into small task packets and executes them within available time slots (Figure 4). This method is applicable as graphics display or data logging does not have to be completed between the sampling intervals, updates of once per machine cycle will meet the requirements of this process.

Fuzzy logic is not particularly computing intensive, but as the size of the rule base increases the program performance will suffer. In a real time system, it would not be possible to perform all of the fuzzy logic tasks within the sampling time (assuming the sample rate is high - measured in milliseconds). In the application considered here, the sample rate should be set high at 8ms, a necessity in fast process control or condition monitoring. If the fuzzy logic inference is required to make a decision every sample then the TS architecture offers no improvements in performance. However, if it is necessary to collate many data points before invoking the fuzzy logic then TS can offer a solution. As many manufacturing and industrial processes are cyclic in nature as opposed to continuous processes such as chemical production, this method can be considered suitably generic for a wide range of industrial applications.

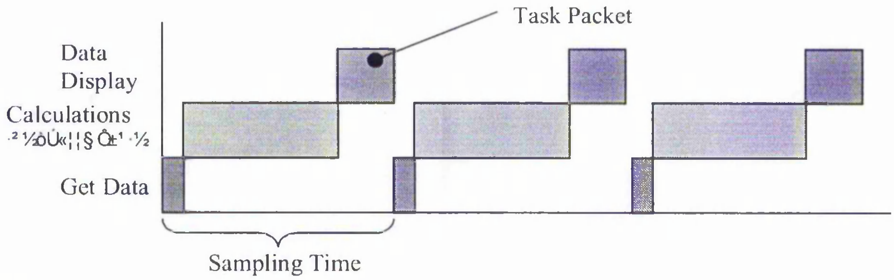


Figure 4. Time Slicing into task packets

The difficulties presented by the TS approach are not how well the task has been performed, but to do so simply, elegantly and robustly. There are some important criteria for consideration. A function must be self-aware of its own progress/state, therefore, it must be self-contained, so that it can be called from anywhere else within the program without any external knowledge of the progress of execution. This eliminates the need for a complex web of variables and pointers describing the possible states of software functions, hence providing a simple but robust solution. This feature will also allow for a certain amount of nestability (i.e. TS functions calling other TS functions), which is important for maintaining good program structure. It should also be noted that the computational effort of deducing the next process state must not be so time consuming as to negate the advantages of time slicing. The biggest obstacle presented by the TS method is developing the software in such a way that functions can be exited from upon part completion and then returned to in the next available CPU time slot to continue execution. The solution is to effectively change the program execution dynamically. The path through TS architecture depends on the progress of individually called TS functions, without using pre-defined loops. The effect of this method is to create a dynamic software architecture. Although the functional programming structure is easily recognisable, the program execution path becomes more convoluted but more efficient in the task at hand.

## 4 Results

The time slicing method for software development was implemented and tested on the condition monitoring test rig. By associating fuzzy input variables with possible machine parameters, a good appraisal of system performance was obtained. The parameters identified were relative offsets between the three encoders, absolute value of the proximity sensor, and a derived relationship between the proximity sensor and the encoders. These were considered good indicators for deducing overall machine condition. The sensitivity of the fuzzy inference engine to the variations of individual parameters was achieved by setting the fuzzy input domains (operational windows) to an estimation of critical tolerance. Hence, the values obtained from the proximity sensor were sensitised to follow small changes in alignment of the reciprocating device.

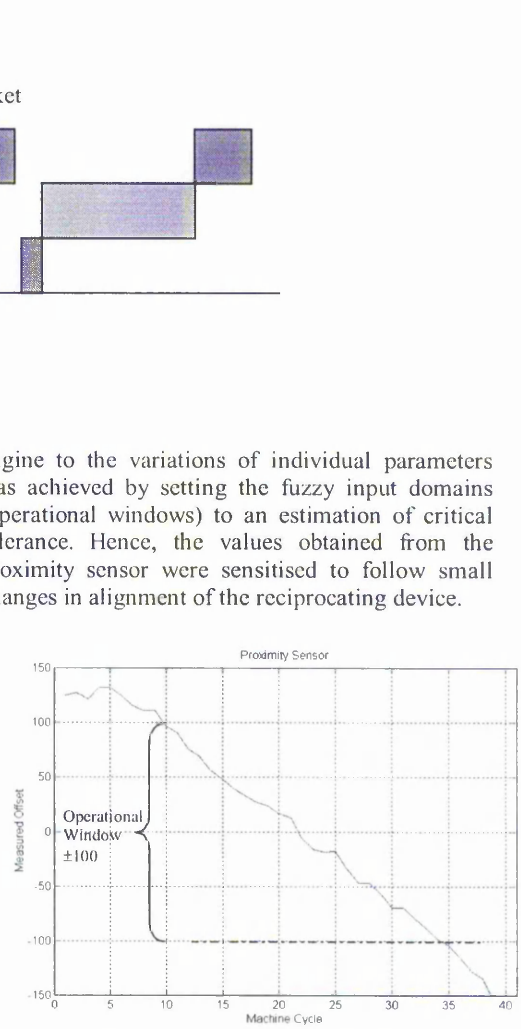


Figure 5. Proximity Sensor Measurements

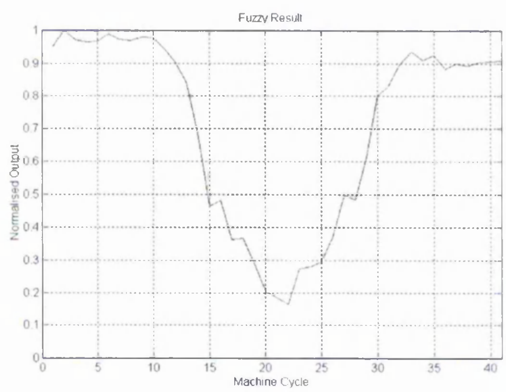


Figure 6. Defuzzified Result

Figures 5 & 6 show the effect on the fuzzy outcome from changes in the proximity sensor from an experimental run of the test rig for 41 simulated machine cycles. The synchronisation error of the optical encoders remained constant during these cycles. The operational window of the proximity sensor was set at ±100 (arbitrary units, proportional to a measured distance of approximately 0.2mm); intersected at cycle 10 and cycle 34 (Figure 5).

As can be seen, the fuzzy result (Figure 6) is unchanged whilst measured values are outside the operational window, cycles 0-9 and 35-41. Once measurements are received within the operational window, the fuzzy result is able to track the changes, smoothly and representatively. Variations in the fuzzy result contour are indicative of measurement noise present in the instrumentation.

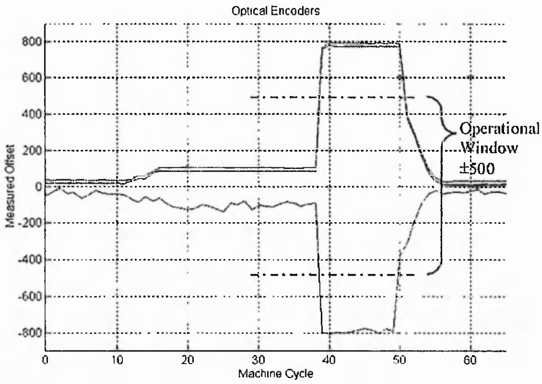


Figure 7. Optical Encoder Offsets

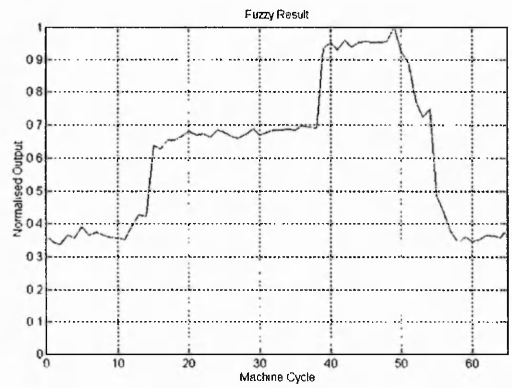


Figure 8. Defuzzified result

Figures 7 & 8 show the effect of changes of offset between optical encoders. Within the test run of 65 cycles, a synchronisation error is introduced into encoder 1. This can be seen in the three traces of Figure 7 at cycle 10; the error also increasing again at cycle 38. The positive traces are the offsets between encoder 1 and encoder 2 and between encoder 1 and encoder 3; the negative trace is the calculated estimate of the offset between the proximity sensor and encoder 1. The operational window of the encoders was set at ±500.

Figure 8 shows the fuzzy result is able to track changes in the offsets and is not confused by the positive and negative values of input parameters. At cycle 38, the encoder offset exceeds the operational window and therefore the fuzzy response flattens off. However in practical situations, the result would have already triggered an action as the machine condition value has moved above its safe steady state value (0.4 in this example). It should be noted that beyond the operational window the actual value is irrelevant, as the parameter concerned is already in failure.

Based on these results, the matrix inference technique appears to provide an accurate and useful means of determining overall machine condition. This therefore provides the groundwork to the next stage of research - interpreting these results to allow suggestions to be made on corrective actions.

## 5 Conclusions

When considering solutions for manufacturing problems, cost-effective condition monitoring has a large appeal. This is especially relevant when the processes under examination are not of high premium, and expensive technological solutions do not provide a reasonable price/performance ratio to the process in question. Therefore, the scope for 'off-the-shelf' technology is broad, covering many manufacturing sectors with packaging being one example. By employing a standard PC, utilising a custom-built interface and with in-house developed software, the work presented here shows a method for implementing a cost-effective solution to many condition monitoring applications.

However, if sample by sample consideration is required then fast, dedicated processors with optimised code are the only answer. Time slicing offers a cost-effective solution for assimilating cyclic information only. This method does not allow a slow processor to outperform faster ones. Nevertheless, it does produce a sound methodology for the design of a real-time condition monitoring system, allowing the incorporation of additional fuzzy logic elements without sacrificing the overall performance.

The fuzzy logic has been implemented within a real-time condition monitoring system. In utilising a matrix inference technique for determining rule output, the results confirm that this approach is valid for risk assessment problems.

The fuzzy condition monitoring system discussed here provides the stepping stone to the development of a fully intelligent condition monitoring system for packaging and other processes. This approach presents itself as a robust and inexpensive means of tackling the problems of wastage and maintenance management in the modern manufacturing environment.

## 6 References

[1] Byrne G, Dornfeld D, Inasaki I, Ketteler G, Konig W, Teti R. 1995, Tool condition monitoring (TCM) - the status of research and industrial application. *CIRP Annals - Manufacturing Technology*, **Vol.44**, No.2, pp.541-567

[2] Plantenberg D H, Lai E and Jeffries M. 1997, A Novel Approach to Reduce Packaging Process Down-Time *Advances in Materials and Processing Technologies, AMPT '97* **Vol. 2**, 630-635.

[3] Chow M, Magnum P M and Yee S O. 1991, A neural network approach to real-time condition monitoring of 7induction motors. *IEEE Trans. On Industrial Electronics*, **Vol. 38**, 6, 448-453.

[4] Aluindigue I E et al. 1993, Monitoring and diagnosis of rolling element bearings using artificial neural networks. *IEEE Trans. On Industrial Electronics*, **Vol. 40**, 2, 209-216.

[5] Lai E, Plantenberg D H, Grant Z R, Hull J B. 1996, Condition Monitoring of plants and equipment with the aid of neural networks. *Proc. Symposium on Mechaincs in Design, University Of Toronto, Canada*, **6-9 May**, 515-522.

[6] Hong G S, RahmanM, Zhou Q. 1996. Using neural network for tool condition monitoring based on wavelet decomposition *International Journal of Machine Tools & Manufacture*, **Vol.36**, No.5, pp.551-566

[7] Javed M A, Hope A D, Littlefair G, Adradi D, Smith G T, Rao B K N, 1996, On-line tool condition monitoring using artificial neural networks *Insight: Non-Destructive Testing and Condition Monitoring*, **Vol.38**, No.5, pp.351-354

[8] Das S, Roy R, Chattopadhyay AB, 1996, Evaluation of wear of turning carbide inserts using neural networks *International Journal of Machine Tools & Manufacture*, **Vol.36**, No.7, pp.789-797

[9] Das S, Bandyopadhyay P P, Chattopadhyay A B, 1997, Neural-networks-based tool wear monitoring in turning medium carbon steel using a coated carbide tool, *Journal of Materials Processing Technology*, **Vol.63**, No.1-3, pp.187-192

[10] Li S and Elbestawi M A. 1996, Tool Condition Monitoring in Machining by Fuzzy Neural Networks, *Journal of Dynamic Systems, Measurement, and Control, Transactions of the ASME*, **Vol. 118**, 665-672.

[11] Li S, Elbestawi M A, 1996, Fuzzy clustering for automated tool condition monitoring in machining, *Mechanical Systems & Signal Processing*, **Vol.10**, No.5, pp.533-550

[12] Rondeau L, Ruelas R, Levrat L, Lamotte M. 1997, A defuzzification method respecting the fuzzification *Fuzzy Sets and Systems* **86**, 311-320

[13] Pressman R S, 1987. Software Engineering - A Practitioners Approach 2nd Edition, McGraw-Hill International Editions, p367-402

[14] Jeffries M, Lai E, Plantenberg D H, Hull J B. 1997, A Fuzzy Approach to the Condition Monitoring of a Packaging Plant. Accepted for presentation at *6th International Scientific Conference "Achievements in Mechanical & Materials Engineering" November 28th - December3rd 1997, Poland/Hungary.*

# Appendix C

## Motion Controller Program Listing

```
/***************************************************************************
 * Module:   IAGV.H
 *
 * Function: Header file for the main program
 *
 **************************************************************************/
#ifndef _IAGV_H
#define _IAGV_H

#include <stdio.h>
#include <scim.h>
#include <spc1.h>
#include <sqd4.h>
#include <sad.h>
#include "gvglobal.h"
#include "gvconst.h"
#include "initdata.h"
#include "nvdata.h"
#include "traction.h"
#include "steering.h"

/* function prototypes */
void main( void );
#endif /* _IAGV_H */
/* EOF */
```

```
/***************************************************************************
 * Program : IAGV.C
 *
 * Function: Calculate steering angles and store them in memory. Switch the
 *           Power for the steering system on and adjust the steering angles
 *           to their theoretical angular position.  Then switch the power
 *           to the drive motor on and capture the data.  When the
 *           travelling distance has been completed switch the power off
 *
 * Compile & link using Borland C v 3.1 or v4.0 only
 *
 * SMALL Memory Model
 *                bcc -c -v -1 -ms iagv.c
 *                slink iagv sqd4 spc1
 *
 **************************************************************************/
#include "iagv.h"
void main( void )
{
      static StructIAGV IAGV = IAGVINIT;
      static PtrStructIAGV PtrIAGV = NULL;
      static StructSQD4 SQD4_01 = SQD4INIT;
      static PtrStructSQD4 PtrSQD4_01 = &SQD4_01;
      static StructSPC1 SPC1_01 = SPC1INIT;
      static PtrStructSPC1 PtrSPC1_01 = &SPC1_01;
#ifdef SIMULATION
      static StructNVIAGV     NVIAGV;
      static FPtrStructNVIAGV FPtrNVIAGV = &NVIAGV;
#else
      static FPtrStructNVIAGV FPtrNVIAGV = MK_FP(STARTNVRAM, 0);
      /* start address of non-volatile RAM */
#endif
      static StructIAGVdata IAGVData0 = IAGVDATAINIT;
      static StructIAGVdata IAGVData1 = IAGVDATAINIT;
      int     OverRun;
      const  char IDmessage[IDLENGTH]="CR(-70,-750) 30-Sep-99";
```

```
printf("\nIAGV version 1.5a 30-09-99 dhp");
PtrIAGV = &IAGV;
PtrIAGV->PtrIAGVdata = &IAGVData0;
PtrIAGV->PtrIAGVdata->PtrNext =
                        PtrIAGV->PtrIAGVdata->PtrPrev = &IAGVData1;
PtrIAGV->PtrIAGVdata = PtrIAGV->PtrIAGVdata->PtrNext;
PtrIAGV->PtrIAGVdata->PtrNext =
                        PtrIAGV->PtrIAGVdata->PtrPrev = &IAGVData0;
PtrIAGV->PtrSQD4 = PtrSQD4_01;
PtrIAGV->PtrSPC1 = PtrSPC1_01;
PtrIAGV->FPtrNVIAGV = FPtrNVIAGV;

PtrIAGV->TargetICR.x=-70.0;   // do not place icr within IAGV envelope
PtrIAGV->TargetICR.y=-750.0;

save_IDstring( PtrIAGV, IDmessage );
if(!init_data( PtrIAGV ))
      return;
if(calc_steertable( PtrIAGV ))
      printf("\n data set:  initialised");
PtrIAGV->PtrSPC1->IOdata[POWER] PWRON PWRSTEER;
printf("\n steering:  ON");
PtrIAGV->PtrSPC1->IOdata[POWER] PWROFF PWRDRIVE;
printf("\n traction:  OFF");
write_spc1( PtrIAGV->PtrSPC1 );
reset_IAGV(PtrIAGV);
if(!get_nvIAGVsettings( PtrIAGV ))
      printf("\n no stored data\n");
printf("\n steering:");
start_timer1();
start_timer2();
while(set_steering(PtrIAGV))
{
      if((OverRun = synchronise())!= 0) // synchronise to SYNC signal
            printf("\r %u",OverRun);    // limits the step rate
      write_spc1( PtrIAGV->PtrSPC1 );   // implement steering action
} //  while(set_steering(PtrIAGV))
printf("  OK");
get_nvIAGVstatus(PtrIAGV);
PtrIAGV->PtrSPC1->IOdata[POWER] PWRON PWRDRIVE;
printf(" traction:  ON\n");
start_timer1();
start_timer2();
for(;;) // ever
{
      if((OverRun = synchronise())!= 0) // synchronise to SYNC signal
            printf("\r %u",OverRun);
      PtrIAGV->PtrSPC1->IOdata[POWER] =
            PtrIAGV->PtrSPC1->IOdata[POWER] SET SYNCFLAG;
      write_spc1( PtrIAGV->PtrSPC1 ); // needed here to set the bit
      read_sqd4( PtrIAGV->PtrSQD4 );
      if(calc_traction( PtrIAGV ))
      {
            if(save_nvIAGVdata( PtrIAGV ) == 0)
            PtrIAGV->PtrSPC1->IOdata[POWER] PWROFF PWRDRIVE;
      } // if(calc_traction( PtrIAGVData))
      if(!set_steering(PtrIAGV)) // only if aligned calc next position
            calc_steering(PtrIAGV);
      set_traction(PtrIAGV);
      PtrIAGV->PtrIAGVdata = PtrIAGV->PtrIAGVdata->PtrNext;
                                         // pointer set to next dataset
      PtrIAGV->PtrSPC1->IOdata[POWER] =
            PtrIAGV->PtrSPC1->IOdata[POWER] CLEAR SYNCFLAG;
```

```
            write_spc1( PtrIAGV->PtrSPC1 ); // implement steering traction
      } // for(;;)
} // void main( void )
/* EOF */

/**************************************************************************
 * Module:   GVCONST.H
 *
 * Function: Global constants for the guided vehicle
 *
 **************************************************************************/
#ifndef _GVCONST_H
#define _GVCONST_H

/* miscellaneous constants */
#define INCSAVETHRESHOLD        5   // minimum increments before start sampling
#define SAMPLESAVETHRESHOLD     2   // set to zero to save every sample
#define STOPWHEELREV            70 // number of wheel 1 revolutions to stop
#define STOPWHEELPOS            100 // maximum 9999

#define MAXSAMPLES              65000L / DATASTRUCTSIZE //leave 536 bytes free
#define TRACTIONTHRESHOLD       2

#define INCFACTOR               2.9 // needs to be a float to display properly

#define STEPSPERRAD             849 // toggles
#define STEPRESOLUTION          1.0 / STEPSPERRAD
#define PI                      3.14159265359
#define IDLENGTH                50  // Length of string to ID data

#define STEERTABLESIZE  120

/* IAGVstatus flag identifier */
#define SAVEDATA        0x01  // Flag to indicate that data needs saving
#define STEEROK         0x02  // Flag to indivate steering position OK

#define PWRON       &= ~
#define PWROFF      |=
#define SET         |
#define SETBIT      |=
#define CLEAR       &~
#define CLEARBIT    &= ~
#define IAGVINIT    {{ 0,0,0,0 },      /* TargetWheelRev[] */\
                    { 0,0,0,0 },       /* TargetWheelPos[] */\
                      NULL,            /* PtrSteerCombi */\
                    { 0, 0 },          /* TargetICR */\
                      0x00,            /* IAGVstatus */\
                      NULL,            /* PtrIAGVData */\
                      NULL,            /* PtrSQD4 */\
                      NULL,            /* PtrSPC1 */\
                      NULL}            /* FPtrNVIAGV */

#define IAGVDATAINIT { NULL,           /* PtrNext */\
                       NULL,           /* PtrPrev*/\
                     { 0,0,0,0 },      /* WheelPos[] */\
                     { 0,0,0,0 }}      /* WheelInc[] */
#define NOOFWHEELS                     4
#define MAXSTEERTOGGLES 3259 // Maximum toggles allowed to limit rotation
typedef enum eWheels { W1, W2, W3, W4 };

#define IAGV_2LE        298   /* half length between wheel axles */
#define IAGV_2WI        210   /* half width between wheel axles */
#define CtoW1x          IAGV_2LE
```

```
#define CtoW1y          IAGV_2WI
#define CtoW2x          -IAGV_2LE
#define CtoW2y          IAGV_2WI
#define CtoW3x          -IAGV_2LE
#define CtoW3y          -IAGV_2WI
#define CtoW4x          IAGV_2LE
#define CtoW4y          -IAGV_2WI
#define COGx            4.27 // cog offset in x direction
#define COGy            3.75 // cog offset in y direction

/* constants for the SCIM88 controller */

/* timer operation control register */
#define T_EN            0x8000      /* enable */
#define T_INH           0x4000      /* inhibit */
#define T_INT           0x2000      /* interrupt */
#define T_RIU           0x1000      /* register in use */
#define T_MC            0x0020      /* maximum count */
#define T_RTG           0x0010      /* retrigger */
#define T_P             0x0008      /* prescaler */
#define T_EXT           0x0004      /* external clock */
#define T_ALT           0x0002      /* alternate compare register */
#define T_CONT          0x0001      /* continuous mode */
#define TTIMER          0.25        /* 16Mhz/4 = 4Mhz : 0.25us */
#define T1MARK          (int)(   5 /TTIMER)  /* mark time in us */
#define T1SPACE         (int)(4995 /TTIMER)  /* space time in us */
                         /* 833us = 1200Hz = max freq for stepper motor */
#define T1CONTROLREG 0x0000|T_EN|T_INH|T_ALT|T_CONT
#define T2MARK          2
#define T2CONTROLREG 0x0000|T_EN|T_INH|T_CONT

#define STARTNVRAM      0x4000

/* constants for the SQD4 interface */
#define SQD4INIT { SQD4BASE01,       /* Base Address for SPC1 */\
            { 0, 0, 0, 0 },/* IOdata */\
            { 2500 * -4, 2500 * -4, 2500 * -4, 2500 * -4 }, /* ENCres */\
            { 2500 * -2, 2500 * -2, 2500 * -2, 2500 * -2 }} /* ENC_2res */

/* constants for the SPC1 interface */
#define PWRSTEER        0x80
#define PWRDRIVE        0x40
#define SYNCFLAG        0x20
#define W1STEP          0x01
#define W1DIR           0x02
#define W2STEP          0x04
#define W2DIR           0x08
#define W3STEP          0x10
#define W3DIR           0x20
#define W4STEP          0x40
#define W4DIR           0x80
#define STEERSTEPINIT { W1STEP, W2STEP, W3STEP, W4STEP }
#define STEERDIRINIT  { W1DIR,  W2DIR,  W3DIR,  W4DIR }

#define SPC1INIT  { SPC1BASE01, /* Base Address for SPC1 */\
                  { 0, 0 }}     /* IOdata */

#endif /* _GVCONST_H */
/* EOF */
```

```
/****************************************************************************
 * Module:    GVGLOBAL.H
 *
 * Function: Global variables for the IAGV system
 *
 ****************************************************************************/
#ifndef _GVGLOBAL_H
#define _GVGLOBAL_H
#include <sqd4.h>
#include <spc1.h>
#include <sad.h>

typedef struct ticrcomplex {
      double      x;
      double      y;
};
typedef struct ticrcomplex icrcomplex;

/* This structure holds the data which is transferred to the
   non-volatile memory */
typedef struct tstructActualData {       /* Structure to hold one data set */
      volatile int      WheelRev[NOOFWHEELS];            /* the revolutions */
      volatile unsigned WheelPos[NOOFWHEELS];  /* the rotational position */
      volatile int      SteerTog[NOOFWHEELS];   /* the steering positions */
      volatile int      DataSampleNo;         /* number of the data sample */
};
typedef struct tstructActualData StructActualData;
typedef struct tstructActualData far *FPtrStructActualData;
#define DATASTRUCTSIZE sizeof(struct tstructActualData)

/* This structure is to be instantiated in the non-volatile RAM so to have
   a record of the situation after power loss */
typedef struct tstructNVIAGV {                   /* Structure to go in NVRAM.. */
      char IDstring[IDLENGTH];                     /* ..ID character string */
      int SamplingTime;                                  /* ..Sampling time */
      int SaveThreshold;
      volatile icrcomplex ActualICR;
      StructActualData Actual;
      FPtrStructActualData Current;       /* ..Pointer to current data set */
      FPtrStructActualData First;           /* ..Pointer to first data set */
      volatile int NoOfDataSetsStored;     /* ..Number of data sets stored */
};
typedef struct tstructNVIAGV StructNVIAGV;
typedef struct tstructNVIAGV far *FPtrStructNVIAGV;
/* Far pointer to structure */

/* Two of the following structures are to be instantiated in DRAM and used
   alternatively so to have a reference to the previous siuation */
typedef struct tstructIAGVdata {
      struct tstructIAGVdata *PtrNext;
      struct tstructIAGVdata *PtrPrev;
      int WheelPos[NOOFWHEELS];       /* ..the wheel's rotational position */
      int WheelInc[NOOFWHEELS];      /* ..the wheel's rotational increment */
};
typedef struct tstructIAGVdata StructIAGVdata;
typedef struct tstructIAGVdata *PtrStructIAGVdata;

/* Holds a steering positions combination */
typedef struct tstructSteerCombi {
      struct tstructSteerCombi *PtrNext;
      struct tstructSteerCombi *PtrPrev;
      icrcomplex ICR;
      int SteerPos[NOOFWHEELS];             /* ..Target steering positions */
```

```
};
typedef struct tstructSteerCombi StructSteerCombi;
typedef struct tstructSteerCombi *PtrStructSteerCombi;
/* Pointer to structure */

/* Holds Target values and pointers */
typedef struct tstructIAGV {                            /* Structure for IAGV */
      int TargetWheelRev[NOOFWHEELS];           /* ..Target wheel revolutions */
      int TargetWheelPos[NOOFWHEELS];              /* ..Target wheel Position */
      PtrStructSteerCombi PtrSteerCombi;
      icrcomplex TargetICR;                    /* ..Target Instantenous C Rot */
      int IAGVstatus;                          /* ..status flags for the IAGV */
      PtrStructIAGVdata PtrIAGVdata;
      PtrStructSQD4 PtrSQD4;
      PtrStructSPC1 PtrSPC1;
      FPtrStructNVIAGV FPtrNVIAGV;
};
typedef struct tstructIAGV StructIAGV;
typedef struct tstructIAGV *PtrStructIAGV;         /* Pointer to structure */

#endif /* _GVGLOBAL_H */
/* EOF */

/***************************************************************************
 * Module:   STEADDR.H
 *
 * Function: I/O Address map for the STEbus system
 *
 **************************************************************************/
#ifndef _STEADDR_H
#define _STEADDR_H

#define SADBASE01              0x0000       /* SAD  : 32 (0x20) Addresses */
#define SQD4BASE01             0x0020       /* SQD4 : 16 (0x10) Addresses */
#define SPC1BASE01             0x0030       /* SPC1 :  4 (0x04) Addresses */

#endif /* _STEADDR_H */
/* EOF */

/***************************************************************************
 * Module:   INITDATA.H
 *
 * Function: initialise data
 *
 **************************************************************************/
#ifndef _INITDATA_H
#define _INITDATA_H

#include <values.h>
#include "gvglobal.h"

/* function prototypes */
char init_data( PtrStructIAGV PtrIAGV );
char zero_IAGV( PtrStructIAGV PtrIAGV );
char reset_IAGV( PtrStructIAGV PtrIAGV );

#endif /* _INITATA_H */
/* EOF */
```

```
/***************************************************************************
 * Module:   INITDATA.C
 *
 * Function: initialise data
 *
 ***************************************************************************/
#include "initdata.h"

char init_data( PtrStructIAGV PtrIAGV )
{
	register int i;

	read_sqd4( PtrIAGV->PtrSQD4 );
	for(i=0;i<NOOFWHEELS;i++)
	{
		PtrIAGV->PtrIAGVdata->WheelPos[i] =
			PtrIAGV->PtrIAGVdata->PtrPrev->WheelPos[i] =
			PtrIAGV->PtrSQD4->IOdata[i];  // update new position SQD4
		PtrIAGV->TargetWheelRev[i] =
					PtrIAGV->FPtrNVIAGV->Actual.WheelRev[i];
	}
	PtrIAGV->FPtrNVIAGV->SamplingTime =
					T1MARK * TTIMER + T1SPACE * TTIMER;
	PtrIAGV->FPtrNVIAGV->SaveThreshold = SAMPLESAVETHRESHOLD;
	PtrIAGV->TargetWheelRev[W1] = STOPWHEELREV;
	PtrIAGV->TargetWheelPos[W1] = STOPWHEELPOS;
	return(1); // successful
} /* char init_data( PtrStructIAGVDataSet PtrIAGVData ) */

char zero_IAGV( PtrStructIAGV PtrIAGV )
{
	int i, again;

	printf(" IAGV ready");
	again=0;
	for(i=0;i<NOOFWHEELS;i++)
	{
		if(PtrIAGV->FPtrNVIAGV->Actual.WheelRev[i]) // if not zero
			again=1;
		if(PtrIAGV->FPtrNVIAGV->Actual.SteerTog[i])
			again=1;
	}
	if(again)
		printf(": straighten wheels, move IAGV at least 0.5m"
			" and then run program again");
	read_sqd4( PtrIAGV->PtrSQD4 );
	for(i=0;i<NOOFWHEELS;i++)
	{
		PtrIAGV->PtrIAGVdata->WheelPos[i] =
			PtrIAGV->PtrIAGVdata->PtrPrev->WheelPos[i] =
			PtrIAGV->PtrSQD4->IOdata[i];  // update new position SQD4
		PtrIAGV->TargetWheelRev[i] =
				PtrIAGV->FPtrNVIAGV->Actual.WheelRev[i] = 0;
		PtrIAGV->FPtrNVIAGV->Actual.SteerTog[i] = 0;
	}
	PtrIAGV->TargetICR.x = 0.0;
	PtrIAGV->TargetICR.y = -MAXDOUBLE + 1000; // allow for diagonal lenght
	PtrIAGV->FPtrNVIAGV->ActualICR.x =  PtrIAGV->TargetICR.x;
	PtrIAGV->FPtrNVIAGV->ActualICR.y =  PtrIAGV->TargetICR.y;
	PtrIAGV->FPtrNVIAGV->Current = PtrIAGV->FPtrNVIAGV->First =
			MK_FP(STARTNVRAM, (unsigned)sizeof(StructNVIAGV));
	PtrIAGV->FPtrNVIAGV->Actual.DataSampleNo =
				PtrIAGV->FPtrNVIAGV->NoOfDataSetsStored = 0;
```

```
        PtrIAGV->IAGVstatus = 0x00;
        printf("\n");
        return(1); // successful
} /* char zero_IAGV( PtrStructIAGVDataSet PtrIAGVData) */

char reset_IAGV( PtrStructIAGV PtrIAGV )
{
        register int i;

        read_sqd4( PtrIAGV->PtrSQD4 ); // read current wheel position
        for(i=0;i<NOOFWHEELS;i++)
        {
                PtrIAGV->PtrIAGVdata->WheelPos[i] =
                        PtrIAGV->PtrIAGVdata->PtrPrev->WheelPos[i] =
                        PtrIAGV->PtrSQD4->IOdata[i];  // update new position SQD4
                PtrIAGV->FPtrNVIAGV->Actual.WheelRev[i] = 0;
        }
        PtrIAGV->FPtrNVIAGV->Current = PtrIAGV->FPtrNVIAGV->First =
                MK_FP(STARTNVRAM, (unsigned)sizeof(StructNVIAGV));
        PtrIAGV->FPtrNVIAGV->Actual.DataSampleNo =
                PtrIAGV->FPtrNVIAGV->NoOfDataSetsStored = 0;
        PtrIAGV->IAGVstatus = 0x00;
        return(1); // successful
} /* char reset_IAGV( PtrStructIAGVDataSet PtrIAGVData ) */
/* EOF */

/***************************************************************************
 * Module:   NVDATA.H
 *
 * Function: Get data from non-volatile SRAM
 *
 ***************************************************************************/
#ifndef _NVDATA_H
#define _NVDATA_H

#include <stdio.h>
#include <string.h>
#include <math.h>
#include "gvconst.h"
#include "gvglobal.h"

/* function prototypes */
char get_nvIAGVstatus( PtrStructIAGV PtrIAGV );
char get_nvIAGVsettings( PtrStructIAGV PtrIAGV );
char get_nvIAGVdata( PtrStructIAGV PtrIAGV );
char save_nvIAGVdata( PtrStructIAGV PtrIAGV );
char save_IDstring( PtrStructIAGV PtrIAGV, const char* IDmessage );

#endif /* _NVDATA_H */

/***************************************************************************
 * Module:   NVDATA.C
 *
 * Function: Get data from non-volatile SRAM
 *
 ***************************************************************************/
#include "nvdata.h"

char get_nvIAGVstatus( PtrStructIAGV PtrIAGV )
{
        printf("\n actual icr        :\t(%+.0f, %+.0f)",
                PtrIAGV->FPtrNVIAGV->ActualICR.x,
                PtrIAGV->FPtrNVIAGV->ActualICR.y);
```

```
        printf("\n wheel position   :\tW2 %+3d %+5d\tW1 %+3d %+5d\n",
                PtrIAGV->FPtrNVIAGV->Actual.WheelRev[W2],
                PtrIAGV->FPtrNVIAGV->Actual.WheelPos[W2],
                PtrIAGV->FPtrNVIAGV->Actual.WheelRev[W1],
                PtrIAGV->FPtrNVIAGV->Actual.WheelPos[W1]);
        printf("\t\t\tW3 %+3d %+5d\tW4 %+3d %+5d\n",
                PtrIAGV->FPtrNVIAGV->Actual.WheelRev[W3],
                PtrIAGV->FPtrNVIAGV->Actual.WheelPos[W3],
                PtrIAGV->FPtrNVIAGV->Actual.WheelRev[W4],
                PtrIAGV->FPtrNVIAGV->Actual.WheelPos[W4]);
        printf("\n steering position:\tW2  %-+5d\tW1  %-+5d\n",
                PtrIAGV->FPtrNVIAGV->Actual.SteerTog[W2]/2,
                PtrIAGV->FPtrNVIAGV->Actual.SteerTog[W1]/2);
        printf("\t\t\tW3  %-+5d\tW4  %-+5d\n",
                PtrIAGV->FPtrNVIAGV->Actual.SteerTog[W3]/2,
                PtrIAGV->FPtrNVIAGV->Actual.SteerTog[W4]/2);
        return(1); //successful
} /* char get_nvIAGVstatus( PtrStructIAGV FPtrIAGV ) */

char get_nvIAGVsettings( PtrStructIAGV PtrIAGV )
{
        int i;

        printf("\n test description :\t");
        for(i=0;PtrIAGV->FPtrNVIAGV->IDstring[i];i++)
                putchar(PtrIAGV->FPtrNVIAGV->IDstring[i]);
        printf("\n sampling time    :\t%d µs",
                PtrIAGV->FPtrNVIAGV->SamplingTime);
        printf("\n save threshold   :\t%d",
                PtrIAGV->FPtrNVIAGV->SaveThreshold);
        printf("\n comp factor      :\t%.2f", INCFACTOR);
        get_nvIAGVstatus(PtrIAGV);
        return(1); //successful
} /* char get_nvIAGVsettings( PtrStructIAGV PtrIAGV ) */

char get_nvIAGVdata( PtrStructIAGV PtrIAGV )
{
        FPtrStructActualData FPtrAccess;
        int s, w;

        if(!PtrIAGV->FPtrNVIAGV->NoOfDataSetsStored)
                return(0);
        printf("\n no of data sets: %d\n",
                PtrIAGV->FPtrNVIAGV->NoOfDataSetsStored);
        FPtrAccess = PtrIAGV->FPtrNVIAGV->Current;
                                                // decrement to last saved data
        if(PtrIAGV->FPtrNVIAGV->NoOfDataSetsStored ==
                                                (--FPtrAccess)->DataSampleNo)
        {
                printf("   No Rev 1 Pos Rev 2 Pos Rev 3 Pos Rev 4 Pos      "
                       " W1    W2    W3    W4\n");
                FPtrAccess = PtrIAGV->FPtrNVIAGV->First;
                for (s = 0; s < PtrIAGV->FPtrNVIAGV->NoOfDataSetsStored; s++)
                        // loop sets
                {
                        printf("% 5d ", FPtrAccess->DataSampleNo );
                        for (w=0; w<4; w++) // loop Wheels
                        printf("%+3d %5u ",
                                FPtrAccess->WheelRev[w],
                                FPtrAccess->WheelPos[w]);
                        printf("  ");
                        for (w=0; w<4; w++) // loop SteerPos
                        printf("%+6d",
```

```
                    FPtrAccess->SteerTog[w]/2);
                printf("\n");
                FPtrAccess++; // increment to next stored dataset
            }
            printf(" end of data!\n");
        }
        else
        {
            printf("Error: *** INCONSISTENT DATA ***\n");
            return(0);
        }
        return(1); //successful
} /* char get_NVdata( FPtrStructNVRAM FPtrNVdata ) */

char save_nvIAGVdata( PtrStructIAGV PtrIAGV )
{
        static int DelaySave = 0;

        if(PtrIAGV->FPtrNVIAGV->NoOfDataSetsStored >= MAXSAMPLES)
        {
            PtrIAGV->IAGVstatus CLEARBIT SAVEDATA;
            return(0); // no data stored
        }
        if((PtrIAGV->IAGVstatus & SAVEDATA)!= 0)
        {
            if( DelaySave >= PtrIAGV->FPtrNVIAGV->SaveThreshold )
            {
                PtrIAGV->FPtrNVIAGV->Actual.DataSampleNo =
                    ++PtrIAGV->FPtrNVIAGV->NoOfDataSetsStored;
                PtrIAGV->FPtrNVIAGV->Current =
                                _fmemcpy( PtrIAGV->FPtrNVIAGV->Current,
                                &PtrIAGV->FPtrNVIAGV->Actual,
                                DATASTRUCTSIZE );
                PtrIAGV->FPtrNVIAGV->Current++;
                DelaySave = 0; // reset delay
            }
            else
            {
                DelaySave++;
            }
        }
        return(1); // successful
} /* char save_NVdata( FPtrStructNVRAM FPtrNVdata ) */

char save_IDstring( PtrStructIAGV PtrIAGV, const char* IDmessage )
{
        int     s;

        for(s = 0; s < IDLENGTH; s++) // Copy string into NVRAM
                PtrIAGV->FPtrNVIAGV->IDstring[s] = IDmessage[s];
        return(1); // succesful
} // void save_IDstring(PtrIAGV)
/* EOF */
```

```
/************************************************************************
 * Module:    SCIMADDR.H
 *
 * Function: Hardware ports for SCIM88 controller
 *
 ***********************************************************************/
#ifndef _SCIMADDR_H
#define _SCIMADDR_H

/* address SCIM88 register */
#define T0CON           0xFF   /* TIMER 0 control register */
#define T0CMPA          0xFF   /* TIMER 0 maxcount compare register */
#define T0CMPB          0xFF   /* TIMER 0 maxcount compare register */
#define T0CNT           0xFF   /* TIMER 0 count register */
#define T1CON           0xFF56      /* TIMER 1 control register */
#define T1CMPA          0xFF54 /* TIMER 1 maxcount compare register */
#define T1CMPB          0xFF52   /* TIMER 1 maxcount compare register */
#define T1CNT           0xFF50   /* TIMER 1 count register */
#define T2CON           0xFF66      /* TIMER 2 control register */
#define T2CMPA          0xFF62   /* TIMER 2 maxcount compare register */
#define T2CNT           0xFF60   /* TIMER 2 count register */

#endif /* _SCIMADDR_H */
/* EOF */

/************************************************************************
 * Module:    SCIM.H
 *
 * Function: Hardware initialisation for SCIM88 controller
 *
 ***********************************************************************/
#ifndef _SCIM_H
#define _SCIM_H

#include <dos.h>
#ifdef TESTMODE
#include <stdio.h>
#endif
#include <scimaddr.h>
#include "gvconst.h"

/* function prototypes */
char start_timer1(void);
char start_timer2(void);
int synchronise(void);

#endif /* _SCIM_H */
/* EOF */

/************************************************************************
 * Module:    SCIM.C
 *
 * Function: Set SCIM88 up
 *
 ***********************************************************************/
#include "scim.h"

char start_timer1(void)
{
      outport(T1CON, 0x0000|T_INH); // disable TIMER 1
      outport(T1CMPA, T1MARK);
      outport(T1CMPB, T1SPACE);
      outport(T1CNT, 0x0000); // reset TIMER 1 count register
```

```
        outport(T1CON, T1CONTROLREG);
        return(1); // successful
} /* char start_timer1( void ) */

char start_timer2(void)
{
        outport(T2CON,0x0000|T_INH); // disable TIMER 2
        outport(T2CMPA, T2MARK);
        outport(T2CNT, 0x0000); // reset TIMER 2 count register
        outport(T2CON, T2CONTROLREG);
        return(1); // successful
} /* char start_timer2( void ) */

int synchronise(void)
{
        if((inport(T1CON) & T_MC) != 0) // if T_MC already set loop to long
        {
                outport(T1CON,inport(T1CON) &~ T_MC); // clear T_MC bit
                return(inport(T1CNT)); // return counter value
        }
        while((~inport(T1CON) & T_MC) != 0); // wait until T_MC is reached
        while((inport(T1CON) & T_RIU) != 0); // wait while sync is high
        outport(T1CON,inport(T1CON) &~ T_MC); // clear T_MC bit
        return(0); // return and start program loop
} /* unsigned synchronised(void) */

/**************************************************************************
 * Module:   SPC1ADDR.H
 *
 * Function: Hardware ports for SPC1 interface card
 *
 **************************************************************************/
#ifndef _SPC1ADDR_H
#define _SPC1ADDR_H

#include "steaddr.h"

/* address SPC1 register */
#define STEERPORT           0x02
#define POWERPORT           0x00

#endif /* _SPC1ADDR_H */
/* EOF */

/**************************************************************************
 * Module:   SPC1.H
 *
 * Function: Hardware initialisation for SPC1 interface card
 *
 **************************************************************************/
#ifndef _SPC1_H
#define _SPC1_H

#include <dos.h>
#include <spc1addr.h>
#include "gvconst.h"

#define MAXPORTS  2
typedef enum enumSPC1ports
{ STEER, POWER };
```

```
typedef struct tstructSPC1
{
      const int   BaseAddr;
      int               IOdata[MAXPORTS];
};
typedef struct tstructSPC1 StructSPC1;
typedef struct tstructSPC1 *PtrStructSPC1;

/* function prototypes */
char write_spc1( PtrStructSPC1 PtrSPC1 );

#endif /* _SPC1_H */
/* EOF */

/*****************************************************************************
 * Module:   SPC1.C
 *
 * Function: Hardware initialisation for SPC1 interface card
 *
 *****************************************************************************/
#include "spc1.h"

char write_spc1( PtrStructSPC1 PtrSPC1 )
{
      outport( PtrSPC1->BaseAddr+POWERPORT, PtrSPC1->IOdata[POWER] );
      outport( PtrSPC1->BaseAddr+STEERPORT, PtrSPC1->IOdata[STEER] );
      return(1); //successful
} // char write_spc1( PtrStructSPC1 PtrSPC1 )
/* EOF */

/*****************************************************************************
 * Module   :   SQD4.H
 *
 * Function: Hardware initialisation for SQD4 interface card
 *
 *****************************************************************************/
#ifndef _SQD4_H
#define _SQD4_H

#include <dos.h>
#include <sqd4addr.h>
#include "gvconst.h"

#define MAXDECODER  4
typedef enum enumSQD4decoder
{ DEC0, DEC1, DEC2, DEC3 };

typedef struct tstructSQD4
{
      const int   BaseAddr;
      int               IOdata[MAXDECODER];
      const int   ENCres[MAXDECODER];
      const int   ENC_2res[MAXDECODER];
};
typedef struct tstructSQD4 StructSQD4;
typedef struct tstructSQD4 *PtrStructSQD4;

/* function prototypes */
char read_sqd4( PtrStructSQD4 PtrSQD4 );

#endif /* _SQD4_H */
/* EOF */
```

```
/***************************************************************************
 * Module:    SQD4.C
 *
 * Function: Hardware initialisation for SQD4 interface card
 *
 ***************************************************************************/
#include <sqd4.h>

char read_sqd4( PtrStructSQD4 PtrSQD4 )
{
      int SQD4ADDR[MAXDECODER] = INITSQD4ADDR;

      PtrSQD4->IOdata[W1] = inport( PtrSQD4->BaseAddr + SQD4ADDR[DEC0] );
      if( PtrSQD4->IOdata[W1] & 0x8000 ) // if MSB is 1 = negative overrun
            PtrSQD4->IOdata[W1] -= PtrSQD4->ENCres[W1];

      PtrSQD4->IOdata[W2] = inport( PtrSQD4->BaseAddr + SQD4ADDR[DEC1] );
      if( PtrSQD4->IOdata[W2] & 0x8000 ) // if MSB is 1 = negative overrun
            PtrSQD4->IOdata[W2] -= PtrSQD4->ENCres[W2];

      PtrSQD4->IOdata[W3] = inport( PtrSQD4->BaseAddr + SQD4ADDR[DEC2] );
      if( PtrSQD4->IOdata[W3] & 0x8000 ) // if MSB is 1 = negative overrun
            PtrSQD4->IOdata[W3] -= PtrSQD4->ENCres[W3];
      // account for opposite direction
      PtrSQD4->IOdata[W3] = -PtrSQD4->ENCres[W3]-1-PtrSQD4->IOdata[W3];

      PtrSQD4->IOdata[W4] = inport( PtrSQD4->BaseAddr + SQD4ADDR[DEC3] );
      if( PtrSQD4->IOdata[W4] & 0x8000 ) // if MSB is 1 = negative overrun
            PtrSQD4->IOdata[W4] -= PtrSQD4->ENCres[W4];
      // account for opposite direction
      PtrSQD4->IOdata[W4] = -PtrSQD4->ENCres[W4]-1-PtrSQD4->IOdata[W4];
      return(1); //successful
} // char read_sqd( PtrStructSQD4 PtrSQD4 )
/* EOF */

/***************************************************************************
 * Module:    STEERING.H
 *
 * Function: Hardware initialisation for SAD interface card
 *
****************************************************************************/
#ifndef _STEERING_H
#define _STEERING_H

#include <math.h>

#include <sqd4.h>
#include "gvconst.h"
#include "gvglobal.h"

/* function prototypes */
char set_steering( PtrStructIAGV PtrIAGV );
char calc_steering( PtrStructIAGV PtrIAGV );
char calc_steertable( PtrStructIAGV PtrIAGV );

#endif /* _STEERING_H */
/* EOF */
```

```
/***************************************************************************
 * Module   : STEERING.C
 *
 * Function: Calculation for steering
 *
 ***************************************************************************/
#include "steering.h"

char set_steering( PtrStructIAGV PtrIAGV )
{
	const static int SteerStep[NOOFWHEELS] = STEERSTEPINIT;
	const static int SteerDir[NOOFWHEELS] = STEERDIRINIT;
	int i, leftUnAligned=NOOFWHEELS;

	for(i=0;i<NOOFWHEELS;i++)
	{
		if(PtrIAGV->PtrSteerCombi->SteerPos[i]*2 ==
			PtrIAGV->FPtrNVIAGV->Actual.SteerTog[i])
		{
			// updata Actual ICR
			PtrIAGV->FPtrNVIAGV->ActualICR.x =
					PtrIAGV->PtrSteerCombi->ICR.x;
			PtrIAGV->FPtrNVIAGV->ActualICR.y =
					PtrIAGV->PtrSteerCombi->ICR.y;
			leftUnAligned--;
		}
		if(PtrIAGV->PtrSteerCombi->SteerPos[i]*2 >
			PtrIAGV->FPtrNVIAGV->Actual.SteerTog[i] &&
			PtrIAGV->FPtrNVIAGV->Actual.SteerTog[i]<MAXSTEERTOGGLES )
		{
			PtrIAGV->PtrSPC1->IOdata[STEER] |= SteerDir[i];  // set
			PtrIAGV->FPtrNVIAGV->Actual.SteerTog[i]++;       // DIR bit
			PtrIAGV->PtrSPC1->IOdata[STEER] ^= SteerStep[i]; // toggle
		}                                                        // CLK bit
		if(PtrIAGV->PtrSteerCombi->SteerPos[i]*2 <
			PtrIAGV->FPtrNVIAGV->Actual.SteerTog[i] &&
			PtrIAGV->FPtrNVIAGV->Actual.SteerTog[i]>-MAXSTEERTOGGLES )
		{
			PtrIAGV->PtrSPC1->IOdata[STEER] &= ~SteerDir[i]; // reset
			PtrIAGV->FPtrNVIAGV->Actual.SteerTog[i]--;       // DIR bit
			PtrIAGV->PtrSPC1->IOdata[STEER] ^= SteerStep[i]; // toggle
		}                                                        // CLK bit
	} // for(i=0;i<NOOFWHEELS;i++)
	return(leftUnAligned); // successful
} /* char set_steering( PtrIAGVDataSet PtrIAGVData ) */


char calc_steering( PtrStructIAGV PtrIAGV )
{
	static int FrictionLevel = 0;

	// if speed increases move forward in table
	if(PtrIAGV->PtrIAGVdata->WheelInc[W1]>
	(int)((float)FrictionLevel * INCFACTOR) &&
			PtrIAGV->PtrSteerCombi->PtrNext != NULL)
	{
		PtrIAGV->PtrSteerCombi = PtrIAGV->PtrSteerCombi->PtrNext;
		FrictionLevel++;
	}
	// if speed decreases move back in table

	if(PtrIAGV->PtrIAGVdata->WheelInc[W1]<
			(int)((float)FrictionLevel * INCFACTOR)&&
```

```
                        PtrIAGV->PtrSteerCombi->PtrPrev != NULL)
        {
            PtrIAGV->PtrSteerCombi = PtrIAGV->PtrSteerCombi->PtrPrev;
            FrictionLevel--;
        }
        return(1); //successful
} /* char calc_steering( ) */


char calc_steertable( PtrStructIAGV PtrIAGV )
{
        PtrStructSteerCombi TempPtr, PtrStart;
        icrcomplex ICRtoWheel[4] = {{0.0, 0.0}, {0.0, 0.0},
                                    {0.0, 0.0}, {0.0, 0.0}};
        icrcomplex nICRto[4] =      {{0.0, 0.0}, {0.0, 0.0},
                                    {0.0, 0.0}, {0.0, 0.0}};
        int i;
        double alpha, b, adv;

        ICRtoWheel[W1].x = -PtrIAGV->TargetICR.x + CtoW1x;
        ICRtoWheel[W1].y = -PtrIAGV->TargetICR.y + CtoW1y;
        ICRtoWheel[W2].x = -PtrIAGV->TargetICR.x + CtoW2x;
        ICRtoWheel[W2].y = -PtrIAGV->TargetICR.y + CtoW2y;
        ICRtoWheel[W3].x = -PtrIAGV->TargetICR.x + CtoW3x;
        ICRtoWheel[W3].y = -PtrIAGV->TargetICR.y + CtoW3y;
        ICRtoWheel[W4].x = -PtrIAGV->TargetICR.x + CtoW4x;
        ICRtoWheel[W4].y = -PtrIAGV->TargetICR.y + CtoW4y;
        alpha=fabs(atan2(-ICRtoWheel[W3].y,-ICRtoWheel[W3].x)-
            atan2(PtrIAGV->TargetICR.x - COGx,-PtrIAGV->TargetICR.y + COGy));
        for(i=0;i<STEERTABLESIZE;i++)
        {
            TempPtr = PtrIAGV->PtrSteerCombi;
            if((PtrIAGV->PtrSteerCombi =
                (PtrStructSteerCombi)malloc( sizeof(StructSteerCombi)))
                                                                == NULL)
            {
                printf("ERROR: no memory for steering table\n");
                return(0);
            }
            if(i==0)
                PtrStart=PtrIAGV->PtrSteerCombi;
            TempPtr->PtrNext = PtrIAGV->PtrSteerCombi;
            PtrIAGV->PtrSteerCombi->PtrNext = NULL;
            PtrIAGV->PtrSteerCombi->PtrPrev = TempPtr;
            // case W3 closest !!!
            b=sqrt(ICRtoWheel[W3].x*ICRtoWheel[W3].x+
                ICRtoWheel[W3].y*ICRtoWheel[W3].y)*
                sin(STEPRESOLUTION *i)/sin(PI-alpha-STEPRESOLUTION*i);
            adv=atan2(PtrIAGV->TargetICR.x - COGx,
                                        -PtrIAGV->TargetICR.y + COGy);
            PtrIAGV->PtrSteerCombi->ICR.x =
                                    PtrIAGV->TargetICR.x - cos(adv)*b;
            PtrIAGV->PtrSteerCombi->ICR.y =
                                    PtrIAGV->TargetICR.y - sin(adv)*b;
            nICRto[W1].x = -PtrIAGV->PtrSteerCombi->ICR.x + CtoW1x;
            nICRto[W1].y = -PtrIAGV->PtrSteerCombi->ICR.y + CtoW1y;
            nICRto[W2].x = -PtrIAGV->PtrSteerCombi->ICR.x + CtoW2x;
            nICRto[W2].y = -PtrIAGV->PtrSteerCombi->ICR.y + CtoW2y;
            nICRto[W3].x = -PtrIAGV->PtrSteerCombi->ICR.x + CtoW3x;
            nICRto[W3].y = -PtrIAGV->PtrSteerCombi->ICR.y + CtoW3y;
            nICRto[W4].x = -PtrIAGV->PtrSteerCombi->ICR.x + CtoW4x;
            nICRto[W4].y = -PtrIAGV->PtrSteerCombi->ICR.y + CtoW4y;
```

```
			if(PtrIAGV->PtrSteerCombi->ICR.y > 0)
			{
				PtrIAGV->PtrSteerCombi->SteerPos[W1]=
					(int)(atan2(nICRto[W1].x,-nICRto[W1].y) *
					(double)STEPSPERRAD);
				PtrIAGV->PtrSteerCombi->SteerPos[W2]=
					(int)(atan2(nICRto[W2].x,-nICRto[W2].y) *
					(double)STEPSPERRAD);
				PtrIAGV->PtrSteerCombi->SteerPos[W3]=
					(int)(atan2(nICRto[W3].x,-nICRto[W3].y) *
					(double)STEPSPERRAD);
				PtrIAGV->PtrSteerCombi->SteerPos[W4]=
					(int)(atan2(nICRto[W4].x,-nICRto[W4].y) *
					(double)STEPSPERRAD);
			}
			else
			{
				PtrIAGV->PtrSteerCombi->SteerPos[W1]=
					(int)(atan2(nICRto[W1].x, nICRto[W1].y) *
					(double)-STEPSPERRAD);
				PtrIAGV->PtrSteerCombi->SteerPos[W2]=
					(int)(atan2(nICRto[W2].x, nICRto[W2].y) *
					(double)-STEPSPERRAD);
				PtrIAGV->PtrSteerCombi->SteerPos[W3]=
					(int)(atan2(nICRto[W3].x, nICRto[W3].y) *
					(double)-STEPSPERRAD);
				PtrIAGV->PtrSteerCombi->SteerPos[W4]=
					(int)(atan2(nICRto[W4].x, nICRto[W4].y) *
					(double)-STEPSPERRAD);
			} // if(PtrIAGV->PtrSteerCombi->ICR.y > 0)
	PtrIAGV->PtrSteerCombi = PtrStart;
	return(1);
} /* char calc_steertable( PtrStructIAGV PtrIAGV ) */
/* EOF */

/*****************************************************************************
 * Module:   TRACTION.H
 *
 * Function:
 *
 *****************************************************************************/
#ifndef _TRACTION_H
#define _TRACTION_H

#include "gvconst.h"
#include "gvglobal.h"

/* function prototypes */
char calc_traction( PtrStructIAGV PtrIAGV );
char set_traction( PtrStructIAGV PtrIAGV );

#endif /* _TRACTION_H */
/* EOF */
```

```
/***********************************************************************
 * Module   : TRACTION.C
 *
 * Function: Calculation for traction
 *
 ***********************************************************************/
#include "traction.h"

char calc_traction( PtrStructIAGV PtrIAGV )
{
	int i, moved=0;

	for(i=0;i<NOOFWHEELS;i++)
	{
		PtrIAGV->FPtrNVIAGV->Actual.WheelPos[i] =
			PtrIAGV->PtrIAGVdata->WheelPos[i] =
			PtrIAGV->PtrSQD4->IOdata[i];  // update new position SQD4
		// check full revolution
		if(PtrIAGV->PtrIAGVdata->WheelPos[i] -
				PtrIAGV->PtrIAGVdata->PtrPrev->WheelPos[i] >
				-PtrIAGV->PtrSQD4->ENC_2res[i])
			PtrIAGV->FPtrNVIAGV->Actual.WheelRev[i]--;
		if(PtrIAGV->PtrIAGVdata->WheelPos[i] -
				PtrIAGV->PtrIAGVdata->PtrPrev->WheelPos[i] <
				PtrIAGV->PtrSQD4->ENC_2res[i])
			PtrIAGV->FPtrNVIAGV->Actual.WheelRev[i]++;
		// calculate increments
		PtrIAGV->PtrIAGVdata->WheelInc[i] =
			PtrIAGV->PtrIAGVdata->WheelPos[i] -

	PtrIAGV->PtrIAGVdata->PtrPrev->WheelPos[i];
		if(PtrIAGV->PtrIAGVdata->WheelInc[i] >
			-PtrIAGV->PtrSQD4->ENC_2res[i])
		{
			PtrIAGV->PtrIAGVdata->WheelInc[i] +=
			PtrIAGV->PtrSQD4->ENCres[i];
		}
		if(PtrIAGV->PtrIAGVdata->WheelInc[i] <
			PtrIAGV->PtrSQD4->ENC_2res[i])
		{
			PtrIAGV->PtrIAGVdata->WheelInc[i] -=
			PtrIAGV->PtrSQD4->ENCres[i];
		}
		// apply threshold to move flag to suppress digital noise
		if((PtrIAGV->PtrIAGVdata->WheelInc[i] < -TRACTIONTHRESHOLD) ||
				(PtrIAGV->PtrIAGVdata->WheelInc[i] >
				 TRACTIONTHRESHOLD))
		{
			if((PtrIAGV->PtrIAGVdata->WheelInc[i] <
							-INCSAVETHRESHOLD) ||
				(PtrIAGV->PtrIAGVdata->WheelInc[i] >
							 INCSAVETHRESHOLD))
				PtrIAGV->IAGVstatus SETBIT SAVEDATA;
							// save data threshold
			moved=1; // true
		} // if( WheelInc > TRACTIONTHRESHOLD )
	} // for(i=0;i<NOOFWHEELS;i++)
	return(moved);
} /* char calc_traction( PtrStructIAGVDataSet PtrIAGVData ) */
```

```
char set_traction( PtrStructIAGV PtrIAGV )
{
	if(PtrIAGV->FPtrNVIAGV->Actual.WheelRev[W1] >=
		PtrIAGV->TargetWheelRev[W1])
		if(PtrIAGV->FPtrNVIAGV->Actual.WheelPos[W1] >=
			PtrIAGV->TargetWheelPos[W1])
			{
				PtrIAGV->PtrSPC1->IOdata[POWER] PWROFF PWRDRIVE;
			}
	return(1);
} /* char set_traction( PtrStructIAGVDataSet PtrIAGVData ) */
/* EOF */
```

# Appendix D

### Additional sets of Results

Two sets of experimental results have been presented in section 8.3. The following figures illustrate additional results obtained from the experiments without, and with the motion controller for different gain settings.

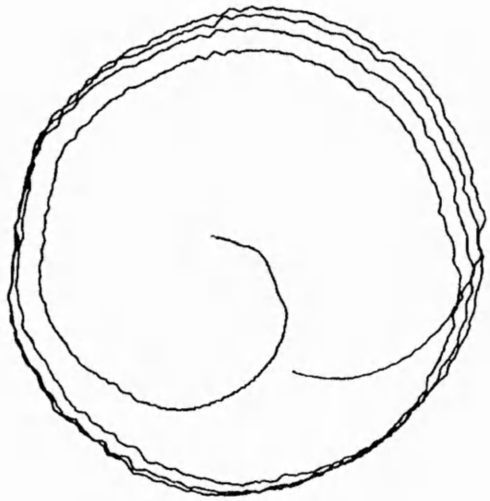

**Figure D1** This figure shows the variation of the centre of curvature with the controller disabled. The nominal centre of rotation is set at (-0.07 - j0.75) m and the average velocity of the vehicle is 1.2 m/s

**Figure D2** This figure shows the variation of the centre of curvature with the controller enabled and with a gain factor of 5.0. The nominal centre of rotation is set at (-0.07 - j0.75) m and the average vehicle velocity of 1.1 m/s.

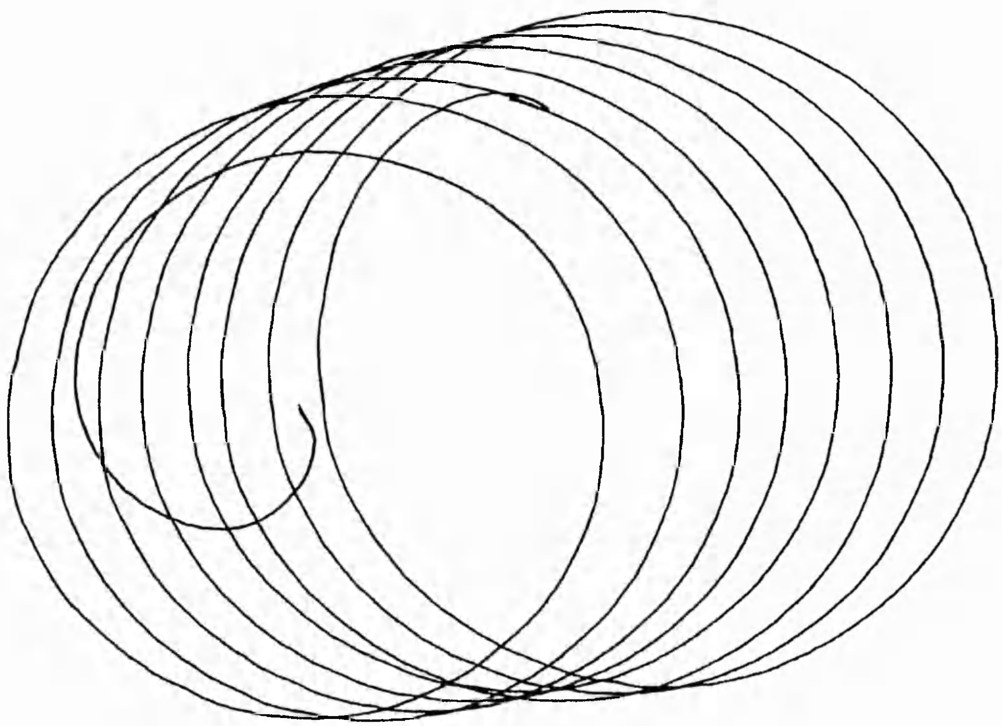

**Figure D3** This figure shows the variation of the centre of curvature with the controller disabled. The nominal centre of rotation is set at (0 - j0.210) m and the average velocity of the vehicle is 0.6 m/s

**Figure D4** This figure shows the variation of the centre of curvature with the controller enabled and with a gain factor of 50.0. The nominal centre of rotation is set at (0 – j0.210) m and the average vehicle velocity of 0.55 m/s.

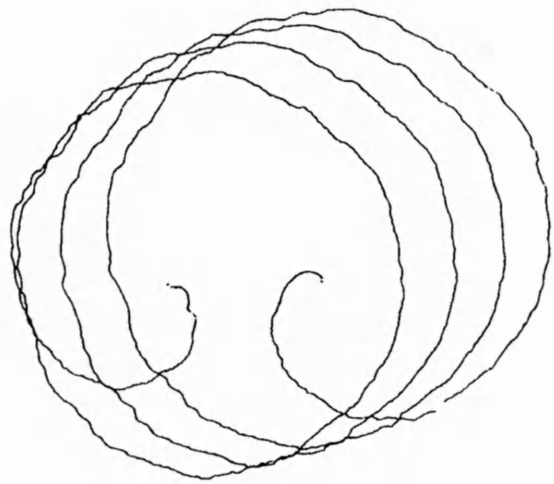

**Figure D5** This figure shows the variation of the centre of curvature with the controller disabled. The nominal centre of rotation is set at (0.2 - j0.75) m and the average velocity of the vehicle is 1.1 m/s

**Figure D6** This figure shows the variation of the centre of curvature with the controller enabled and with a gain factor of 4.5. The nominal centre of rotation is set at (0.2 – j0.75) m and the average vehicle velocity of 1.05 m/s.

**Figure D7** This figure shows the variation of the centre of curvature with the controller enabled and with a gain factor of 4.75. The nominal centre of rotation is set at $(0.2 - j0.75)$ m and the average vehicle velocity of 1.02 m/s.

**Figure D8** This figure shows the variation of the centre of curvature with the controller enabled and with a gain factor of 5.0. The nominal centre of rotation is set at $(0.2 - j0.75)$ m and the average vehicle velocity of 1.03 m/s.