Analysis and redesign of an antilock brake system controller

P.E. Wellstead
N.B.O.L. Pettit

Indexing terms: Antilock braking, Automotive control, Nonlinear control systems, Hybrid systems, Relay control systems

Abstract: Antilock braking (ABS) systems are a combination of hydromechanical braking components with switching solenoids and relays which operate the brakes in response to wheel and vehicle speed information. The control law which calculates the braking is a complex set of logical rules designed to allow for variations in road conditions and the nonlinear nature of tyre friction characteristics. The piecewise linear action of the actuator components (solenoids and relays) and the logical control rules make it difficult to analyse the dynamical behaviour of an ABS using conventional methods. The paper describes the analysis and subsequent redesign of an ABS using a new method for studying the dynamical behaviour of piecewise linear systems. The method is applied to a conventionally designed ABS controller. Undesirable performance features, not evident from conventional simulation and testing procedure, are found and the controller is redesigned to give a satisfactory dynamical performance.

1 Introduction

The paper takes a model of an antilock braking system and shows how it can be analysed and redesigned using a piecewise linear system analysis tool. Background on the analysis ideas can be found in [1–4].

1.1 Background: ABS – antilock braking systems

Antilock braking systems are now a commonly installed feature in road vehicles. They were first developed in the 1950s for aircraft, but were too expensive for road vehicles [5]. The first ABS systems for vehicles appeared in the late 1960s but were commercially unsuccessful, although the research associated with these products was important for a number of other commercial products [6].

Although ingenious mechanical systems have been developed [7], the real growth of ABS technology in the 1970s was made possible by integrated electronics and microcomputers [8], and as a result ABS controllers have moved inevitably to microcomputer based methods [9]. Indeed, despite current developments of electric braking [10], all current mass production ABS units comprise a hydraulic circuit, with additional solenoids and a control computer. The control algorithm consists of a set of logical rules (a rule base), stored in a microcomputer, which switches the brake on and off according to the sensed wheel speeds. Because of the range of operating conditions, the performance analysis of an ABS rule based controller is complex (a rule based controller typically has several hundred rules) and is normally achieved by repeated simulations and track trials [11]. However, the complexity of the rule base means that, even with the most extensive test programme, it is improbable that all the operating conditions are tested by conventional simulation and track tests. In this paper we show that an alternative exists. Specifically, by using a special piecewise linear analysis tool, it is possible to study the logic controller performance in a way which reveals characteristics which are not revealed by conventional validation procedures.

1.2 Background: piecewise linear systems

A piecewise linear system component is one in which the system characteristics change abruptly in response to the input or some other system variable. Many everyday control system components are piecewise linear – relays, saturating amplifiers, static friction and hysteresis are examples. In addition, many modern computer control system implementations are also piecewise linear. Specifically, programmable logic controllers (PLCs), rule based controllers and certain intelligent control system methods are based upon decision rules which change the controller characteristics abruptly. As a result the controllers act as piecewise linear systems. In the same spirit, scheduling of controller actions over a range of dynamical regions also results in a piecewise linear system.

When the piecewise linear actions in a control system are limited to a small number of events, say switching between one or two linear systems, then the dynamical behaviour can often be analysed using existing techniques [12–14]. Similarly, when there are no dynamical components in the system (or the dynamics are 'slow' compared to the piecewise linear behaviour) then the system performance can be analysed using static logic network analysis. However, many current control systems have extensive piecewise linear and dynamical elements which interact strongly, and as a result their dynamical behaviour is difficult to predict. Thus we

IEE Proceedings online no. 19971441

Paper first received 3rd June 1996 and in revised form 22nd May 1997

P.E. Wellstead is with the Control Systems Centre, UMIST, PO Box 88, Manchester M60 1QD, UK

N.B.O.L. Pettit is with the CF-T, Danfoss A/S, 6430 Nordborg, Denamrk

have the situation that a growing number of practical control systems exist where the performance cannot be completely predicted or analysed beyond simple cases. The system considered here (an ABS brake controller) is typical of this class of complex logic/dynamic systems. As implied in Section 1.1, it consists of a fast dynamical subsystem (comprising the brake hydraulics, wheel and tyre inertias) and a complex set of logical rules which must ensure satisfactory control.

2 Analysis of piecewise linear systems

This Section is an overview of the analysis of piecewise linear (PL) systems. The analysis aims to trace all the dynamical paths that are valid for a given system and capture these possible dynamic paths in a systematic way. For a general system, the method is complex and requires computer assistance.

2.1 The concept behind the analysis technique

The attractions of piecewise linear systems in control have been recognised for a long time [15, 16] since many nonlinearities such as saturation and relays are expressed easily as PL functions. The interest in revisiting this type of system model is that logic can also be incorporated into the PL framework. This means PL models can act as a unified framework for modelling dynamical systems with nonlinearities and logic. In its simplest form a PL system is described as a set of convex polytopes $\mathbf{P}_i \in \Re^n$ [17–19] each containing some linear system of the form:

$$\dot{x} = A_m x + b_m \qquad x \in P_m \tag{1}$$

and $\mathbf{P}_i$ form a partition of $\Re^n$ s.t.:

$$\cup \mathbf{P}_i = \Re^n, \quad \mathbf{P}_i \cap \mathbf{P}_j = \emptyset \quad i \neq j \tag{2}$$

The problem has been that the geometric interpretation leads to a complex picture of 'boxes' stacked together in state space with each box containing a different linear dynamic system. Any global analysis must somehow identify the dynamical behaviour in each box and then link them together to form a global picture of the dynamics. Fig. 1 illustrates this geometric interpretation.

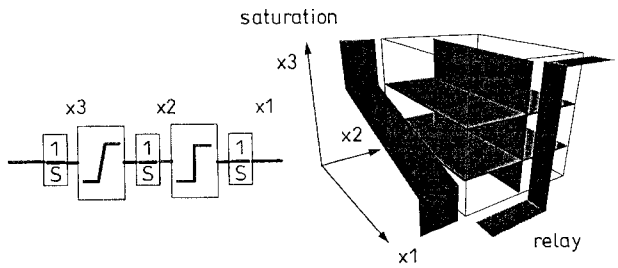

Fig. 1 *Geometric interpretation of a piecewise linear system*

In Fig. 1 the block diagram shows a three-state system with two PL functions; a saturation followed by a relay. In state space, the system will be in three dimensions. One axis will be split by two planes due to the two breakpoints that appear in the saturation. The other will be split by one plane due to the relay, since, although the relay has two breakpoints, they occur at the same instance in the input. As a result, the state space will comprise six linear regions. Fig. 1 shows how the PL functions of the system result in switching surfaces in state space. These surfaces act as boundaries of convex polytopes that bound each linear dynamic region. In general, a model can be constructed that ensures all regions will be convex [3].

The analysis approach developed is described in [1–4, 20]. The idea is to capture each region's dynamics in a simple connected graph. This is repeated for all regions, the regions are then pieced together and a global representation of the system is achieved. To see how this is done, consider one of the linear regions within the PL state space. The initial information describing the region will be the bounding hyperplanes of the convex polytope (γ, c) and the internal dynamics ($\dot{x} = Ax + b$). The adjacent regions will act to drive external dynamics towards the region boundary or away from it. Fig. 2 illustrates one region considered in isolation.

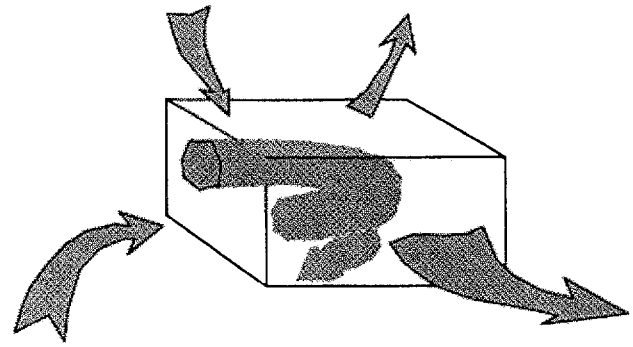

Fig. 2 *Dynamical behaviour of one region in a PL system*

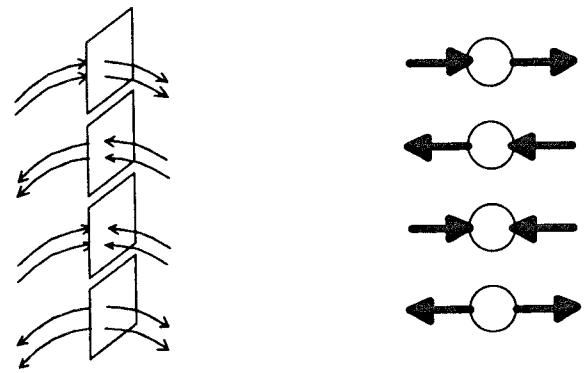

Fig. 3 *Partitioning of a boundary according to the dynamics either side*

The behaviour of the dynamics as they approach the boundary of a region can be categorised into four principal types: they can enter the region, exit the region, converge to a boundary or diverge from a boundary. These are illustrated in Fig. 3. Partitioning the boundaries of a region into these characteristic behaviours is a straightforward matter and results in convex partitions of the boundary. As illustrated in Fig. 3, these partitions can be represented as nodes or points, with directed arrows symbolising the characteristic dynamic patterns. Note that one behaviour shown, that of convergence to a boundary, typically identifies a sliding condition in the model.

Once the partitions are identified, the next step is to identify how the dynamics move within one region. The idea used is simple. Each partition where dynamics enter a region is projected through the region using that region's dynamics. In this way it is possible to find out which partitions the dynamics will exit from, given they enter a region by a particular partition. The result is to form a connected graph showing how partitions are connected via a regions dynamics.

The final step is to aggregate all the region graphs to form a global graph. The principle steps in the analysis procedure (illustrated in Fig. 4) are straightforward in concept. It is important to note, however, that the computer algorithms used to realise the analysis are far from straightforward [3].

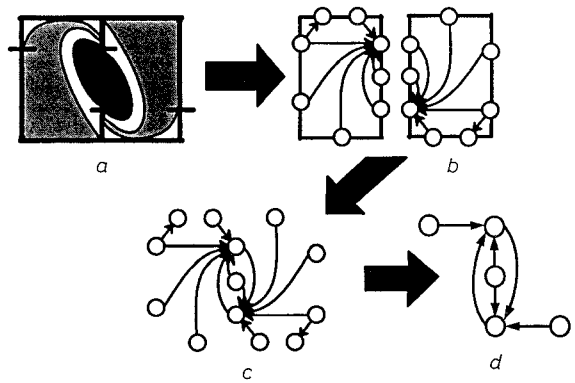

Fig. 4 *Illustration of the analysis concept*
(*a*) Initial system dynamics in state space with the piecewise linear boundaries superimposed
(*b*) Separate analysis of each region, identifying the dynamic connections between boundaries
(*c*) Merging the individual node graphs to form a global node graph of the system
(*d*) Applying graph simplification routines to analyse the final graph

The preceding discussion is intended to illustrate how the behaviour of a piecewise linear system can be represented in a linear graph. The advantage is that, regardless of the number of states of the model or the complexity of the piecewise linear elements, the final result is always a connected graph. This is then available to be processed using graph theory, a well-developed area with tools capable of handling very complex graphs. In some sense the nature of the phase portrait is retained. The system is still depicted by a visual representation of all possible dynamics in the system. Finally, the graph retains a close connection with the model. The nodes are directly related to the logic or PL functions in the model and the connections relate to the dynamics. This allows a clear physical interpretation of the graph and shows how logic and dynamics interact within the system.

For a more detailed description and examples of the piecewise linear tool, the reader is referred to [3, 4].

3 ABS principle

The aim of an antilock braking system is to improve the braking performance by maintaining the tyre braking torque at or near its maximum value. Nowadays commercial ABS controllers are almost all rule based and have as their control objective the limit cycling of tyre braking torque about its maximum value [21]. The basic idea can be explained with reference to Fig. 5 which shows a simplified form of a typical curve relating tyre torque and wheel spin. The braking torque, or tyre adhesion, is at its highest between wheel spin values marked A and B on the Figure. If the wheel spin increases beyond B, the wheel 'locks', tyre adhesion decreases and more importantly the driver loses the ability to steer the vehicle. The locked wheels will have a coefficient of friction for sliding that will be lower than the static coefficient of friction for the rolling wheels. So, for instance, in rear braking, unless braking is exactly in a straight line on a level road, the lateral component of forces on the vehicle will meet less resistance from the rear wheels causing the rear to swing round in an uncontrolled manner. The objective of an ABS controller is to keep the wheel spin between A and B, i.e. maximise braking torque without exciting those problems associated with locked wheels. A basic rule based controller that will do this is

```
if wheel spin > B then brake pressure = 0
if wheel spin < A then brake pressure = maximum
```

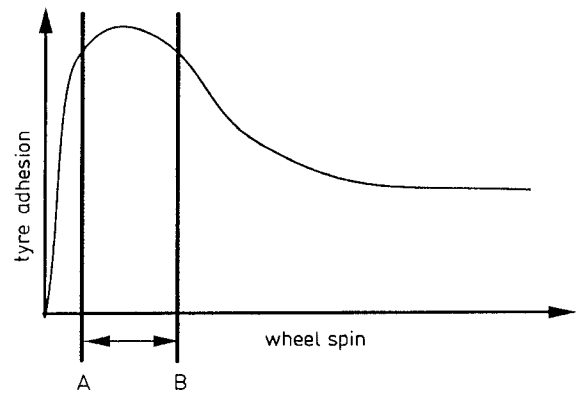

Fig. 5 *Tyre adhesion curve used in ABS*

In reality the tyre torque curve changes rapidly and by large amounts, depending on, among other things, the conditions of the road, weather, vehicle and tyres [22, 23]. Furthermore, sudden 'discrete event' changes will occur if the wheels move over different surface types. For instance an icy road will cause the curve to flatten and the distinct peak will disappear altogether. This means that selecting thresholds like A and B in Fig. 5 is not possible. A more sensible controller would be

```
if wheel spin increasing & brake force increasing
then brake pressure = max
if wheel spin increasing and brake force decreas-
ing then brake pressure = 0
```

Although this is still an incomplete rule set, it nevertheless embodies the key idea that the rules should track the position along the tyre torque curve and act accordingly. The main problem is that neither brake force nor wheel spin can be measured. Typically, braking force is deduced by measuring pressure of the brake hydraulics before the four ABS valves. This information, coupled with the valve history, allows a rough estimate of the braking force. One way of determining wheel spin is to measure the wheel speed of the four wheels and use logic to infer the wheel spin. To make the best of this poor measurement data, the measurements are coupled with system knowledge. For example, if a passenger car wheel registers a deceleration greater than that equivalent to gravity, then the wheel is almost certainly locked. This is because the mechanics of a passenger car show that its deceleration limit is typically in the region of 0.8g.

Because of the additional rules and information, a practical ABS controller will have a high degree of complexity involving perhaps hundreds of rules [24, 25]. In the following Section we explain how these rules might evolve in a simplified ABS model and further show how the controller behaviour can be analysed.

4 Simplified ABS model

The model is 'simplified' because it aims to capture a few core nonlinearities associated with an ABS system. The main nonlinearity being the tyre adhesion curve. It also sets out to control the brakes using a logic controller that contains basic features found in an ABS controller. This model is then analysed to show that a dynamic system, combining system nonlinearities with logic control can be analysed. In [1], a number of more traditional nonlinear analysis techniques were shown to be ineffective with this type of model, demonstrating the need for a different approach. This paper aims to validate one such new approach.

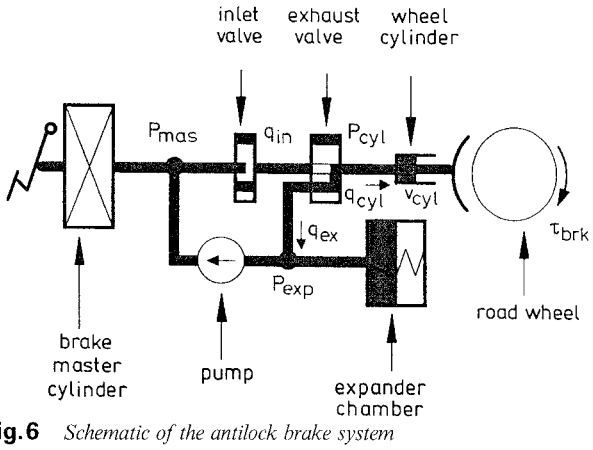

Fig. 6 *Schematic of the antilock brake system*

4.1 The brake subsystem

As shown in Fig. 6, a simple antilock brake can be modelled using three states. This Figure defines the main variables of the system and shows its principal operating regimes. There are three types of control action: (a) pump; (b) hold; and (c) dump. The principle of operation is that if the driver presses the brakes hard, then once the ABS control computer has identified the brakes are on hard and approaching their locked condition, the ABS system is activated. With reference to Fig. 6, the 'pump' action can be associated with the condition:

(1) pump activated, (2) inlet valve set open and (3) exhaust valve closed

The 'pump' action increases the braking torque applied to the wheel. The 'hold' action closes the inlet and exhaust valves, holding whatever pressure is applied to the wheel cylinder. The 'dump' action opens the exhaust valve, dumping fluid into the expander chamber and releasing pressure to the wheel cylinder. The 'pump' phase can be scheduled to allow different rates of pump action.

The schematic shown in Fig. 6 can be used to construct the model of an ABS. Specifically, Fig. 7 shows a block diagram which captures the essential dynamic and significant nonlinear features of an antilock brake, but at the same time reflects the features found in a working system. The model is built from two subsystems, one for the braking dynamics and one for the tyre and wheel dynamics. The key system nonlinearities in the model are: (a) the pressure–volume relationship ($v_{cyl} \rightarrow P_{cyl}$) in the wheel cylinder; (b) the road angular velocity effect on torque or road load effect ($\omega_r \rightarrow \tau_{rld}$) and (c) most importantly, the relationship between tyre torque and wheel spin ($\omega_s \rightarrow \tau_{tyr}$). The control 'input' is shown as a switching function that can select either 'pump', 'hold' or 'dump' modes.

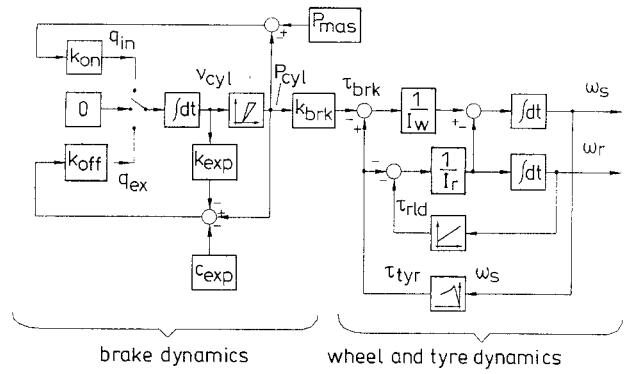

Fig. 7 *Block diagram of the antilock brake*

The equations of the system can be decomposed [26] to give the following mathematical model of the systems behaviour.

Assuming linear hydraulic restrictance, the inlet and exhaust valve flows are, respectively:

$$q_{in} = \frac{P_{mas} - P_{cyl}}{R_{in}}, \quad q_{ex} = \frac{P_{cyl} - P_{exp}}{R_{ex}} \tag{3}$$

The valve restrictances are

$$k_{on} = \frac{1}{R_{in}} \qquad k_{off} = -\frac{1}{R_{ex}} \tag{4}$$

Wheel cylinder volume is given by

$$v_{cyl} = \int (q_{in} - q_{ex})dt \tag{5}$$

Assuming a linear relationship between brake pressure and brake torque :

$$\tau_{brk} = k_{brk} P_{cyl} \tag{6}$$

Expander chamber pressure is given by

$$P_{exp} = c_{exp} - k_{exp} v_{cyl} \tag{7}$$

The wheel cylinder pressure is given by

$$P_{cyl} = \begin{cases} k_{pv}(v_{cyl} - v_{ded}) & \text{when } v_{cyl} < v_{ded} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

The wheel and vehicle inertias give the following expressions. The wheel acceleration is given by

$$\dot{\omega}_w = \frac{\tau_{tyr} - \tau_{brk}}{I_w} \tag{9}$$

where I_w is the road wheel inertia.

The vehicle acceleration is given by

$$\dot{\omega}_r = \frac{-(\tau_{tyr} - \tau_{rld})}{I_r} \tag{10}$$

where I_r is the 'inertia' of the vehicle as seen by the road wheel.

Finally, the wheel spin speed is given by

$$\omega_s = \omega_w - \omega_r \tag{11}$$

where τ_{brk}, the brake torque, is obtained from the brake model and the tyre and road load torques, τ_{tyr} and τ_{rld}, are given by the piecewise linear functions discussed below. It will be convenient to rearrange the equations to give spin speed as a state because spin speed is simply a linear combination of the wheel and road speeds. Rearranging the relevant equations gives:

$$\begin{aligned} \dot{\omega}_s &= \dot{\omega}_w - \dot{\omega}_r \\ &= \frac{\tau_{tyr} - \tau_{brk}}{I_w} + \frac{\tau_{tyr} + \tau_{rld}}{I_r} \\ &= \frac{\tau_{tyr}(I_w + I_r)}{I_r I_w} - \frac{\tau_{brk}}{I_w} + \frac{\tau_{rld}}{I_r} \end{aligned} \tag{12}$$

A general autonomous state model of the system dynamics can be deduced from eqns. 3–12:

$$\begin{bmatrix} \dot{\omega}_s \\ \dot{v}_{cyl} \\ \dot{\omega}_r \end{bmatrix} = \begin{bmatrix} c_{tyr}\frac{I_w+I_r}{I_w I_r} & -not_{ded}\frac{k_{pv}k_{brk}}{I_w} & \frac{c_{rd}}{I_r} \\ 0 & -not_{ded}k_{pv}(k_{on}+k_{off})-c_{exp}k_{off} & 0 \\ \frac{-c_{tyr}}{I_r} & 0 & \frac{-c_{rd}}{I_r} \end{bmatrix} \cdot \begin{bmatrix} \omega_s \\ v_{cyl} \\ \omega_r \end{bmatrix} + \begin{bmatrix} k_{tyr}\frac{I_w+I_r}{I_w I_r}+v_{ded}\frac{k_{pv}k_{brk}}{I_w} \\ not_{ded}k_{pv}v_{ded}(k_{on}+k_{off})+k_{exp}k_{off}+P_{mas}k_{on} \\ \frac{-(k_{tyr}+k_{rd})}{I_r} \end{bmatrix} \quad (13)$$

where the parameters c_{tyr} and k_{tyr} depend upon the piecewise linear segment of the tyre torque characteristic; c_{rd} and k_{rd} depend upon the road load characteristic; k_{on} and k_{off} are the inverse restrictances of the ABS modulator valves and not_{ded} is a Boolean value that indicates when the brake wheel cylinder is not in its dead zone, i.e.

$$not_{ded} = 0 \quad \text{for all } v_{cyl} < v_{ded}$$
$$not_{ded} = 1 \quad \text{for all } v_{cyl} \geq v_{ded} \quad (14)$$

It is important to reiterate that this model has been developed to capture basic ABS features. As a result, dynamics are highly simplified and issues such as how the road angular velocity is measured in a real vehicle are not addressed. As the main aim is to demonstrate the analysis of mixed dynamic/logic systems, it was felt the model omissions were acceptable.

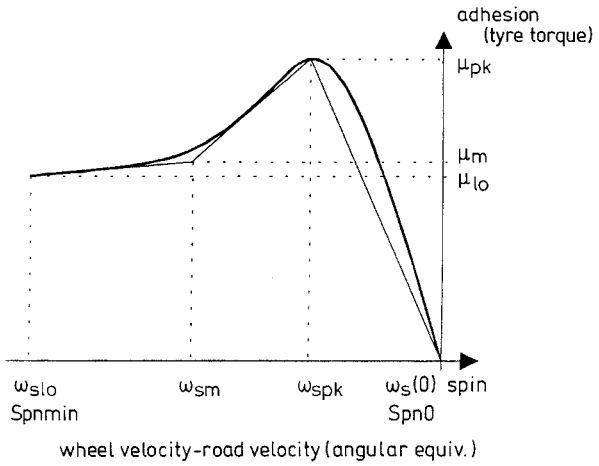

Fig. 8 *Characteristic tyre adhesion curve*

4.2 Tyre modelling

A piecewise linear approximation to the tyre torque versus spin speed is shown in Fig. 8. This curve is usually given as a continuous nonlinear curve that varies for differing road conditions. The true tyre curve has a complex shape and changes with road and vehicle conditions. For a review of recent developments in tyre modelling see [27, 28]. The curves used in this analysis were derived by fitting a double exponential function to data supplied by an industrial collaborator. In order to handle the nonlinearity in the analysis, the curve is approximated by a PL function which is designed to capture the peak characteristic and the tail slope of the curve. There are four key wheel spin references that are used in the controller design:

Spn0 ($\omega_s(0)$) – the zero spin or no braking reference

ω_{spk} – the spin at the peak tyre torque point

ω_{sm} – the spin at the 'knee' of the curve

Spnmin (ω_{slo}) – the lower bound on the spin.

If the spin reaches Spnmin, then normal ABS operation is considered to have failed.

Since the tyre curve varies for differing road conditions, four curves were used and four graphs generated through the analysis. The curves used are shown in Fig. 9 and are labelled as $\mu = 1$ (high adhesion) through to $\mu = 0.2$ (icy conditions), $\mu = 0.5$ is considered to be the nominal road adhesion condition for tarmac. This article will present just two results, those for $\mu = 0.2$ and $\mu = 0.5$, since these will be sufficient to demonstrate the advantages of the analysis method. A more detailed explanation of the modelling process and complete analysis is given elsewhere [1].

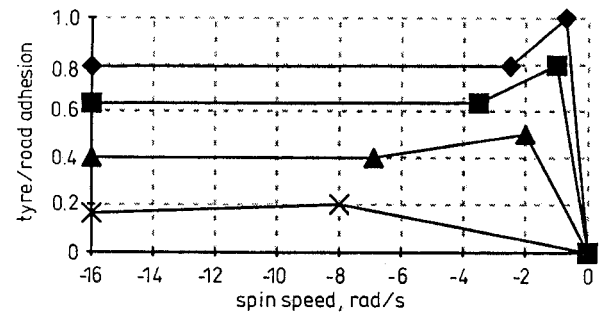

Fig. 9 *Tyre adhesion curves used in the PL analysis*
—◆— $\mu = 1$
—■— $\mu = 0.8$
—▲— $\mu = 0.5$
—×— $\mu = 0.2$

5 Initial controller design

The controller aims to generate a limit cycle in the dynamics that move the wheel spin back and forth around the maximum of the tyre torque curve. We will use the term 'stable region' to indicate the nonsliding condition (wheel spin between 0 and ω_{spk}). The term 'unstable region' will indicate the locked wheel condition. The control principle is to set three threshold curves that approximate the true tyre characteristic, but to also allow for uncertainty in the tyre characteristic. These thresholds represent values that are the difference between the tyre and brake torques. One threshold curve (acc1) sits 'below' the true curve, the other two (acc2 and acc3) are close together and sit 'above' the tyre torque curve. The control aim is to drive the spin acceleration variable $\dot{\omega}_s$ using the 'pump' control action until the acc2 threshold is reached (Phase_B 'on'). The 'hold' control action is then used to maintain the torque such that it tracks acc2 (Phase_B 'hold'). This is depicted as the two parts of the 'Phase_B' curve in Fig. 10. Once the ω_{spk} value is reached, the tyre adhesion will decrease. Since the braking torque is maintained using the 'hold' action, the acceleration measurement will maintain its value as spin increases negatively and the acc2 threshold will drop below it. The acceleration curve will hold steady until it hits the acc3 threshold, this confirms that the ω_{spk} value has been reached and triggers the 'dump' action. This switches off the brake and unlocks the wheel as shown in 'Phase_C' of Fig. 10. When sufficient decrease in braking torque has occurred, acc1 is crossed and the brake pressure set to 'hold' (indicated by 'Phase_A' in

Fig. 10). Once acc1 is re-crossed, this indicates that the wheel spin is again in the stable region of the tyre adhesion curve, and 'Phase_B' can be repeated.

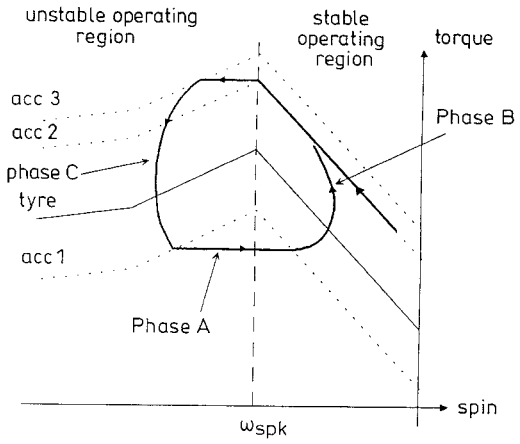

Fig. 10 *Illustration of the controller action*

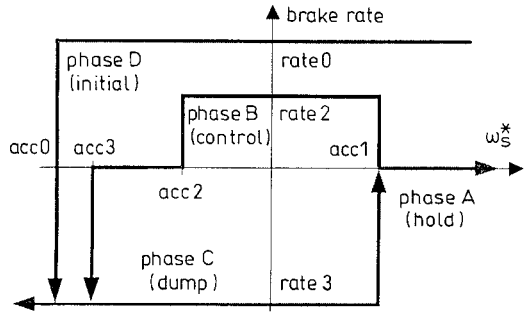

Fig. 11 *Controller loop*

Further scheduling is required in setting the rate at which the pressure is applied and removed from the braking cylinder. This determines how fast 'Phase_B' and 'Phase_C' operate. The rule logic can be shown as a plot of spin acceleration against the 'restrictance' applied to the valves to regulate flow to and from the brakes. This is shown in Fig. 11. An initialisation phase 'Phase_D' is also shown. This will be activated just once in the control cycle when the driver activates the ABS through hard braking.

The controller rule set used to achieve the desired limit cycle is given below:

```
If dwspin > acc1 AND phase ~= Phase_D
  THEN phase -> Phase_A
If dwspin < acc1 AND phase == Phase_A
  THEN phase -> Phase_B
If dwspin < acc3 AND phase ~= Phase_D
  THEN phase -> Phase_C
```

(15)

where dwspin is the spin acceleration ($d\omega_s/dt$) and Phase_D is the initialisation state (see below). The control is then determined by the state of phase:

```
If phase == phase_A
  THEN rate -> 0
If phase == phase_B AND dwspin > acc2
  THEN rate -> rate2
If phase == phase_B AND dwspin < acc2
  THEN rate -> 0
If phase == phase_C
  THEN rate -> rate1
```

(16)

Three additional rules are also used. The first is an absolute bound on the spin velocity (NB wspin = ω_s) and aims to prevent it from becoming excessively large by switching to the turn-off phase (Phase_C).

```
IF wspin < spnmin, THEN phase -> phase_C
```

(17)

where spnmin is a large negative value. The other additional rules are used for initialisation. In this context it is useful to note that a large number of rules are often required just to initiate the ABS controller limit cycle. In this case two rules are used:

```
IF wspin > Spn0 THEN phase -> phase_D
IF wspin < Spn0 AND dwspin < acc0
  AND phase == phase D,
  THEN phase -> phase_C
```

(18)

Phase_D is the initial phase in which the brakes are turned on at the maximum rate – it is the 'ABS off' condition with rate0 corresponding to the rate with the inlet valve fully open. Hence:

```
IF phase == phase_D THEN rate -> rate0
```

(19)

Rule set 18 provides both entry and exit conditions to ABS operation. The condition on spin is that it must remain at a small value greater than Spn0, which is a small negative limit. The condition on acceleration is that it must be less than acc0, a large negative value. When the deceleration is large and the spin greater than Spn0, the wheel is judged to have broken away and the ABS control is brought into action by switching to the recovery phase (Phase_C). The combined rules and dynamics create a state event table summarising all the system state transitions (Table 1).

Table 1: State event table for the ABS controller

State event	phase_A	phase_B (on)	phase_B (hold)	phase_C	phase_D
dwspin < acc0 & wspin < spn0					phase_C
dwspin < acc1	phase_B (on)				
dwspin > acc1		phase_A		phase_A	
dwspin < acc2		phase_B (hold)			
dwspin > acc2			phase_B (on)		
dwspin < acc3			phase_C		
wspin < spnmin	phase_C	phase_C	phase_C		phase_C
wspin > spn0	phase_D	phase_D	phase_D	phase_D	

The controller was implemented in ACSL as this has an effective structure for dealing with logic. The system dynamics, as given in eqn. 13, are embedded in a derivative block of ACSL. The rules are implemented by assigning ACSL schedules to ω_s and $\dot{\omega}_s$, i.e. wspin and dwspin, respectively. These variables are scheduled in the derivative block against the tuneable variables acc0 to acc3 for dwspin and Spn0, spnmin for wspin. Flags are used to track the history of the control phases activated. A lookup table in the discrete block then allows the braking rate to be selected from the tuneable variables rate0 to rate2. The lookup logic is given by Table 1.

Once the controller rule set is fixed, then the remaining parameters of the controller that require tuning are the scheduling variables to trigger the rules. These are the four accelerations, `acc0` to `acc3` and the two spin limits, `Spn0` and `spnmin`. There are then the three rate variables controlling the braking rate, `rate0` to `rate2`. This reflects a large range of choice in controller variables and illustrates the formidable controller tuning problems which the designer of rule based controllers faces. In practice the designer has little alternative but to conduct hundreds of simulation runs, testing combinations of parameter values in an attempt to select a parameter set containing the required performance and robustness conditions. The results of a design obtained in this 'trial and error' manner are illustrated by two typical simulations given in Figs. 12–17. The controller phase plot of Fig. 13*b* can be related to Fig. 11. The value 4 in Fig. 13*b* indicates `Phase_D` in Fig. 11, 3 indicates `Phase_C`, 2 indicates `Phase_B` and 1 indicates `Phase_A`. Figs. 12, 13 and 14 show responses for a road surface of $\mu = 0.2$ and Figs. 15, 16 and 17 responses for $\mu = 0.5$.

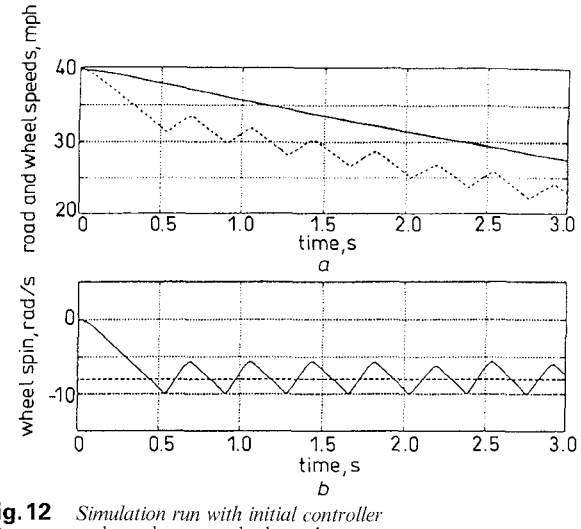

Fig. 12 *Simulation run with initial controller*
(*a*) ——— road speed – – – – wheel speed
(*b*) Wheel spin
$\mu = 0.2$

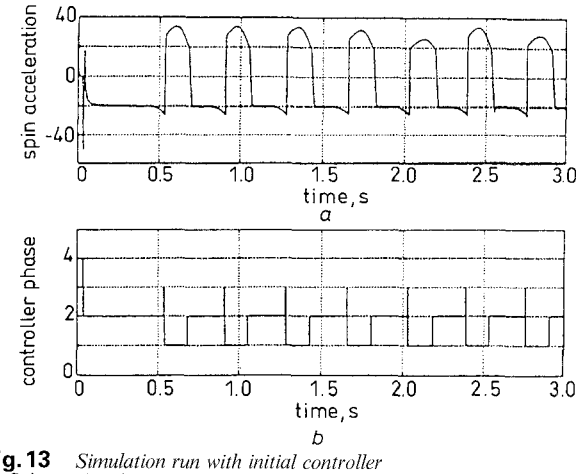

Fig. 13 *Simulation run with initial controller*
(*a*) Spin acceleration
(*b*) Controller phase
$\mu = 0.2$

Figs. 12–17 illustrate some of the problems in designing complex systems using simulation. On first inspection, the figures look very promising. The key plots for tyre and brake torque show the ABS system quickly

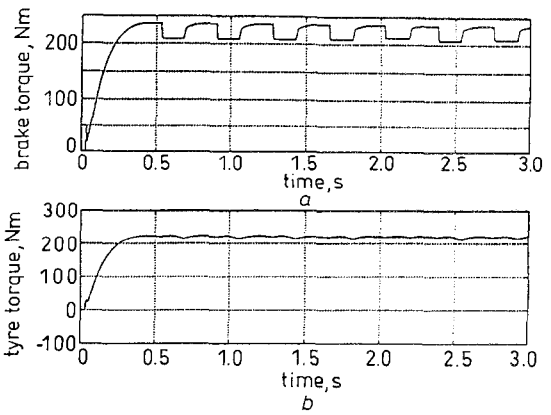

Fig. 14 *Simulation run with initial controller*
(*a*) Brake torque
(*b*) Tyre torque
$\mu = 0.2$

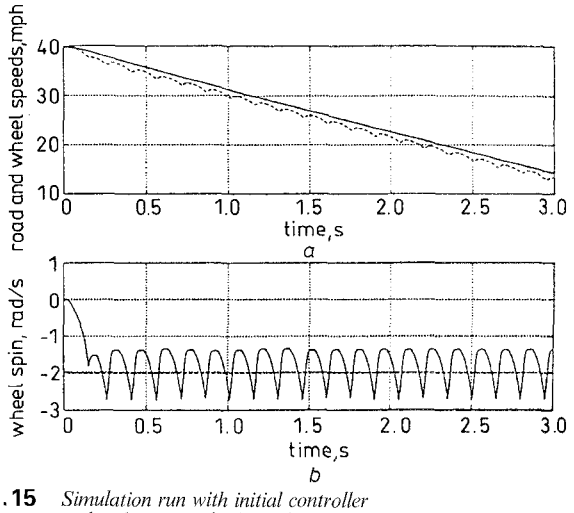

Fig. 15 *Simulation run with initial controller*
(*a*) ——— road and – – – – wheel speed
(*b*) Wheel spin
$\mu = 0.5$

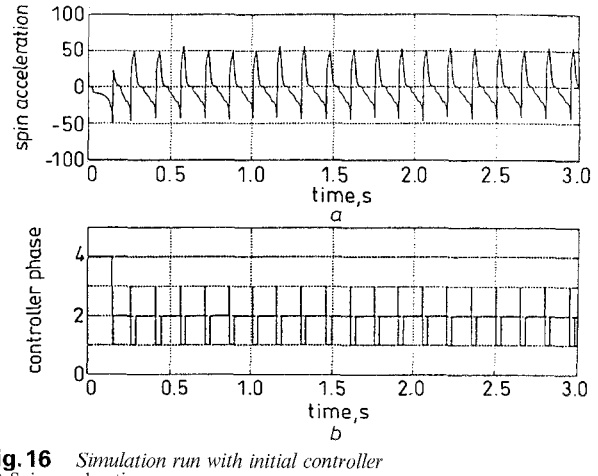

Fig. 16 *Simulation run with initial controller*
(*a*) Spin acceleration
(*b*) Controller phase
$\mu = 0.5$

bringing the braking torque to a maximum and holding it there for the duration of the ABS operation. However, these plots provide no information as to the functioning of the controller with the system. To attempt to give this information, simulations of the controller phase, spin acceleration, wheel spin and road and wheel speeds are given. The controller appears to cycle around its phases in the manner predicted through the design. The controller action together with the plots of

tyre and brake torques behave in a similar manner for $\mu = 0.2$ and $\mu = 0.5$. This is also a good indication that the design works. The plots of wheel spin and spin acceleration show the effect of the limit cycle behaviour and seem to be within acceptable limits of variation. If there is any slight concern, it may be that the wheel spin cycling for $\mu = 0.5$ is a little excessive, but it is still acceptable. As can be seen, even with detailed simulation, the designer can extract little more than qualitative information from the simulations. The ability of the designer to interpret this information using past experience then becomes critical in order to achieve good design. We will return to the plots after the analysis and design cycle presented in the paper. This will allow us to examine the simulation using the insights provided by the analysis.

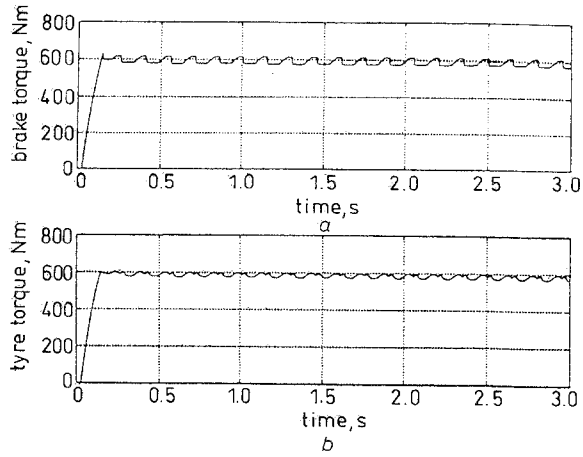

Fig. 17 *Simulation run with initial controller*
(*a*) Brake torque
(*b*) Tyre torque
$\mu = 0.5$

6 Analysis of the initial controller

As mentioned in the previous Section, only two road conditions are considered here. In many ways the road adhesion $\mu = 0.5$ represents a normal road so this will be examined first. The full connection graph for the system is not shown here. This is because, as an example, the $\mu = 0.5$ road generates 44 regions with something like 215 nodes. It is meaningless to display such a large graph, since no useful information can be obtained by simple visual inspection. Indeed, automatic information extraction methods are required, see [20]. The key limit cycles that are related to the control action can be extracted from the full connection graph to produce a reduced graph. In all the graph plots, a grid is drawn of wheel spin along the x-axis and a combination of torque acceleration and valve position along the y-axis. This allows the graph to be related back to the control cycle shown in Figs. 10 and 11 because the rules depend only upon wheel spin and rate of change of wheel spin. Fig. 18 shows the key paths found for $\mu = 0.5$.

In Fig. 18 the principal control cycle has Phase_B moving from B3 through B8 to B9. At this point, acc3 is reached and Phase_C ('dump') is triggered. This drives the trajectories down to C6 where acc1 is crossed and Phase_A begins. This holds the trajectories through A6 until acc1 is recrossed and Phase_B restarts. This cycle is as expected from the control design and simulation. The other cycles in the graph also make sense, since it is expected that for certain initial conditions, the dynamics cross the ω_{sm} threshold and as a worst case, cross the safety rule triggered by the dynamics reaching spnmin. All these conditions are shown to return to the main control loop within the graph.

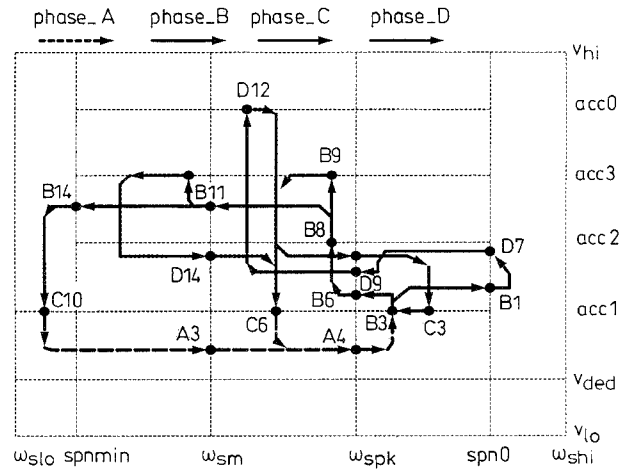

Fig. 18 *Key dynamic paths for $\mu = 0.5$*

In the main, the graph confirms the behaviour which was planned during the design/simulation iteration. However, the graph shows up one unexpected pattern. During the design process the control performance criterion was for Phase_B to drive the dynamics to the acc2 threshold and hold them at this level until ω_{spk} is crossed. The wheel spin should then increase until acc3 is reached and the 'dump' action initiated. The graph tells a slightly different story. Phase_B only manages to achieve an acceleration torque between acc1 and acc2 before ω_{spk} is reached (see B3 → B6 in the graph). The acc2 level, crossed at node B8, is reached after ω_{spk} is exceeded. The dynamics then increase until acc3 is reached. This implies that the control action spends much longer in the unstable part of the torque curve than was anticipated in the design, since the acceleration torque must cross both acc2 and acc3 in the unstable part of the curve.

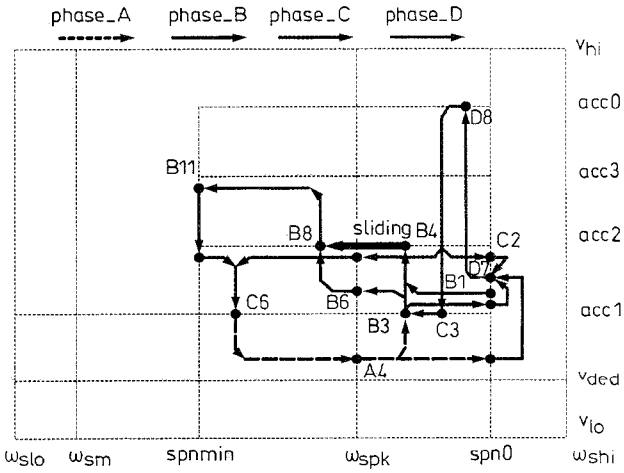

Fig. 19 *Key dynamic paths for $\mu = 0.2$*

Fig. 19 shows the key cycles when $\mu = 0.2$. This pattern shows a number of differences from the expected control cycle. The first is that conditions for a sliding mode exist at node B4 (in the full graph B4 is a node with all connections converging to it as illustrated in

Fig. 3). Specifically an unplanned and unpredicted sliding mode from B4 to B8 exists. The problem of acc2 not being crossed before ω_{spk} is reached is again shown in the path B3 → B6 → B8. However, a potential path B3 → B4 → B8 exists where acc2 is reached before ω_{spk}. More disturbing is the fact that the only method of reaching Phase_C or the 'dump' action is for ω_s to increase until Spnmin is reached. As mentioned previously, this is an emergency rule and for ω_s to reach Spnmin means a long time is spent in the unstable region of the tyre adhesion curve.

The analysis shows a clear failure in terms of acceptable control action in the ABS. Moreover, it is one which was neither predicted by *ad hoc* design nor detected in simulation. However, given these analysis results, we can revisit the simulations in Figs. 12–17. The problem of hitting the spnmin rule for $\mu = 0.2$ is one that might have been picked up by looking at the wheel spin plot (Fig. 12*b*). Here the wheel spin is seen to touch the spnmin limit of –10. However, as the simulation is of a low friction surface, the peak of the tyre/torque curve is close to this limit anyway (Fig. 9). Indeed, for many initial conditions, the designer would expect the spnmin rule to fire, and so would not be overly concerned to see the limit reached. The other problems are almost impossible to infer from simulation. There is no evidence of the sliding behaviour as only certain initial conditions excite this behaviour. The designer must hope one of the simulations catches the behaviour and this did not occur here. Given the knowledge that the controller spends too much time in the unstable region of the curve, it is possible to look at the controller phase plots of Fig. 13*b* and say that the time spent in level '1' is too long. However, this is not something easily inferred *a priori*.

The node graphs highlight a number of problems in the control design. In summary, these are:

(a) The acceleration control shown in Phase_B needs to be improved so that the design threshold acc2 is reached before ω_{spk} is crossed. This relates to increasing the time the cycle spends in the stable side of the adhesion curve.

(b) The sliding behaviour for $\mu = 0.2$ should be removed.

(c) The wheel spin needs to avoid the Spnmin rule, particularly for $\mu = 0.2$.

7 Redesign of the controller

The first problem to be addressed is the sliding mode. Examination of the acc2 rule suggests that the sliding behaviour occurs whenever Phase_B results in the acceleration torque reaching acc2 before ω_{spk} is crossed. The alternative is to prevent acc2 being reached and increase the time spent in the unstable part of the adhesion curve. This leads to the conclusion that a *designed* sliding mode is to be preferred, since it should be relatively robust to different values of μ. This effect is achieved by increasing the rate2 value for turning the brakes on ('pump' action) (this is to guarantee that the acceleration reaches acc2), including hysteresis and dividing acc2 into two thresholds (acc20 and acc21). This avoids chattering on the sliding mode. An unwanted consequence of increasing the set rate rate2 for Phase_B is a reduction in time spent in the stable part of the curve. This can be resolved by providing two thresholds to replace acc1. Thus the controller switches from Phase_C to Phase_A at acc10 and from Phase_A to Phase_B at acc11. Thus acc11 can be scheduled to ensure Phase_B starts well within the stable portion of the curve. These changes are illustrated in Fig. 20*a*. The modified illustration of the rule operation is shown in Fig. 20*b*.

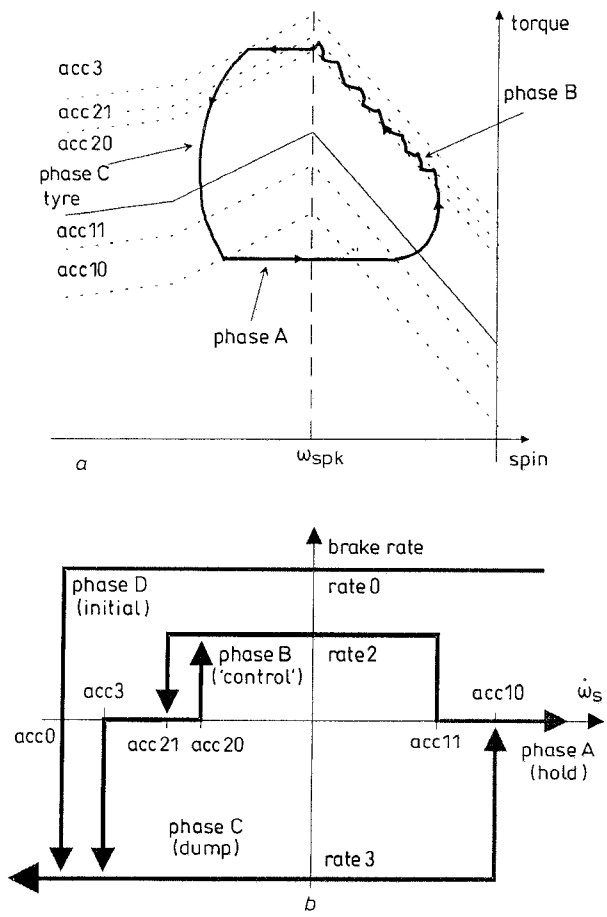

Fig. 20 *Redesigned controller loop*
(*a*) Modified controller action
(*b*) Modified controller schedule

The sliding mode hysteresis design aims to give a (relatively) low frequency chatter while maintaining tight control. To ensure that the sliding mode is attained in all cases rate2 must be chosen so that the spin acceleration can always reach acc21 whilst operating in the stable region of the tyre torque curve. To see how redesigned variables were chosen, consider a simplified model of the operation in phase_B on the stable side of tyre torque curve that can be obtained from eqn. 12:

$$\dot{\omega}_s = \frac{\tau_{tyr}(I_w + I_r)}{I_r I_w} - \frac{\tau_{brk}}{I_w} + \frac{\tau_{rld}}{I_r} \qquad (20)$$

The tyre torque in the stable region increases with decrease in spin speed. Assuming τ_{brk} is increasing linearly with time and that τ_{rld} is constant gives the simplified equation:

$$\dot{\omega}_s = -k_1\omega_s - k_2\text{rate2}t + c \qquad (21)$$

where k_1 will depend on μ, k_2 will depend upon the brake pressure at which the linearisation is performed and c is a constant which will depend upon the road speed about which the approximation is made. To examine the scheduling of the thresholds, the variation of acceleration with μ is needed. The acceleration will

be constant when

$$k_1\omega_s = -k_2\texttt{rate2}t$$
$$\Rightarrow \omega_s = -\frac{k_2\texttt{rate2}t}{k_1}$$
$$\Rightarrow \dot{\omega}_s = -\frac{k_2\texttt{rate2}}{k_1} \quad (22)$$

To achieve sliding motion in phase_B, a (negative) spin acceleration is required that will drive the controller past the acc2 threshold under all conditions, i.e. $\dot{\omega}_s <<$ acc2 then from eqn. 22 we obtain

$$\texttt{rate2} \gg -\frac{k_1}{k_2}\texttt{acc2} \quad (23)$$

The values for k_1 and k_2 can be derived from the parameters of the state equations (eqn. 12) for each operating point. In the sliding mode assume that the mean acceleration is approximately equal to acc2. In phase_B a duration of about 0.1 s and a change of spin in that period of 5% of the peak torque spin is chosen for design, i.e.

$$\texttt{acc2} \approx \frac{d\omega_s}{dt} \approx \frac{\Delta\omega_s}{\Delta t} = \frac{0.05\omega_p}{0.1} = \frac{\omega_p}{2} \quad (24)$$

Since the ABS should operate around the spin peak this suggests a schedule of

$$\texttt{acc2} = \omega_s/2 \quad (25)$$

Combining eqns. 22 and 23 and inserting values for k_1, k_2 and ω_{spk} for each case of μ gives

$$\texttt{rate2} \gg 0.0094 \quad (26)$$

so choose

$$\texttt{rate2} = 0.02 \quad (27)$$

Note that the rate2 should ideally be scheduled; however, this would be difficult with the current ABS modulators.

For an overshoot (past ω_{spk}) between 1% and 10% a suitable choice is

$$\texttt{acc3} = 0.3\omega_s - 2.2 \quad (28)$$

Note that only the value for acc2 has been chosen. The control algorithm requires acc20 and acc21 to be specified. For simplicity these are made linear combinations of acc2 and acc3:

$$\texttt{acc20} = \texttt{acc2} + \Delta(\texttt{acc2} - \texttt{acc3})$$
$$\texttt{acc21} = \texttt{acc2} - \Delta(\texttt{acc2} - \texttt{acc3}) \quad (29)$$

The value of Δ determines the frequency of the switching along the sliding mode. Here Δ was chosen to be 0.1. In retrospect, the frequency achieved is rather high. The values of acc10 and acc11 depend upon the value of the spn0 threshold; if the undershoot is too large, the spn0 threshold will be hit during a normal cycle which is undesirable. The difference in acceleration thresholds can be reflected into the difference of spins on entry and exit of phase_A. With spn0 chosen to be –0.25, acc10 was set to 50 and acc11 to 10.

The system with modified controller was analysed for $\mu = 0.5$ and the resulting graph constructed. The analysis was carried out in the most 'critical' dynamic regions first, these regions being identified from the analysis of the system with the initial controller. This approach resulted in a dangerous dynamic pattern being revealed by the partially created graph before analysis of the system was completed. This pattern, illustrated in Fig. 21, was confirmed to exist for $\mu = 0.8$ and 1.0 as well as $\mu = 0.5$, with all three tyre adhesion curves showing the same pattern of instability within their respective graphs.

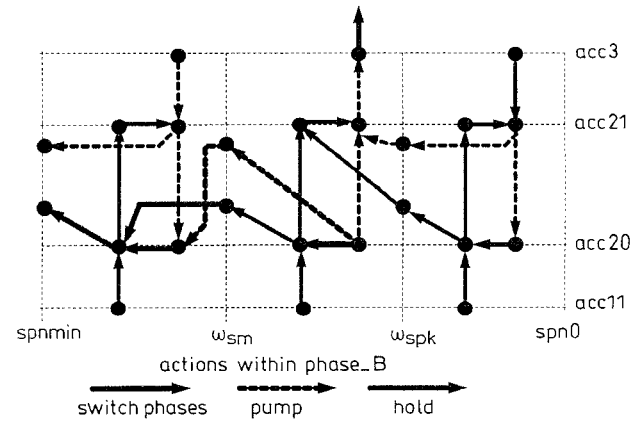

Fig. 21 *Problem with the redesigned controller for $\mu \geq 0.5$*

Fig. 21 shows that for normal operation, the ABS behaves as expected. If Phase_B is entered by acc11 being triggered for wheel spin between Spn0 and ω_{spk} then the graph shows the acceleration torque being driven to the sliding mode and entering a cycle between acc20 and acc21. In this cycle the pressure to the brake cylinder is alternately switched on at rate2 and held. This cycle continues until ω_{spk} is exceeded, then the dynamics are able to exit out via the acc3 rule and begin Phase_C. However, a totally unexpected instability occurs for initial conditions where $\omega_s > \omega_{spk}$. In this case, there is a strong likelihood of the dynamics being driven past ω_{sm}, the knee of the adhesion curve. Once this occurs, a dangerous limit cycle is generated between the acc20 and acc21 thresholds for which the only escape is to wait until ω_s exceeds the Spnmin safety rule. This kind of behaviour can occur if, for instance, the car hits a slippery patch while braking. This could result in the initial conditions for the ABS to suddenly appear around ω_{sm}, inducing the instability in the controller.

Once a problem such as this is diagnosed from the graph, its cause can be deduced from the model. This again is a result of the analysis and the fact the nodes in the graph can be related directly back to the system model. It is unlikely that (with no prior information) an instability such as this would have been spotted by checking the equations. The problem is due to the scheduling of the acceleration thresholds acc20, acc21 and acc3. At high spin speeds, the acceleration threshold decreases faster than the spin acceleration. That is, when operating in Phase_B with the controller holding the brake pressure constant and the spin speed becoming more negative, the acceleration threshold is decreasing with spin at a higher rate than the spin acceleration. Therefore the acceleration threshold is never exceeded and the controller locks up until the Spnmin spin threshold is passed.

To see this consider eqn. 21. When the brakes are being held, the spin acceleration can be approximated by

$$\dot{\omega}_s \approx -k_1\omega_s + c \quad (30)$$

Meanwhile, the controller is waiting for the acc3 threshold to be crossed, i.e.

$$\dot{\omega}_s < m_{\texttt{acc3}}\omega_s + c_{\texttt{acc3}} \quad (31)$$

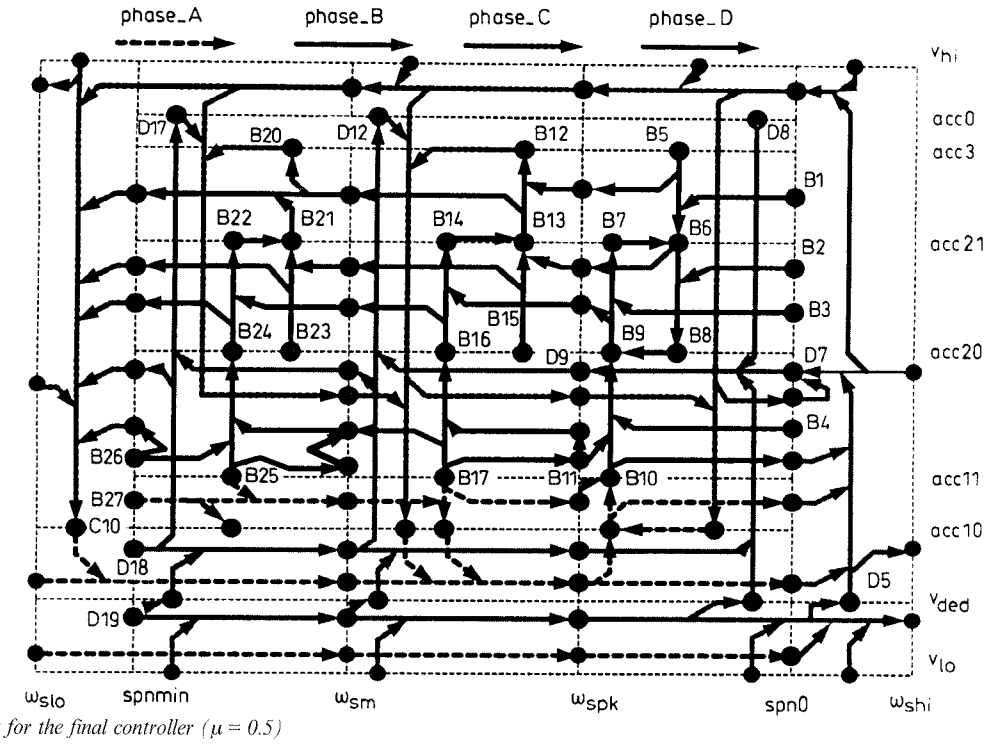

Fig. 22 *Analysis for the final controller ($\mu = 0.5$)*

where m_{acc3} and c_{acc3} are the gradient and offset of the acc3 threshold that is scheduled against ω_s. Combining these two equations gives the inequality

$$-(k_1 + m_{acc3})\omega_s < c_{acc3} - c \qquad (32)$$

The problem arises when ω_s becomes more negative and the term $(k_1 + m_{acc3})$ is also negative; then the inequality cannot be true and the controller cannot switch to Phase_C and release the brakes. Thus when the gradient of the tyre torque function has become very shallow and k_1 is small, eqn. 32 cannot be satisfied. This problem is due to incorrect scheduling of the thresholds acc20, acc21 and acc3, particularly when wheel spin > ω_{spk}. Once diagnosed this problem is easily overcome by choosing different threshold values to ensure eqn. 24 is satisfied.

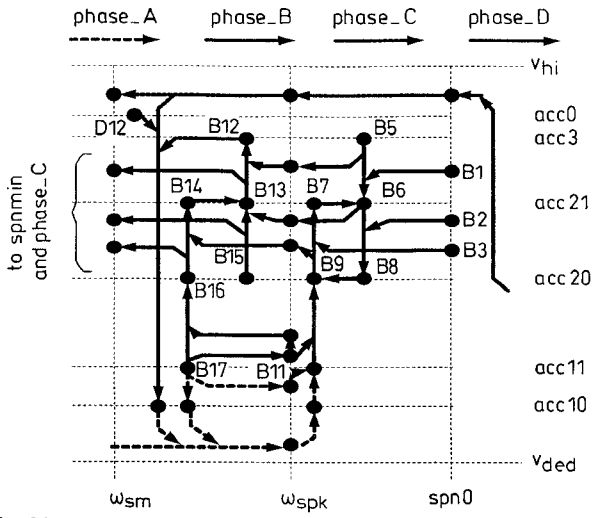

Fig. 23 *Key control loop for $\mu = 0.5$*

8 Final controller design

The complete graph for the final controller design is too large. Instead a simplified version is given in Fig. 22, which shows the graph for $\mu = 0.5$. The key paths of the controller are shown in Fig. 23.

Fig. 23 shows Phase_B being triggered at acc11 (node B11). The dynamics are driven to the sliding mode between acc20 and acc21 and oscillate until spin exceeds ω_{spk}. This is shown by the cycle around B7 ⇒ B6 ⇒ B8 ⇒ B9. The continued sliding cycle after ω_{spk} in the initial, unsuccessful, redesign (shown in Fig. 20) no longer exists, since no cycle occurs around B13 ⇒ B14 ⇒ B15 ⇒ B16. Instead, the dynamics move towards the acc3 threshold (B12) and trigger Phase_C. Fig. 22 shows that the sliding mode after ω_{sm} has disappeared and the acceleration can increase and reach acc3 through B20 at this point. In effect, the graph shows that for $\mu = 0.5$ the controller rules are well integrated and no dangerous conditions are apparent. This result is confirmed with $\mu = 0.2$. Fig. 24 shows the corresponding graph.

Fig. 24 shows that the principal control action has a very similar graph to that for $\mu = 0.5$. The graph shows that Phase_C can be reached through both the acc3 and Spnmin rules as intended. Closer investigation confirms that the preferred path is via the acc3 rule (B12 in the graph).

To confirm that the improved control gives good responses, two simulation runs using the final controller are shown in Figs. 25–30. The interesting thing to note is that this second set of simulation runs does not differ significantly in appearance from the first set shown in Figs. 12–17. This reinforces one of the tenets of this paper, specifically that direct testing or repeated simulation is insufficient to reveal the actions of a piecewise linear system, in this case a logic controller. Given the knowledge from the analysis it is interesting to look at the differences between initial and final controller simulations (Figs. 12–17 and Figs. 25–30, respectively). The most obvious difference is that the final controller exhibits a much faster limit cycle than before. This is a byproduct of the design and in principle, the cycle frequency can be made slower in the final design so that it matches the initial design. More significant are the differences in wheel spin when comparing the plots for $\mu = 0.2$ and $\mu = 0.5$. In the final control-

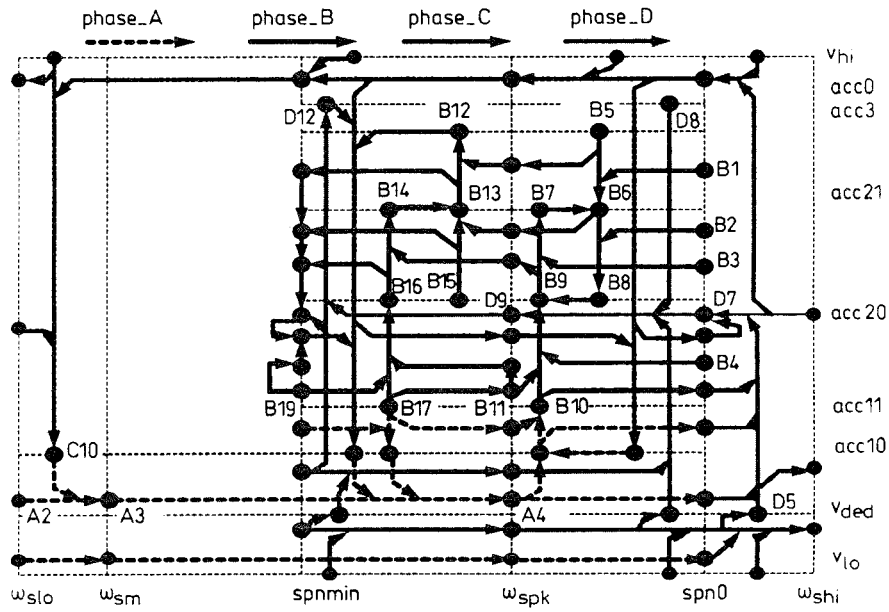

Fig. 24 *Analysis of the final controller ($\mu = 0.2$)*

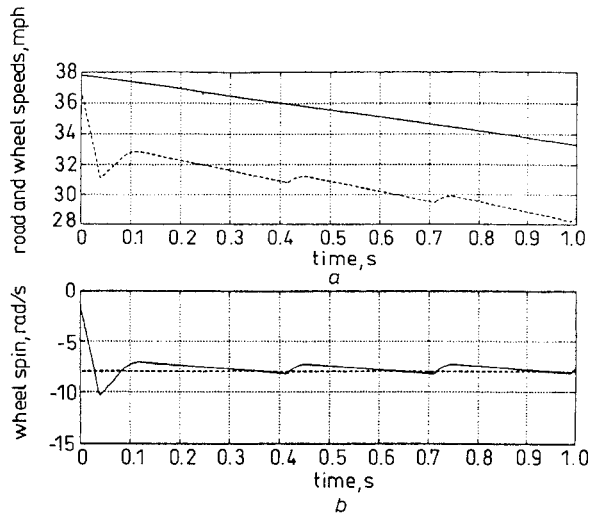

Fig. 25 *Simulation run with final controller*
(*a*) ——— road and - - - - wheel speed
(*b*) Wheel spin
$\mu = 0.2$

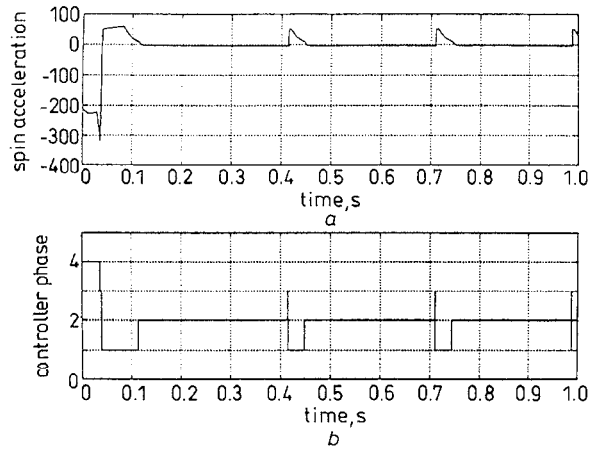

Fig. 26 *Simulation run with final controller*
(*a*) Spin acceleration
(*b*) Controller phase
$\mu = 0.2$

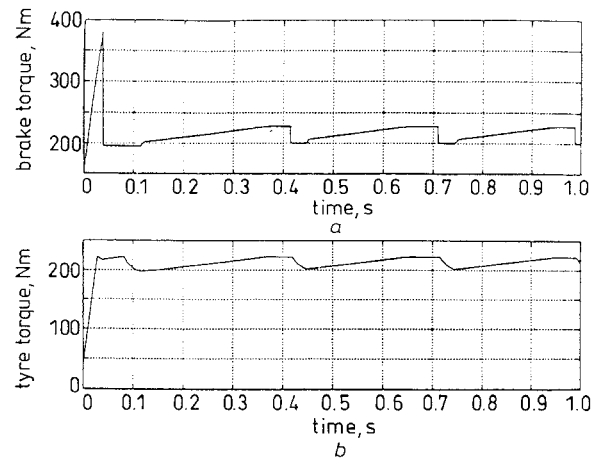

Fig. 27 *Simulation run with final controller*
(*a*) Brake torque
(*b*) Tyre torque
$\mu = 0.2$

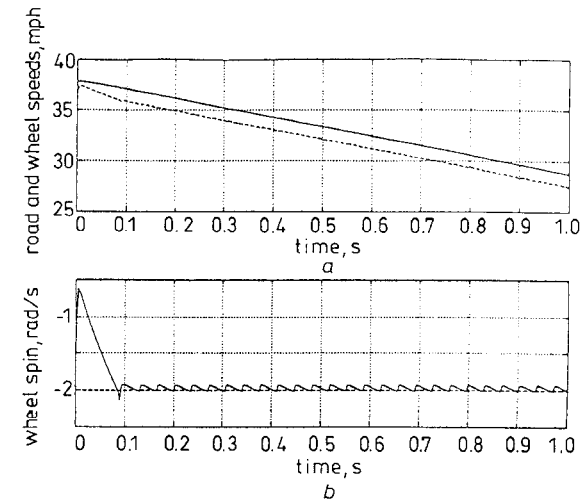

Fig. 28 *Simulation run with final controller*
(*a*) ——— road and - - - - wheel speed
(*b*) Wheel spin
$\mu = 0.5$

ler, the wheel spin does not dip much below the peak spin value for the tyre torque curve and it is clear the wheel spin in these plots spends most of its time in the stable part of the curve. This is not so obvious in the initial controller plots (Figs. 12*b* and 15*b*). Similarly the final controller spends much less time in its level '1' or Phase_A mode relative to the other controller phases (see the controller phase plots) than the initial control-

ler. However, the actual tyre torque and braking torque plots for the initial and final controller differ very little.

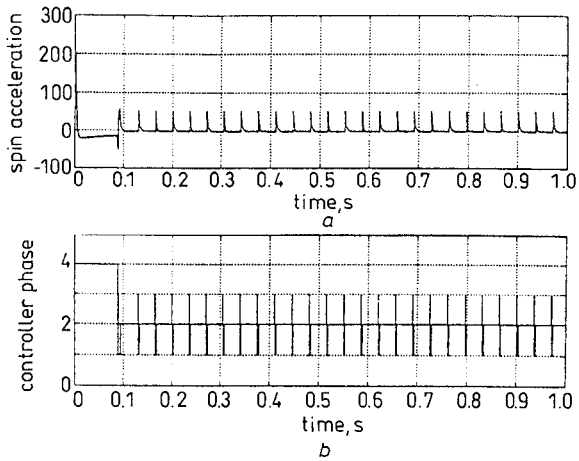

Fig. 29 *Simulation run with final controller*
(*a*) Spin acceleration
(*b*) Controller phase
$\mu = 0.5$

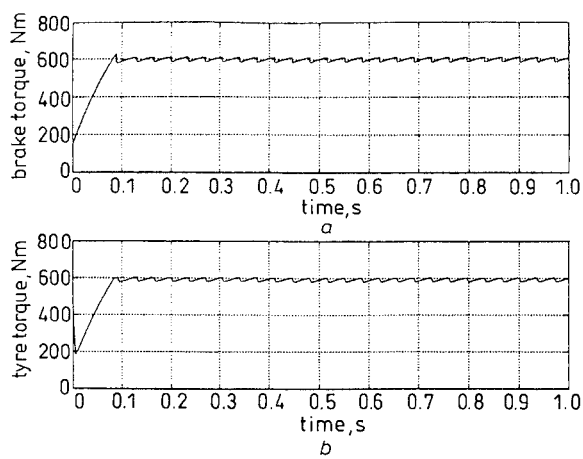

Fig. 30 *Simulation run with final controller*
(*a*) Brake torque
(*b*) Tyre torque
$\mu = 0.5$

9 Summary

The theme of this paper is that, except for the simplest systems, piecewise linear dynamical systems cannot be adequately analysed using conventional simulation tools. However, by using a special piecewise linear analysis tool, it is possible to study in detail the performance of a logic controller in a way which reveals characteristics not visible by simulation, field trials or static analysis. To this end we have applied the technique to a typical industrial piecewise linear system. Specifically we have discussed a model for an ABS controller and applied it to a simplified model of the wheel and brake. The model has been created to capture underlying features of an ABS system although much of the complexity of an accurate 'real life' model has been missed. An initial controller design was carried out using extensive simulation together with intuition as to the effect of the rules on the dynamics. The result was a controller with around 10 rules. These were then analysed using the PL analysis principles outlined in Section 2 and detailed in [3, 4]. The PL analysis produced a linear connected graph which contains information concerning the possible performance of the PL system. In the case of the ABS, the graph highlighted weaknesses in the rules which were not apparent in the initial simulations. The controller rules were consequently redesigned. The new graph showed a further problem; an instability in the rules that was not obvious through the design. Once identified, the source of the instability could be found through the model equations and a new rule design generated which removed the instability. This design produced a high degree of consistency in the graphs for different values of road/tyre friction and showed no inconsistencies in the rule operations. Thus the use of the PL analysis to create a graph linking the controller logic with the closed-loop system dynamics was crucial to the design of the rule base and to the subsequent redesign of the controller laws.

Having shown the analysis working for a simplified model, the next step would be to look at a more realistic model with higher order dynamics and a more complex rule base.

10 Acknowledgments

We would like to highlight the contribution of Dr Russell Wilson-Jones to the paper. He has been an integral part of the ideas behind the work. We thank the EPSRC for supporting the piecewise linear systems research which underpins this paper. The technical and financial support of Lucas Automotive is also acknowledged; special thanks go to M. Appleyard, Professor P. Extance, and Dr P.G. Scotson. Finally we would like to thank Danfoss A/S for its encouragement in writing up the research.

11 References

1 WILSON-JONES, R.: 'A generalised phase portrait for piecewise linear systems'. PhD thesis, Control Systems Centre, UMIST, 1993
2 BESSON, V., PETTIT, N.B., and WELLSTEAD, P.E.: 'Representing piecewise linear systems for analysis and simulation'. Proceedings of 3rd IEEE conference on *Control applications*, 1994, pp. 1815–1820
3 PETTIT, N.B.O.L.: 'The analysis of piecewise linear dynamical systems', Vol. 3 of UMIST Control Systems Centre Series, (Research Studies Press, John Wiley and Sons, Taunton, 1995)
4 PETTIT, N.B.O.L., and WELLSTEAD, P.E.: 'Analyzing piecewise linear dynamical systems', *IEEE Control Syst. Mag.*, October 1995,
5 MADISON, R.H., and RIORDAN, H.E.: 'Evolution of the sure track brake system'. Proc. SAE, 1969, (690213)
6 STEWART, E.E., and BOWLER, L.L.: 'Road testing of wheel slip control systems'. Proc. SAE, 1969, (690215)
7 FARR 1985
8 GERSTENMEIER, J.: 'Electronic control unit for passenger car anti-skid (ABS)', *Proc. I. Mech. E*, 1981, (Paper C186/81)
9 LEIBER, H., and CZINCZEL, A.: 'The potential and the problems involved in integratred anti-lock braking systems'. Proceedings of a conference on *Anti-lock braking systems for road vehicles*, (I. Mech. E. Conf. Publ. 1985-8, London, 1985)
10 LIN, W.C., DOBNER, D.G., and FRUECHTE, R.D.: 'Design and analysis of an antilock brake control system with electric brake actuator', *Int. J. Veh. Des.*, 1993, **14**, (1), pp. 13–43
11 SRINIVASA, R., GUNTER, R.R., and WONG, J.Y.: 'Evaluation of the performance of anti-lock brake systems using laboratory simulation techniques', *Int. J. Veh. Des.*, 1980, **1**, (5), pp. 467–485
12 ATHERTON, D.P.: 'Nonlinear control engineering' (Van Nostrand, London, 1983)
13 COOK, P.A. 'Nonlinear control systems' (Prentice Hall, London, 1986)
14 TSYPKIN, Y.Z.: 'Relay control systems' (Cambridge University Press, Cambridge, 1984)
15 FLÜGGE-LOTZ, I.: 'Discontinuous and optimal control' (McGraw Hill, New York, 1968)
16 KALMAN, R.: 'Physical and mathmatical mechanisms of instability in nonlinear automatic control systems', *Trans. ASME*, 1957, **79**, pp. 553–566
17 SONTAG, E.D.: 'Nonlinear regulation: the piecewise linear approach', *IEEE Trans. Autom. Control*, 1981, **26**, pp. 346–358

18 BANKS, S.P., and KHATHUR, S.A.: 'Structure and control of piecewise linear systems', *Int. J. Control,* 1989, **50**, pp. 667–686
19 McMULLEN, P., and SHEPHARD, G.C.: 'Convex polytopes and the upper bound conjecture', London Mathematical Society, Lecture Notes Series 3, (Cambridge University Press, 1971)
20 PETTIT, N.B.O.L., MANAVIS, T., and WELLSTEAD, P.E.: 'Using graph theory to visualise piecewise linear systems'. Proceedings of 3rd European *Control* conference, Rome, 1995, pp. 1631–1636
21 LEIBER, H., and CZINCZEL, A.: 'Antiskid system for passenger cars with a digital control unit'. Proc. SAE, 1979, (790458)
22 BAKKER, E., PACEJKA, H.E., and LIDNER, L.: 'A new tyre model with applications in vehicle dynamics studies'. Proc. SAE, 1989, (890087)
23 PADOVAN, J., and PADOVAN, P.: 'Modelling tyre performance during anti-lock braking', *Tire Sci. Technol.,* 1994, **22**, (3), pp. 182–204
24 HUSSAIN, S.F.: 'Digital algorithm design for a wheel lock control system'. Proc. SAE, 1986, (860509)
25 BREARLY, M., PHILLIPS, M.I., PRESCOTT, R.D. and ROSS, C.F.: US Patent 4,852,953, 1989
26 WELLSTEAD, P.E.: 'Introduction to physical system modelling' (Academic Press, London, 1979)
27 PACEJKA, H.B., and SHARP, R.S.: 'Shear force development by pneumatic tyres: a review of modelling aspects', *Veh. Syst. Dyn.,* 1991, **20**, pp. 121–176
28 PAUWELUSSEN, J.P., and PACEJKA, H.B. (Eds): 'Smart vehicles' (Swets & Zeitlinger, Netherlands, 1995)