

System Design Document

Jaime Martin, Nilesch Humbad, Steve Kansa, Dan Forbes

Anti-lock Brake System

March 24, 1999

Contents

1	Introduction	3
1.1	System Objectives	3
1.2	Hardware, Software, and Human Interfaces	3
1.3	Design Constraints and Limitations	4
2	System Architecture	4
2.1	Topology	5
2.1.1	Topology Model Description	7
2.2	Data Stores	7
2.3	Software Control	8
3	GUI Description	8
4	System Design Description	12
4.1	Object Model	12
4.1.1	Object Model Description	12
4.1.2	Data Dictionary	17
4.2	Dynamic Model	18
4.2.1	Dynamic Model Description	18
4.3	Functional Model	24
4.3.1	FunctionalModelDescription	24
5	References	27
5.1	System Documentation	27
5.2	Technical References	27

1 Introduction

Originally developed for aircraft, anti-lock brake systems (ABS) work by limiting pressure at any wheel that decelerates rapidly. Therefore, maximum stopping force can be applied without brake lockup. In operation, the wheelspeed sensors, located at each wheel, send electronic pulse signals to the control unit. Should rapid deceleration be detected, indicating wheel lockup, the computer signals the valve unit to limit the hydraulic pressure to the wheel cylinder. This is accomplished by diverting some of the fluid into the accumulator, or a small reservoir. When the brakes are not being applied, the fluid is pumped out by the return pump.

1.1 System Objectives

The anti-lock braking system is a safety mechanism designed to prevent skidding and help drivers maintain steering control during an emergency stopping situation in which wheel lockup occurs. The purpose of ABS is to provide controlled stopping by maintaining maximum tire to road friction and also allow steering control while braking. In contrast, conventional brakes do not allow for steerability in this situation because of wheel lockup. When a driver applies the brakes, the braking force (which decelerates the car) and the cornering force on each wheel (which keeps the car on the road) interact against one another. The tire is pulled in the direction of the overall force, which is the result of these two forces. The wheel can only transmit a limited overall force based on the condition of the tires, the road, and the wheel contact force. In extreme braking situations, the braking force will overcome all other forces and cause one or more wheels to lock up. This results in a complete loss of cornering forces in the locked wheels.

With ABS, the four wheelspeed sensors (one for each wheel) detect rapid deceleration and thus detect wheel lockup. By means of its digital controller and hydraulic fluid, ABS in effect pumps the brakes as much as 10 times per second. The driver is then able to maintain steering control because the tires continue to rotate without sliding. The system can selectively pump only the tires that are slipping, maintaining maximum brake pressure on the rest of the tires.

To ensure a high level of safety, ABS has the ability to check itself for errors and turn off when it determines a fault, leaving normal braking unaffected. This is accomplished by redundant processors which can perform error checking on each other and various components in the system. When an error is detected, ABS stores the error code in memory, allowing for more efficient repairs.

1.2 Hardware, Software, and Human Interfaces

The major hardware components of ABS include the input circuit, ROM, two caches, four wheelspeed sensors and four solenoid valves. The wheelspeed sensors provide wheel frequencies to the input circuit where they are converted to square-wave output signals. Two CPUs exist, one for interaction with the front brakes, and the other for interaction with the rear brakes. A distributed operating system supports multithreading to exploit the parallelism of the two CPUs. Applications in ABS include the error data monitor, voltage monitor, controller logic and input conversion. The signals from the input circuit are converted into digital words and sent to the controller logic where they are used for the calculation of slip and deceleration. These values determine the proper positioning commands to be sent to the solenoid valves. The error data monitor and voltage monitor are testing applications for the anti-lock braking system. The error data monitor performs a test of the CPUs each time the car is started, when the brakes are used and at system shutdown. Error messages are relayed to the voltage monitor which governs the warning lamp and deactivates ABS. The warning lamp alerts the driver to problems in ABS. The voltage monitor also produces error

codes when voltage levels are either too high or too low. All error codes are stored in a cache for access by a technician.

1.3 Design Constraints and Limitations

Constraints to the anti-lock braking system include the following:

- The wheelspeed sensor gives input to the controller. Rapid deceleration causes the ABS system to start. The system must check for skidding hundreds of times a second. If one wheel is decelerating faster than the others, lockup can be caught before it happens.
- The system must control a valve that interfaces with the wheel cylinders. It results in 'pumping' of the brakes, as much as 10 times per second.
- This relieving of pressure within the wheel cylinder can be accomplished by diverting some of the fluid into a small reservoir.
- The ABS must re-pump this reservoir's fluid back into the main fluid reservoir.
- When the vehicle is initially started, the ABS goes through a test sequence. Another test sequence is needed every time the brakes are applied. The system must evaluate itself. If an error is found, the system turns itself off, and sends an error light to the dash.

The ABS system is designed to calculate the maximum braking pressure that can be applied to the wheels before lockup occurs. This is accomplished through the use of a previously loaded ROM table of ideal braking specifications. The table is created using sample data from laboratory test cars in simulated conditions. These values are computed with an appropriate safety margin to allow for some variation in car maintenance. Performance of the system can be negatively effected when conditions on the vehicle, such as tire wear, tire pressure and brake wear, fall below accepted norms. ABS will not work accordingly if the user pumps the brakes manually. It is recommended that ABS car owners hold the brake down and allow ABS to perform the pumping actions, which is its entire purpose.

2 System Architecture

The architecture of ABS is broken into many subsystems with a peer-to-peer relationship. The layering and partitioning of these subsystems defines the interaction of these subsystems, and is illustrated by the system topology diagram. Another architectural property of ABS is that software control is event-driven. The control resides in several independent objects concurrently.

ABS conforms to a real-time system architectural framework. To put this in more concrete terms, a real-time system is one that is timing crucial; i.e., time constraints on actions are particularly tight and the slightest timing error cannot be tolerated. At highway speeds, fractions of a second can mean tens of yards. Clearly, real-time responsiveness is a necessary condition of the system design. ABS has been designed to make as many as ten solenoid adjustments per second.

Real-time systems, by definition, are also interactive systems. An interactive interface is a system that is dominated by interactions between the system and external agents, such as humans, devices or other programs. In the case of ABS, the only external agent is the brake pedal.

2.1 Topology

The Anti-lock brake subsystems represent a peer-to-peer relationship. For example, ABS has two CPU's, one for controlling the front brakes and one for the wheel brakes; both communicate with various applications through one operating system.

The subsystems are layered in the system topology to represent abstraction. Abstraction increases from the lower layers up to the higher layers. Each layer is thus built in terms of the one below it, and knowledge is one-way. This means layers know only about subsystems below them. ABS layering conforms to a closed architecture, each layer is built only in terms of the immediate lower layer. Portability and modularity are the advantages of using a closed architecture.

Partitions divide a system into independent, weakly-coupled subsystems which provide specific services. In other words, the vertical partitions of the system topology represent functional decomposition.

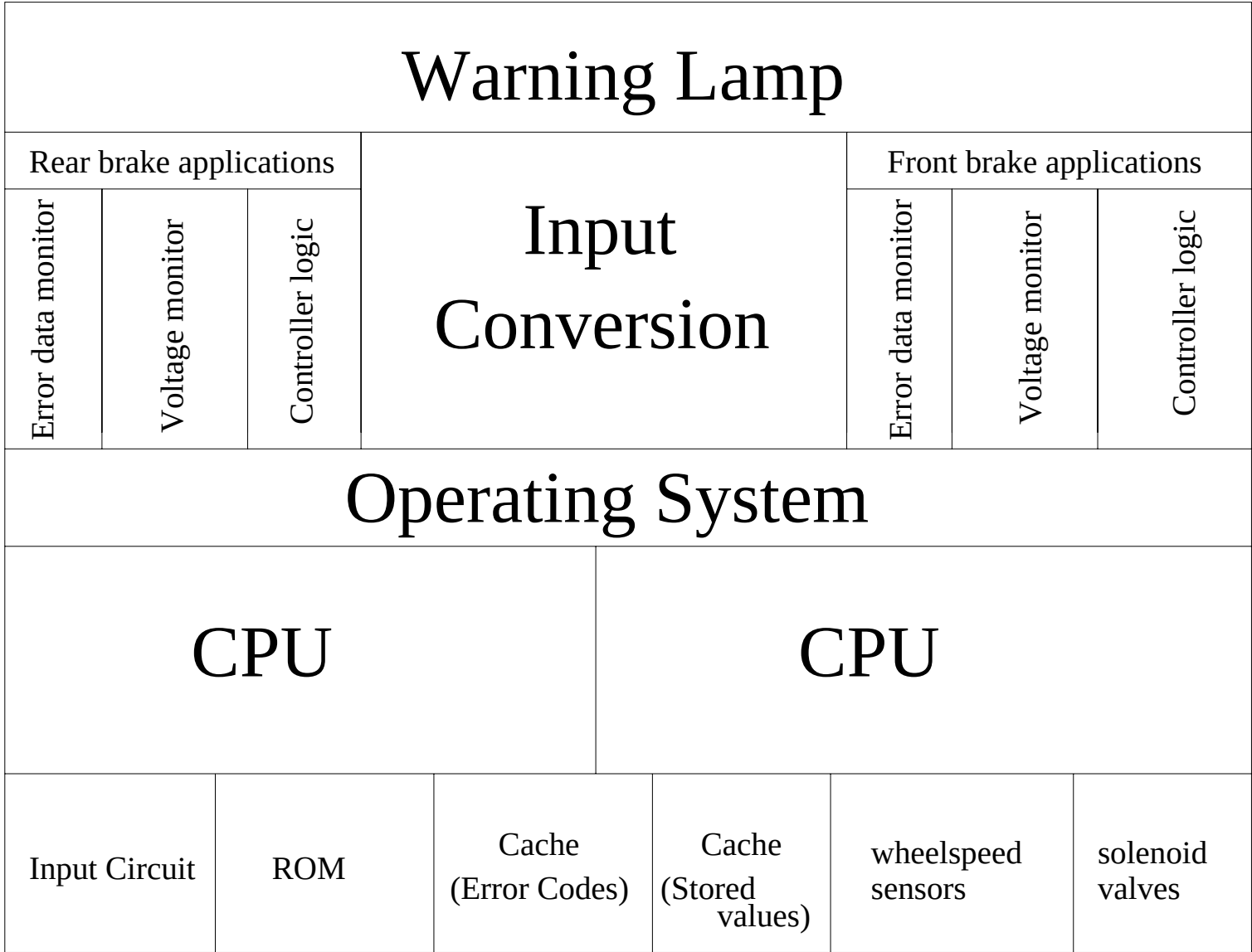


Figure 1: Topology Model

2.1.1 Topology Model Description

The lower layer of the ABS topology represents hardware units. They are partitioned into the input circuit, which converts the sinusoidal AC voltage from the wheelspeed sensors into square-wave output signals; a cache holding a table of error codes to be replaced on a first-in-first-out basis; a cache holding previously calculated deceleration and speed; ROM containing a table correlating speed values with average deceleration and slip rates used for detecting wheel lockup; the wheel-speed sensors; and the solenoid valves.

Another hardware layer represents the two central processing units (CPU). One CPU controls rear brake applications while the other controls front brake applications. Two CPUs are used in order for the front and rear brakes to use the applications concurrently.

Immediately above the CPU layer is the operating system layer. ABS has a distributed operating system. A distributed operating system is necessary to facilitate both CPUs. The distributed operating system supports multithreading to enhance application throughput and application responsiveness. Multithreading also exploits the parallelism of the two CPUs.

The layer above the operating system represents system applications. They are partitioned into the rear brake applications, the front brake applications, and the input conversion subsystems. Both the rear and front brake applications consist of the error data monitor, voltage monitor and controller logic applications. The error data monitor tests data used to make sure the system is working properly. For example, identical test inputs are sent to the front and rear CPUs and the end calculated values are compared. Should they differ, an error code is generated and sent to the voltage monitor which shuts down the entire ABS system and sends the code to an error code cache in memory for future evaluation. The voltage monitor also monitors voltage levels of ABS components and issues an error code should values deviate from the norm. Any time the voltage monitor issues an error code, it shuts down the system and turns on the warning lamp.

The controller logic application calculates the controlled variables slip, deceleration and wheel speed for the wheels based on the processed digital word from the input conversion application. These values are used to determine proper positioning commands for the solenoid valves. If wheel lockup is detected based on these values, the controller logic needs to inform the solenoid valves to start pumping, thus releasing pressure.

Both CPUs have access to the input conversion application which takes the square-wave output signal from the input circuit and converts it into digital words for control signals to the controller logic.

The top layer in the ABS topology represents the user interface. The warning lamp alerts the driver to an error in ABS, thus indicating it is inoperative.

2.2 Data Stores

Three main data stores exist in the anti-lock braking system. The first is located in ROM and holds a table of ideal braking specifications. There is a correlation between braking conditions and set vehicle speeds. Braking conditions are defined by the coefficient of friction and a maximum deceleration rate. Vehicle speed defines a mean braking deceleration, total braking time and a braking distance. This table is referenced for comparison with currently calculated values to determine if wheel lockup will occur.

A memory cache is used to hold a table of previously calculated speeds, slip and deceleration to be used for new calculations of these values. The cache incorporates a first-in-first-out replacement algorithm because older values are only needed for one calculation after their own use.

The third, and final, memory store is also a cache. This cache holds the error codes issued by the

voltage monitor. Each error code is time stamped to determine which ones need to be evaluated by the service technician. A first-in-first-out replacement method is also used for the error code cache.

2.3 Software Control

ABS is an event-driven system. In an event-driven system, the system's activation is triggered by some external event. Control resides in a dispatcher or monitor which activates the system when certain conditions have been met. For ABS, there are several primary dispatchers. The first is the error data monitor. Each time the brake is applied, a test is performed by the error data monitor. This test compares the output values of two identical CPUs, which are simultaneously receiving the same input. If there are any discrepancies, a message is relayed to the voltage monitor, which will turn on the warning lamp. The voltage monitor is also a dispatcher which activates the warning lamp when voltage levels are not within close tolerances. Finally, there is the controller logic, which constantly receives input from the input conversion application. When the controller logic detects potential wheel lockup, based on the slip and deceleration values, it sends proper positioning commands to the solenoid valves.

Concurrency is another feature of the anti-lock braking system. In concurrent systems, control resides within several independent objects simultaneously, with each object performing a separate task. In ABS, the wheelspeed sensors take input while the solenoid valves are acting in response to previous output.

The CPUs themselves operate concurrently with respect to each other. While one CPU handles data from the rear brakes, the other CPU handles data from the front brakes. Besides the obvious speed advantage this affords, there is also a considerable safety advantage, as there are separate error data monitors and voltage monitors. This translates to increased performance in error detection and recognition. The major purpose of concurrency with the CPUs in ABS is so the front and rear brakes can perform the controller logic applications simultaneously. This allows for detection of wheel lockup in the front and rear brakes at the same time, without a time loss for reactions. Without this concurrency, wheel lockup could be occurring in the rear brakes while the front brake values are being processed. The loss of time to detect the lockup in the rear brakes would cause late reaction time by the solenoid valves, and possibly a car crash thus making the entire anti-lock braking system worthless. In safety-critical systems, where fractions of a second can mean the difference between life and death, one can't be too particular about system choices.

3 GUI Description

The ABS user interface is composed of two key elements: the brake, and the ABS error lamp that is located on the dashboard. These two components are the only elements of ABS that give user 'feedback.' The pumping action of the brakes, when ABS is activated, can be considered a part of user 'feedback.' However, since this action is a result of the brakes being pressed, it is not included as part of the user interface. Furthermore, this action underlies the user interface, so it is not ignored in the overall system.

The user interface can be more clearly understood by looking at the ABS prototype. Figure 2 shows what the prototype looks like upon instantiation. The prototype has four user changeable fields: brake pressure as a percentage, potential wheel lockup selection, road condition selection and the speed of the motor vehicle. The brake system in the prototype is modeled by first specifying

a brake pressure percentage and then by eventually clicking the brake button. The user specifies the brake pressure, but the brake button will be pressed after all other user changeable values are adjusted. The speedometer is shown in the prototype as a reference to the relationship between wheelspeed frequency and wheel lockup. The speedometer is not part of the ABS user interface because ABS retrieves its wheelspeed frequency values from its wheelspeed sensors, not the actual speedometer. However, the speedometer provides a useful way of showing the relationship between wheel speed and wheel lockup, and thus fits very well in the prototype. The speed, which is at the default value of 60 mph can also be adjusted in the prototype by the user. Furthermore, the prototype allows for the selection of one of four road conditions: normal (default), rain, snow and ice. In general, the worse the road conditions, the more likely ABS will be activated. The selection of the wheels that will 'potentially' be locked up is made possible because there is no way of telling which wheel will lock up. Thus, the wheels that are selected in this part of the prototype are the only wheels that can lock up if ABS is activated. After all the user definable values are adjusted accordingly, the brake button is clicked. In the case of an error, the ABS error lamp lights up, resulting in ABS shutdown (leaving normal braking unaffected). If ABS is activated based on the brake pressure, speed and road conditions, the wheels that are selected for potential wheel lockup are thus locked up. The prototype shows wheel lockup and ABS activation by lighting up an 'ABS Active' light. The reset button changes all the values and fields in the prototype to their default values.

Figure 3 shows an example scenario of ABS being activated. The brake pressure is set to 70 and three are chosen for potential wheel lockup. Based on all the data given, wheel lockup is detected in wheels one and three. When the button is pressed, ABS is activated.

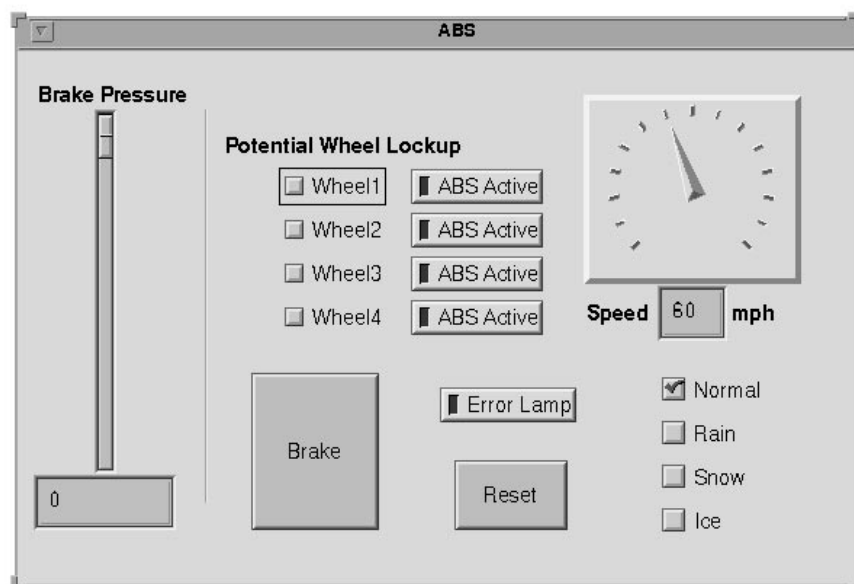


Figure 2: Prototype Figure 1

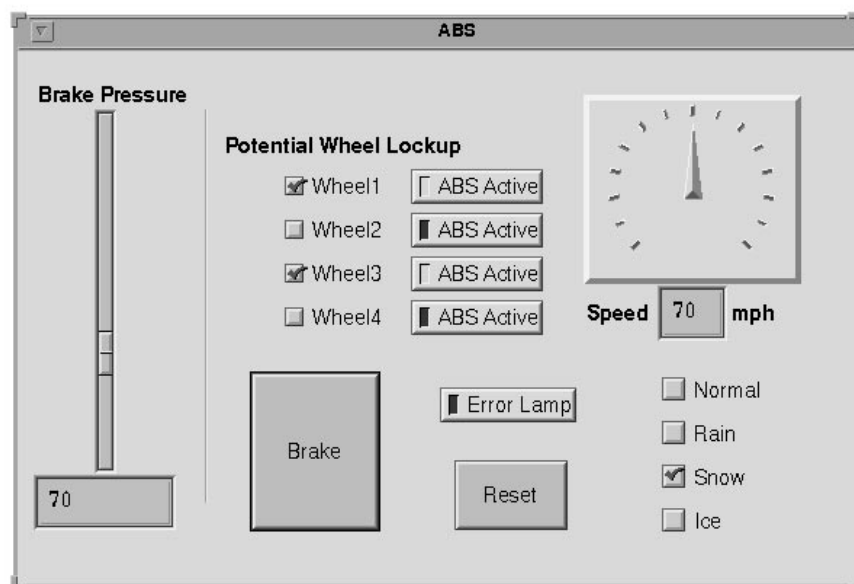


Figure 3: Prototype Figure 2

4 System Design Description

The System Design Description contains the OMT diagrams. There are three different design diagrams, the Object Model, Dynamic Model and Function Model. The Object Model describes the structure of objects in the system. The model shows an object's identity, attributes, operations and relationships to other objects in the system. The Dynamic Model describes aspects of the system concerned with time and sequencing of operations. The model represents the different states the system can occupy and events that transition the system from state to state. The Functional Model is concerned with the transformation of data in the system. The model shows the flow of data from inputs, through operations and data stores, to outputs. The models provide three different, but related way of looking it the system as a whole.

4.1 Object Model

4.1.1 Object Model Description

Changes have been made to the object models since the analysis document. These changes were necessary because components originally designed as hardware, such as the monitoring circuit, have been changed to software applications. Also, certain objects have been removed, such as LSI circuits, and their functionality has been joined with other other objects. In the case of the LSI circuits, their functionality as described in the analysis document, is now merged with the error data monitor. To accommodate the changes made in the object models, the data dictionary has also been revised.

The previous object models describe the objects in the system, and their relationships. Figure 4 is an overall look at the objects and their aggregation. Aggregation is the "part-whole" relationship in which objects representing the components of something are associated with an object representing the entire assembly. In the ABS object model, wheelspeed sensors have a pole pin that has a magnet. All of the "hasA" relationships are represented by a diamond.

Generalization is the relationship between a class and one or more refined versions of it. These relationships are represented in the object model by triangles. For example, the outlet valve is a type of valve. Valve is the superclass and outlet valve is a subclass inheriting from it.

The same relationships are shown in the other three models. These figures also describe associations between objects. The numbering shows the direction of the associations. Also labeled in these models are the attributes and operations of the objects. Figure 5 describes relationships of the sensors with the input circuit, as well as associations with other sensor objects. The wheelspeed sensor has an attribute named wheelFrequency which represents the wheel frequency value.

Figure 6 represents associations in the controller of the ABS, and its interaction with the solenoid valves and warning lamp. There also exists numerous links between each object of the controller. Certain operations present include on and off for the warning lamp and voltage monitor. Attributes include slip and deceleration used by the controller logic to determine the positioning commands for the solenoid valves. The connection between the HPM and various braking system components is represented in Figure 7. The solenoid valves perform three operations, pressueBuildup, pressureRelease and pressureHold.

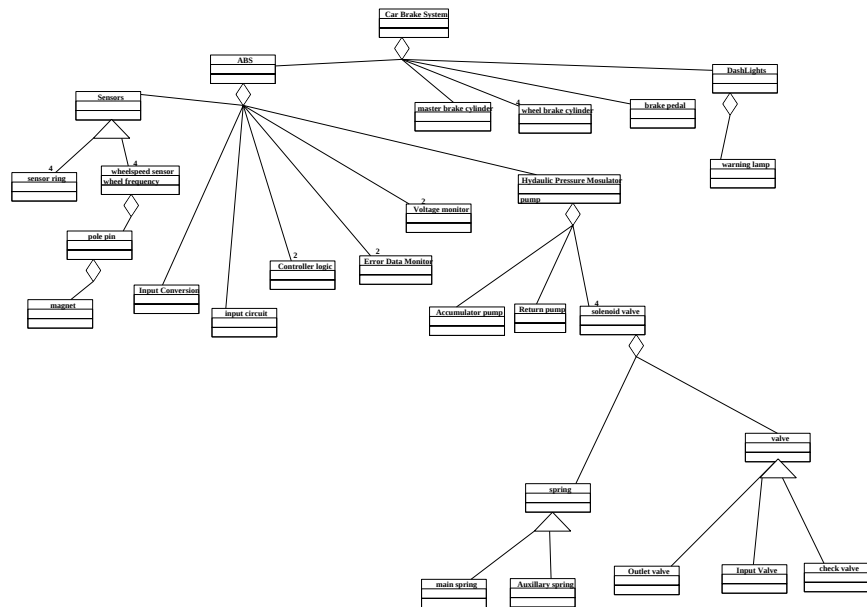


Figure 4: Object Model

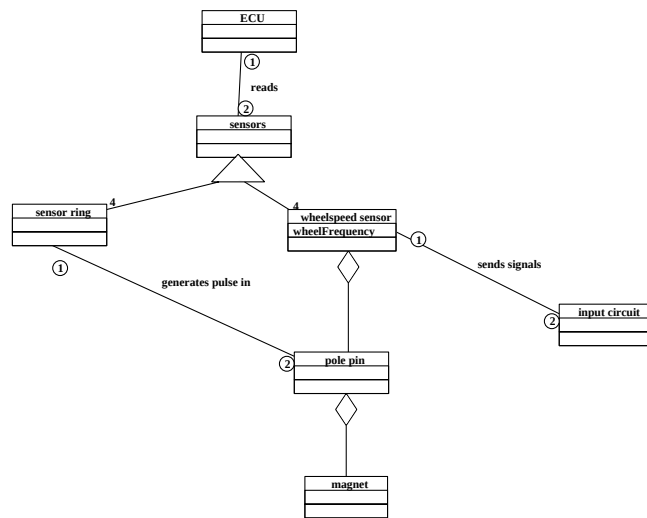


Figure 5: Sensor Object Model

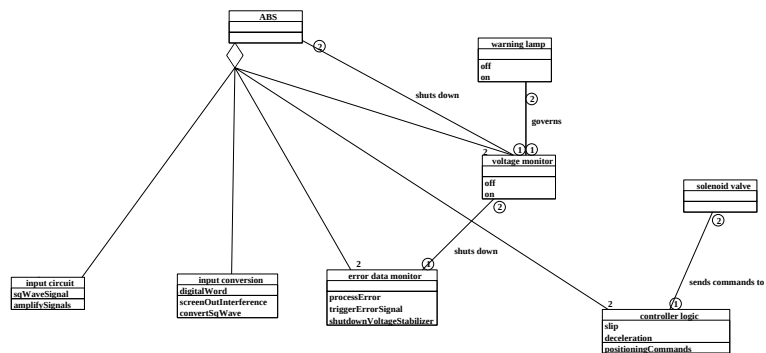


Figure 6: Controller Object Model

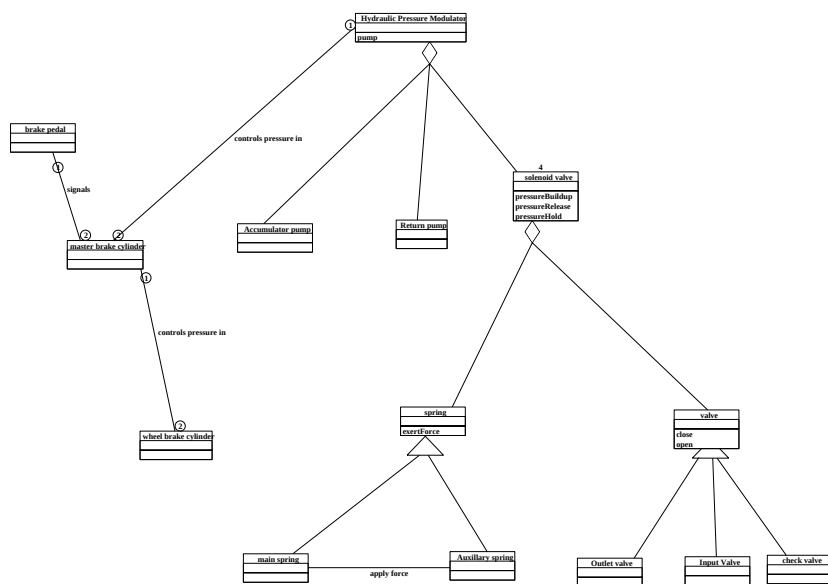


Figure 7: HPM Object Model

4.1.2 Data Dictionary

1. ABS: Used to enhance braking safety, the anti-lock braking system prevents the wheels from locking in response to excess brake pressure. The ABS exercises direct control over the operation of the service brakes.
2. Input Circuit: This circuit converts the sinusoidal AC voltage from the wheelspeed sensors into square-wave output signals.
3. Controller Logic: Calculates the controlled variables "slip" and "deceleration" for the wheels based on the processed wheel speed (digital words from the input conversion application). Based on these values, the controller logic sends the appropriate positioning commands to the solenoid valves.
4. Error Data Monitor: Used for error recognition and evaluation. Responds to faults by triggering an error signal and shutting down the voltage monitor to deactivate the system.
5. Warning Lamp: Alerts the vehicle operator to the fact that ABS is inoperative.
6. Input Conversion: This is where the square-wave signal is converted into digital words.
7. Voltage Monitor: Monitors and stabilizes voltage supply, incorporates a low voltage recognition function to switch off the system in the event of inadequate vehicle power supply or fault memory. Also governs the warning lamp.
8. Hydraulic Pressure Modulator: Implements commands issued by the Controller Logic. The HPM uses solenoid valves to automatically control pressure levels at the wheel-brakes.
9. Return Pump: Transmits the emerging brake fluid through the accumulator back to the master brake cylinder as pressure is released from the wheel-brake cylinders.
10. Accumulator: Temporarily absorbs flow surges that occur as brake fluid is discharged during the pressure-release phase.
11. 3/3 Solenoid Valves: Control pressure modulation processes in the wheel-brake cylinders during active ABS intervention. Each wheel has a solenoid valve.
12. Wheelspeed Sensors: Provide signals (wheel frequencies) to the input circuit as the basis for determining the wheels' rotational speeds.
13. Pole Pin: Connected to a permanent magnet producing an electric field that extends outward to the sensor ring, generating a voltage in the sensor's winding to provide an index of wheel speed.
14. Sensor ring: Pulse rotor joined to the wheel hub.
15. Main Spring: Located in the solenoid valves. Due to high tension and pressure from the auxiliary spring, the main spring opens the input valve. Likewise, when the pressure is no longer exerted, the input valve closes.
16. Auxiliary Spring: Located in the solenoid valve. It exerts forces on the main spring for control of the input valve.

17. Armature: Located in the solenoid valve. Its interaction with the springs controls the opening and closing of the outlet valve.
18. Input Valve: Located in the solenoid valve. It reacts to incipient wheel lockup by interrupting the connection between the brake master cylinder and the wheel-brake cylinders of the wheels concerned in order to prevent additional increases in pressure.
19. Outlet Valve: Located in the solenoid valve. The outlet valve opens to release the excess braking pressure. It is a passage between the affected wheel-brake cylinder and the accumulator.
20. Master Brake Cylinder: Displaces hydraulic brake fluid under pressure to the rest of the brake system, triggered when the brake pedal is depressed.
21. Wheel-Brake Cylinder: Converts hydraulic fluid pressure into mechanical force. There is one for each wheel. When the brake pedal is depressed, the pistons in the master cylinder force the fluid through the brake lines and into the cylinders at each wheel.
22. Brake Pedal: Located on the left side of the accelerator pedal, stepping on the brake pedal begins the process of slowing down or stopping a vehicle.

4.2 Dynamic Model

4.2.1 Dynamic Model Description

The dynamic model has been changed in the System Design Document to elaborate on the error checking and transitions between states during the braking process.

Figure 8 is a high level look at the states and events in ABS. When the car is started or the brake pressed, the controller activates an error checking test. Should the system check return positive results, the anti-lock braking system is enabled. In case of errors and test failure, only the normal service brakes will be in use and ABS is disabled.

At a lower level, Figure 9 models the concurrent actions that take place in the ABS controller. The first section represents the voltage monitoring function of the controller. If the voltage to the sensors pumps or error light falls out of a range of preset norms then ABS is shut off and an error code is stored. The second section represents the calculation of wheel deceleration, slip and speed values by the controller logic.

Figure 10 represents the transitions between the Pressure Build-up, Pressure Holding and Pressure Reduction stages during the time the brake is pressed. The controller determines the transitions between the states based on deceleration, wheel slip and wheel speed values. During the transitions, the controller sends commands to open and close the input and outlet valves of all four solenoid valves. These commands and actions control the braking force on the wheels. Depending on external conditions, different wheels can be in different states at any give time during braking. When the brake is pressed and the error test has been passed the controller sends the Open Input Valve command to the solenoid valves and the Pressure Build-Up state is entered. In this state the controller monitors the deceleration rates of the wheels. When the deceleration rate of a wheel exceeds a preset maximum level, the controller sends the Close Input Valve command and the Pressure Holding state is entered. Pressure Build-up can also be exited when the brake is released by the driver.

In the Pressure Holding state the controller monitors the wheel deceleration and wheel slip rate. If the deceleration rate falls back into an acceptable range then the Pressure Build-up state is re-entered. If the wheel slip rate exceeds the maximum level, the controller sends the Open Out-Let command, looks up the ideal wheel speed from the table in memory and enters the Pressure Reduction state.

In the Pressure Reduction state the controller compares the current wheel speed with the ideal value. When the current wheel speed is equal to the ideal value, the controller sends the Close Outlet Valve and Open Input Valve commands and re-enters the Pressure Build-up state. If the driver releases the brake during this stage, pressure reduction by the solenoid valves is no longer necessary, and this state is exited.

Figure 11 represents the test on the two CPUs each time the car is started or the brake is pressed. During the test the two CPUs are sent identical inputs. If the system is working correctly, the two CPUs should produce identical outputs. If the outputs are identical the system is enabled, otherwise the system is shut down and an error code is stored.

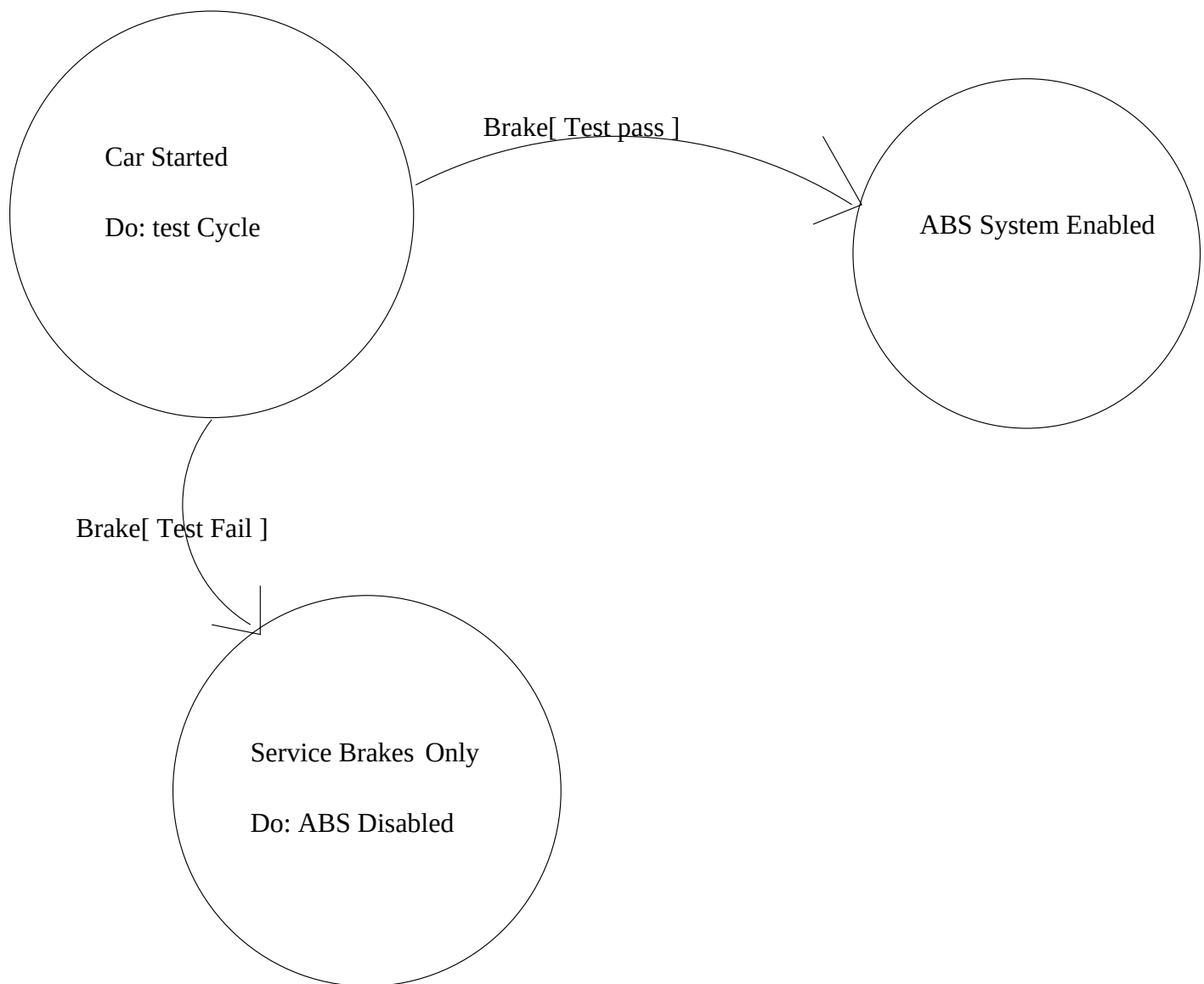


Figure 8: High Level

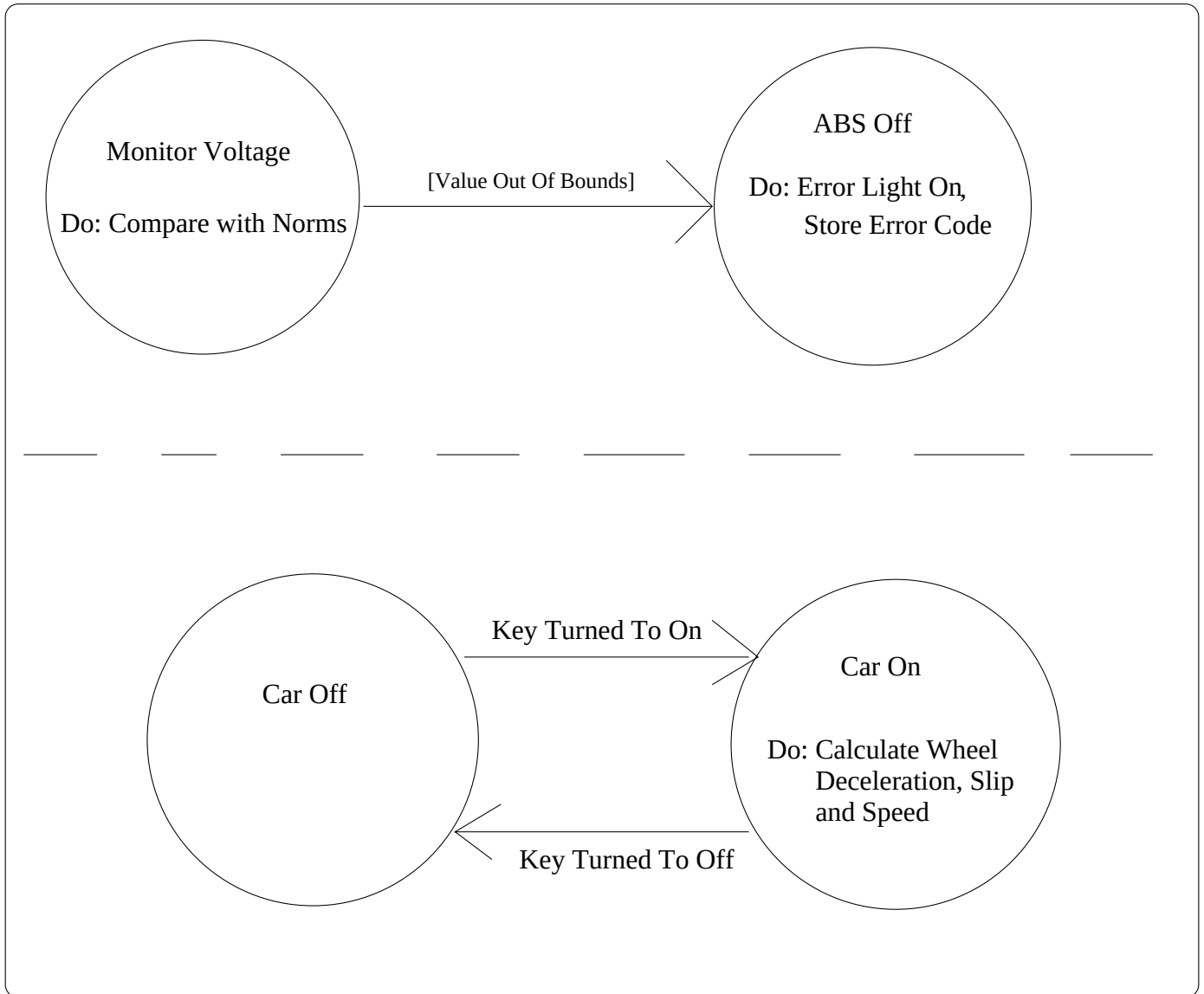


Figure 9: Current Processes

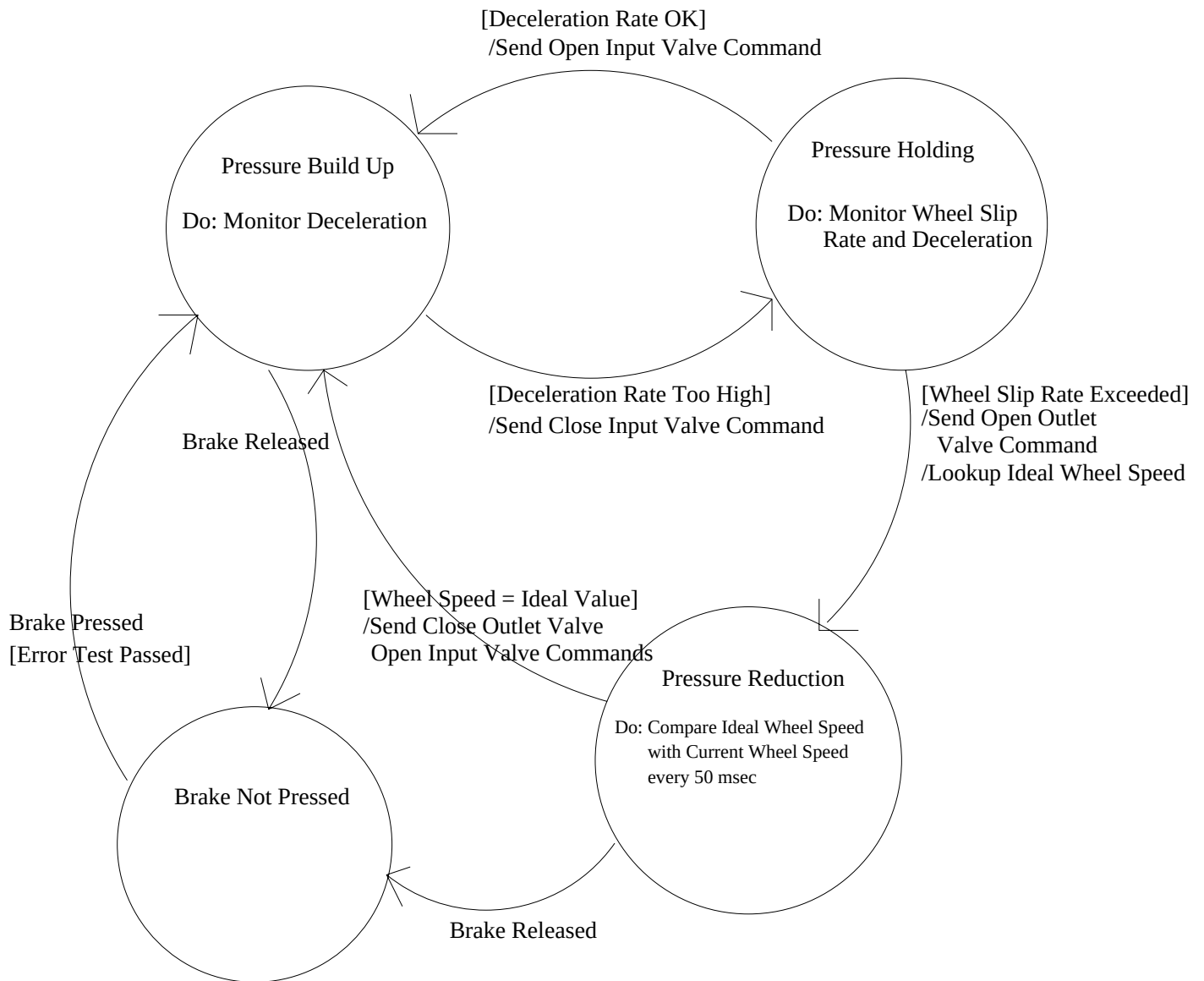


Figure 10: Braking Actions

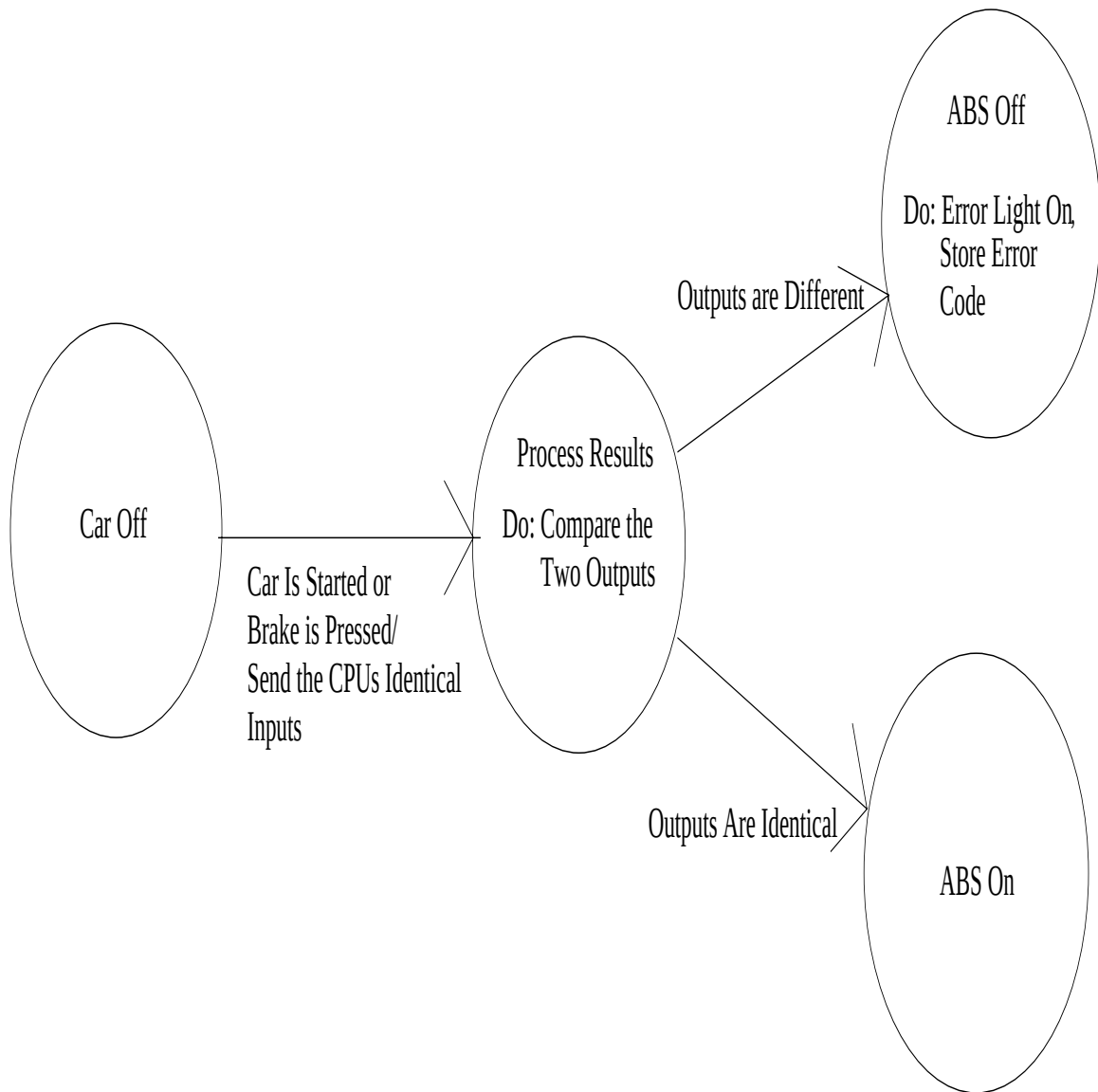


Figure 11: Controller Test

4.3 Functional Model

4.3.1 FunctionalModelDescription

The functional model of ABS shows the relationships among the processes and their respective input and output data transformations. The functional process loops continuously when the brakes are applied, thus the ABS functional model represents a very brief moment of time in real-time system performance. The lower level functional model has been modified to include all data stores and show more detailed data interaction.

The high level functional model shows the main input actor, output actors and data store. The wheelspeed sensors (input actor) input the solenoid voltage frequencies to the main process, the controller logic process. This process provides all of the computations of ABS, and results in three different outputs.

The lower level functional model shows detailed data interaction and computations. The wheelspeed sensors continuously provide a sinusoidal voltage frequency to the input circuit which executes two processes. First the input circuit takes the sinusoidal voltage frequency and applies filtration and amplification procedures to make a 'clean' analog signal. Then the analog signal is converted to square-wave output signals that represent wheel frequency in the input circuit. After they leave the input circuit, the square-wave output signals are then converted to digital words through an analog to digital converter. If an error occurs during the conversion, a fault signal is sent to an error handling process which writes an error code to memory and sends a safety relay signal to an error lamp. The error lamp, which is located on the dash board, lights up indicating that ABS is shut down due to an error. In general, all errors are processed in a similar fashion by this error handling process which results in the following actions: ABS shuts down, the error lamp lights up and the error code is written to memory.

At this point, the controller logic takes control of the data signal and eventually outputs control commands for the solenoids. In the case of an error, a safety relay is sent to an error lamp and the error condition is stored in memory. The controller logic calculates slip and deceleration based on the digital words and the previous slip and deceleration values that are retrieved from cache. The cache also allows for fast data writes to memory when the new slip and deceleration values are stored. Errors that possibly occur result in a fault that is sent to the error handling process. The new slip and deceleration values are then compared to ideal values specified in the ROM. Errors that may occur in this process are also accounted for by the error handling process. Based on the comparisons, if ABS is to be activated, control commands are sent out by the controller. Otherwise, if ABS does not need to be activated, the controller does not output anything and ABS repeats the functional process from start. In the case of ABS needing to be activated, positioning commands are sent from the controller logic to the solenoid valves and thus cause a pumping action to occur in the hydraulic system. ABS at this point is 'turned on'.

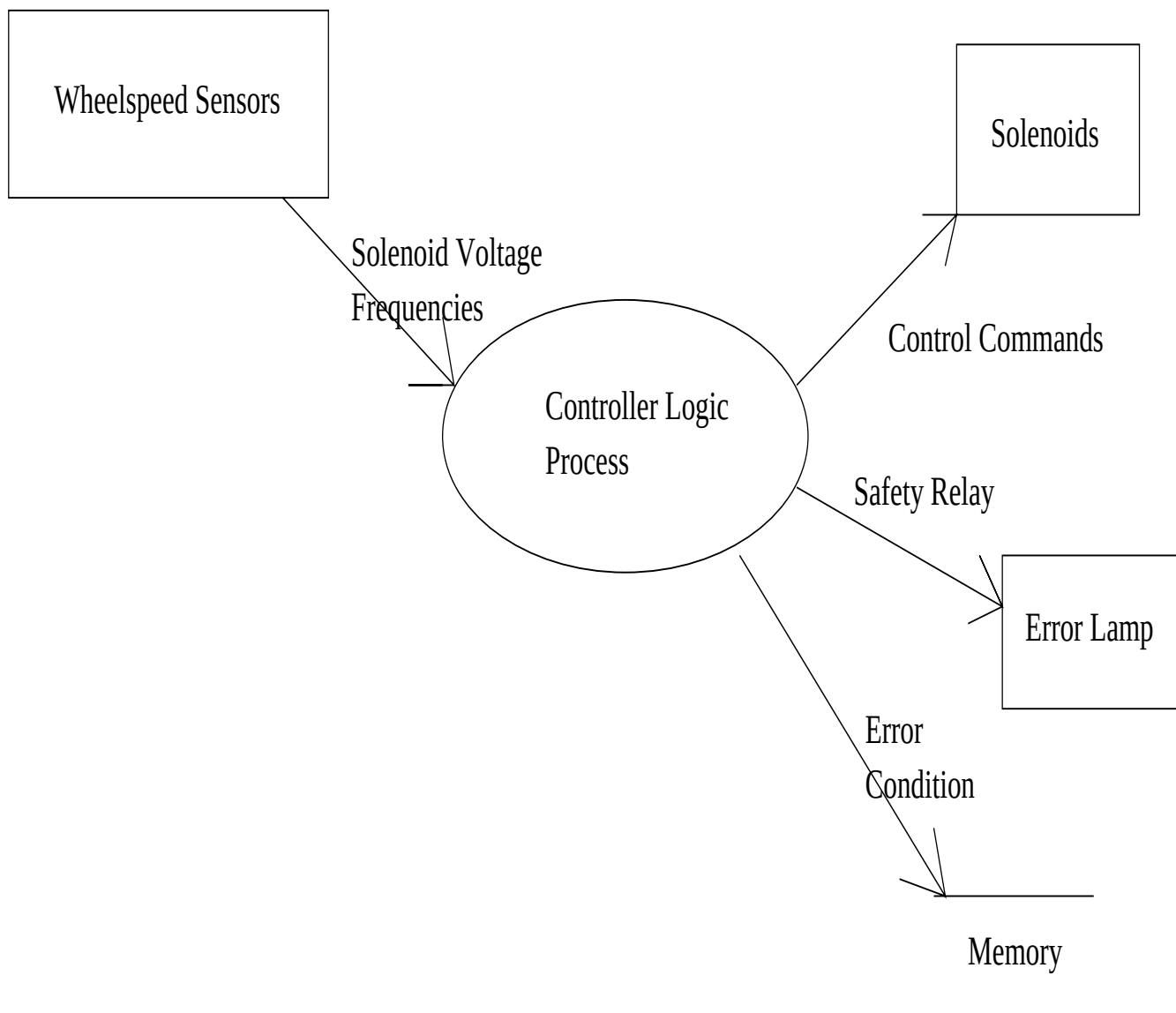


Figure 12: High Level Functional Model

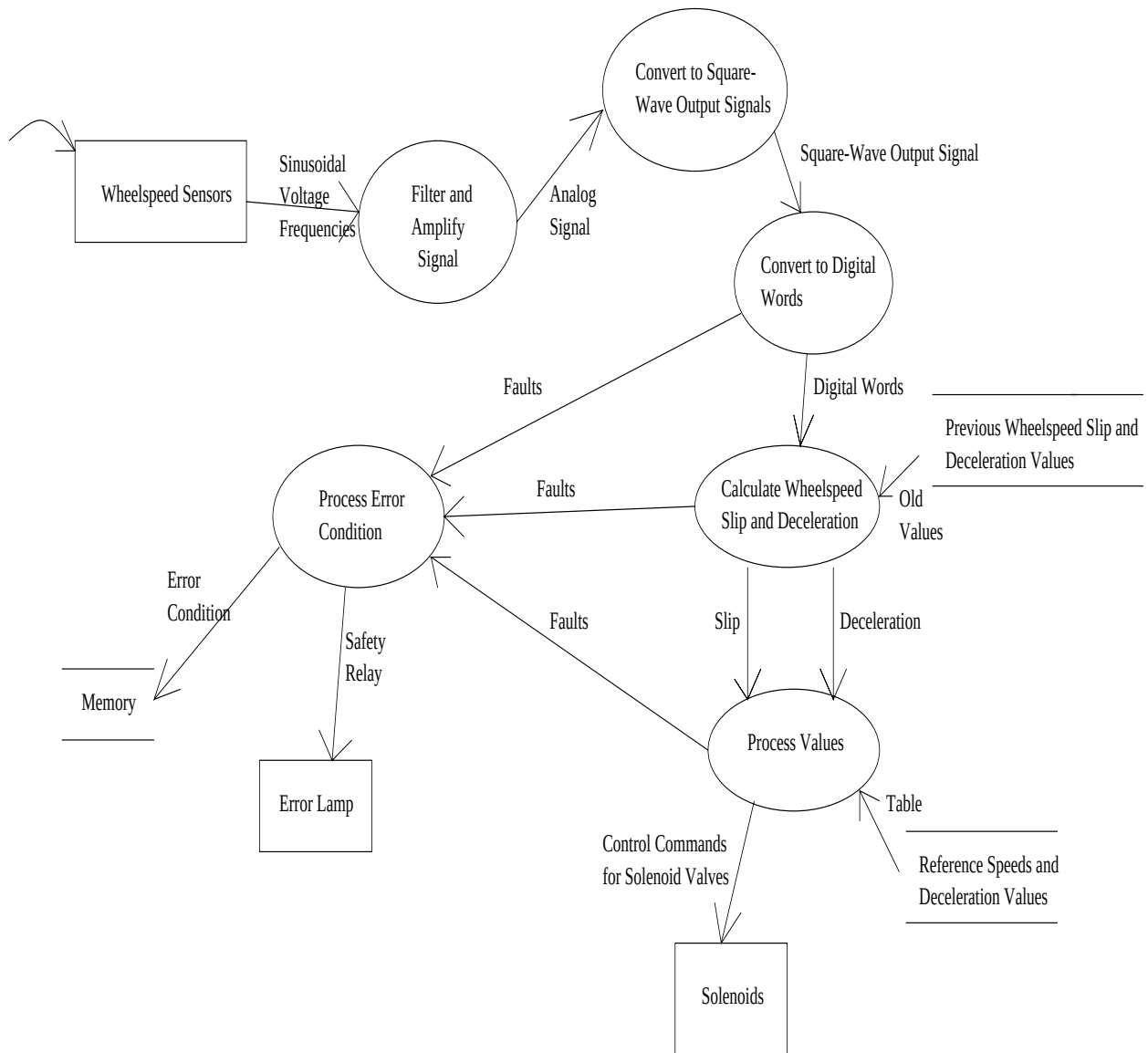


Figure 13: Low Level Functional Model

5 References

5.1 System Documentation

- Pressman, Roger. Software Engineering: A Practitioner's Approach, McGraw-Hill, 1987.
- Rumbaugh, James. Object-Oriented Modeling and Design. New Jersey: Prentice-Hall, 1991.
- Coursepack: CPS410 Operating Systems, Mutka, Matt, Spring 1997.
- Autoshop Online <http://www.autoshop-online.com/auto101/labs.html>
- Informative Graphics Corp. <http://www.innerbody.com/innerauto/htm/auto.html>, "Anti-lock Braking System." 1997.

5.2 Technical References

- Society of Automotive Engineers. Automotive Braking Systems. 1st edition. Massachusetts: Bosch, Robert, 1995.