



This is to certify that the

thesis entitled

DESIGN AND DEVELOPMENT OF A
MICROPROCESSOR-BASED STEERING CONTROLLER
FOR AN OVER-THE-ROW APPLE HARVESTER

presented by

Bruce Lanier Upchurch

has been accepted towards fulfillment
of the requirements for

Masters degree in Agricultural Eng.

A handwritten signature in cursive script, reading "Truman C. Schubert".

Major professor

Date November 10, 1980



OVERDUE FINES:
25¢ per day per item

RETURNING LIBRARY MATERIALS:
Place in book return to remo
charge from circulation reco

DESIGN AND DEVELOPMENT OF A MICROPROCESSOR-BASED
STEERING CONTROLLER FOR AN OVER-THE-ROW-APPLE HARVESTER

By

Bruce Lanier Upchurch

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Department of Agricultural Engineering

1980

ABSTRACT

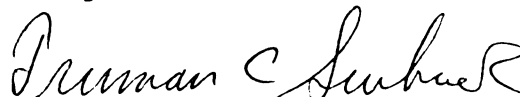
DESIGN AND DEVELOPMENT OF A MICROPROCESSOR-BASED STEERING CONTROLLER FOR AN OVER-THE-ROW APPLE HARVESTER

By

Bruce Lanier Upchurch

The purpose of this research was to design and develop a microprocessor-based steering controller to provide improved steering maneuverability of an over-the-row apple harvester. The controller allowed the machine operator to select from five steering modes: 1) automatic, 2) front only, 3) four way, 4) crab, and 5) rear only. Upon recognizing the selected steering mode, the controller performed the necessary steering control by executing the appropriate steering algorithm. Sensors were designed to provide wheel and tree position information. The microprocessor-based controller successfully performed steering coordination for conventional operator steering. The automatic steering system successfully controlled the machine's movement such that the tree was kept within 15 cm of the machine's centerline during movement over a straight and curved row.

Approved



Major Professor

Approved



Department Chairman

ACKNOWLEDGMENTS

The author wishes to express his thanks to his advisor, Dr. Truman C. Surbrook, for his valuable time, guidance, and critical evaluation throughout the course of this research. He would also like to thank Dr. Bernard R. Tennes of the USDA who provided funds and support for this work. The author would also like to acknowledge other faculty members, technicians, and fellow graduate students for their helpful suggestions and comments. Finally, a very special thanks to my wife, Mary Ann, whose patience, moral support, and many sacrifices have helped make this thesis possible.

TABLE OF CONTENTS

CHAPTER	Page
LIST OF FIGURES	iii
1. INTRODUCTION	1
2. LITERATURE REVIEW	5
3. OBJECTIVES	11
4. DESCRIPTION OF DESIGN	13
4.1 Description of Operation	13
4.2 Description of Steering Modes	16
4.3 Description of Sensors	17
4.4 Description of Hardware	28
4.5 Description of Software	36
5. TESTING AND MODIFICATIONS	47
5.1 Testing of the Steering Modes	47
5.2 Testing of the Auto-Steer Mode	50
5.3 Modifications	52
6. RESULTS	57
6.1 Steering Mode Results	57
6.2 Auto-Steer Results	63
7. SUMMARY AND CONCLUSIONS	70
8. SUGGESTIONS FOR FURTHER RESEARCH	73
REFERENCES	74

LIST OF FIGURES

FIGURE	Page
1. Front view of the over-the-row apple harvester on which the steering controller was implemented.	14
2. Schematic of the harvester's steering controller showing how the various components were interconnected.	15
3. Diagram representing the four main steering modes: (a) front only ($\theta_r = 0$); (b) four way ($\theta_r = -\theta_f$); (c) crab ($\theta_r = \theta_f$); and (d) rear only ($\theta_r = \theta_d$).	18
4. Diagram showing the sequence of events which occur when automatic steering is selected.	19
5. The four tree sensors used to detect the position of the tree with respect to the centerline of the harvester.	20
6. The absolute shaft encoders used to detect the angular wheel position.	20
7. The apparatus that allowed the sensor arm to pivot about an axis.	23
8. An absolute shaft encoder used to determine the actual position of the wheels and sensor arms.	23
9. The Gray code used in the (a) wheel and (b) sensor arm encoders.	25
10. Circuit diagram of the comparator circuit used to convert the array into a digital form.	27
11. Circuit diagram used to convert a 6-bit Gray code into a 6-bit natural binary. To perform a 4-bit conversion, only the four most significant bits are used.	29

12.	Diagram of the controller's section which selectively input data from the various sensors.	30
13.	Diagram of the controller's output section which was responsible for the actual turning of the wheels.	32
14.	Diagram of the controller's memory section showing both volatile and nonvolatile parts.	34
15.	Schematic of the steering mode selection circuit.	37
16.	Flowchart of the main program used in the steering controller.	38
17.	Flowchart of the auto-steer subroutine.	40
18.	Flowchart of the front only subroutine.	43
19.	Flowchart of the four way subroutine.	44
20.	Flowchart of the crab and rear only subroutine.	46
21.	A single turn 100K ohm potentiometer was connected to the shaft of the wheel encoder. The output of the potentiometer was an analog voltage representing the angular position of the wheel.	48
22.	Diagram showing the placement of the tree stands in the (a) straight and (b) curved paths. These two paths were used to test the ability of the controller to keep the machine centered over the row.	51
23.	The new sensor arm used with the new auto-steer algorithm which produced a different code for each 2.54 cm (1 in.) offset.	53
24.	Flowchart of the new auto-steer algorithm which centered the machine by proportional control.	56
25.	The front and rear wheel steering responses when the steering wheel was rotated at (a) fast and (b) slow speeds. The front wheels were controlled by the movements of the steering wheel while the rear wheels were positioned by the controller.	58

- | | | |
|-----|---|----|
| 26. | The rear wheel response to changes in the front wheel position during the front only steering mode. The front wheels were steered by the operator, and the rear wheels were controlled by the steering controller. | 60 |
| 27. | The rear wheel response during the crab steering mode while the front wheels were steered by the operator. | 61 |
| 28. | The rear wheel response to changes in the front wheel position during the four way steering mode. The front wheels were controlled by the operator, and the rear wheels were positioned by the controller. | 62 |
| 29. | Diagram showing the centerline path of the front and rear of the over-the-row apple harvester while traveling over a straight row. The altered auto-steer algorithm was used to perform the necessary control. | 64 |
| 30. | Diagram showing the centerline path of the front and rear of the over-the-row apple harvester while traveling over a curved path. The altered auto-steer algorithm was used to perform the necessary control. | 65 |
| 31. | Diagram showing the centerline path of the front and rear of the over-the-row apple harvester while traveling over a straight row. The new auto-steer algorithm which positioned the wheels according to the amount the machine was off center was used to perform the centering tasks. | 67 |
| 32. | Diagram showing the centerline path of the front and rear of the over-the-row apple harvester while traveling over a curved row. The new auto-steer algorithm which positioned the wheels according to the amount the machine was off center was used to perform the centering tasks. | 68 |

1. INTRODUCTION

Over the past twenty-five years, the U.S. apple production has shown a sizeable increase, and this increase is expected to continue (11)^{1/}. Although the production of apples is increasing, the amount of land available for fruit plantings is decreasing each year due to the building of residences, expressways, and factories (8). In order to achieve increased production with less land, the grower is planting a controlled size tree which produces more fruit than the "old standard" trees. Not only does the controlled size tree produce more fruit, but more trees are planted per unit area.

In Michigan 66,157 acres are planted with apple trees. Of these acres, 46 percent of the trees are the "old standard" type and 54 percent of the "new standard" type as of 1978 which is an increase of 35 percent over the past five years (9). Carlson (3) refers to the "new standard" tree, which now is known as a compact tree, as a tree with a base of eight feet and a height of ten feet. Each year, a larger percentage of the trees planted are of the compact type.

^{1/} Numbers in parentheses refer to appended references.

When compact trees are properly spaced, they reach maximum production at an earlier age and produce more bushels of fruit per acre. The compact trees when compared to the "old standard" trees can be more economically pruned, sprayed, and harvested (16).

Since the grower is losing skilled labor to industry due to the large difference in hourly wages between agriculture and manufacturing, the grower needs to find a way to increase productivity so that a higher wage can be offered to the skilled workers. Before the grower can increase productivity, a way to increase the harvesting rate must be found. There are devices available which help increase a picker's productivity; however, the devices are not enough. A mechanical harvester is a better solution, because the harvester would allow the workers to harvest more acres than they would with mechanical aids. By harvesting more acres, the grower's productivity increases which would allow a higher wage to be paid to the skilled workers.

Today, most of the mechanical harvesters are built around the shake-and-catch idea. In the shake-and-catch system, a shaker is used to loosen the fruit by vibrating the trunk or scaffold limbs and a padded catching surface which covers the entire area under the tree is used to catch the fallen fruit (10). The harvesters are either a two section harvester which stops and joins at each tree for the harvesting operation or an over-the-row harvester

which can continuously move over the row of trees while harvesting the fruit. The two section harvester requires two drivers, one for each section, while the over-the-row harvester requires only one driver. In orchards with "old standard" trees, the present stop and catch harvester is efficient, because it can efficiently harvest 40 trees per acre with yields of 20 bushels per tree. When harvesting an orchard with "new standard" trees, the shake-and-catch harvester's efficiency decreases since 10 to 40 times more compact trees can be planted per unit area than the "old standard" trees (15). The over-the-row harvester is ideal for harvesting compact trees, because when the trees are planted in a high density orchard, a type of hedgerow is formed which makes it difficult for a two section harvester to operate (1).

Presently, the USDA at Michigan State University is working on a continuous over-the-row harvester. In addition to harvesting, the machine will also perform the pruning and spraying operations. For all operations, the machine must remain centered over the trees to maximize the operation being performed (14). During each operation, the machine operator has numerous machine functions to check; therefore, it would be helpful if the machine followed a row of trees which would eliminate the operator having to drive while watching the functions of the machine. Since the driver is located on top of the machine so that the operations can be monitored, the driver does not have visual contact

with the tree after it enters the machine.

As a beginning of a total automatic guidance system, a centering controller is needed which will keep the machine centered over a tree after it enters the machine. The purpose of this research is to design and develop a steering controller which is the first step in the solution of making the machine operator a machine supervisor.

2. LITERATURE REVIEW

Automatic guidance is not a new area of research. Over the past few years, the amount of research on automatic guidance of farm machinery has increased, because the guidance system would help prevent early fatigue, which is caused by wearisome driving of the operator.

Kirk (7) reported on a furrow-following tractor guidance system. The furrow was prepared by a previous pass of the machine and was detected by a mechanical arrangement extending in front of the tractor. The sensor followed the furrow and sent an analog voltage to an electronic circuit for comparison with a reference voltage. The voltage from the sensor was proportional to the distance the tractor was from the furrow due to the angular displacement of the sensor. If the two voltages were unequal, an analog voltage proportional to the difference was sent to a motor. Through a mechanical linkage to the steering wheel, the motor rotated the steering wheel. The speed of rotation depended upon the analog voltage supplied to the motor. This guidance system used only a front sensor which could not prevent machine deviation from a small radius curved path. The furrow would not be an acceptable sensing device for orchard applications, because the making of a furrow in the orchard could cause severe erosion problems.

For use on a swather, the mechanical sensor of the furrow-follower was replaced by an infra-red sensor. The sensor distance from the crop edge influenced the intensity of the reflected light. First the signal from the sensor was linearized, and then it was fed to a motor controller and amplified. The controlled motor positioned the swather steering wheel by means of a chain drive. The movement of the steering wheel caused the swather to move left or right in order to keep the machine a certain distance from the swath edge (6). The swather guidance was used on rear only steer and could only follow curved paths with large radii. Since the sensor itself had to be kept within fourteen inches of the target and the target had to be a continuous crop row, the infra-red sensor would have little use in orchard applications.

Busse (2) reported on an automatic steering mechanism that was developed by the Class Company^{2/} for use on self-propelled combines and forage harvesters. The row sensor had two swivel mounted sensors which were spring loaded to hold the bars in a neutral position. As the crop row entered the machine, each sensor rotated a certain amount depending upon the row location. The outputs of the sensors,

^{2/} Trade names are used in this paper only to provide specific information. Mention of a specific trade does not constitute an endorsement of the product by the author.

which were analog voltages with the magnitude representative of the amount of rotation, were compared in an operational amplifier control unit. The electronic control unit compared the results of the comparison between the two sensors with the location of the steering wheels before making a steering correction. A steering correction was achieved by activating one of two steering actuators which were solenoid actuator valves. When they were activated, a left or right turn of the steering wheels resulted. The sensing method of the Claas system was applicable only to a continuous row crop. Although the steering actuator was a solenoid valve, it could be used to make steering corrections; since, the solenoid valve actuator in the Claas system allowed smooth steering corrections.

A digital controller for automatic tractor guidance was reported by Schafer (12), in which the guidance system utilized a set of antennas mounted on the tractor to sense the location of a buried wire which was excited by a low current, low frequency signal. The output of the sensing circuits after conditioning were two logic signals, one for a too far left error and one for a too far right error. These two steering error signals were inputs into a digital controller which determined the direction of steering. The output of the digital controller was fed to a digital-to-analog converter, the output of which activated a proportional flow control servo-valve. The analog voltage was proportional to the flow through the valve which determined

the rate at which the steering corrected. Although the digital controller performed as well as any analog controller in the past, the application of buried cable sensing is not feasible for orchards, because the cost of setting up an orchard with the cable is too expensive.

Since most of the past work involving automatic guidance systems for farm machinery dealt with either front or rear only steering, Shukla (13) investigated the tracking accuracies of different types of steering systems for automatic guidance systems. By using a computer model and an experimental vehicle to verify the model, the best tracking accuracy occurred when four wheel steering was used with both a front and rear sensor where the front sensor guided the frontwheels and the rear sensor the rear wheels; however, there was little change in the tracking accuracy when only a front sensor was used to guide both the front and rear wheels. Front wheel steering with a front sensor was less accurate than any four wheel steering, but the front steering proved to be more accurate than the rear steering with a front sensor. During testing, the sensor was in continuous contact with the desired path. Since a tree row is not a continuous contact path, modifications would be required before the results could be applicable to an orchard situation.

With a multiple mode steering system, the decision making device can become very complex. As the microprocessor technology improves and the costs decrease, the use of a microprocessor-based controller as the decision maker

becomes more feasible. Due to the microprocessor's programmability, its possible use in diverse applications gives the processor a wide range of applications. By program control, the processor can execute complex control operations because of its ability to perform computations and logical operations. When comparing digital and analog computations, digital computations can be performed with a higher degree of precision because of the lack of drift which is common to analog circuits (4).

Due to the versatility of the microprocessor, El-Ibiary (5) reported on a self-adjusting microprocessor-based electrohydraulic position controller. The position control system consisted of a solenoid valve, hydraulic cylinder, microprocessor, and position encoder which provided a feedback signal to the microprocessor of the load position. By program control, the desired and actual load positions were input into the microprocessor, processed and then the processor activated a solenoid which caused a change in the load position. Sometimes, the actual and desired positions differed due to fluid leakage and compression caused by system changes in temperature, pressure, friction, and load. Since the cylinder position was dependent upon the time interval the solenoid valve was activated, the processor recognized the deviations between the desired and actual positions and altered the time interval of activating the solenoid valve. The self adjusting program considered the effect the changes in the system

parameters had on the load position and compensated for the error by either expanding or shortening the duration the valves were on the next time the same conditions existed. The microprocess would be an acceptable processing unit for automatic guidance, because the processor could calculate the desired path of the machine by using different sets of data.

3. OBJECTIVES

The overall objective of this project was to design and develop a microprocessor-based controller for an over-the-row apple harvester that would allow the driver to select any one of the five steering modes (see Figure 3):

- a) front only
- b) rear only
- c) four way
- d) crab
- e) automatic

The controller would recognize the steering mode selected and control the steering wheels accordingly. The specific objectives of the research project were:

1. Design the hardware required for the microprocessor-based controller including the interfacing of the present machine components.
2. Develop the steering algorithms for the controller which would perform the necessary control for a system.
3. Design a sensor that would detect the location of the tree in relationship to the centerline of the machine.

4. Evaluate the performance of the controller when compared to the system specifications:

- a) When using automatic steering on a straight and curved path with 3.04 m (10 ft.) tree spacing, the tree was to be within 15 cm (6.0 in.) of the machine's centerline 1.83 m (6.0 ft.) ahead of the rear wheels when traveling at 1.6 km/hr (1.0 mi/hr).
- b) The conventional operator steering modes performed by the controller should be a stable control of a steering system which uses solenoid valves as the steering actuators.

4. DESCRIPTION OF DESIGN

4.1 Description of Operation

A microprocessor-based steering controller was designed and used to steer an over-the-row apple harvester (Figure 1). The harvester was designed by the USDA at Michigan State University, and the harvester measured 5.96 m (19 ft.) long, 4.57 m (15 ft.) wide, and 3.67 m (12 ft.) high. Being an over-the-row type harvester, the machine had to be large enough to allow a tree to pass through the machine with clearance on all sides.

Because of the machine's large size, it was more conveniently driven and steered hydraulically. The steering system (Figure 2) was composed of a microprocessor-based controller, position sensors, solenoid valves, and hydraulic actuators. By using the Claas steering valve, the front wheels were steered either by turning the steering wheel or by activating one of the two solenoid valves. During all steering modes, rear wheel positioning was accomplished by the controller activating and deactivating the rear solenoid valves. The angular position of each wheel changed as the flow through the activated solenoid valve caused either an elongation or retraction of a pair of hydraulic linear

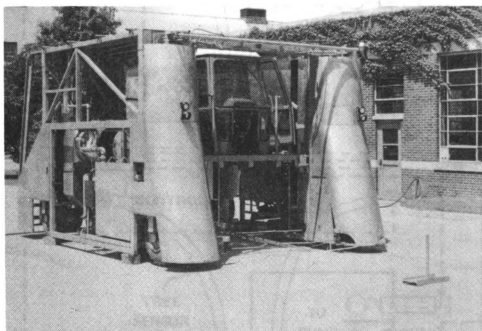


Figure 1: Front view of the over-the-row apple harvester on which the steering controller was implemented.

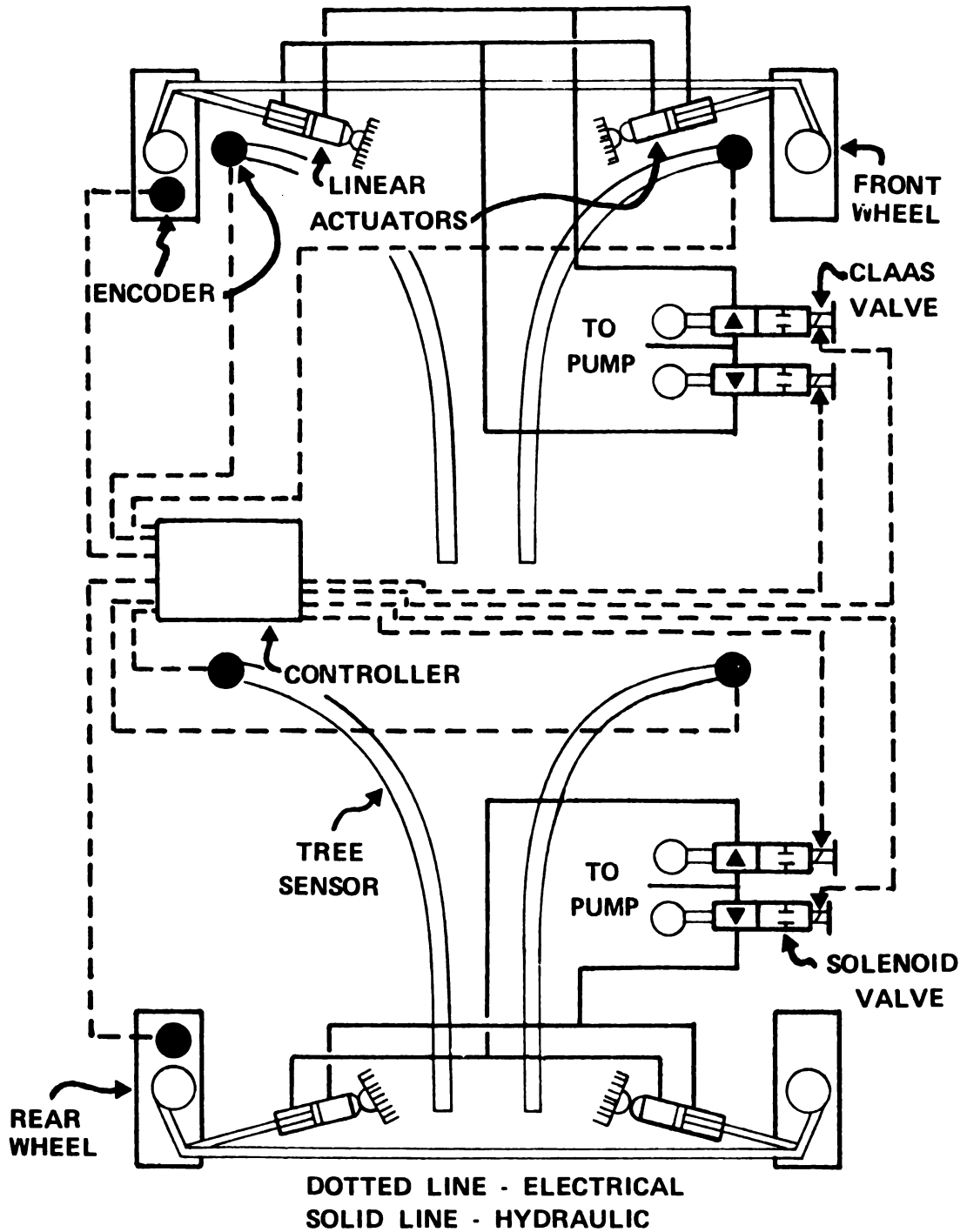


Figure 2: Schematic of the harvester's steering controller showing how the various components were interconnected.

actuators. To ensure that both wheels rotate equally, a tie rod connected each pair.

A microprocessor-based controller was the main processing unit which determined the direction of turn for the wheels. After recognizing the mode of steering selected by the operator, the processor selectively entered data from the various position sensors. The front and rear wheel position data was used for all steering modes; however, the tree sensor data was only needed during the automatic steering mode. The controller then processed the entered data. If a turn was required, the appropriate solenoid valve was activated causing the selected pair of wheels to turn.

4.2 Description of Steering Modes

By using a microprocessor as the processing unit in the steering controller, the operator was provided with a choice of five different steering modes. The four main steering modes (Figure 3) were front only, four way, crab, and rear only while the fifth mode was automatic steer (Figure 4). During the four main steering modes, the front wheels were positioned by the operator when he turned the steering wheel. The rear wheels' position was dependent upon the fronts' except during rear only where the rear position was determined by a position dial in the cab. In the front only mode (Figure 3a), the machine was steered only by positioning the front wheels; therefore, the rear wheels were kept straight such that $\theta_r = 0$. The four way mode (Figure 3b)

required the angular position of the rear wheels to be equal to but in the opposite direction of the front, $\theta_r = -\theta_f$. While in the crab mode (Figure 3c), the rear wheels were positioned to equal the front in both magnitude and direction, $\theta_r = \theta_f$. The rear only mode (Figure 3d) allowed the operator to select the position of the rear wheels regardless of the front wheels' position. The rear wheels were turned to the selected position such that $\theta_r = \theta_d$. When automatic steer (Figure 4) was selected, the front and rear wheel steering was dependent upon the location of the tree with respect to the two pairs of tree sensors. If the tree was not centered within the machine, a steering correction occurred and continued until the tree was centered; however, a steering correction did not occur until the tree sensors detected an off center tree.

4.3 Description of Sensors

The steering controller's sensors were used to detect the tree position with respect to the machine's centerline (Figure 5) and the angular position of the front and rear wheels (Figure 6). There were two pairs of tree sensors where one pair was used to detect the tree in the front half of the machine while the other pair sensed the tree in the rear half. Each tree sensor was composed of a spring loaded sensor arm and an absolute shaft encoder. When a tree pressed against the sensor arm, the arm pivoted at the pinned end, and the angular position of the bar was an output of a

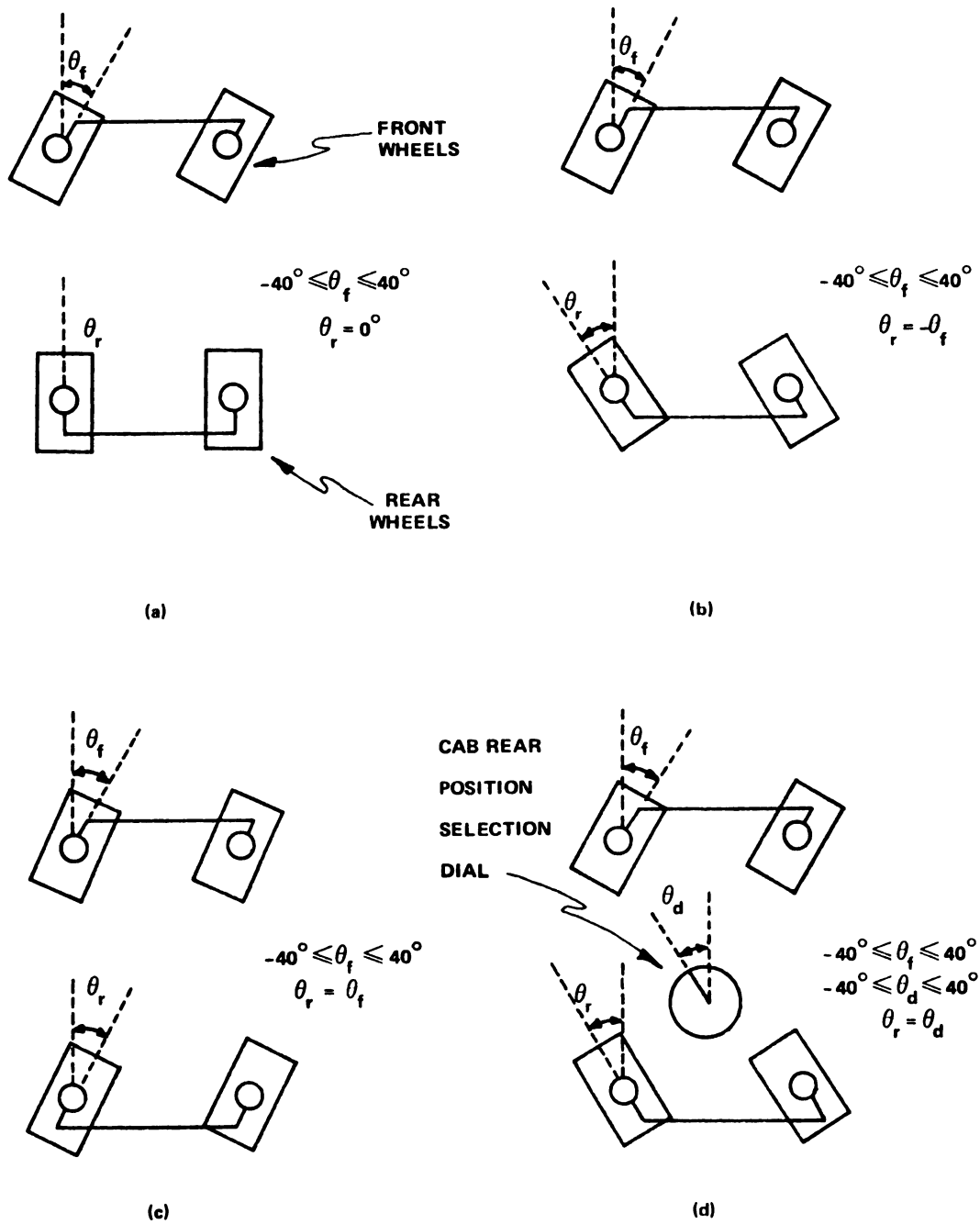


Figure 3. Diagram representing the four main steering modes:
 a) front only ($\theta_r = 0$); b) four way ($\theta_r = -\theta_f$);
 c) crab ($\theta_r = \theta_f$); and d) rear only ($\theta_r = \theta_d$).

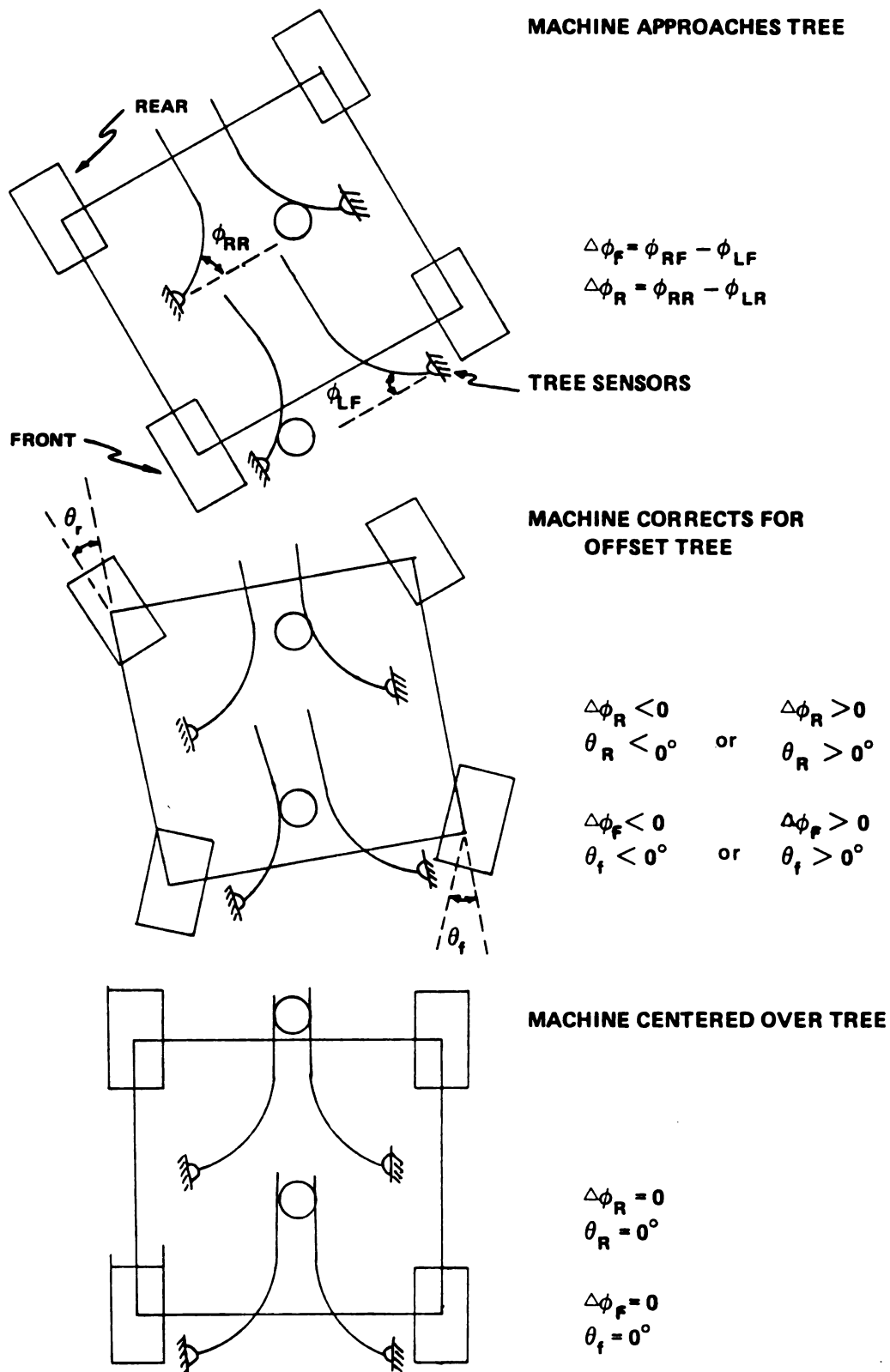


Figure 4: Diagram showing the sequence of events which occur when automatic steering is selected.

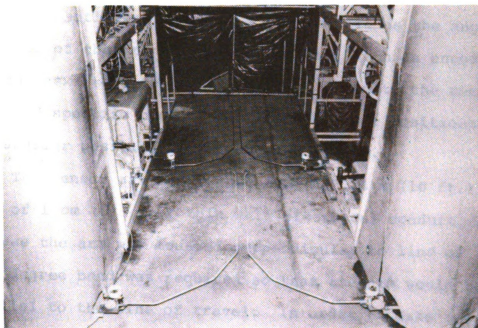


Figure 5: The four tree sensors used to detect the position of the tree with respect to the centerline of the harvester.

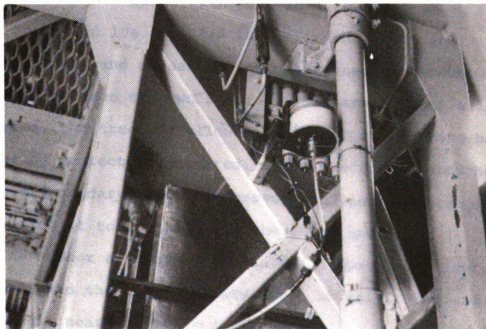


Figure 6: The absolute shaft encoders used to detect the angular wheel position.

shaft encoder. A shaft encoder was used to detect the angular position of the sensor arm as well as the angular position of the front and rear wheels. With an encoder mounted next to a front and rear wheel shaft, the encoder produced specific codes for different wheel positions as the angular position of the wheels changed.

The sensor arm was formed from a 3.04 m (10 ft.) section of 1 cm (0.5 in.) thin wall electrical conduit. Because the arm was mounted perpendicular to line of travel, a 90 degree bend was required so that the arm would be parallel to the line of travel. In order to make a gradual 90 degree change from the pivot end in the bar, bends were made at 29 cm (11.5 in.), 88 cm (34.7 in.), and 134 cm (52.7 in.) from the pivot end on the front arms. The rear arms had bends at 27 cm (10.5 in.), 65 cm (25.5 in.), and 104 cm (41.0 in.) from the pivot end. The degree of bend at each point determined the amount of free area between the two front or rear arms. With a large free area, the tree was allowed to be off center more before a steering correction occurred. When the tree approached the outer boundary, the tree pressed against the sensor arm causing it to rotate about the pivot end.

The sensor arm was attached to an apparatus (Figure 7) which allowed the arm to rotate about an axis. The end of the conduit nearest the bends was welded to a vertical 2 cm (1 in.) shaft. The shaft was mounted between two sets of bearings and perpendicular to the plane containing the

sensor arm so that the shaft rotated with the arm. To ensure that the sensor arm returned to the mechanical stop after being displaced, a spring was placed around the shaft and attached to the arm. In order to have available the amount the sensor arm rotated, a shaft encoder was mounted on the shaft of the arm.

Absolute shaft encoders were used to determine the angular position of the sensor arms, front wheels, and rear wheels. The digital encoder (Figure 8) was selected, because it could be interfaced directly to the microprocessor without having any analog-to-digital converters which would allow signal errors resulting from voltage drifts. The encoder consisted of a digital code and an optical sensor which read the code. The code was mounted around the inside of a 10 cm (4 in.) inner diameter polyvinyl chloride (PVC) pipe. The pipe section was attached to the mainframe of the machine so that the code remained stationary with respect to the machine. With the optical sensor attached to a 1 cm (0.5 in.) shaft which was connected to the sensor arm or wheel shaft, any rotation by the arm or wheel caused the optical sensor to rotate. Gears were chosen to connect the sensor arm shaft to the encoder shaft such that a 5 degree rotation of the sensor arm produced a 2.5 degree rotation of the optical sensor resulting in a 2 to 1 rotation ratio. For the wheel encoder, a 2 cm (0.9 in.) spool was attached to the encoder's shaft. A small cable was wrapped around the wheel shaft and the spool so that any rotation by the

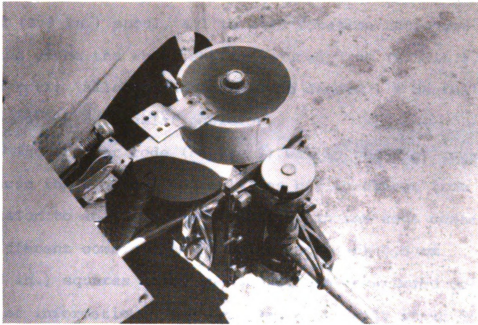


Figure 7: The apparatus that allowed the sensor arm to pivot about an axis.

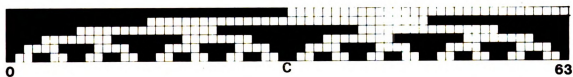


Figure 8: An absolute shaft encoder was used to determine the actual position of the wheels and sensor arms.

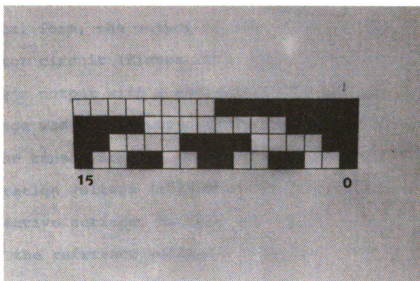
wheel produced a rotation in the encoder. By using the 2 cm (0.9 in.) spool, a 1 to 2.93 rotation ratio resulted since the wheel shaft measured 7 cm (2.7 in.). This ratio allowed the encoder to detect 1.8 degree changes in the wheels' position.

The encoder code (Figure 9) was in the Gray code form, because the code allowed only one bit to change from one position to another. Each position in the code contained a different combination of white and black 0.5 cm (0.2 in.) squares where one square was considered one bit of information. The code was originally drawn doubled its required size and then reduced photographically. By reducing the code photographically, the code was placed onto semi-gloss paper which allowed the white squares to be reflective to light and the black squares nonreflective. For the wheel encoders, each position in the code was formed by a column of six squares while the sensor arm encoder had a column of four squares. With a column of six bits of information, there were sixty-four different code combinations which allowed the processor to recognize 1.8 degree changes in the wheels. The sensor arm encoder had sixteen different code positions and detected 5 degree changes in the arm.

The optical sensor for the two wheel encoders each contained a column of six Fairchild FPA106 light reflection emitter/sensor arrays while the four sensor arm encoders each had a column of four. An infra-red emitting diode and a silicon npn phototransistor with an infra-red filter were



(a)



(b)

Figure 9: The Gray code used in the a) wheel and b) sensor arm encoders.

housed within the sensor array. The array measured 6.09 mm (0.24 in.) wide, 4.95 mm (0.19 in.) long, and 4.95 mm (0.19 in.) high. The infra-red filter on the phototransistor prevented visible light from entering the device; therefore, the effects of daylight on the switching of the transistor were eliminated. If a white square from the code was in view of the array, the infra-red light from the diode was reflected off the surface and caused the phototransistor to enter saturation. On the other hand, a black square absorbed the emitted light, and the phototransistor remained in cutoff.

In order to obtain a signal from the sensor array in digital form, the output of the array was an input to a comparator circuit (Figure 10). This circuit compared the array's output with a set reference voltage. The reference voltage was set between the two different voltage outputs of the sensor array. The output of the comparator reached saturation voltage (+5V) when the array was in view of a reflective surface, because the array's output was greater than the reference voltage. When the array was in view of a nonreflective surface, the input to the comparator from the array was less than the reference voltage causing the comparator's output to become zero (0V). Since the comparator's output could only have two possible states (0V or +5V) the output was in digital form where each comparator produced one binary bit of information. The digital information from the comparators was not easily usable by the

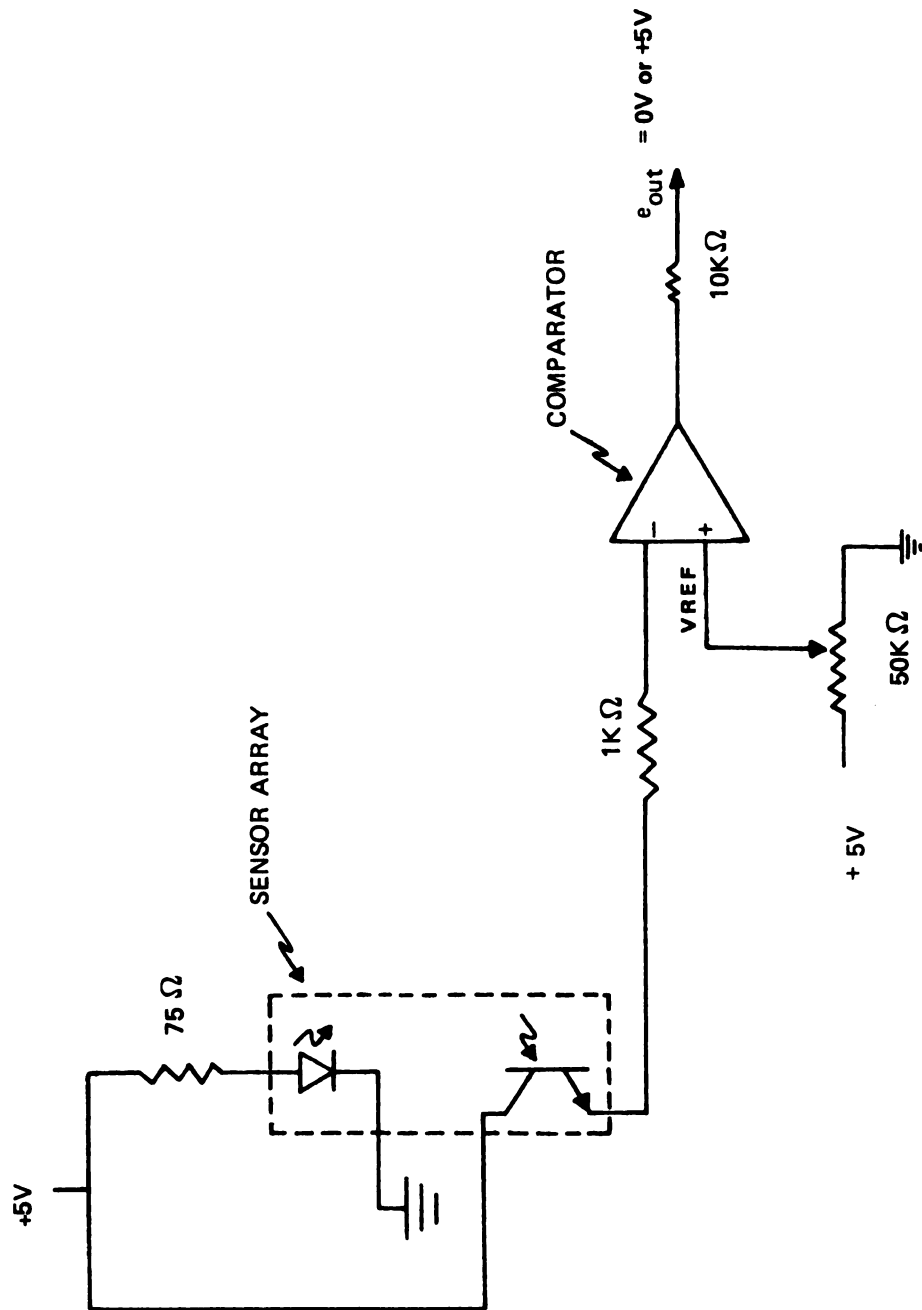


Figure 10: Circuit diagram of the comparator circuit used to convert the array into a digital form.

microprocessor, because the information was in the Gray code format. To obtain the code in natural binary, the Gray code was converted to the natural binary format by the use of exclusive-or gates (Figure 11).

4.4 Description of Hardware

The switching of the steering solenoid valves was accomplished by the hardware of the system. To determine which valves were activated, the hardware, microprocessor and several support chips, made the decision during the execution of the control programs. The steering controller hardware was subdivided into four sections: 1) input, 2) output, 3) memory, and 4) steering mode selection. By program control, the microprocessor determined the steering mode and then selectively input the data from the various position encoders into memory. After processing the entered data, the processor initiated a change in the wheels' position by outputting a specific code for a certain turn.

The controller input section (Figure 12) allowed only one of seven sets of data from the various sensors to be entered into the processor and memory at one time. This selection process was achieved by using a 3-to-8 digital decoder to decode the three input/output command lines (N0, N1, and N2) from the processor. When the processor executed any input or output instruction, the three command lines were set to a particular binary code. To distinguish between an input and output operation, the decoder was enabled by the

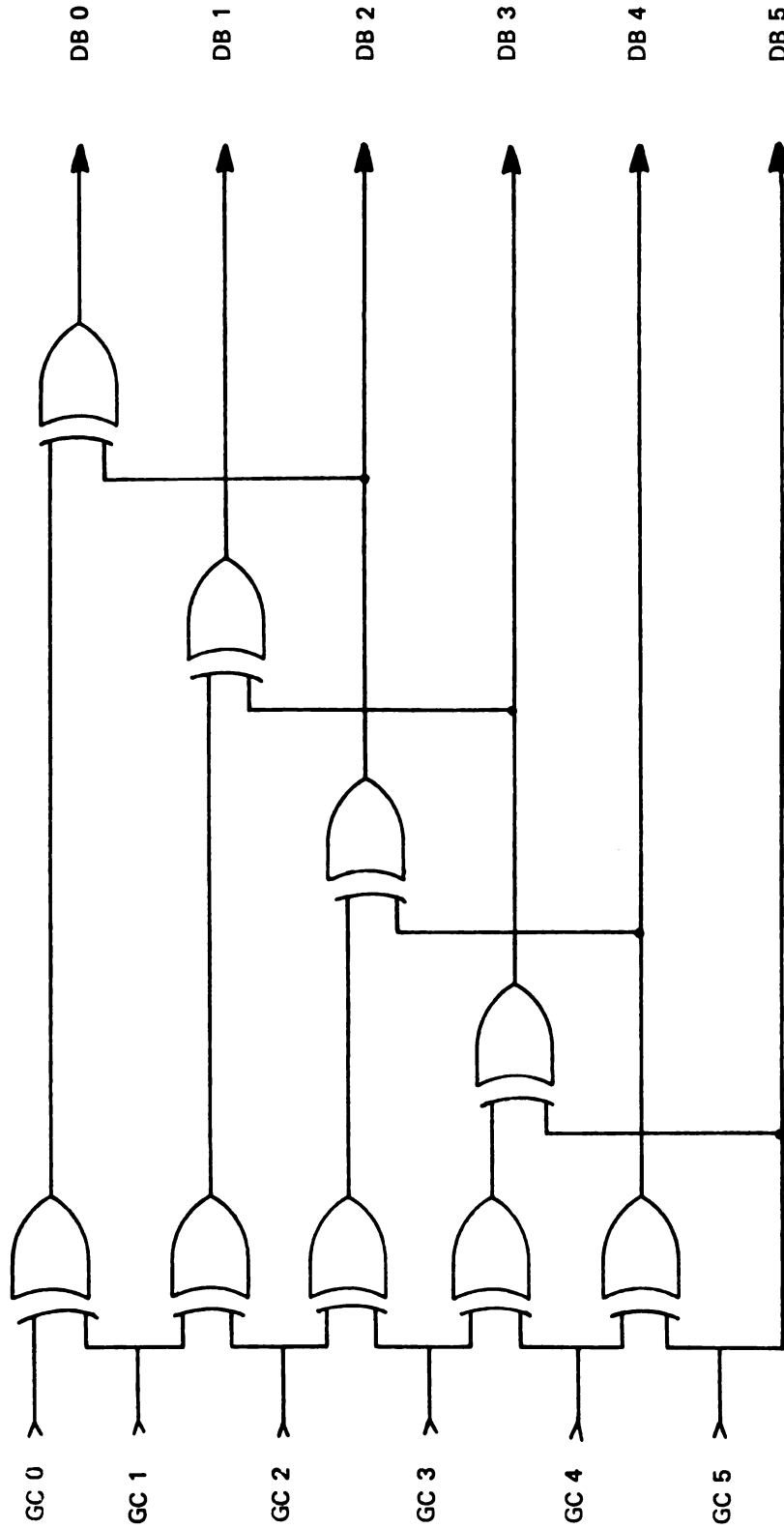


Figure 11: Circuit diagram used to convert a 6-bit Gray code into a 6-bit natural binary. To perform a 4-bit conversion, only the four most significant bits are used.

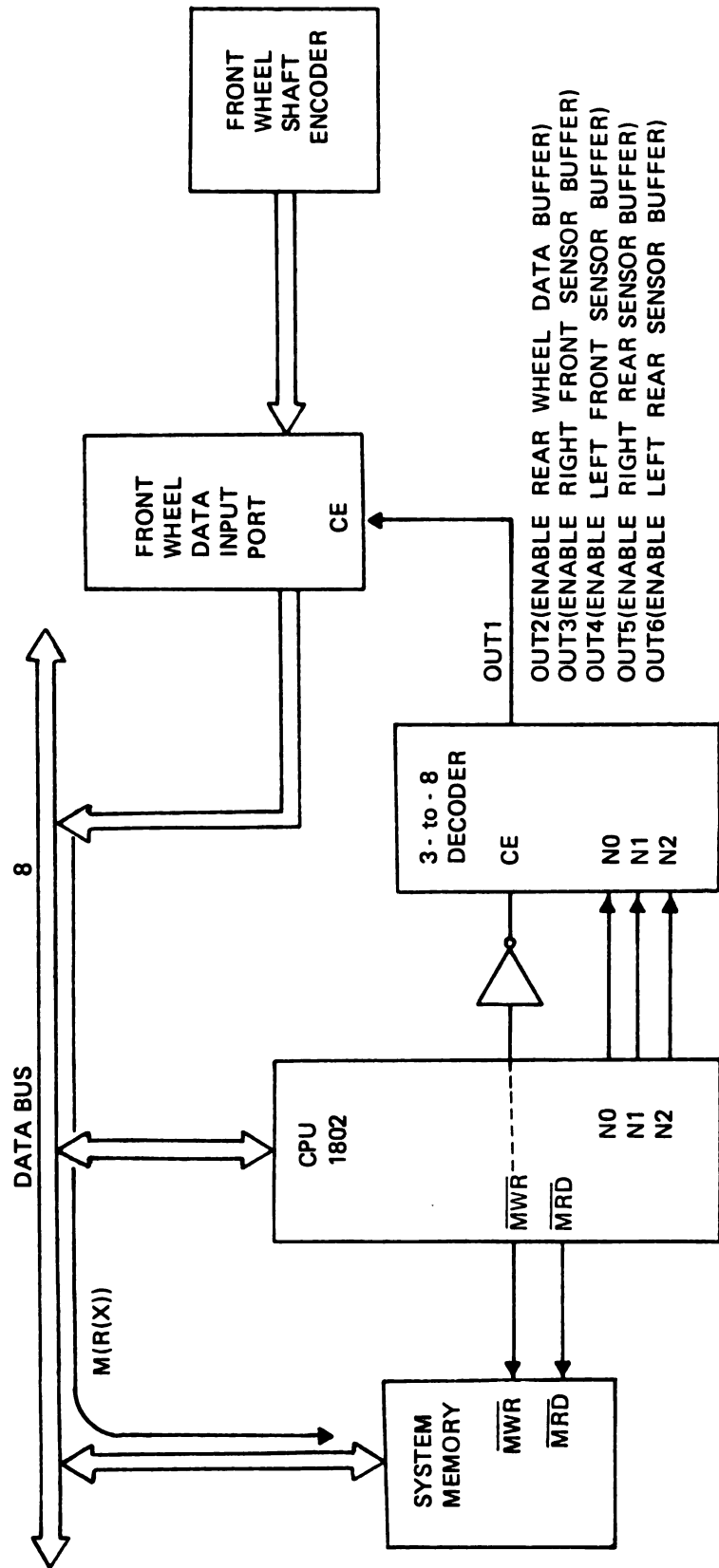


Figure 12: Diagram of the controller's section which selectively input data from the various sensors.

inverse of the memory write ($\overline{\text{MWR}}$) line which was low during an input. When the input decoder was enabled, the three command lines were decoded, and a pulse was output on a particular line of the decoder. The pulse from the decoder enabled the tri-state output of one of the seven input buffers. By enabling the buffer output, the data in the buffer was now on the data bus. With the information on the data bus, the data was put into the D register of the processor and written into memory at the location pointed to by the X register. If the data was needed at a later time, it was available in memory at a specific location.

If a turn was required after processing the new data, the controller output section (Figure 13) determined the direction of turn for the front and rear wheels. To distinguish between a front and rear correction, a 3-to-8 digital decoder was used to decode the three command lines from the microprocessor. If a front correction was needed, the binary code for 1 appeared on the three lines while binary code for 2 appeared for a rear correction. The decoder was enabled by the inverse of the $\overline{\text{MRD}}$ line which went true (OV) during an output instruction. During the execution of an output instruction, the outputs of the decoder, OUT 1A ($\overline{\text{N2}} \cdot \overline{\text{N1}} \cdot \overline{\text{NO}} \cdot \overline{\text{MRD}}$) and OUT2A ($\overline{\text{N2}} \cdot \text{N1} \cdot \overline{\text{NO}} \cdot \overline{\text{MRD}}$) dictated whether a front or a rear steering operation was needed by enabling one of two additional 3-to-8 digital decoders. The OUT1A pulse signaled for a front steering operation while OUT2A indicated a rear one.

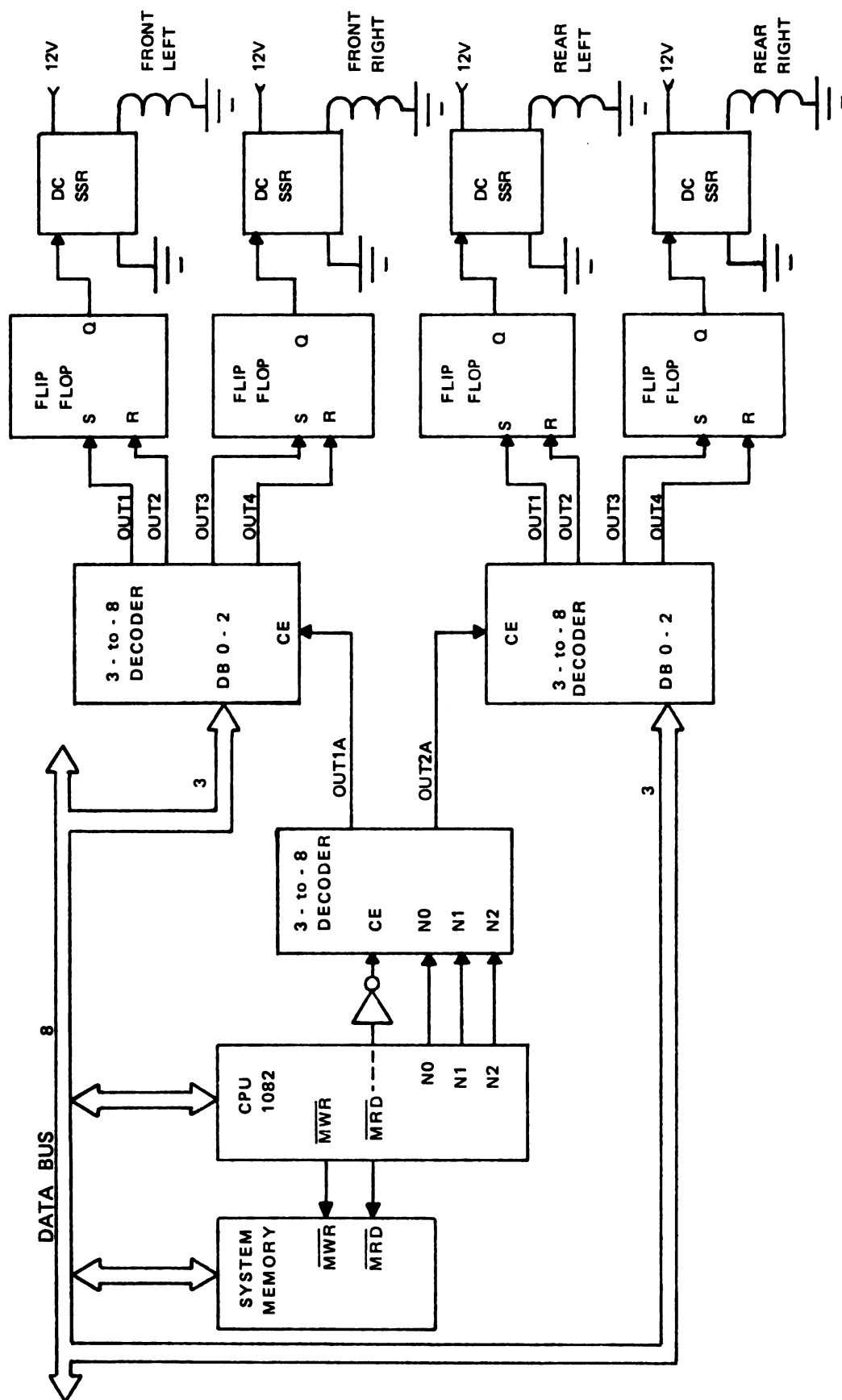


Figure 13: Diagram of the controller's output section which was responsible for the actual turning of the wheels.

When one of the two decoders was enabled by one of the OUT lines, the decoder decoded the contents on the data bus during the execution of the output instruction. The data originated from the memory location pointed to by the X register. This data contained a specific binary code such that after the enabled decoder decoded the data, a steering operation resulted. For instance, if the data consisted of a binary 1, a left turn resulted while a binary 2 stopped a left turn. The output pulse from the decoder was used to either set or reset a flip-flop. For example, the OUT1 pulse from the data decoder set a flip-flop while OUT2 reset the same flip-flop. By setting and resetting the flip-flop, a solid state relay (SSR) was activated and deactivated respectively. The switching of the SSR caused the switching of the hydraulic solenoid values.

The controller's memory section (Figure 14) was composed of a volatile and nonvolatile section. An erasable programmable read only memory (EPROM) was used for the nonvolatile memory and occupied the first 2K (2,048) bytes of the controller's memory. The EPROM contained the different steering control programs, because of the EPROM's ability to retain its contents when power was disconnected. With the data from the various sensors changing, the data was temporarily stored in the thirty-two (32) byte section of volatile or random access memory (RAM) which was located after the EPROM section of memory. If the RAM lost power, the contents of the memory were also lost. As the data

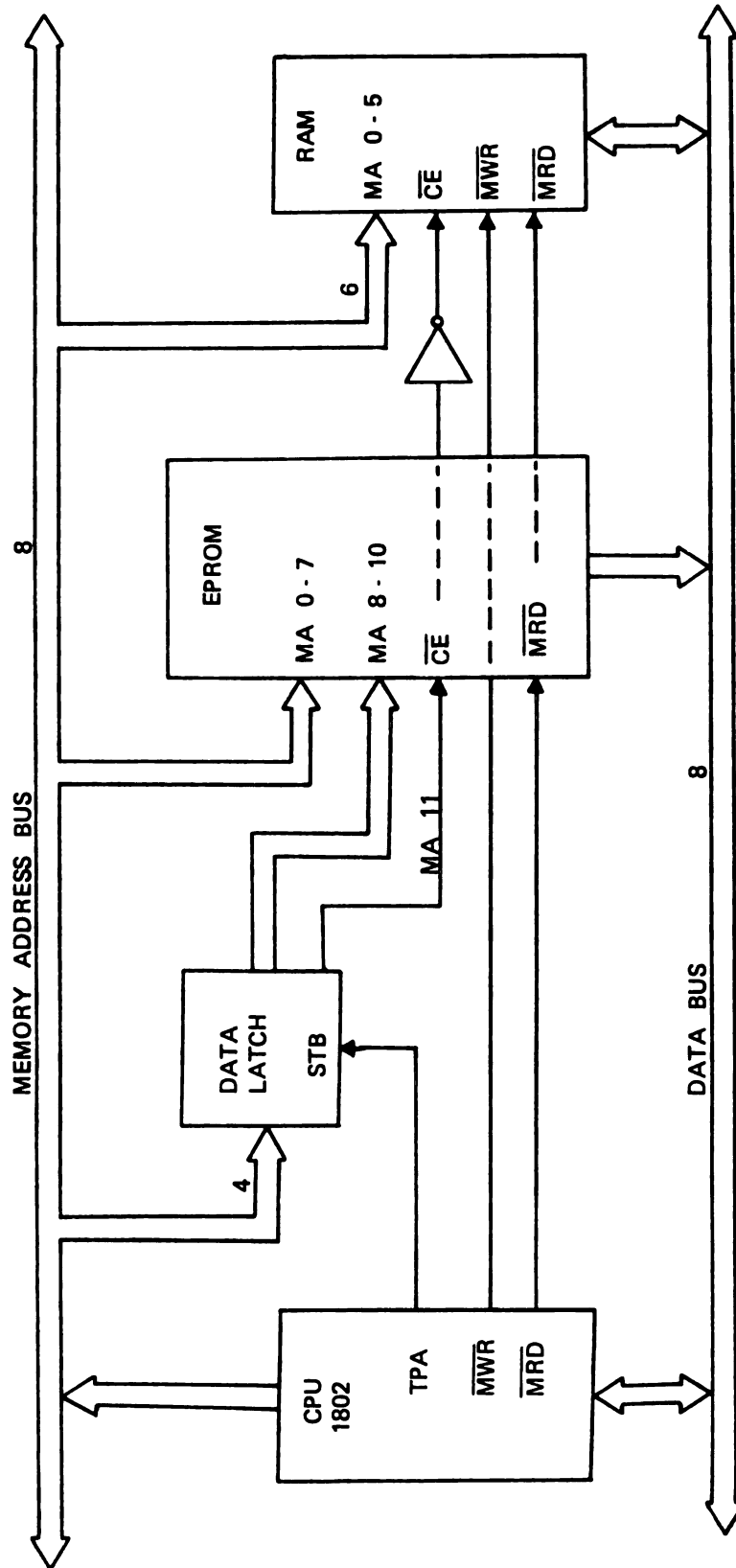


Figure 14: Diagram of the controller's memory section showing both volatile and non-volatile parts.

changed, the new data was written over the old or previous bytes of data.

The microprocessor addressed any one of 65K (65,536) memory locations. The sixteen (16) bit address was placed on the address bus in two eight(8) bit segments. The upper byte of the memory address was on the address bus during the first half of the machine cycle and strobed into a data latch by the processor's timing pulse TPA. Immediately following the TPA pulse, the lower byte of the address was put on the address bus which now gave a full sixteen (16) bit address.

The EPROM contained the hexadecimal memory locations 0000 to 07FF while the RAM memory locations were at 0800 to 081F. The proper memory section was selected by the twelfth address bit where a low bit selected the EPROM and a high, the RAM. With a low twelfth bit, the first eleven bits addressed a memory location in EPROM, and the contents of the addressed location were placed on the data bus during the $\overline{\text{MRD}}$ pulse. A high twelfth bit disabled the EPROM and enabled the RAM. When the RAM was enabled, the first six address bits addressed one of the thirty-two (32) RAM locations. If a $\overline{\text{MWR}}$ pulse occurred while the RAM was enabled, the contents on the data bus were written into the addressed memory location. During a $\overline{\text{MRD}}$ pulse, the contents of the addressed memory location in RAM or EPROM were placed on the data bus for use by the microprocessor and decoders.

The machine operator selected a steering mode by using

a dial (Figure 15) which was mounted in the cab. To select any mode except auto-steer, the driver simply turned the dial. By changing the dial position, a different status flag on the microprocessor was high (+5V). Since the same control subroutine was used for crab and rear only steer, the selection of rear only enabled the rear dial data buffer and disabled the front one. In the other positions, the front data buffer was enabled. To enter the auto-steer mode, a push-button was pushed. The pushbutton set a flip-flop whose output went high and was connected to one of the four status flags. The flip-flop was reset by a pulse either from a foot switch or from the Claas valve when the driver turned the steering wheel approximately 60 degrees.

4.5 Description of Software

The steering controller's software was composed of a main program and four different control subroutines: 1) auto-steer, 2) front only, 3) four way, and 4) crab and rear only. The software determined the selected steering mode and then controlled the position of the wheels accordingly.

The main program (Figure 16) was responsible for determining the selected steering mode. By testing the voltage state (0V or +5V) of the four input/output status flags, the main program detected the steering mode chosen by the driver. For the selected mode, a particular status flag was true or +5V. After testing the flags and determining the true flag, the main program branched to one

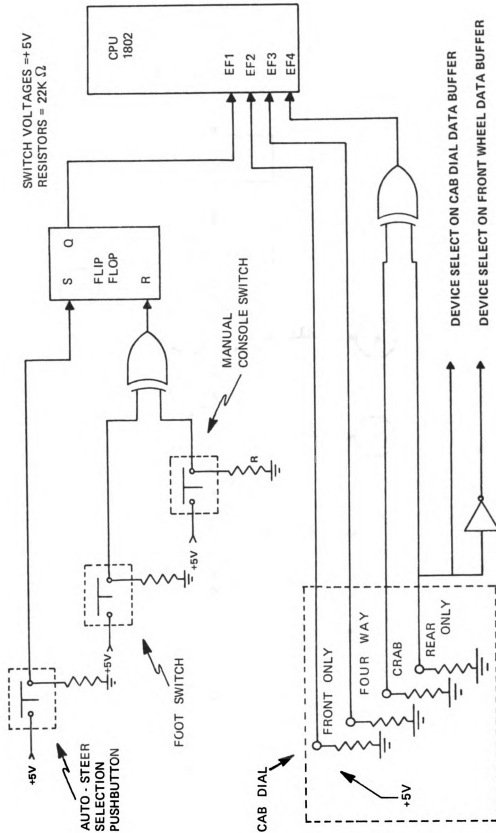


Figure 15: Schematic of the steering mode selection circuit.

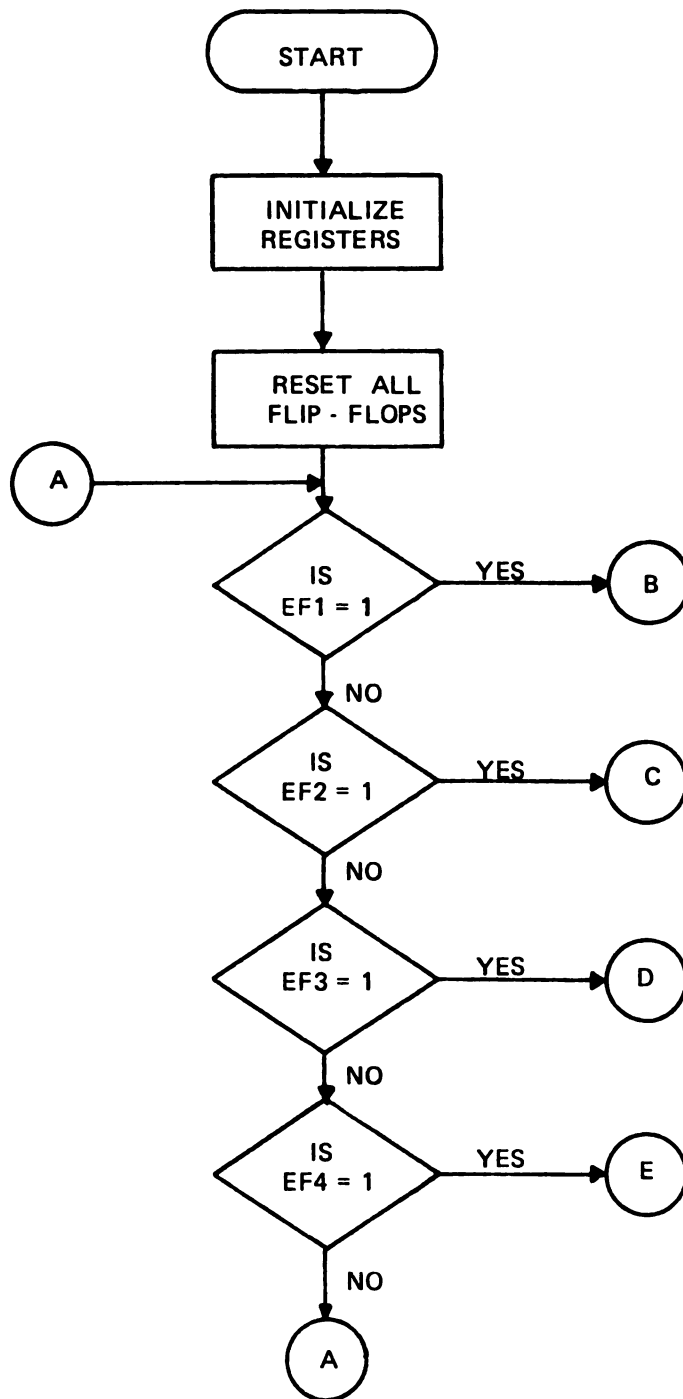


Figure 16: Flowchart of the main program used in the steering controller.

of the four steering control subroutines which performed the necessary control.

The centering of the machine over the tree row was controlled by the auto-steer subroutine (Figure 17) Before beginning the actual centering algorithm, the front and rear wheels were straightened. This was accomplished by entering the wheel position data into the processor, and then comparing the entered data with the center position code. With a 6-bit code, the center position was located at 32 or 20 hexadecimal. If the wheel position was less than 32, then a right turn was initiated and continued until the position was 32. On the otherhand, a left correction occurred when the position data was greater than the center code.

After straightening the wheels, any further steering corrections were dependent upon the data from the sensor arms. First, the data from the two front sensors were entered into the processor and compared. If the two sets of data were not equal, a front correction occurred if the result from the comparison was greater than the result from the previous comparison. The steering correction continued to occur until the difference between the two sensors decreased at which time the wheels were turned in the reverse direction which was toward center. As soon as the wheels were centered, the valves were deactivated. The control of the rear wheels was performed in the same manner as the front except the rear corrections were dependent upon the

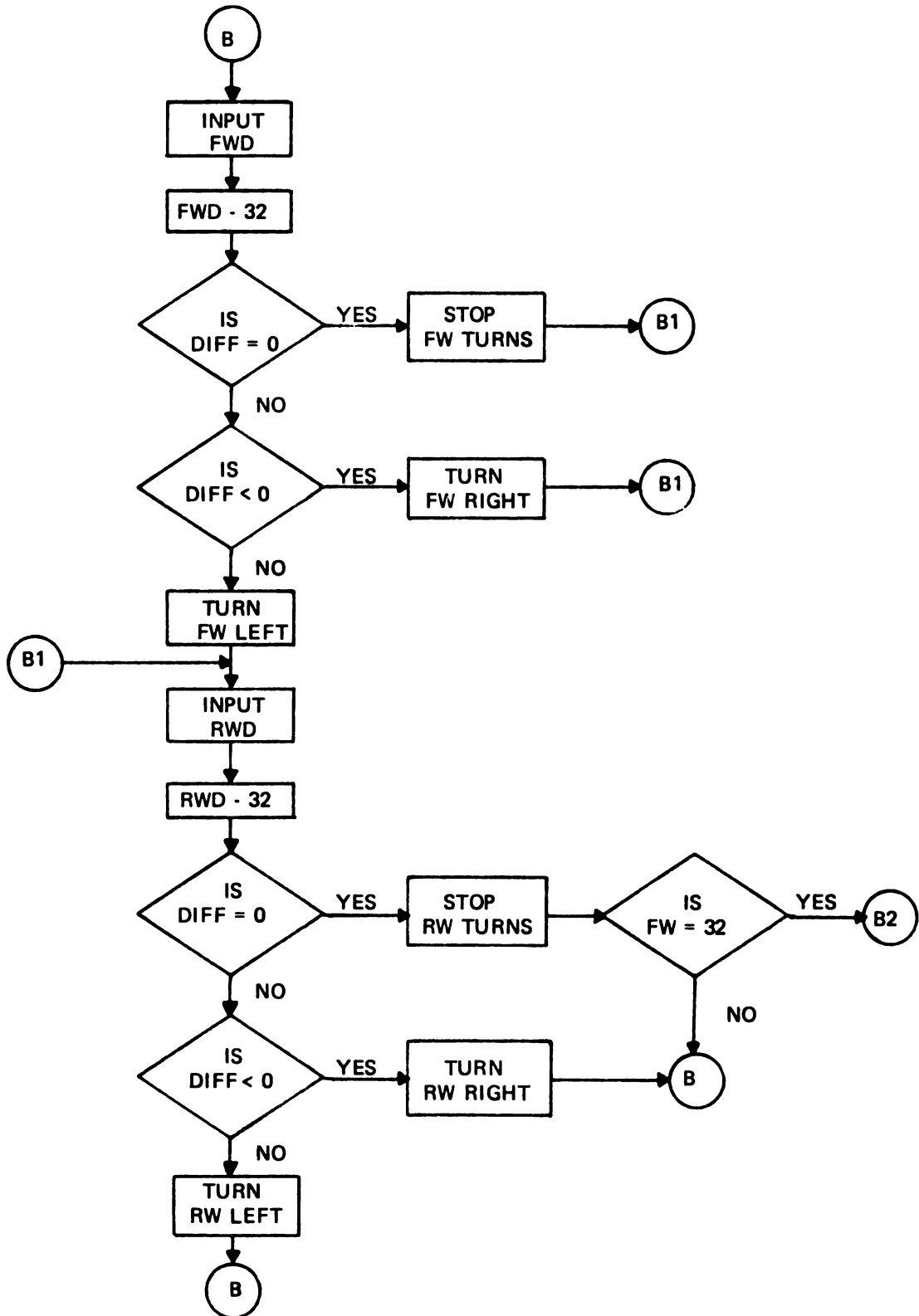
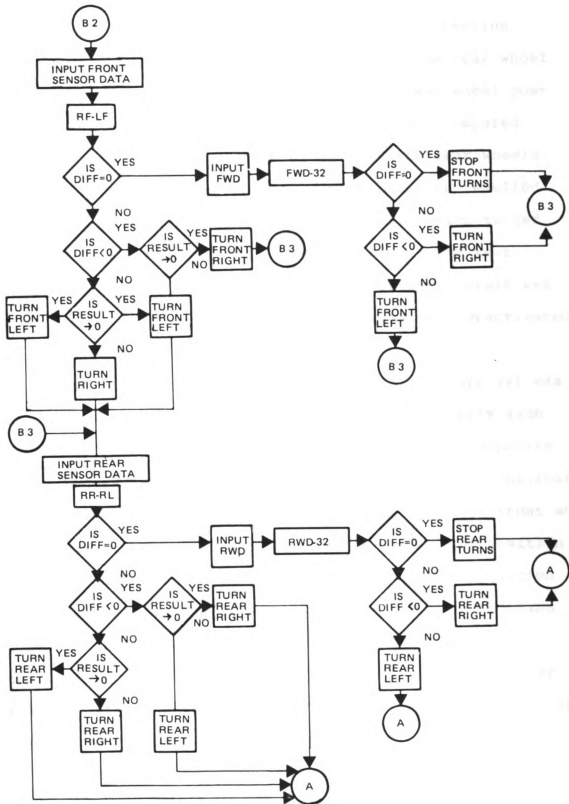


Figure 17: Flowchart of the auto-steer subroutine.

Figure 17 (continued)



two rear sensors.

During the front only mode, the machine was steered only by the front wheels; therefore, the rear wheels were kept at the center position. The front only steering algorithm (Figure 18) continuously checked the rear wheel position to assure proper placement. The rear wheel position data was entered into the processor and compared with the center position code of 32. If the rear wheels were to the left of center, a negative position resulted from the comparison, and the solenoid valve which turned the rear wheels to the right was activated. The rear wheels were turned to the left when a positive result was the outcome of the comparison. All valves were deactivated when the rear wheels were on center.

When the four way steering subroutine (Figure 19) was executed, the controller positioned the rear wheels such that they were turned the same amount but in the opposite direction as the front. The direction of turn for the rear wheels was dependent upon the location of both the front and rear wheels. By complementing the centered front position code and adding 1 to the complement, the correct position code for the rear wheels was obtained. The rear position code was then compared to the desired code to determine the direction of turn for the steering adjustment. After the comparison, a negative result caused a right turn while a positive result produced a left turn.

The crab and rear only subroutine (Figure 20)

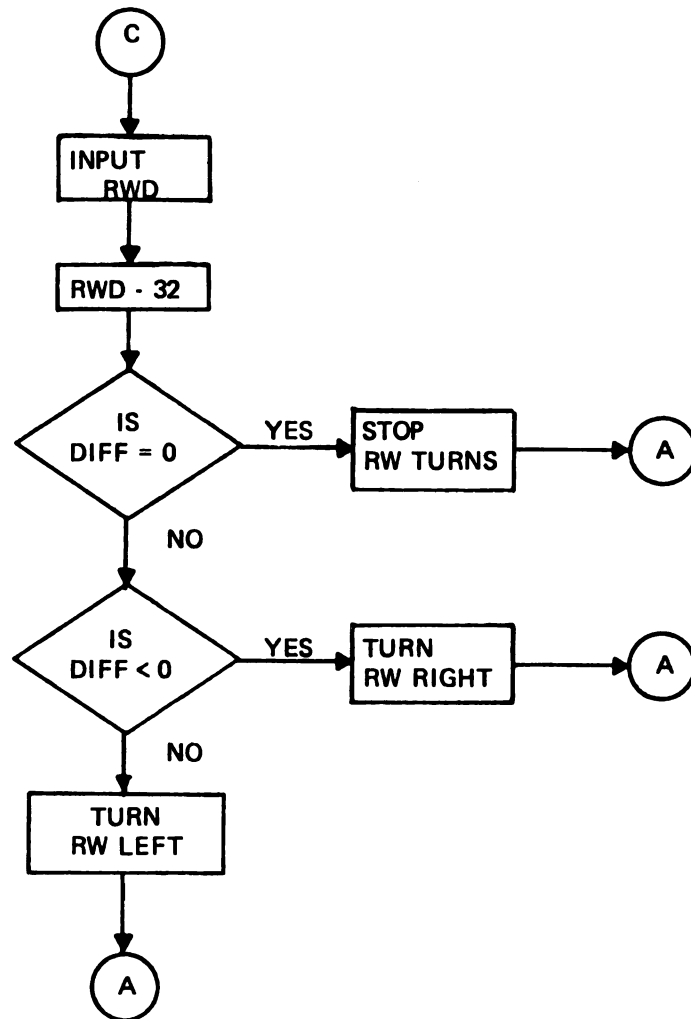


Figure 18: Flowchart of the front only subroutine.

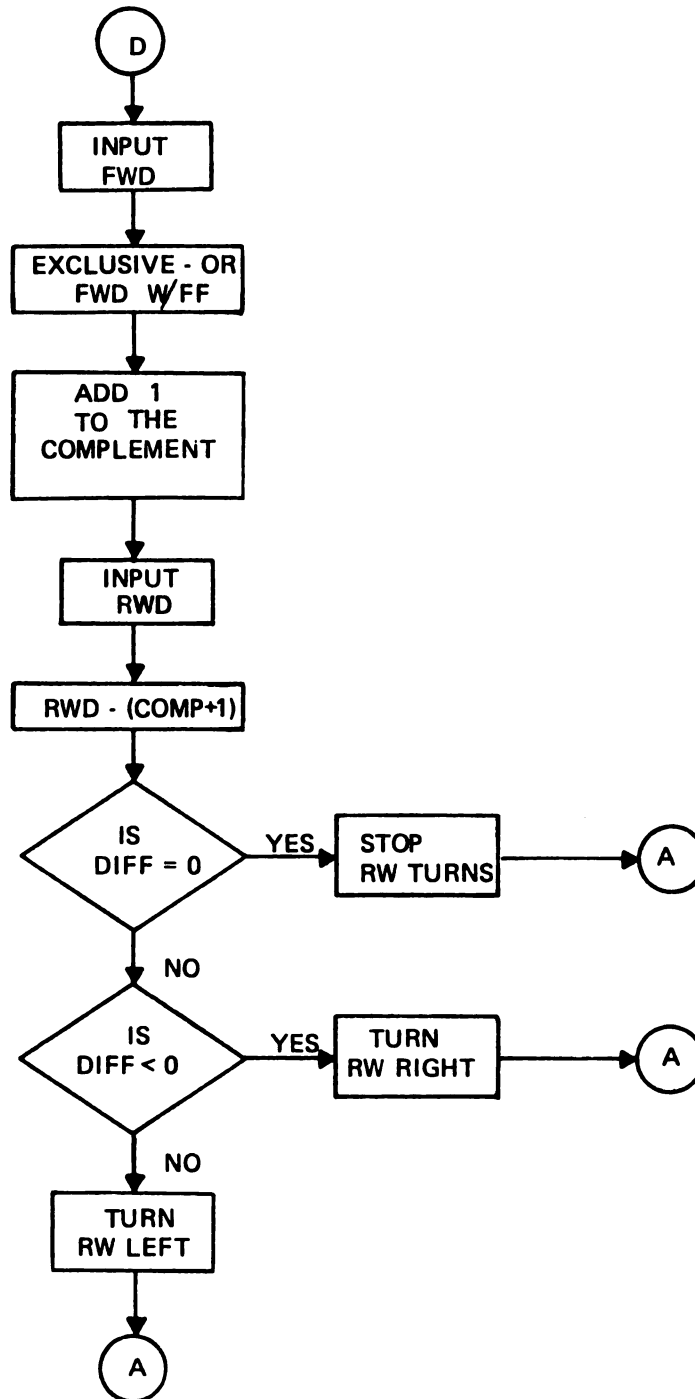


Figure 19: Flowchart of the four way subroutine.

performed the necessary control so that the rear wheel position was equal to the front or dial position depending upon whether crab or rear only was selected. The same routine was executed for both steering modes; however, the position data from the correct sensor was determined by hardware. With the desired position code entered into the processor, the entered data was compared with the rear position code. If the rear code was not equal to the desired position, a steering adjustment was initiated and continued until a zero difference between the two codes was obtained.

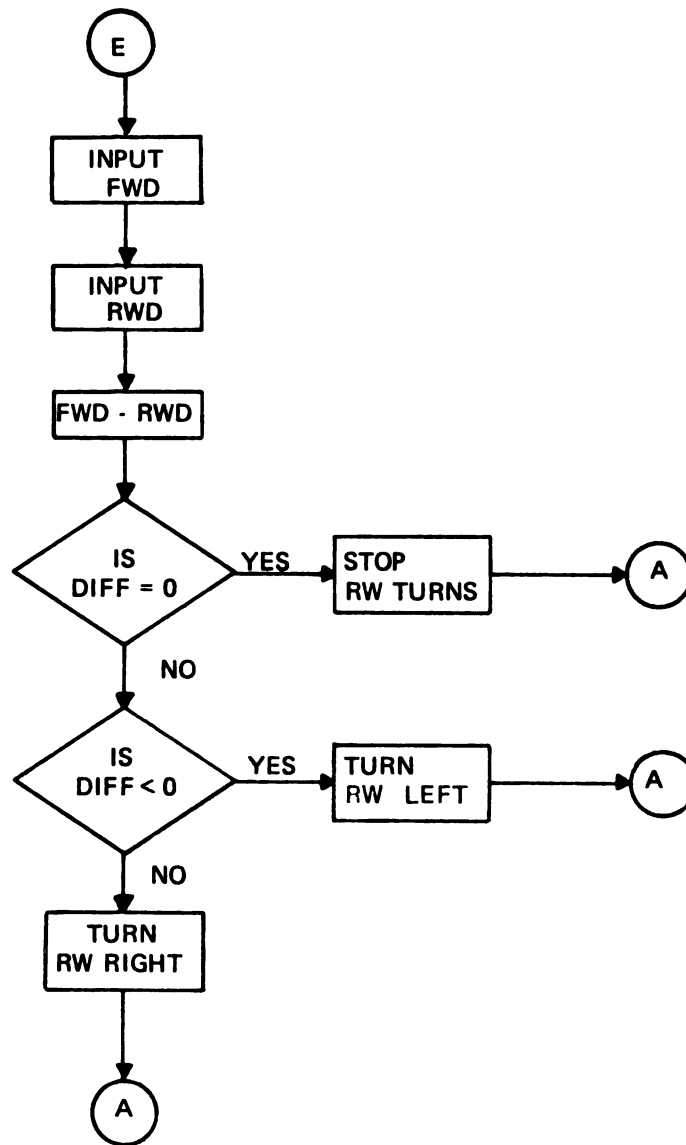


Figure 20: Flowchart of the crab and rear only subroutine.

5. TESTING AND MODIFICATIONS

Since the designed steering controller performed two different steering operations, the testing of the steering controller was divided into two separate phases. In the first stage, the different steering modes were examined to determine the ability of the rear wheels to follow the front. The second testing phase investigated the ability of the machine to remain centered over a tree while moving down the row.

5.1 Testing of the Steering Modes

The ability of the steering controller to position the rear wheels to a desired position was investigated for three steering modes: front only, four way, and crab. The rear only mode was not tested, because the rear only and crab modes used the same steering algorithm. To determine the wheels' position, a single turn 100K ohm potentiometer was mounted above the front and above the rear wheel encoders (Figure 21). The shafts of the encoder and potentiometer were connected together so that any rotation by the encoder caused an equal rotation by the potentiometer. The potentiometer output was an analog voltage which represented a



Figure 21: A single turn 100K ohm potentiometer was connected to the shaft of the wheel encoder. The output of the potentiometer was an analog voltage representing the angular position of the wheel.

particular wheel position.

The different steering mode tests were performed with the machine on blocks so that the wheels were not in contact with the ground. This set-up allowed the wheels to respond to steering corrections without any loading effects which could differ between the front and rear wheels. The angular velocity of the front wheels was variable and was dependent upon the rotation speed of the steering wheel. On the other hand, the rear wheels rotated at an almost constant rate, because the flow through the activated solenoid valves to the cylinders was constant. During the steering tests, the driver continuously steered the front wheels by turning the steering wheel at different rates. The rear wheels were positioned by the steering controller, and the position was dependent upon the selected steering mode and the front wheel position.

For comparing the front and rear wheel positions, the analog voltages from the two potentiometers were recorded and plotted. The two voltages were continuously recorded by a Racal 4D Instrumentation Recorder. At the end of the tests, the recorded voltages were input into a Century 460(GPO) Oscillograph. The oscillograph simultaneously plotted the two position voltages so that the front and rear wheel positions could be compared in time.

5.2 Testing of the Auto-Steer Mode

The auto-steer mode was examined to determine the ability of the machine to remain centered over a row. In order to determine the position of the machine while traveling, two chalk markers were attached to the frame of the machine. One marker was placed 44.5 cm (17.5 in.) ahead of the front wheels, while the other marker was located 53.34 cm (21.0 in.) ahead of the rear wheels. Both markers were 127.635 cm (50.25 in.) from the centerline of the machine. After a pass down the row, the distance from the marked path to a reference line was measured every 0.3048 m (1 ft.) along the path. The centerline of the machine was determined from the measurements and was plotted with the row so that the travel of the machine could be compared with the tree row.

An imitation tree row was set-up with tree stands (Figure 22) and was about 30.48 m (100 ft.) in length. The tree stands for the second run were formed by welding a 64 cm (25.5 in.) high 4 cm (1.6 in.) diameter pipe to an I-beam while the stand for the first run used 11 (4.5 in.) diameter pipe. The stands were placed 3.048 m (10 ft.) apart for both the straight and curved paths. The straight path (Figure 22a) was formed by placing the stands in alignment with each other. To produce an offset in the path for the curved path (Figure 22b) the third, fourth, fifth, sixth, seventh, and eighth stands were set 20 cm (8.0 in.), 35 cm (14.0 in.), 50 cm (20.0 in.), 50 cm

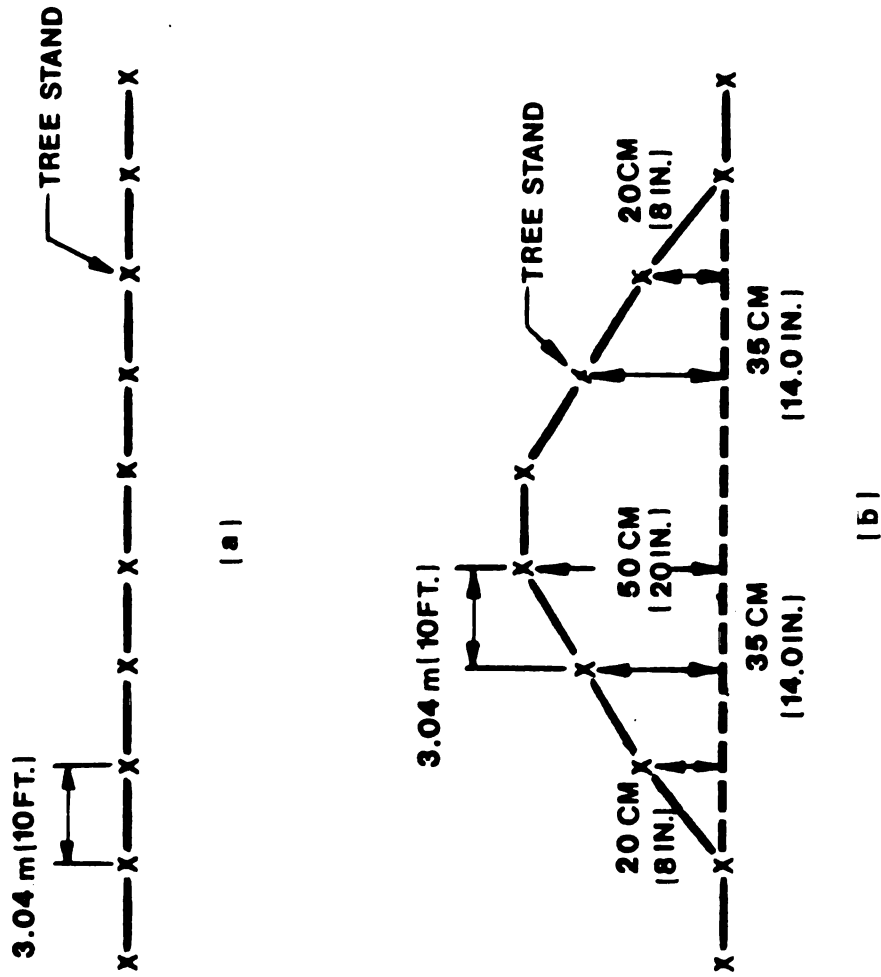


Figure 22: Diagram showing the placement of the tree stands in the (a) straight and (b) curved paths. These two paths were used to test the ability of the controllers to keep the machine centered over the row.

(20.0 in.), 35 cm (14.0 in.), and 20 cm (8.0 in.) respectively to the left of the centerline of the straight course. This curved path was used to examine the controller's ability to keep the machine centered over the tree stands. Since the auto-steer mode would be used mostly during the harvesting operation, all runs were made at approximately 1.6 km/hr (1.0 mi./hr).

5.3 Modifications

During the preliminary testing of the original design, some modifications were required to produce an acceptable steering controller. When passing over the tree stands, the machine under corrected for an off center stand. The under correcting was mainly due to the tree sensor design. As the offset stand passed through the machine, the sensor arm deflection decreased even when the stand remained the same amount off center. The decreasing deflection of the sensor arm caused the code to also decrease which gave incorrect information to the steering controller; therefore, the tree sensors needed to be altered.

To help eliminate the undershooting caused by the code automatically decreasing, a sensor arm was designed so that the code remained constant for a constant offset. This uniformity along the sensor was accomplished by using a parallel four-bar linkage mechanism. The sensor arm (Figure 23) consisted of three sections of 1 cm (0.5 in.) conduit. Two sections were mounted parallel to each other and

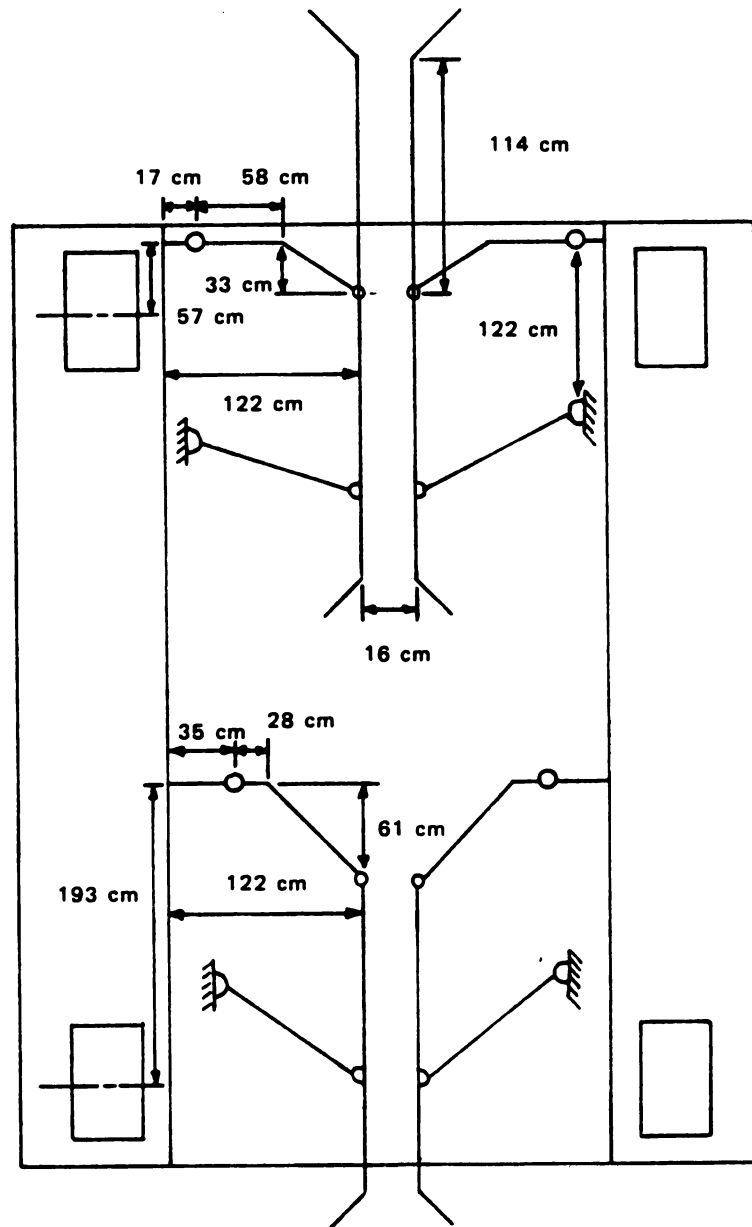


Figure 23: The new sensor arm used with the new auto-steer algorithm which produced a different code for each 2.54 cm (1 in.) offset.

1.21 m (4.0 ft.) apart. One of the two sections was attached to the pivot point while the other section was pinned to a bracket on the main frame. Both sections were placed at a 45° angle toward the center of the machine. The third section was pinned to the two free ends and was parallel to the center of the machine. The distance between the two sensor arms was 16 cm (6.5 in.) and could be varied by changing the angle of the two parallel bars. By using the parallel linkage, a different code occurred for each 2 cm (1.0 in.) offset.

The testing of the parallel linkage design proved that a software modification was now needed in the controller. With the initial auto-steer algorithm and parallel linkage, the machine did not make a complete steering correction. The machine corrected until the result between the two sensors started to decrease at which time the wheels were started back toward their center position. The result slowly decreased, but never reached zero. Since the difference between the two sensor arms never reached zero until the stand left the sensor arm area, the wheels passed by their center position causing the machine centerline to move away from the row. This movement made steering corrections very difficult, because the amount the machine needed to correct kept increasing from stand to stand. To eliminate the wheels from passing through center, two steering algorithms were written and tested.

The first algorithm alteration was made by

adding a few steps to check the wheel position as the wheels approached their center position. The wheel position check was used to insure that the wheels never passed by the center position when the result continued to decrease. If the result never reached zero, the wheels were stopped two codes short of the center position. This allowed the machine to make an additional correction until the result was zero. When the result reached zero, the wheels were centered without causing much of an overshoot.

The second change involved using a new steering routine for the main centering algorithm and extending the front sensor arms. The new program (Figure 24) calculated the desired wheel position by adding or subtracting the difference between the two sensor arms and the center position code for the wheels. The actual wheel position was then compared to the calculated one. If the two positions were not equal, a steering correction was initiated. The front sensor arm extenders were added so that the tree could be detected 111 cm (44 in.) ahead of the front sensor pivot point.

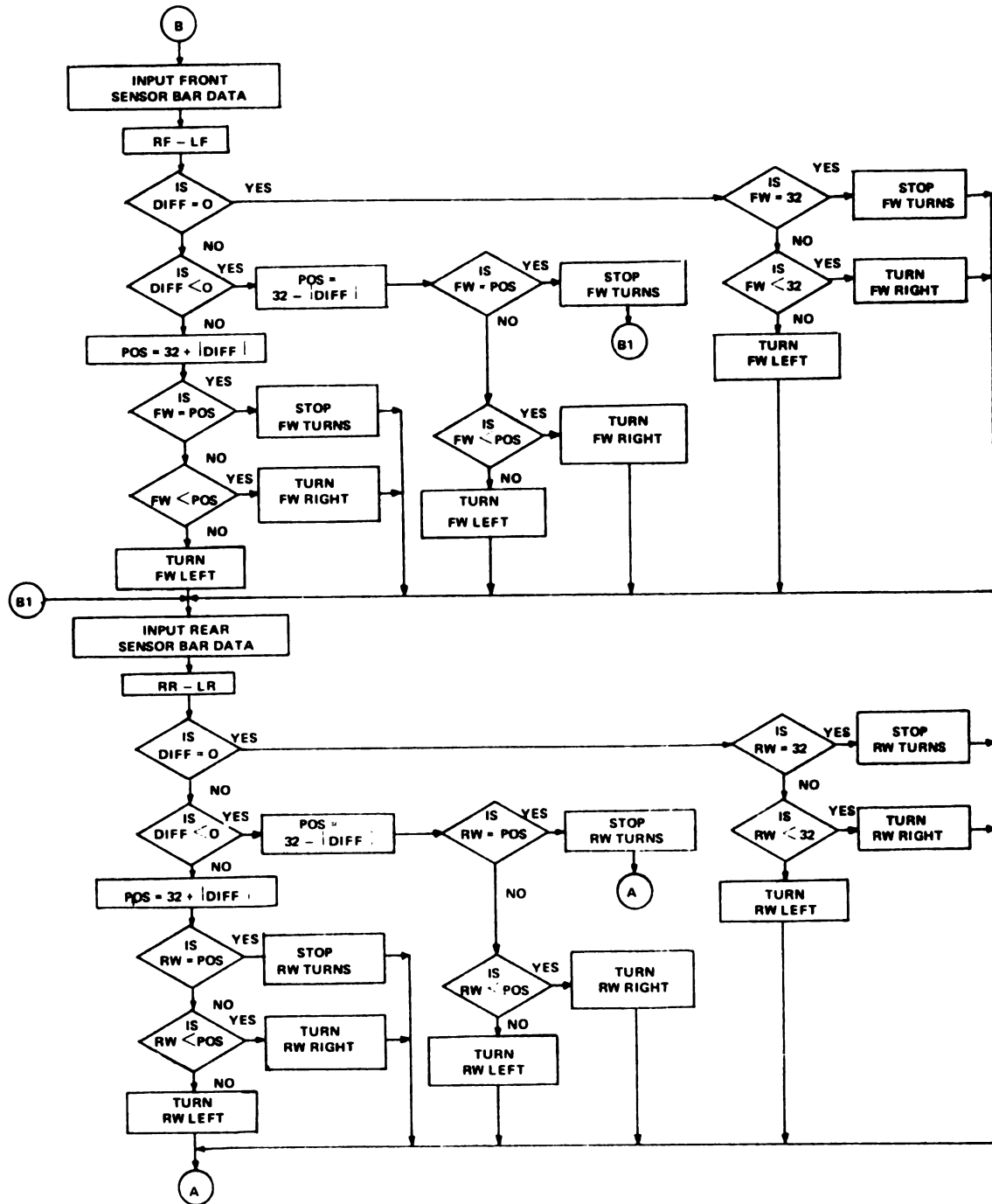


Figure 24: Flowchart of the new auto-steer algorithm which centered the machine by proportional control.

6. RESULTS

This section discusses the results obtained from testing the steering controller. The first part discusses the results obtained from the steering mode tests where the ability of the controller to position the rear wheels according to the position calculated by the processor. The second part evaluates the performance of the auto-steer algorithm as the machine passed over a straight and then a curved path.

6.1 Steering Mode Results

Since the front wheels were steered at different speeds, the rear wheel response to a front wheel change in position varied. The rear wheel response was observed as the front wheels made an 80° swing at a fast (Figure 25a) and then a slow (Figure 25b) angular velocity. The two responses showed that there was a varying time lag between the movements of the front and rear wheels. The front wheels' angular velocity varied up to $16.6^{\circ}/\text{sec}$. while the rear rotated at a constant $15.3^{\circ}/\text{sec}$. At the maximum angular velocity, the difference of $1.3^{\circ}/\text{sec}$. caused a 0.2 second time lag between the front and rear wheels when a 80°

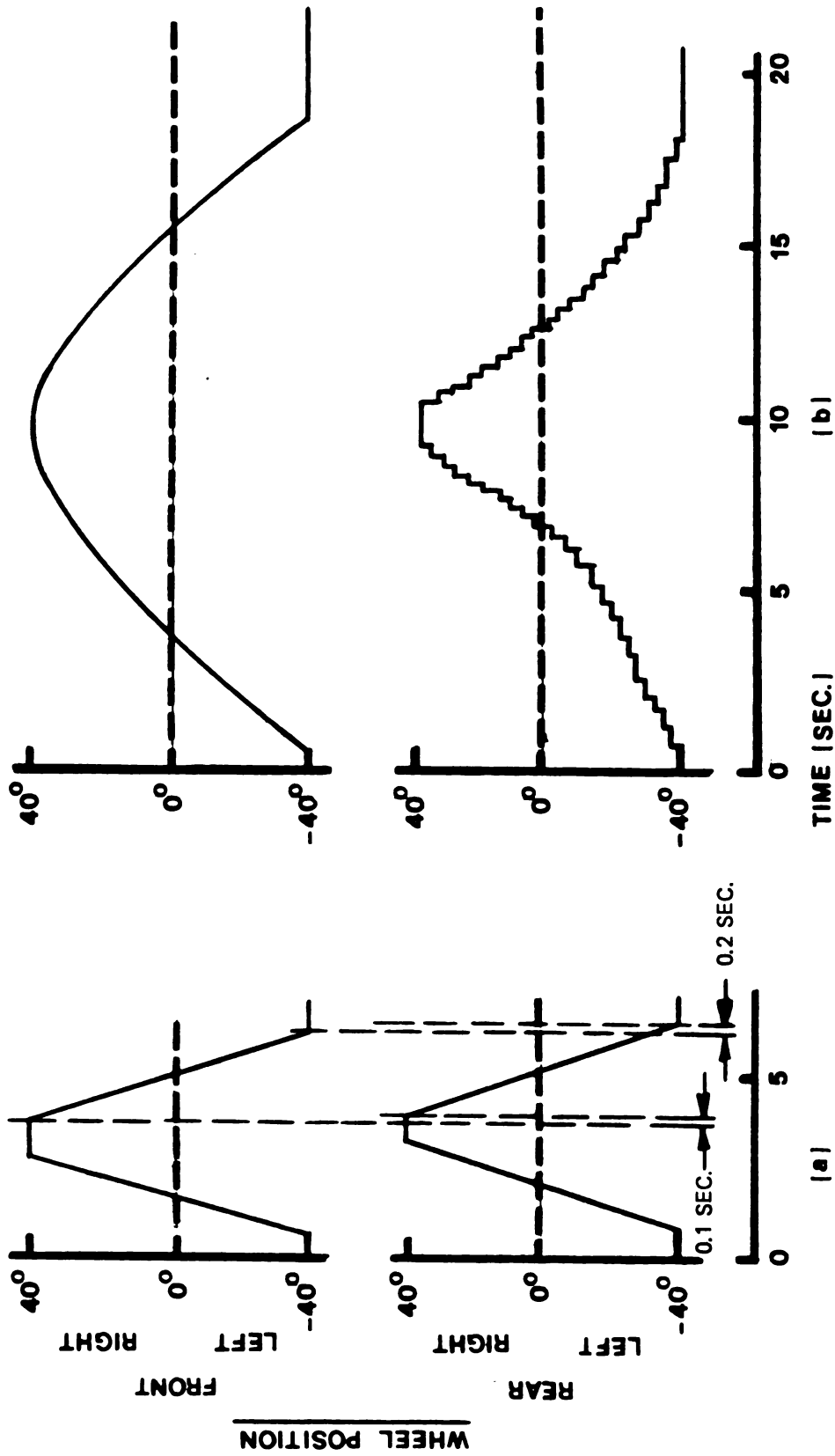


Figure 25: The front and rear wheel steering responses when the steering was rotated at a) fast and b) slow speeds. The front wheels were controlled by the movements of the steering wheel while the rear wheels were positioned by the controller.

swing was made by the wheels.

Not only was there a time lag due to the different angular velocities, but a time lag was also caused by a delay in the rear wheel movement. As the front wheels started to change positions, there was a minimum of a 0.1 second pause before the rear wheels began to correct to match the front. The time delay was due to the time required for the front wheel code to change from the code which was equal to the rear's and was dependent upon the angular velocity of the front wheels.

The rear wheel response to different steering patterns was dependent upon the angular velocity of the front wheels. At the beginning of any steering sequence, the rear wheels lagged the front by 1 code or 1.8° . This lag increased or decreased as the steering operation progressed. If the front wheels rotated at a velocity greater than the rear (Figure 25a), the lag increased, because the rear was unable to keep up with the front. The lag did not increase when the angular velocities of the front and rear wheels were equal. When the front wheels rotated slower than the rear (Figure 25b), the rear reached the next code and stopped during the entire rotation, because the rear wheels rotated at a faster speed and reached a code equal to the front's before the front had time to change to the next code.

The results from the three steering mode tests (Figure 26, 27, and 28) showed the controller's ability to position the rear wheels according to the mode selected. During the

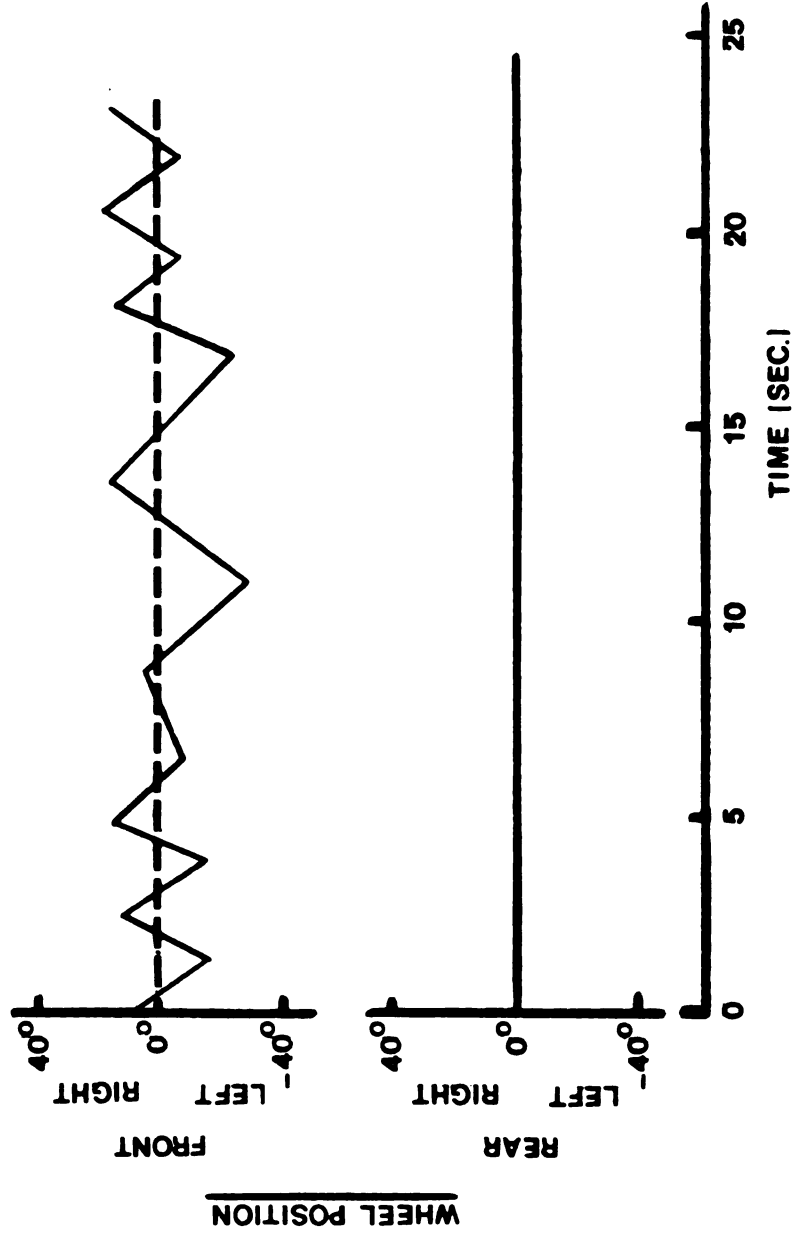


Figure 26: The rear wheel response to changes in the front wheel position during the front only steering mode. The front wheels were steered by the operator, and the rear wheels were controlled by the steering controller.

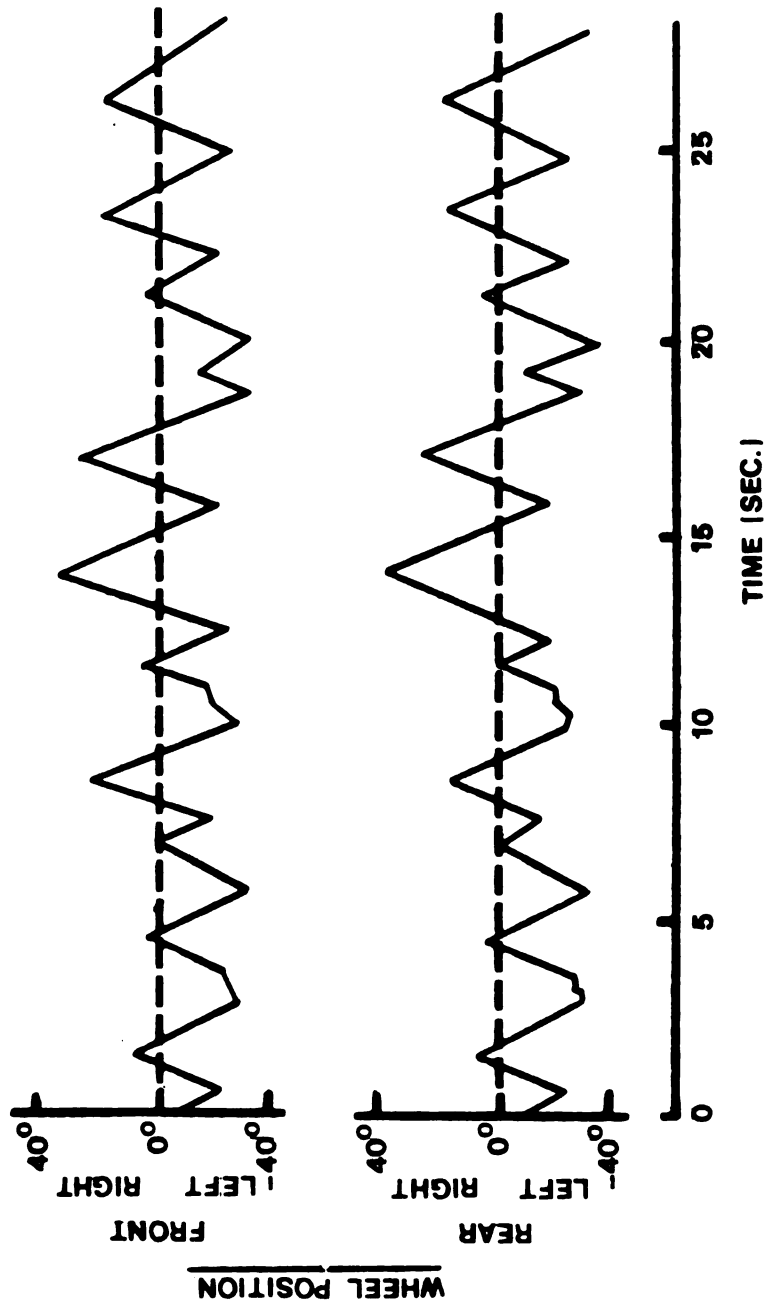


Figure 27: The rear wheel response during the crab steering mode while the front wheels were steered by the operator.

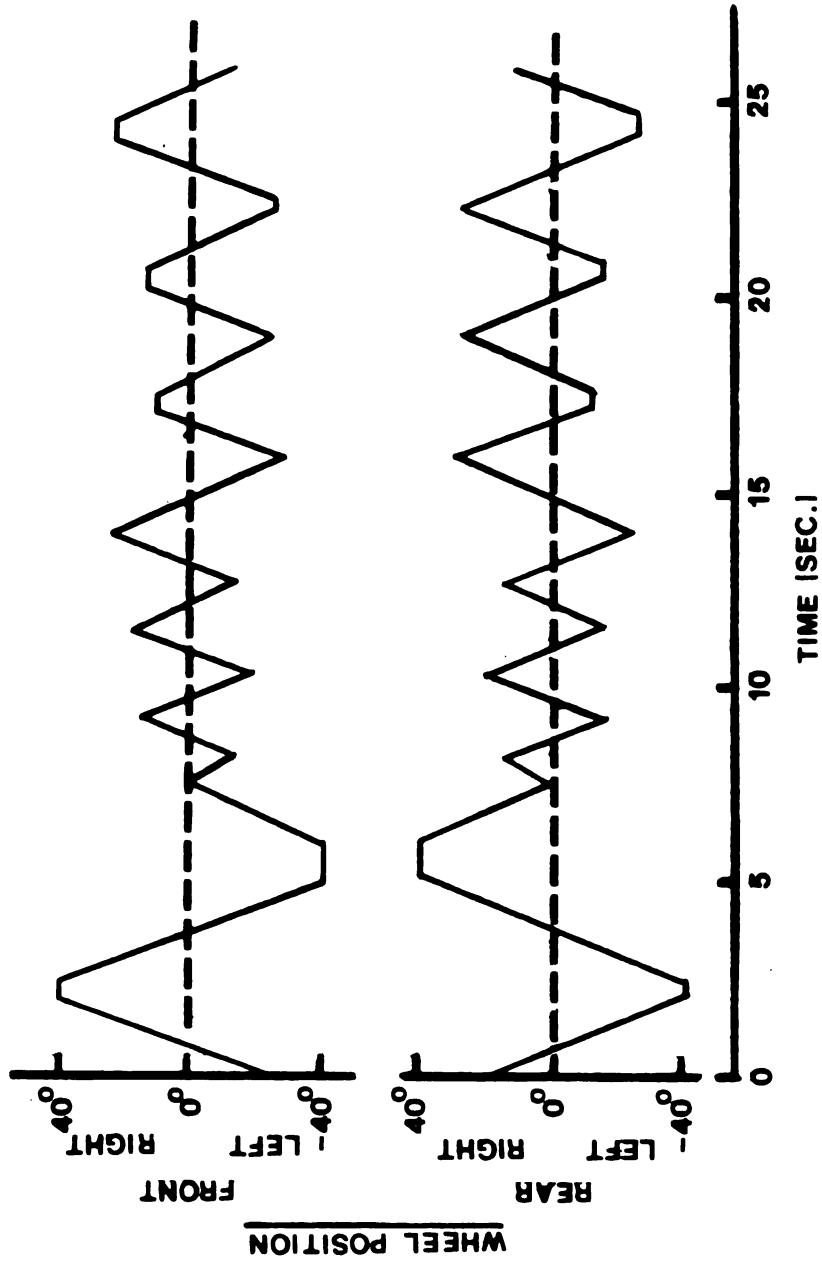


Figure 28: The rear wheel response to changes in the front wheel position during the four way steering mode. The front wheels were controlled by the operator, and the rear wheels were positioned by the controller.

front only mode (Figure 26), the controller kept the rear wheels in the center position regardless of the front position. The rear wheels closely followed the front during the crab steering mode (Figure 27); however, the rear wheels did lag behind the front. While in the four way steering mode (Figure 28), the rear wheels were turned the same amount as the front except in the opposite direction. Both steering modes showed the controller's ability to position the rear wheels without rear wheel oscillations. During the crab and four way modes, the lag between the front and rear was dependent upon the angular velocity of the front wheels. The different angular velocities was due to the hydraulic system and not the controller. Although a time lag was present, the controller did perform the necessary steering control which was needed during the different steering modes.

6.2 Auto Steer Results

The altered version of the original auto-steer algorithm successfully controlled the movements of the machine so that the machine followed the straight (Figure 29) and curved (Figure 30) paths. During each pass down the course, the machine oscillated along the desired line of travel. This oscillation was caused by the stand pressing against the opposite sensor arm at the end of each steering correction. Since there was a 16 cm (6.5 in.) span between the sensor arms and a 11 cm (4.5 in.) diameter stand, a 5 cm (2.0 in.) over correction by the machine allowed the stand to be detected by the other sensor arm. Each steering

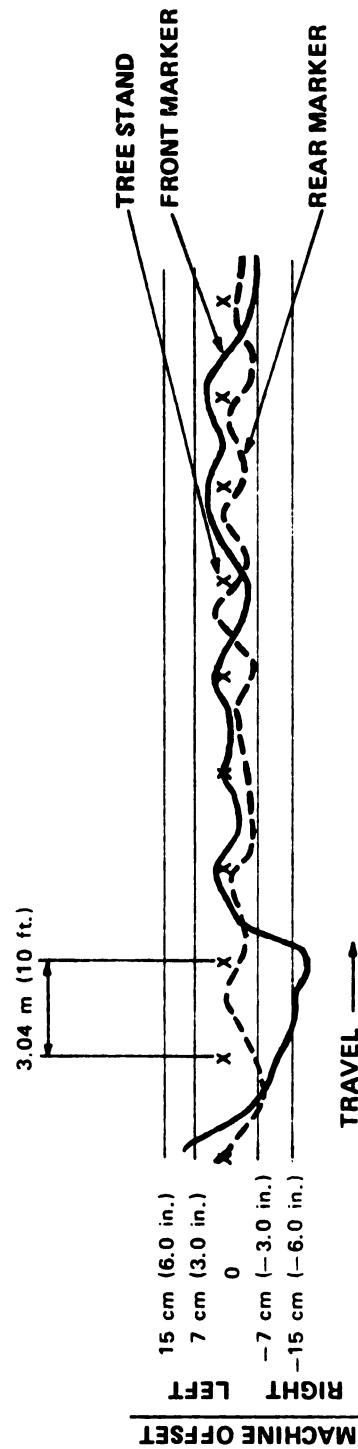


Figure 29: Diagram showing the centerline path of the front and rear of the over-the-row apple harvester while traveling over a straight row. The altered auto-steer algorithm was used to perform the necessary control.

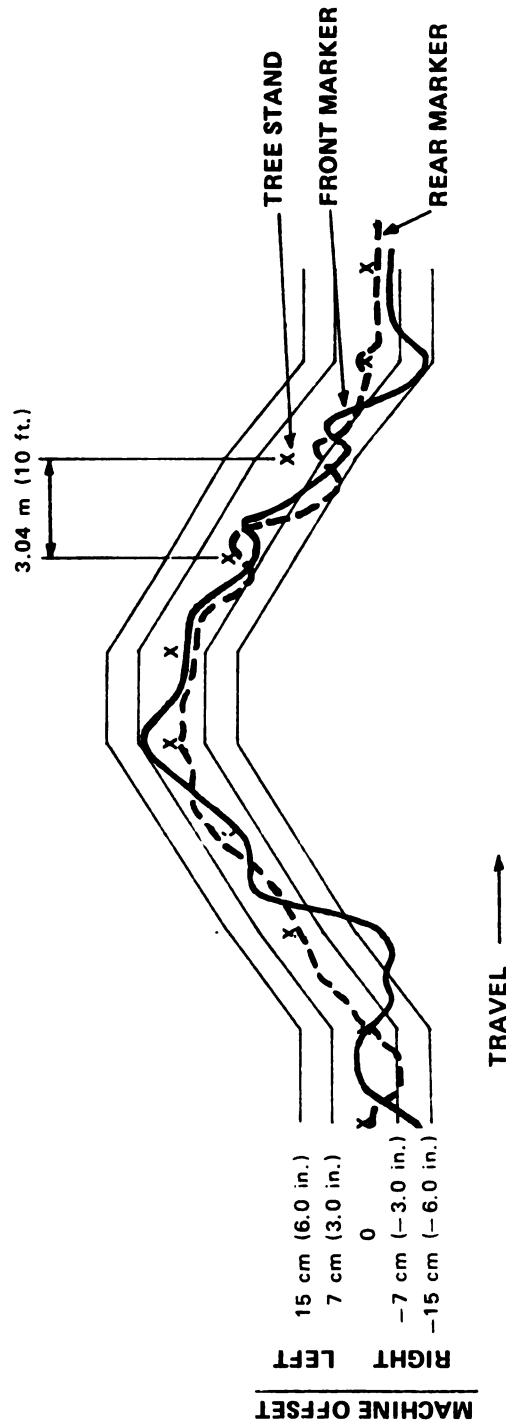


Figure 30: Diagram showing the centerline path of the front and rear of the over-the-row apple harvester while traveling over a curved path. The altered auto-steer algorithm was used to perform the necessary control.

correction usually over corrected 2 cm (1 in.) to 3 cm (2.0 in.).

During the two passes, the rear marker was closer to the desired path than was the front's. The front marker varied up to 25 cm (10 in.) from the tree stand as it entered the front sensor arms while the rear marker was kept within 5 cm (2.0 in.) of the tree stands. One reason for this difference was the mounting locations for the markers in relation to the sensor arms. For the front, the marker was located near the tree stand entrance of the sensor arms and marked the location of the machine as it approached a tree. On the other hand, the rear marker was located where the tree stand needed to be very close to the centerline of the machine. The front marker marked the machine centerline before a steering correction occurred while the rear marked the location near the end of a rear correction. When a 25 cm (10 in.) offset occurred, the machine traveled 0.91 m (3 ft.) at 1.6 km/hr (1.0 mi./hr.) before becoming centered.

With the new auto-steer algorithm and the front sensor arms extended, the controller successfully maneuvered the machine over the straight (Figure 31) and curved (Figure 32) paths and was able to keep the machine's centerline within 6 cm (2.5 in.) of the desired position without much oscillation in the steering corrections. This was due to the large gap between the sensor arms and the tree stand. By having a large gap, the machine could over correct without

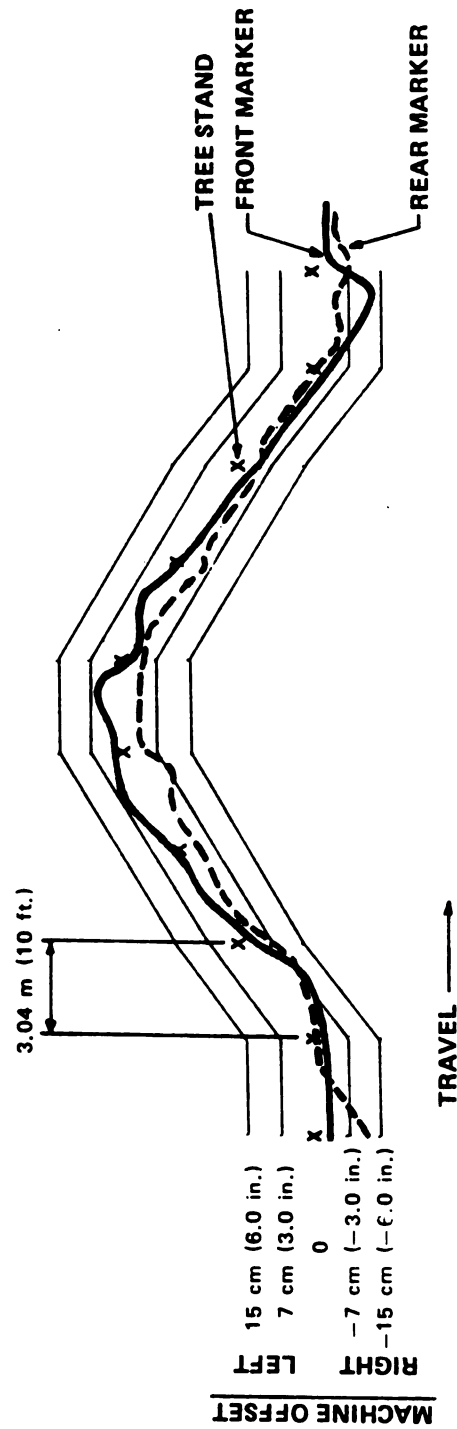


Figure 32: Diagram showing the centerline path of the front and rear of the over-the-row apple harvester while traveling over a curved row. The new auto-steer algorithm which positioned the wheels according to the amount the machine was off center was used to perform the centering tasks.

the tree stand depressing the other sensor arm. By using extender arms on the front sensors, the front started to correct for the offset before reaching the tree which allowed additional time for the front to react to large offsets. With this additional correction time, the front corrections put the machine very close to the tree stand which allowed the rear to correct for only small offsets.

Both auto-steer algorithms performed the necessary control to keep the tree within 15 cm (6.0 in.) of the machine's centerline when passing over a straight and curved row. Several passes were made over each row so that the machine was positioned differently at the start of each run. During each run, the offset tree was detected, and the machine successfully corrected for the offset.

By using a microprocessor-based control unit, the multi-steering mode system designed was simplified resulting in a lower cost control system.

7. SUMMARY AND CONCLUSIONS

The testing results show that a microprocessor-based steering controller can provide the necessary control for a multi-steering mode system. During the three steering mode tests, 1) front only, 2) four way, and 3) crab, the controller positioned the rear wheels according to the selected steering mode and the front wheel position. The auto-steer mode successfully kept the machine centered over the tree row, and was able to reposition the machine when an offset stand was encountered.

The microprocessor-based steering controller successfully maneuvered the rear wheels during the three steering modes. The rear wheels were positioned very closely to the desired position which was dependent upon the selected steering mode and the front wheel position. A time delay between the front and rear wheel movements occurred in the four way and crab steering modes; however, most of the delay was caused by the different angular velocities of the wheels. The portion of the delay caused by the controller was very small and occurred at the beginning of each steering sequence when the controller was waiting for the front code to change before initiating a rear wheel movement. At the

end of an 80 degree wheel rotation, the rear wheels were always less than 0.3 second behind the front. Although there was a time delay between the front and rear wheel movements, this delay was very small in ordinary steering movements, because most steering changes were executed in small angular movements. In actual steering situations, a time delay is expected due to the different loading effects between the front and rear wheels.

During the auto-steer mode, the controller kept the machine's centerline within 7 cm (3.0 in.) of the tree stands. Although some oscillation occurred when the altered auto-steer algorithm was used, the rear of the machine was centered over the tree stand as the machine passed over the row. The new auto-steer algorithm kept the machine centered over the tree stands with very little overshooting in the steering corrections. By using the extended front sensor arms, the controller was able to center the front of the machine before reaching the next tree stand.

The results from this research indicate that a microprocessor-based steering controller did satisfactorily perform the necessary steering control for a multi-steering system. By using a microprocessor in the controller, the controller did process numerous machine parameters before initiating a steering change. Not only did the controller process information, but it allowed the operator to choose one of a number of steering modes. Since the operator loses sight of the tree under the machine, the controller did

process the tree's location with respect to the machine's centerline and corrected for any offset.

The specific conclusions which could be derived from this research project were:

1. A microprocessor-based steering controller did perform the necessary control for a multi-steering mode system.
2. To detect offset of a crop row which was not a continuous row, a parallel four-bar linkage was a functional mechanical sensor. This linkage could be extended so that it would always be in contact with at least one plant.
3. By using a sensor pair on the front and one on the rear of the machine, the automatic steer satisfactorily controlled the machine such that the tree was within 15 cm (6.0 in.) of the machine's centerline. The independent front and rear steer during auto-steer allowed the trees within the machine to be located near the machine's centerline while traveling over a curved row.
4. A microprocessor-based controller provided stable control for the steering system which uses solenoid valves as the steering actuators.

8. SUGGESTIONS FOR FURTHER RESEARCH

This research project is only a start in making a machine supervisor out of the driver. In order to aid the driver even more, further study is needed in the following areas:

1. An acceptable contactless tree sensor which can detect the tree row so that the machine will still follow the row when spaces occur due to missing trees,
2. Incorporate the steering controller into a machine controller which will monitor and control several machine operations in addition to the steering operations.

REFERENCES

REFERENCES

1. Allhouse, Gary W. and Morrow, Charles T., Over-the-Row Harvester for Dwarf Fruit Trees. 1971. ASAE Paper No. 71-649.
2. Busse, W., Coenenberg, H., Feldman, and Crusinberry, T.F., The First Serial Produced Automatic Steering System for Corn Combines and Forage Harvesters. Proceedings of the International Grain and Forage Harvesting Conference. September 1977. pp. 43-47.
3. Carlson, Robert F., Old and New Rootstocks, Interstems, and "New Standard". International Dwarf Fruit Tree Association. Compact Fruit Tree, Vol. 12, June 1979. pp. 2-5.
4. Decker, R.L., Microcomputer Control of Mobile Hydraulic Systems. Proceedings of the National Conference on Fluid Power. November 1979. pp. 193-197.
5. El-Ibiary, Y.M. and Nikiforuk, P.N., Design and Performance of a Self Learning Microprocessor-Based Electrohydraulic Position Control System. Proceedings of the National Conference on Fluid Power. November 1979. pp. 215-218.
6. Kirk, T.G. and Krause, A.E. Swather Edge Guide Steering Control System. 1975. ASAE Paper No. 75-1029.
7. Kirk, T.G., Zoerb, G.C., and Wilson, J.N., A Furrow-Following Tractor Guidance System. 1976. ASAE Paper No. 76-001.
8. Michigan Department of Agriculture. Michigan Agricultural Statistics. 1979.
9. Michigan Department of Agriculture. Michigan Fruit Tree Survey. 1978.
10. Ontario Department of Agriculture and Food. Harvesting, Storing, and Packing Apples. Publication No. 431. pp.14-15.
11. Ricks, Donald J. and Pierson, Thomas R., U.S. Apple Supplies: Trends and Future Projections.

Department of Agricultural Economics, Michigan
State University, East Lansing, Michigan 48824
p. 3.

12. Schafer, Robert L. and Young, Roy E., A Digital Controller for Automatic Tractor Guidance. 1976. ASAE Paper No. 76-1068.
13. Shukla, Lal N., Goering, Carrol E., and Day, C. LeRoy, Effects of Tractor Parameters on Automatic Steering. 1970. Transactions of ASAE, Vol. 13, No. 4, pp. 678-681.
14. Tennes, Bernard R., Burton, C. L., and Levin, J. H., Concepts for Mechanizing High-Density-Orchard Fruit Culture. 1976. Transactions of ASAE, Vol. 19, No. 1, pp. 35, 36, and 40.
15. Tennes, Bernard R., Agricultural Engineer USDA. Personal Correspondence. 1980.
16. Westerling, David L. and Costante, Joseph F., Improving Apple Production in Northwest Massachusetts. 1976. ASAE Paper No. 76-2044.