DYNAMIC AUTONOMOUS GROUND VEHICLE RE-ROUTING IN AN URBAN ENVIRONMENT USING A PRIORI MAP INFORMATION AND LIDAR FUSION

Christopher Mentzer
George McWilliams
Kristopher Kozak, PhD
Southwest Research Institute
San Antonio, TX

ABSTRACT

As part of an Internal Research and Design effort to take existing disparate technologies and integrate them into a single autonomous vehicle to advance the state-of-the-art in unmanned ground vehicle autonomy, SwRI has developed a data representation and routing algorithm to deal with the complexities of interconnecting urban roadways and the static and dynamic hazards in such an environment. The program was designed to utilize data from a Route Network Definition File (RNDF), which contains a priori roadway network data. Using its known location and a given destination, the vehicle determines the shortest route to completion. If, during traversal of that route, the vehicle detects an obstacle in its path using its on-board sensors, it will dynamically re-route its path whether that requires changing lanes on a multiple lane road or turning around completely and finding a different route if the path is completely blocked.

INTRODUCTION

The Southwest Safe Transport Initiative

In 2006, Southwest Research Institute (SwRI®) initiated a multi-million dollar R&D program called the Southwest Safe Transport Initiative (SSTI). The SSTI program was begun to build on SwRI's history of supporting the automotive industry and developing technologies for Intelligent Transportation Systems (ITS) and Unmanned Aerial Vehicles (UAV). The initiative aims at improving vehicle performance and traffic safety by integrating sensors, computers and mobile technologies into vehicles with the long-term focus of developing autonomous vehicle technologies. This focus on autonomous vehicle technology development is driven by internal funding provided to research the current state-of-the-art in this area and then develop a demonstration platform capable of autonomously navigating safely in an urban and trafficked environment.

The chosen demonstration platform is a commercially available 2006 Ford Explorer retrofitted with the sensors, actuators, and computing hardware needed to sense the state of the world and make intelligent decisions based on that information (Figure 1. and Figure 2). Commercial off-the-shelf hardware is primarily used to accomplish these tasks, so the main focus is on increasing the state-of-the-art in areas such as sensor processing, world modeling, situational awareness, knowledge representation, value judgment, dynamic vehicle control, route and path planning, and behavior generation.

Figure 1. SSTI Vehicle

Figure 2. SSTI Vehicle Internal Hardware

Some of the off-the-shelf hardware includes:

- Ibeo Alasca XT Laser Scanners and Fusion System
- Oxford RT3052 Global Positioning System (GPS) / Inertial Navigation System (INS)
- EMC AEVIT Drive-by-Wire System (for steering and throttle/brake actuators)
- High Resolution Prosilica EC1350C Camera
- Intel Core 2 Quad and Duo Blade Cluster – ExtremeNode EN-8740 by PCW MicroSystems
- Technocom 5.9 GHz Dedicated Short Range Communication (DSRC) radio
- dSPACE Autobox with Matlab Simulink/CarSim

Autonomous Navigation in Urban Environments

The SSTI vehicle was designed to drive autonomously in an urban environment. Urban driving includes following vehicles, passing vehicles, responding to objects such as pedestrians and bicyclists, merging, yielding, and negotiating complex intersections. The basic component of urban navigation is creating a route from a starting point to a destination point. While in-vehicle GPS systems currently do this for human-driven vehicles, more detail is needed for an autonomous vehicle. By including details such as number of lanes and locations of intersections, the autonomous vehicle has enough information to robustly plan and navigate the entire route from start to finish.

IMPLEMENTATION

Overview of the System Architecture

SSTI has implemented a variant of the 4D/RCS reference model architecture [1] to modularize the components performing the tasks of an autonomous vehicle navigation system. This modular approach allows the slower, more deliberative software components that are computing the long-term route to be separated from the faster, reactive software components that are quickly changing the vehicle's local path to perform actions like avoiding obstacles. shows the navigation system's nodes.

Route Management. The Route Management Node's primary responsibility is to provide an interface to the user allowing the user to choose a destination for the vehicle, either through a map interface, or by providing a Mission Data File, a file type used in the DARPA Urban Challenge. It will then calculate the route based on a priori information stored in a Route Network Definition File (RNDF). The Route Management Node uses the road network points defined in the RNDF to calculate the shortest path from the current position of the vehicle to the location specified by the user. The shortest route is calculated using Dijkstra's algorithm [2]. A list of RNDF Segments is created from the shortest route data. RNDF Segments are a set of individual lanes represented in the RNDF file. The lanes pulled out of the RNDF file and represented as RNDF Segments are lanes that are on the shortest path to the destination, as well as all neighboring lanes that the vehicle can use for passing. The RNDF segments are then sent to the Advanced Navigation Node.

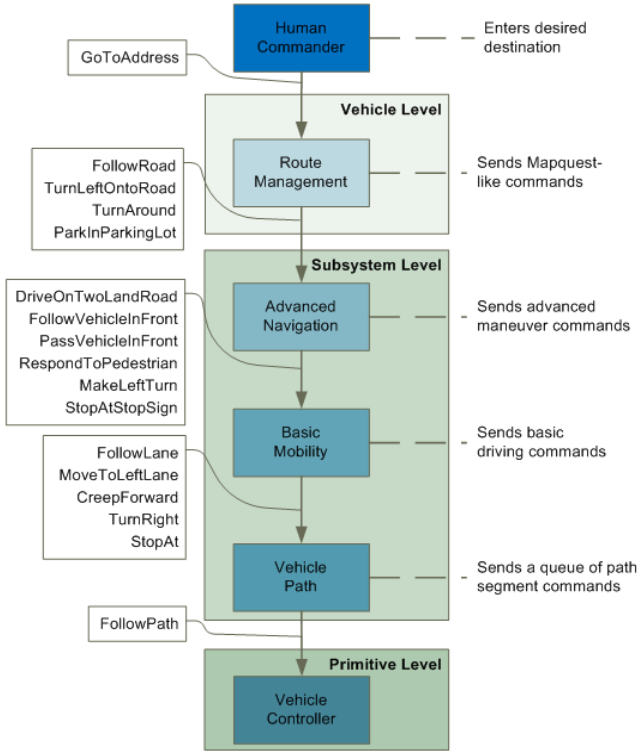

Figure 3. SSTI Navigation System Nodes

Advanced Navigation. The Advanced Navigation Node receives the RNDF Segments from the Route Management Node. When the Advanced Navigation Node receives the RNDF Segments, it converts them into an object called the Multi-Lane Route. The Multi-Lane Route, discussed in the next section, is then used to decide how to negotiate intersections, pass vehicles, or respond to pedestrians. Based on sensor data and the Multi-Lane Route, the Advanced Navigation node generates commands to the Basic Mobility node to perform lower level actions.

Basic Mobility. The Basic Mobility Node receives the commands from the Advanced Navigation Node. It generates behaviors that will allow the vehicle to drive in a specific lane, change lanes, make turns, go around objects, follow other vehicles, and stop at stop signs. It uses data from the Multi-Lane Route to create a Route Segment List. The Route Segment List is a series of lanes from the Multi-Lane Route that constitute the planned route ahead of the

vehicle. Inside of each Route Segment in the list are a set of waypoints. When combined, the waypoints in each route segment make up the desired route of the vehicle. The commands generated from this node will be sent to the Vehicle Path Node.

Vehicle Path. The Vehicle Path Node receives commands from the Basic Mobility Node, which includes information about the near-term desired route the vehicle should travel. It then creates the local path segments, which are compact representations of the local route that can be interpreted by the real-time controller. Specifically, this information contains the desired steering angle, heading, and location of the vehicle along the path. It sends these path segment commands to the Vehicle Controller Node. It also generates desired states of the vehicle's auxiliary functions such as the turn signals and shifter.

Vehicle Controller. The Vehicle Controller Node receives commands from the Vehicle Path Node, which contains information about the desired local path segments and speed profile of the vehicle. It uses this information to generate desired steering and speed commands at each point along the path. It also uses sensor information to perform Advanced Cruise Control (ACC). The ACC ability is a low-level function to follow other vehicles at a specified distance. This allows the higher-level nodes of the architecture to perform more advanced routing and re-routing without worrying about simple driving tasks like following a vehicle at a target distance. This node also uses vehicle and sensor state information to perform safety checks and generate safety behaviors. It sends the steering information to the Steering Controller and the speed information to the Speed Controller.

Multi-Lane Route Class

The Multi-Lane Route (MLR) Class is used to represent not only the shortest path to the destination but also all of the data pertaining to the different segments (individual lanes of a particular roadway) included in that shortest path. The MLR is an array of the segments included in the shortest path as well as all of the segments pertaining to the neighboring lanes, opposing lanes, next lanes in the sequence, and previous lanes in the sequence. The MLR additionally contains the information of how these segments correspond to each other, i.e. if a segment has a lane to the left also traveling in the same direction there is a pointer from that segment to the segment representing the left lane. The information in the MLR is populated at the time of the shortest route calculation and passed to the Central Intelligence Module (CIM) for use in real time route decision making for behaviors such as passing slow moving vehicles and 3-point turns in the case of blocked paths.

General Route Following

The MLR is used by the CIM in situations where obstacles are not encountered to simply manage if lane changes are required to make a left or right turn to continue on the shortest route to the destination. At the time of shortest route calculation, roadways with multiple lanes are simplified to include a minimal lane change distance to any point on the network where one segment can be accessed from another (i.e. a left or right turn). This simplification allows for faster calculation of the shortest route while still enforcing the physical vehicle limitations pertaining to the minimum turn radius.

While enforcing the minimum turn radius ensures that the vehicle can achieve the designated lane changes, the minimum lane change distance does not generally provide the most natural and comfortable lane changes, especially at higher speeds. That being the case, the CIM utilizes the MLR during the actual traversal of the route to 1) keep track of remaining distance on the current segment before it ends and 2) determine if and how many lane changes will be required to make a turn onto the next segment in the route. The CIM will implement a change over a longer distance based on speed if it is possible to provide for a smoother, more comfortable lane change. If there is not enough distance for a longer lane change, the vehicle will have to slow down to accomplish the minimal lane change distance to reach the next segment.

Lane Changes for Obstacles

In addition to using the MLR to follow the predetermined route, the CIM utilizes the MLR to determine if and when the vehicle should perform lane changes when obstacles are detected. The vehicle currently detects other objects in its environment using scanning LIDAR. If an object is located and determined to be on current desired path of the vehicle, the vehicle must then react to that obstacle. At this point, the CIM looks at the speed of the obstacle and the remaining distance on the path to determine the best course of action. If the obstacle is not in the lane of the segment that would ultimately be required to exit to another segment, then the CIM simply force the vehicle to change lanes earlier than it would if it had determined that a route lane change was needed to reach that exit location due to the remaining distance on that segment.

If the obstacle is determined to be stationary and in the lane needed to exit the segment the CIM analyzes the MLR for possible courses of action. First the CIM determines if there are adjacent lanes to the current lane traveling in the same direction to either the right or the left. If a lane exists on both sides of the current lane, the CIM will first check to see if either lane is blocked. If both adjacent lanes are blocked, the CIM will check all lanes going in that direction to determine if multiple lane changes would navigate the

vehicle around the obstacle. If all lanes in one direction are blocked, the vehicle will go into 3-point turn behavior which will be described later in this document. If one lane is available, the vehicle will determine if it can change into that lane, pass the obstacle and have enough room to change lanes back into the current lane to reach the exit from that segment. If so, the vehicle will modify the route to change lanes around the obstacle (see Figures 4 and 5) and then change back after the obstacle. If there is not enough distance, the vehicle will check for other lanes that might be used to travel around the obstacle and reach the goal. If no lanes are found that will get the vehicle back to the lane with the exit, then the vehicle will go into the 3-point turn behavior. If both lanes adjacent to the current blocked lane are available, the vehicle will select the lane that provides the shortest distance to the exit, including lane changes.

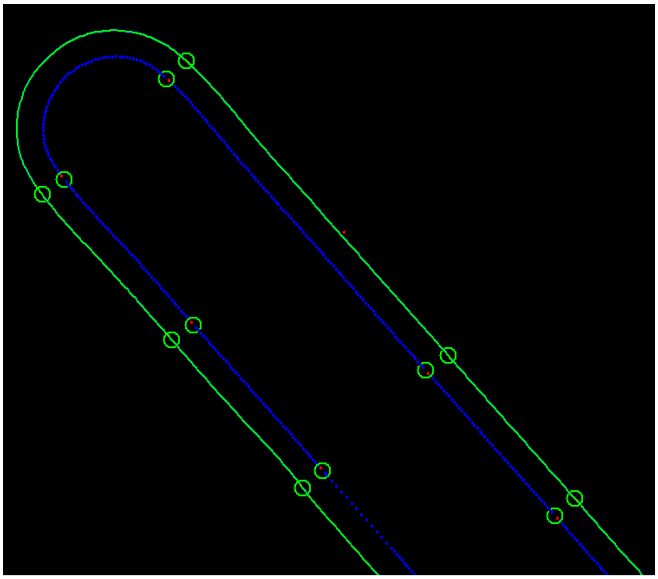

Figure 4. Pre-Planned Route

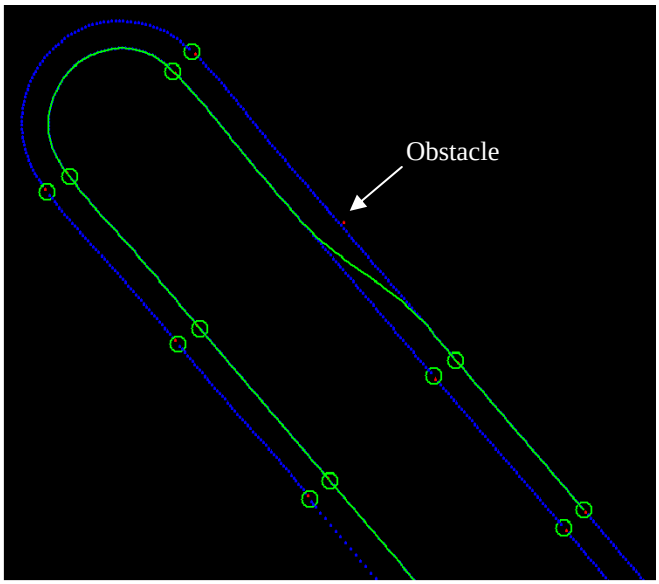

Figure 5. Lane Change Due to Obstacle

In the case where the obstacle is in the current lane of the vehicle and is moving along that lane, the CIM additionally takes the speed of that obstacle into account to determine how it will react. In the case where the obstacle is moving faster than the desired speed of the vehicle, the vehicle will continue on its current course because a collision is not imminent. If the obstacle is moving at a speed slower than the desired speed of the vehicle, the CIM will look at the difference between the speed of the obstacle and vehicle and then determine if there is enough distance to move around the obstacle, assuming it will maintain its current speed, and move back into the lane in front of the obstacle before it reaches its exit.

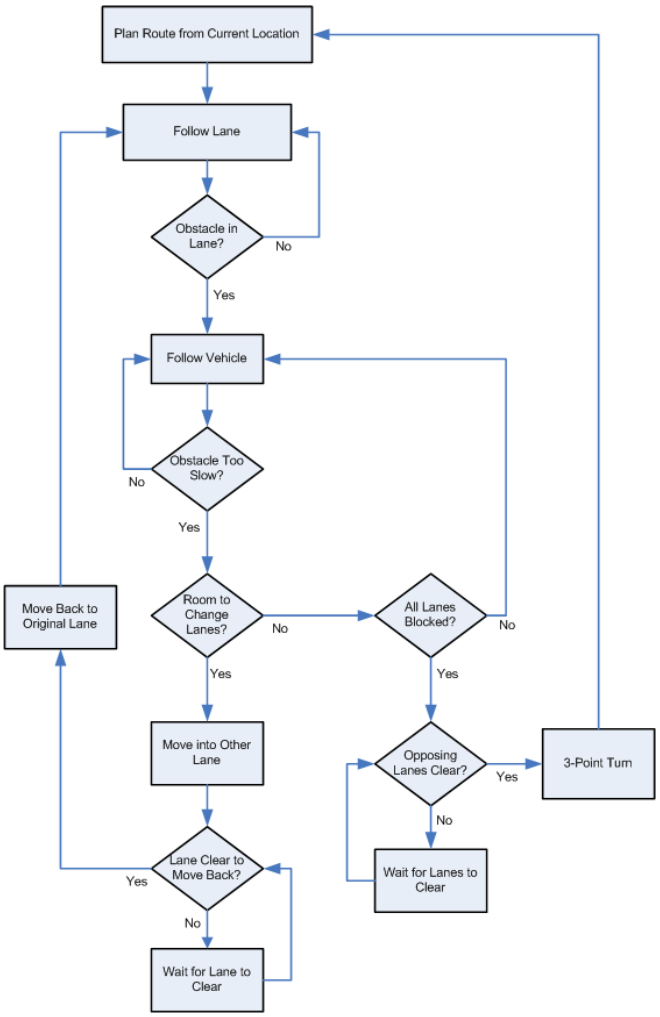

Figure 6. Lane Change Due to Obstacle

In addition to this distance check, the checks performed in the case where the obstacle is stationary listed above are also performed. If there is not enough room to pass the obstacle and make it back in the lane for an exit, the vehicle will slow down and follow the obstacle at its current speed. If the CIM determines that it can pass a vehicle based on the assumption that the obstacle will travel at a fixed speed and changes lanes then the obstacle speeds up, the vehicle will slow down if there is not enough room to pass and pull back in the lane behind the obstacle after it passes. Figure 6 shows a flowchart of the process that occurs when the vehicle decides whether a lane change or 3-Point Turn is needed.

3-Point Turns

In the case where all of the lanes are blocked in one direction of a segment, the CIM will go into the 3-point turn behavior. In this behavior, the vehicle will first back up, if there is nothing behind it, to get a better view of all the lanes to ensure that they are blocked. Additionally, if one of the further lanes is actually open, the vehicle will have enough room to change lanes to get over to that lane. Backing up will also give the vehicle enough room to perform a 3-point turn if it has gotten too close to the obstacle in front of it. If, after backing up, the vehicle still determines that all of the lanes in its current direction are blocked on that segment, the CIM will then use the data in the MLR to locate the lanes on the segment that travel in the opposing direction. These opposing lanes are then checked for moving obstacles approaching the current position of the vehicle and stationary objects within the area where the 3-point turn would occur. Once the opposing lanes are found to be clear, the vehicle will then check to see if the distance from the current lane to the furthest opposing lane is greater than two times the minimum turn radius of the vehicle. If this is the case, then the vehicle will perform a U-turn into the furthest opposing lane. In the case where the furthest lane is closer than two times the minimum turn radius, the vehicle will start a 3-point turn maneuver. In this maneuver, the vehicle will move forward at its minimum turn radius until it reaches the edge of the furthest opposing lane. The vehicle will then back up at its minimum turn radius to the right to a minimum distance required to complete the turn in the next forward move or until it reaches the other edge of the roadway. After that, the vehicle will move forward with the turn radius required to end up in the furthest opposing lane with a heading that matches that of the lane. If the vehicle cannot reach this heading on this move, which might be the case for narrow two-lane roads, the vehicle will perform as many back and forth maneuvers until it reaches the desired state. Figure 7 shows a situation where the vehicle will make a 3-point turn. Both lanes going in the north-west direction are blocked, so the vehicle will make a three-point turn when it gets within a certain range of the obstacles. Figure 8 shows the start of that 3-point turn.

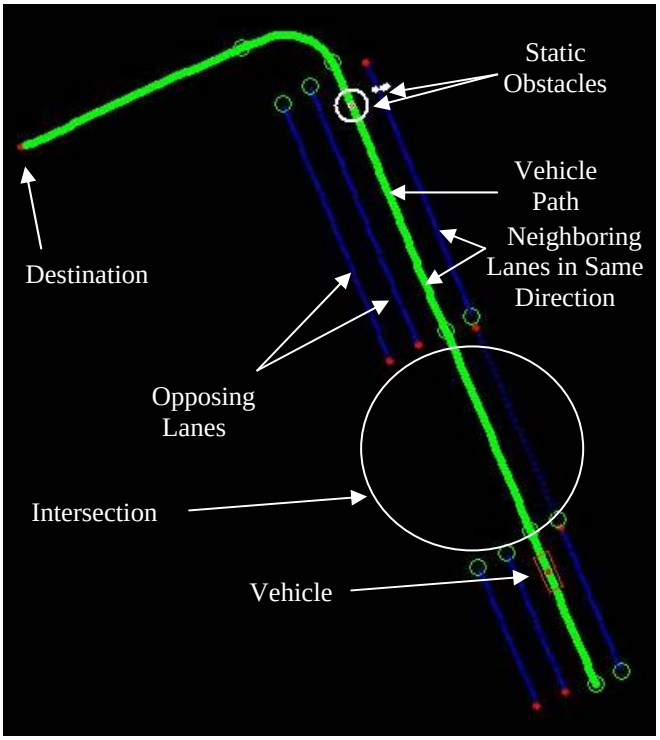

Figure 7. Blocked Lanes

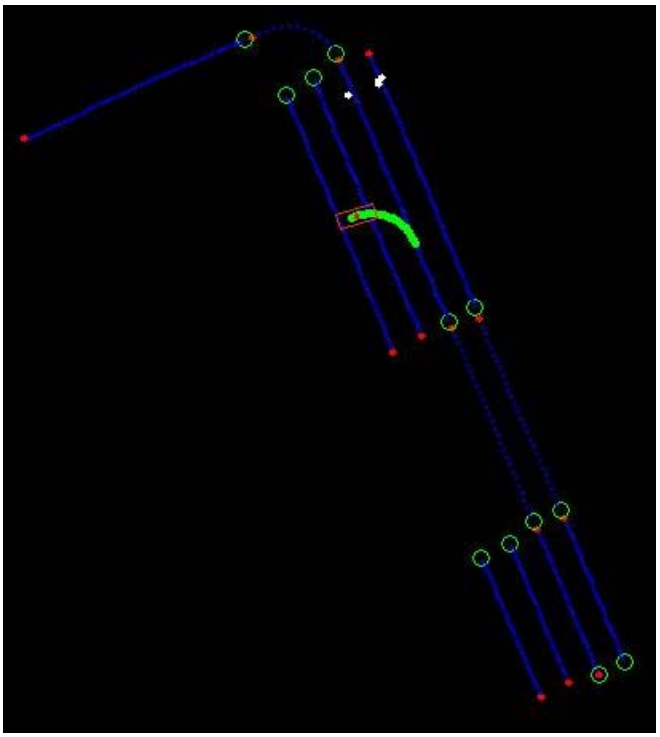

Figure 8. Start of the 3-Point Turn

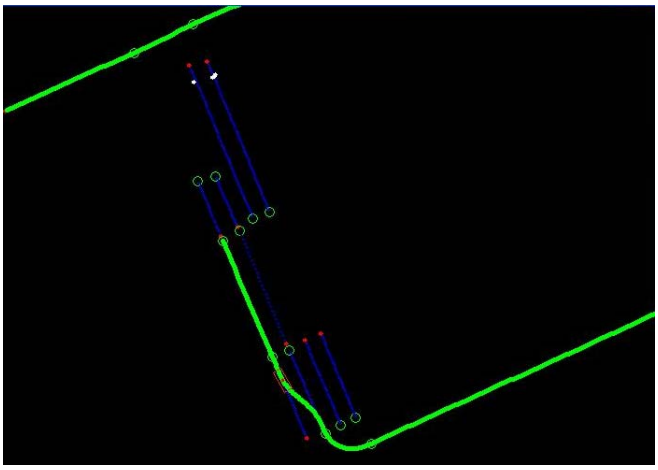

Figure 9. Re-Routed Path Excluding Blocked Segments

Once the vehicle is in the furthest opposing lane and has completed the 3-point turn maneuver, the CIM sends a message to the Route Management Node requesting a new shortest route to the previously designated destination. In this message, the CIM excludes all of the points that were blocked that triggered the 3-point turn to prevent paths going through those points from being included in the new route. Once this new route is determined, a new MLR is generated for the new shortest path and sent to the CIM to start following again. Figure 9 shows the new route the vehicle generates after it has done the 3-point turn.

ACKNOWLEDGMENTS

The authors wish to acknowledge the technical contributions from the following SSTI project team members: Steve Dellenback, Roger Lopez, Ryan Lamm, Dan Pomerening, Joe Steiber, Paul Avery, Mike Brown, Bapi Surampudi, Kevin Alley, and Joshua Curtis. This research was funded under Southwest Research Institute IR&D project 10-R9648.

REFERENCES

[1] Raj Madhavan, Elena R. Messina and James S. Albus, Intelligent Vehicle Systems: A 4D/RCS Approach, Nova Science Publishers, January 15, 2007, pp. 6-12.

[2] E.W. Dijkstra. “A note on two problems in connexion with graphs”, Numerische Mathematik, pp. 269-271, 1959.