Proceedings of the 1986 Winter Simulation Conference
J. Wilson, J. Henriksen, S. Roberts (eds.)

MODELING AGV SYSTEMS

Deborah A. Davis
Systems Modeling Corp.
Calder Sq. P.O. Box 10074
State College, PA 16805

ABSTRACT

Computer simulation is often used as an analysis tool during the design of Automated Guided Vehicle (AGV) systems. However, because of the complexities inherent in automated material handling systems, general-purpose simulation languages must be used creatively to capture the desired detail in the model. This paper presents some general concepts which can be used to model AGV systems. Also, some of the critical concerns which must be addressed in a simulation analysis of an AGV system are presented and discussed.

1. INTRODUCTION

In recent years, automated systems of many varieties have been implemented successfully in manufacturing. In many instances, automation has been added in stages to a non-automated facility, in which case the integration of the old and new production elements must be carefully considered during the design phase. Entire facilities which are highly automated also have been created. In both cases, the system complexity has resulted in considerably greater attention being paid to design.

Analysis of automated systems, in which interactions among many elements are significant contributors to overall performance, is well suited to simulation. AGV systems are particularly sensitive to the interactions among the vehicles on the track, and to the level of coordination between the AGV system and the rest of the manufacturing system. Because of the dynamic behavior of the AGV system, queueing theory-based analyses tend to give overly optimistic predictions. Math programming procedures, many based on the transportation problem, have been presented as solutions to some AGV design problems; however, the problem size for most industrial applications is prohibitively large, requiring very long computer runs. Simulation models, which can incorporate great detail, and which in many cases can be developed on a microcomputer, have emerged as the analysis tool of choice during AGV system design.

2. CHARACTERISTICS OF AGV SYSTEMS

An Automated Guided Vehicle system is a material handling system in which driverless, battery-powered carts are moved by means of an electronic, chemical, or optical signal from a path which has been installed in the floor. Chemical and optical guidance methods are used in "clean" environments, such as office buildings and electronics assembly areas, where other equipment is not likely to disturb the guidance path. However, in many manufacturing environments, a well-protected electronic track is installed in the shop floor, since chemical or optical signals would be frequently disrupted. The electronic path is established by cutting into the floor and installing a wire guide, which generates a magnetic field that the vehicles track. Because of the strong protection provided by the floor, the electronic tracks suffer few disruptions. Changing the track is a costly endeavor, though.

The vehicles themselves may be towing vehicles, semi-automatic pallet trucks, or unit load carriers. Towing vehicles have one or more carts attached to them, and are usually loaded and unloaded manually. Pallet trucks also are often loaded by an operator, who also programs into the vehicle its destination. Loads are placed directly on the unit load carriers, rather than on forks or tines. The unit load carriers are able to load and unload material automatically, and are the most commonly used automated vehicles.

Control of AGVs within most manufacturing systems is accomplished by means of a link to a central computer. The central computer often contains system-wide information, and directs the vehicle where to go to pick up a load, where to take the load, and might even redirect the vehicle while it is in transit to perform another task. Depending upon the type of system, communication may occur only at specified locations, or in more advanced systems, at any vehicle position.

3. SIMULATION MODELING OF AGV SYSTEMS

Simulation modeling is used in many phases of the design process. AGV system characteristics include three for which simulation is commonly used: track layout, vehicle fleet size, and system control logic. When various track layouts are being compared, a simulation model can assist in identifying bottlenecks and areas of high congestion. Once the layout is established, more detailed simulation models are used to compare various control schemes, and to evaluate the impact of the number of vehicles on system performance.

For most analysis purposes, the simulation model should closely resemble the physical system, especially regarding the interactions among the vehicles. As in any simulation analysis, however, the quality of the input to the model, including estimates for vehicle speeds, part process times, etc. will determine how much detail should be incorporated in to the model. Models developed early in the design process will be more general than those which are created during actual system implementation, since system data is still rough. As the system is fine-tuned and information is more exact, the simulation model should become a more detailed description, so that decisions are made based on as exact a replica of the physical system as possible.

4. TOOLS FOR MODELING AGV SYSTEMS

Many characteristics of AGV systems render them difficult to model. The following sections describe some of these problems and techniques for modeling them using general-purpose simulation languages. For illustration purposes, sections of model code will be presented in the SIMAN language. However, the concepts and their solutions are intended to be general enough to be independent of the application language.

4.1 Avoiding Vehicle Collisions

In many material handling systems, including AGV systems, individual transport devices share the areas of space on which they move. If an AGV is stopped, other vehicles which are moving on that track must wait until the halted unit moves. Likewise, if sections of track intersect, only one AGV may pass through the intersection at a time.

In a simulation model, resources are used to avoid collisions between AGV units while they move within the system. Resources are fixed-capacity facilities which may be allocated to one or more simulation entities (jobs). If the AGV track is separated into a number of consecutive resources, then allowing only one entity at a time to have a resource manages AGV movement. Physical AGV systems often resemble this "zoning" very closely. Segments of track are divided into non-overlapping zones, in which no more than one vehicle is permitted at a time to avoid collisions.

The two-loop simple AGV system in Figure 1 will be used for illustration. The entire AGV track is unidirectional, with the direction of flow indicated by arrows on the track. In this system, there are six load/unload stations (circled) at which AGVs pick up or deposit jobs. Also, there are three extra stations (boxed, numbers 7-9) which are used to complete the division of the system into zones. A small number of stations were placed along the track to keep the example at a reasonable size. Additional zoning stations would allow more precise control of the AGVs.

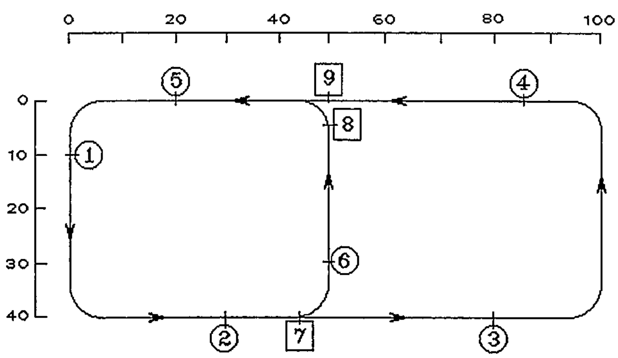

Figure 1: Sample System AGV Layout

Within the simulation model, when a vehicle is to be sent from a load station to its destination unload station, it must undergo a series of movements through the track zones. When it reaches the station at the end of one zone, it must wait until the next zone is clear before it can continue its journey. To accomplish this, the entity in the model seizes the resource which represents the "next" track zone. When the entity has been given the resource, it releases the track zone it just passed through, and proceeds to the station at the end of the next zone.

To move an AGV from one station to another, the simulation also must know what path it should follow. A from-to table is used to identify to what station the vehicle should be sent next, in order to get from where it is to its final destination. The model executes a sequence of moves from one station to another, always finding the next station by looking at where to go next to get to the final destination. The table in Figure 2 maps the sample system. A single entry in a row indicates that to move from the row station number to any other station in the system, the vehicle always must move next to the specified station. For example, to move from station 1 to any other station, the AGV first must travel to station 2.

		TO								
		1	2	3	4	5	6	7	8	9
	1	2								
	2	7								
	3	4								
	4	9								
FROM	5	1								
	6	8								
	7	6	6	3	3	6	6	7	6	6
	8	5								
	9	5								

Figure 2. AGV Travel Table

Zones are numbered such that the resource index number corresponds to the station at the end of the zone. For example, an entity which is currently in track zone 1 must seize resource number 2 before it can move the AGV to station number 2. When it is given the resource, it releases resource 1, over which it previously passed, and moves to station 2. The following code segment illustrates the logic required to move an AGV which has just arrived at a station to the next station en route to the final destination.

```
ENDZONE   ASSIGN:'CURRENT ZONE'='AGV LOC';
          BRANCH,1:
            IF,'AGV LOC'.EQ.'FINAL DEST',ARRIVED:
            ELSE,CONTINUE;
;
CONTINUE  ASSIGN:'NEXT ZONE'=TF('CURRENT ZONE',
            'FINAL DEST');
          QUEUE,'NEXT ZONE';
          SEIZE:TRACK('NEXT ZONE');
          RELEASE:TRACK('CURRENT ZONE');
          MOVE:AGV('VEHICLE NO'),'NEXT ZONE':
            NEXT (ENDZONE);
```

(Note: Synonyms are used to substitute descriptive strings for SIMAN attributes and variables. The synonym definitions for the sample code are: AGV LOC = LT(1,A(4)), FINAL DEST = A(5), NEXT ZONE = M, CURRENT ZONE = A(3), VEHICLE NO = A(4). Also, the SIMAN table look-up function, TF, may need to be adjusted if multiple data tables are included.)

When a vehicle arrives at the end of a zone, its controlling entity is sent to the block labeled ENDZONE. (The final destination of the entity is stored in 'FINAL DEST'.) First, the zone number that the entity currently has, 'CURRENT ZONE', is set to the vehicle's location, 'AGV LOC'. If the current location equals the destination station number, then no more transports are required, so the entity is sent to a block labeled ARRIVED to be unloaded. (The code for unloading is not included.)

If the current zone is not the final destination, then the vehicle is somewhere in transit, and should be moved again toward the destination station. At the block CONTINUE, the number of the next zone to be traveled over is found by looking into the from-to table (Figure 2). The vehicle is located at 'CURRENT ZONE' and wants to move to 'FINAL DEST', so the entry returned by the from-to table is the next station number (also the next zone number) for the vehicle to visit. This value is assigned to 'NEXT ZONE'.

The entity then attempts to seize the resource which controls the next section of track. In SIMAN, the QUEUE-SEIZE combination of blocks is used to allocate resources to entities. If the resource TRACK('NEXT ZONE') is available, the entity seizes it and enters the block following the SEIZE block. If the resource is busy, the entity waits in the 'NEXT ZONE' queue until the TRACK unit is released by the entity which currently has it.

Once the entity is given the resource, it releases the track zone it just moved through, clearing the way for any vehicles behind it to enter that area. Then, the entity moves the vehicle, AGV('VEHICLE NO'), to the station at the end of 'NEXT ZONE'. When the AGV arrives at 'NEXT ZONE', the entity is sent again to ENDZONE, and repeats the above procedure.

The AGV control is structured to read as much information as possible from data tables. By setting few fixed parameters within the model code, any changes to the AGV system design require only modifications to the data tables. The model does not need to be adjusted. This is especially useful as more stations are added to shrink zone sizes, which affords more exact control of the AGVs.

4.2 Mapping the AGV System

The from-to table in Figure 2 partially describes the AGV system to the simulation model. However, other information such as distances between stations and AGV velocities are also required. The AGV velocities can be kept in global variables, to be used for any vehicle movement. Distances between stations are required at two points in the simulation model. First, to calculate the time required to move an AGV from one point to another, the model must have access to the distance between the points (as well as the AGV's velocity). Also, if the allocation of vehicles to parts is based on the distance the AGVs are from the load station, the model must be able to calculate how far the vehicle must travel to get to the part location.

The system map can be defined in at least two ways. Since the from-to table already contains the sequence of stations a vehicle must visit to go from anywhere in the system to any other location, a table which contains only the distance between adjacent stations would completely define the system. When moving an AGV, using this table would not add any modeling burden, since vehicles only travel between adjacent stations. However, when comparing the length of travel of various vehicles from their locations to the part station, use of this table would require the model to trace through the sequence of stations to get from "there" to "here," summing the distances along the way. Figure 3 illustrates this type of table for the sample AGV system.

The alternate mapping table contains a complete from-to distance matrix, listing the total travel distance from any station in the system to any other station. For non-adjacent stations, the table entry is the sum of the distances between the stations which must be traversed to get from the vehicle location to the part station. This method simplifies functions within the simulation model, but increases the amount of data required to fully describe the system. Complete mapping of the sample system is presented in Figure 4.

4.3 Control Logic: Selecting an AGV

When a job completes processing at a station or arrives to the system, an AGV is assigned to pick up the part, if any vehicles are available. If there are none, the part enters a queue, awaiting assignment when a vehicle is freed. If only one vehicle is available, it is allocated to the job and assigned to pick it up. However, if more than one AGV is available, some rule must be applied to assign a vehicle to the job.

		TO								
		1	2	3	4	5	6	7	8	9
	1	-	60	-	-	-	-	-	-	-
	2	-	-	-	-	-	-	15	-	-
	3	-	-	-	75	-	-	-	-	-
	4	-	-	-	-	-	-	-	-	35
FROM	5	30	-	-	-	-	-	-	-	-
	6	-	-	-	-	-	-	-	30	-
	7	-	-	35	-	-	15	-	-	-
	8	-	-	-	-	35	-	-	-	-
	9	-	-	-	-	30	-	-	-	-

Figure 3: Condensed Distance Table

		TO								
		1	2	3	4	5	6	7	8	9
	1	0	60	110	185	150	90	75	115	220
	2	120	0	50	125	90	30	15	55	160
	3	170	230	0	75	140	260	245	285	110
	4	95	155	205	0	65	185	170	210	35
FROM	5	30	90	140	215	0	120	105	145	250
	6	90	150	200	275	60	0	165	25	310
	7	105	165	35	110	75	15	0	40	145
	8	35	125	175	250	35	155	140	0	275
	9	60	120	170	245	30	150	135	175	0

Figure 4: Expanded Distance Table

The rule most commonly applied is "Nearest Vehicle," which assigns the AGV with the shortest travel path to the job's location, in an attempt to move jobs through the system as quickly as possible. Other rules are intended to balance workload among the vehicles. The "Least Utilized Vehicle" rule sends the available vehicle which has the lowest utilization to pick up and deliver the job. Utilization can be defined as fewest minutes moving or fewest minutes loaded, and often only the current shift is included. The "Longest Idle Vehicle" rule selects the available vehicle which has been idle the longest.

In the simulation model, the reqired vehicle information includes the velocity, location, and status of each unit. When a job arrives to the system, it examines the status of all vehicles, to determine whether any are idle and active. If more than one vehicle is available, then one of the above rules must be applied to select a vehicle.

Nearest Vehicle. To find which available unit is closest, the entity must examine the system map (see MAPPING THE AGV SYSTEM) for the minimum distance from each available AGV's location to the job station. When the closest available vehicle has been identified, then it is allocated to the entity and moved to the entity's station. The model code which finds the closest vehicle (using the system mapping in Figure 4) follows.

```
          ASSIGN:J=1;
          ASSIGN:'MIN DISTANCE' = 99999;
TOPLOOP   ASSIGN:'AGV LOC' = LT(1,J);
          BRANCH,1:
            IF,'VEHICLE STATUS' .NE. 'AVAILABLE'
               .OR. TF('AGV LOC','JOB LOC') .GT.
               'MIN DISTANCE',CHKNEXT:
            ELSE,NEWMIN;
;
NEWMIN    ASSIGN:'MIN DISTANCE'=TF('AGV LOC',
            'JOB LOC');
          ASSIGN:'BEST VEHICLE NO' = J;
;
CHKNEXT   ASSIGN:J=J+1;
          BRANCH,1:
            IF,J .GT. 'FLEET SIZE',GOT IT:
            ELSE,TOPLOOP;
```

(Note: Synonyms are used to substitute descriptive strings for SIMAN attributes and variables. The synonym definitions for the sample code are: VEHICLE STATUS = IT(1,J), AVAILABLE = 0, MIN DISTANCE = X(49), FLEET SIZE = X(50), BEST VEHICLE NO = A(4), AGV LOC = X(48), JOB LOC = M. Also, the SIMAN table look-up function, TF, may need to be adjusted if multiple data tables are included.)

The variable J is used as a loop counter, identifying which vehicle number the entity is examining. MIN DISTANCE stores the distance from the closest vehicle examined to the job, and is initialized to an arbitrarily large number (99999). At the block labeled TOPLOOP, the location of vehicle unit J is stored in AGV LOC. Then, the first branch checks whether the vehicle being examined is both idle and closer to the job than the pending MIN DISTANCE. If either of these conditions is not met, then the entity branches to the block labeled CHKNEXT to examine the next vehicle. However, if the current vehicle, unit J, is both available and closer than the pending nearest vehicle, the entity branches to NEWMIN, where it updates both the distance of the closest unit (MIN DISTANCE) and the nearest vehicle's number (VEHICLE NO.).

At CHKNEXT, the vehicle counter is incremented, and compared with the total number of vehicles, stored in FLEET SIZE. If all vehicles have been examined, the entity branches to GOT IT, where it will be assigned to the chosen vehicle (VEHICLE NO.) and will move the AGV to its location, a procedure similar to that described in Avoiding Vehicle Collisions. If the vehicle counter does not exceed the fleet size, then the entity branches back to the block, TOPLOOP to examine the next unit.

Least Utilized Vehicle. Two methods may be used to determine which vehicle has the lowest utilization. In the first, a vehicle status variable is kept for each AGV, which has a value of 0 while the vehicle is idle, and a value of 1 when busy. As mentioned above, "busy" is defined either as moving, or as loaded. The status variable is updated within the model at the appropriate times to reflect the applied definition. Time-persistent statistics kept on each of the status variables provide the percent of time the vehicle has been busy. The minimum average value of these time-persistent statistics tells which vehicle has the lowest utilization.

The second alternative sets aside a variable for each vehicle which records the total amount of time the vehicle has been busy. In the model, each time a "busy" period is initiated, a second variable for that vehicle is assigned the time the busy period began. When the busy period ends, the beginning time is subtracted from current time, and the result is added to the variable which is recording the total busy time for the vehicle. To decide which vehicle has been least utilized, the unit with the minimum total busy time is selected.

Least Utilized Vehicle. As in the second method presented for the Least Utilized Vehicle rule, a variable must be kept for each vehicle which records the time that the most recent busy period began. At the point where the AGV selection is made, then, the unit which has the minimum value of that variable is selected (i.e. the unit which has been idle the longest). An advantage of using the second method for Least Utilized Vehicle is that the Longest Idle Vehicle rule could be used as a tie-breaker, or vice versa.

4.4. Control Logic: Vehicle Dispatching

After an AGV drops off a part, it is assigned the next task to perform, or to wait until it is again required. If there is only one job waiting for loading, then the vehicle is assigned to pick up the job and move it to its next operation. However, if there are many jobs waiting for transport, a dispatching rule must be applied to determine where the vehicle should be sent.

Two simple rules which are commonly applied are First Come-First Served and Earliest Shop Arrival Time. First Come-First Served attempts to minimize the time that jobs spend waiting for vehicles, while Earliest Shop Arrival Time is intended to minimize the amount of time jobs spend in the shop. Two other rules which try to reduce the possibility of system blocking are Maximum Outgoing Queue Size and Minimum Remaining Outgoing Queue Space. Finally, the Shortest Travel Distance rule calculates how far each vehicle must travel to reach each job's station (ignoring the effects of traffic), in an attempt to minimize the proportion of time the vehicles travel empty.

First Come-First Serve and Earliest Shop Arrival Time. To represent the First Come-First Serve and Earliest Shop Arrival Time rules, the simulation model inserts an entity into a queue when it requires an AGV to pick it up. This queue provides a list of all jobs awaiting vehicle assignment. For First Come-First Serve, the queue is ranked First In-First Out, and the vehicle removes the first entity in the queue. Earliest Shop Arrival Time is similar, except that the entities in the queue should have an attribute which holds the shop arrival time. Then, either the queue ranking rule is specified as the minimum of that attribute value, or the queue is searched for the entity with the minimum. The selected entity is removed and allocated the vehicle.

Maximum Outgoing Queue Size and Minimum Remaining Outgoing Queue Space. The two outgoing queue size rules require that the simulation model record in variables the size of the outgoing queue at each department. If separate simulated queues are defined for each station's outgoing queue, then the number of entities in the queue tells how many jobs are awaiting AGV pick-up. The Maximum Outgoing Queue Size rule simply requires the simulation model to determine which station's outgoing queue is largest, then to assign the AGV to pick up the first job in that queue. For the Minimum Remaining Outgoing Queue Space, another set of variables (or data parameters) should hold the outgoing queue size at each station. To determine where the AGV should be dispatched, the model selects the station which has the smallest difference between the queue size and the number of jobs currently in the queue, then assigns the first job in that queue to the vehicle.

Shortest Travel Distance. The Shortest Travel Distance rule is similar to the Nearest Vehicle Rule for AGV selection. All entities awaiting AGV assignment should be placed in a single queue. Using the table in Figure 4 to map the AGV system, the entity which has the minimum value of the table entry corresponding to moving the AGV "from" its current location "to" the entity location should be selected. In all cases, the procedure for moving the vehicle from its current location to the job's station is similar to that described in Avoiding Vehicle Collisions.

4.5 Controlling Vehicle Speed

At the broadest level of detail, AGVs are assumed to have a constant velocity for all movements within the system. In practice, many vehicles travel at a faster rate when unloaded than when carrying a job. Also, some vehicles experience decreases in velocity as the battery wears down. In all systems, for a vehicle to start, stop, or negotiate turns, there are distinct deceleration and acceleration phases. Depending upon the decisions which are to be made based on the simulation analysis, as well as the actual significance of these factors in overall performance, it may not be necessary to include these details in the model. However, for systems which are sensitive to these factors, the model should represent them as accurately as possible.

To change the velocity of a vehicle depending upon whether it is loaded or unloaded, two variables are required. They may either be defined to be one of the velocities (loaded or unloaded) and a factor for converting to the other (e.g. loaded velocity is 80% of unloaded velocity), or simply the unloaded and loaded velocities. Within the model, the time required to travel through a zone is calculated as the length of the zone divided by the appropriate velocity; travel times for the various system control rules are determined similarly.

If vehicle velocity is significantly impaired by battery wear, then a small submodel for each vehicle may be used to adjust the individual AGV's velocity according to a predefined function. The rest of the model remains unchanged, since any movement of the vehicle will be tied to its current velocity.

Modeling the acceleration and deceleration that vehicles undergo in stops, starts, and turns is accomplished in a set of constructs. To reflect the slow-down a vehicle undergoes when stopping, and its start-up acceleration, the entity controlling the vehicle is examined at the end of each movement through a zone. If there is a vehicle currently traveling through the next zone, then the vehicle delays for a short period before seizing the next resource, representing the deceleration phase. When it is allocated the next zone, it undergoes another small delay, to account for the required acceleration. These small delays may be entered as constants, and if there is a significant difference between the acceleration/deceleration time for a loaded vehicle versus an unloaded one, a multiplication factor may be used to represent the motion accurately.

To include the added time required to move a vehicle through a section of path with a turn, the entry in the distance table for that zone may be increased so that the actual time (when the distance is divided by the velocity) is accurate. Since the simulation model is time-driven, rather than actually distance- driven, this will result in accurate travel times. Also, if the nearest vehicle rule is applied for allocating AGVs to parts, the calculations will give the unit which has the smallest travel time, rather than distance. However, the skewed data may be difficult to maintain if there are track layout changes, or if the model is modified by someone else at a later date.

If either of these problems causes concern, an alternate way is to establish a separate table, either listing which zones include turns and the factor which should be multiplied to the distance to give the correct travel time, or completely defining all stations and the travel factors (a full from-to matrix).

4.6 Redirecting In-Transit Vehicles

In AGV systems with more advanced control capabilities, a vehicle which is in transit to one destination may be redirected by the controlling computer to perform an alternate task. This is often used to move "hot" jobs through the system as quickly as possible, or when an AGV is on its way to pick up a job far away, and a closer job requests it. Depending upon the communications setup, the computer can send a message to the AGV either only at specified points within the system (typically pickup and dropoff stations and intersections) or independent of the AGV's location. For modeling purposes, the scope will be limited to the first case, and the assumption will be made that the computer can redirect the vehicle when it passes any control point (station).

The event of redirecting the vehicle to a new destination is triggered in the model by a job requesting transport via the AGV system to its next destination. The model must examine the criteria established for deciding whether to redirect a vehicle to this new assignment, such as whether the job is late or is holding up other production, or whether an AGV which is traveling empty will pass this job's station en route to its current assignment.

When the decision has been made to redirect a vehicle to a job, the entity representing the new job sets a flag for the selected vehicle to tell it where it has been reassigned. The entity then waits to be allocated the AGV when the vehicle receives the signal.

Elsewhere in the model, when any AGV moving to pick up a job arrives at the end of a zone, it checks its own redirect flag to see whether it has been reassigned to another destination. If it has, the vehicle is freed, allowing the "hot" job to be allocated that unit. The original entity returns to the vehicle assignment queue, to await allocation to another AGV.

The new entity then moves the vehicle to its location, as described in Avoiding Vehicle Collisions. Since all vehicle movement is controlled simply by moving the vehicle to the next station en route to the job's location or final destination, the code which controls any vehicle movement can be used for all cases, simply by changing the final destination. For the new entity to redirect the AGV to its station, then, it undergoes the same sequence of events as any other entity moving a vehicle, with the final vehicle destination set to the entity's pickup location.

4.7 Modeling Vehicle Staging

Some AGV systems employ a staging control scheme, such that all idle vehicles are sent to a common area, where they await their next assignment. In this case, the simulation model must be able to detect when an AGV has no work awaiting it, and then send it to the staging station. When an entity arrives at its final destination and is unloaded from the AGV, it checks all vehicle assignment queues. If there are no other entities waiting for pick-up by a vehicle, then a dummy entity is spawned to return the AGV to the staging area. This dummy entity simply sends the AGV through the same sequence of steps as in any other transport, until the vehilce arrives at the staging area. There, the entity frees the AGV and is disposed.

4.8 Modeling Vehicle Failure

When an AGV fails somewhere on the track, other vehicles typically cannot pass it, but rather must wait until the vehicle is repaired or moved out of the way. If there are many vehicles in the system, failures have the potential to cause significant reduction in performance, and may impact the number of vehicles required to meet production demands.

The structure of the simulation model presented allows vehicle failures to be represented by simply stopping the vehicle at the end of a track zone. In the physical system, AGVs can break down anywhere on the track. However, since the zoning concept is relied on in the simulation model for any vehicle movement, limiting failures to the end of the zone should not provide significantly different results.

Breakdowns of any resource in a simulation model, whether it is a machine, worker, or material handling device, are most often treated by creating submodels which schedule breakdown events to occur according to a random distribution. In the AGV model, breakdowns are accomplished by halting the vehicle at the end of its transport. When the vehicle arrives at the end of a zone, it can either examine a variable which tells it whether it should stop where it is (the variable is set by the entity in the breakdown submodel), or if the facility is available in the modeling language, the vehicle itself may be halted, holding the track zone resource until it is repaired. While it is halted on the track, no other vehicles may enter that zone, since the resource controlling the zone status is held by the stopped AGV.

If vehicle repair entails moving the AGV to the side and repairing it, then a special event is added to the model to relinquish the track resource that the AGV was on when it failed, and later to re-seize that zone when the vehicle is reactivated. If the failure can be quickly repaired, such as a vehicle losing the track signal, then a simple delay by the entity controlling the AGV should be sufficient. However, if the maintenance time is lengthy, special code is required to determine what to do with the part (e.g. remove it manually or dispatch another vehicle to pick it up).

Failures of one AGV presents a special control problem when moving other vehicles, also. More advanced systems include the capability to examine a track section for traffic and/or failed vehicles, and to redirect the unit to a secondary route if there is congestion in the primary path. To accommodate this in the simulation model, data describing alternate paths must be entered, either as a secondary from-to matrix or a series of lists. Whenever a vehicle enters an intersection, it checks for congestion on the path ahead, and if there is significant traffic or a failed vehicle, it follows the secondary path (if one exists).

4.9 Collecting Useful Statistics

Many types of statistics may be kept on simulation runs for analysis of AGV systems. The most common include utilization of the vehicles and some measure of AGV blocking. The utilization statistics are collected by use of time-persistent statistics on status variables. In some languages, vehicle status is updated automatically; if this is not the case, global variables should be set aside. The average values of these statistics record the percent of time that individual vehicles were idle, moving empty, moving loaded, failed, etc.

AGV blocking is measured for the model constructs presented through queue statistics. The average of a time-persistent statistic marking the number of entities in a track resource queue provides the percent of time any vehicle was blocked at that station, since no more than one entity ever would have entered the queue at a time. This blocking information is a valuable measure of system congestion. Other job data, such as throughput, waiting times, flowtimes, etc. may be collected on the simulation runs as well.

10. CONCLUDING REMARKS

As companies increase the amount of automation used in their manufacturing facilities, the need for flexible material handling systems has arisen. Automated Guided Vehicle (AGV) systems have been employed successfully in many industries. However, designing these systems, in which the interactions among the components can have significant impact, is a difficult task. Computer simulation has emerged as the key tool for evaluating the performance of AGV systems.

This paper presented a number of characteristics of AGV systems which are somewhat difficult to model in general-purpose simulation languages, and methods for accurately solving the problems. Some approximations were made for certain control elements; however, these represent the physical system closely enough that they do not detract from the model's performance. AGV system tracks are divided into zones for modeling purposes. The size of the zones determines how closely the simulation resembles its physical counterpart. Because almost the entire system description is kept in data tables, changing the track layout or other system parameters is relatively easy, requiring only data modifications, rather than alterations to the model code. The data-driven nature of the model structure suggests an interactive graphics front-end for data input.

REFERENCES

Egbelu, Pius J. (1982). A Design Methodology for Operational Control Elements for Automatic Guided Vehicle Based Material Handling Systems. Unpublished Ph.D. dissertation, Dept. of Industrial Engineering and Operations Research, Virginia Polytechnic Institute and State University, Blacksburg, Virginia.

Egbelu, Pius J. and Tanchoco, Jose M. A. (1984). Characterization of Automatic Guided Vehicle Dispatching Rules. *International Journal of Production Research* 22, 359-374.

Egbelu, Pius J. and Tanchoco, Jose M. A. (1983). Designing the Operations of Automatic Guided Vehicle System using AGVSim. *Proceedings of the 2nd International Conference on Automated Guided Vehicle Systems*, Stuttgart, W. Germany .

Frazelle, Ed (1985). Suggested Techniques Enable Multi-Criteria Evaluation of Material Handling Alternatives. *Industrial Engineering*, 42-48.

Glenney, Neil E. and Mackulak, Gerald T. (1985). Modeling and Simulation Provide Key to CIM Implementation Philosophy. *Industrial Engineering*, 76-94.

Grosseschallau, W. and Heinzel, R. (1983). A New Planning Method for AGVS with Computer Graphics. *Proceedings of the 2nd International Conference on Automated Guided Vehicle Systems*, Stuttgart, W. Germany.

Harmonosky, Catherine M. and Sadowski, Randall P. (1984). A Simulation Model and Analysis: Integrating AGVs with Non-Automated Material Handling. *Proceedings of the 1984 Winter Simulation Conference*, 341-347.

Maxwell, W. L. and Muckstadt, J. A. (1982). Design of Automatic Guided Vehicle Systems. *IIE Transactions* 14, 114-124.

Newton, Dave (1985). Simulation Model Calculates How Many Automated Guided Vehicles Are Needed. *Industrial Engineering*, 68-78.

Roth, Brenda S. (1984). An Investigation of an Automatic Guided Vehicle System. *1984 Annual International Industrial Engineering Conference Proceedings*, 218-224.

AUTHOR'S BIOGRAPHY

DEBORAH A. DAVIS is a Software Engineer with Systems Modeling Corp. Since joining Systems Modeling in 1984, she has been involved in development of the CINEMA animation system and applying SIMAN to industrial problems. She received a B.S. in Industrial Engineering in 1984 and is a candidate for an M.S. degree in Industrial Engineering and Operations Research at The Pennsylvania State University. Her current interests include simulation language development, integrating animation with simulation, and analysis of automated material handling systems. She is a member of Tau Beta Pi, Alpha Pi Mu, and IIE.

Deborah A. Davis
Systems Modeling Corp.
Calder Sq. P.O. Box 10074
State College, PA 16805
(814) 238-5919