Proceedings of the 1995 Winter Simulation Conference
ed. C. Alexopoulos, K. Kang, W. R. Lilegdon, and D. Goldsman

DEVELOPMENT OF A GENERIC SIMNET II SIMULATION PACKAGE FOR UNIDIRECTIONAL, SIDELOADING AGV SYSTEMS

Roger McHaney

College of Business Administration
101 Calvin Hall
Kansas State University
Manhattan, KS 66506

ABSTRACT

The use of simulation in the design stages of complex materials handling systems has become increasingly prevalent in recent years. Computer simulation has become so important to manufacturing system designs that Pritsker (1992) calls simulation *"the premier technique of industrial engineering"*. It is upon this foundation the project described in this paper has been constructed. This project describes the development of a generic simulation which can be used to model unidirectional, sideloading automatic guided vehicle (AGV) systems. Guidepath, programming logic and queuing logic are all represented through databases. New systems can be input only through changes to the database. The simulation system was developed using the SIMNET II modeling language (Taha 1988, 1992). An implementation example is also provided.

1 INTRODUCTION

Discrete event simulation has traditionally been used as a tool for studying the behavior of dynamic real-world systems. This is particularly true in the instances of manufacturing and materials handling applications. This project describes the development of a generic AGV system simulation written with the SIMNET II modeling language. The developed system is fully functional but limited to particular types of AGV systems. It is intended to be used with unidirectional, sideloading vehicles.

The use of a package such as this will allow rapid modeling of proposed AGV vehicle systems. It should be used in the conceptual design stages of engineering projects. Simulations with greater levels of detail can be constructed based on this framework.

2 AGV SYSTEMS

AGVs are generally battery-powered, driverless vehicles that travel along paths which consist of wires buried in the floor. They are used for materials handling tasks. A central computer or distributed PLC system dispatches empty vehicles, optimize routing, inhibits collisions, prevents system gridlocks, and provides empty vehicle management. Part of the mechanism for performing these tasks are magnet codes (or telsor cards) located in the floor at regular intervals and prior to intersections. These codes (also known as control points) maintain vehicle separation and identify vehicle location to the central computer. Vehicles will often have sonic or optic sensors to prevent collisions with path obstructions and other AGVs. Communications with the central computer will usually be implemented with radios or transmitted through the guidepath wires buried in the floor (Sadowski 1987; Quinn 1987).

Simulations of AGV systems consist of several major parts. These are:

1) Floor Logic
2) Vehicle Programming Algorithms
3) Through-put Modeling

Each part of a general AGV simulation is discussed in detail.

2.1 Floor Logic

Floor logic can be broken into two distinct categories: vehicle routing and a control point layout. Each category represents data which needs to be incorporated into the simulation system.

2.2 Control Points

Control points are placed to facilitate vehicle movement through the system. The idea is to allow vehicles to travel as freely as possible and at the same time avoid collisions with other AGVs. The guidepath is reproduced as a system of control points connected by line segments. Vehicles move along one segment at a time. At each control point they receive instructions to either stop and wait for the segment ahead to clear, pick up or drop off a load, turn a corner, advance to a new destination, or simply continue straight ahead. Segments are represented by switches and facilities. The facilities block following AGVs during movement. The switches are used to block following AGVs during queuing, pick-ups or drop-offs.

2.3 Routing

Vehicle routing is logic associated with each guidepath intersection. This logic enables the AGV to arrive at its destination in the most efficient manner. Most often this means taking the shortest path, however at times a longer route through a less traffic-congested area is more advantageous. These algorithms can be implemented using a database. An AGV's next point is selected based on its current location and destination. Table 1 illustrates a route database.

Table 1: Route Database

```
!-----------------------------------------!
!   ROUTE DATABASE                        !
! 1 - CURRENT AGV LOCATION                !
! 2 - AGV DESTINATION (0-DOESN'T MATTER)  !
! 3 - NEXT LOCATION FOR AGV               !
!-----------------------------------------!
     ROUTES;1-10/NS/5,1,1;
                    5,0,6;
                    1,0,7;
                    6,0,7;
                    7,2,2;
                      .
                      .
                      .
                    21,0,5:
```

Once a route has been established, segment distances are obtained from a segment database. Travel time is calculated by dividing segment length by AGV speed. Table 2 illustrates.

Table 2: Segment Database

```
!-----------------------------------------!
!  SEGMENT DATABASE                       !
!  1 - CURRENT AGV LOCATION               !
!  2 - NEXT AGV LOCATION                  !
!  3 - DISTANCE OF SEGMENT                !
!-----------------------------------------!
       SEGMENTS;1-10/NS/1,7,15;
                        2,9,15;
                        6,7,10;
                        7,8,10;
                          .
                          .
                          .
                        21,5,15:
```

2.4 Vehicle Programming Algorithms

In order for an AGV system to provide peak service, it is crucial to develop an algorithm for moving the product. This algorithm usually involves two major areas of concern: dispatching vehicles to pick up loads and managing empty vehicles in a manner that minimizes deadhead (empty vehicle) travel.

2.5 Parking and Empty Vehicle Management

Following completion of a load move, a vehicle at a drop-off point checks for any loads to be picked up at qualifying stations. If no loads are ready to be picked up, the vehicle is assigned an intermediate or final parking location. Both the qualifying stations and the parking locations are represented in terms of databases.

When a vehicle arrives at an intermediate parking location, a query of qualifying pick-up stations continues. If no loads are ready to be picked up, the vehicle either advances to the next parking location or remains at its current location until a load does qualify.

Table 3 illustrates the AGV programming database. It is keyed by load drop-off or parking locations and shows which pick-up stations qualify for service when a vehicle is at the current location. If no loads are ready to be picked up, the vehicle performs the parking logic. Zeroes are used to fill in positions with no qualifying stations.

Table 3: AGV Parray

```
!-----------------------------------------!
! PROGRAMMING ARRAY DATABASE              !
!   1 -  CURRENT AGV LOCATION             !
!   2 THROUGH N -                         !
!       PICK-UP LOCATIONS AGV QUALIFIES   !
!       TO BE ASSIGNED                    !
!                  FOR LOAD PICK UP.      !
!-----------------------------------------!
      PARRAY;1-10/NS/1,1,0,0;
                     2,2,0,0;
                     3,0,0,0;
                     4,0,0,0;
                     5,0,0,0:
```

Table 4 illustrates the vehicle queuing array. It is keyed by parking or drop-off locations. The potential parking locations are listed in order of preference. The last entry in a list is either be a final vehicle queuing location or an infinite length queue. Zeroes are used to fill in positions with no qualifying stations.

Table 4: AGV Qarray

```
!-------------------------------------------!
!  QUEUING ARRAY DATABASE                   !
!  1 -  CURRENT AGV LOCATION                !
!  2 THROUGH N -  QUEUING LOCATIONS AGV     !
!    QUALIFIES TO BE ASSIGNED FROM MOST TO  !
!    LEAST PREFERABLE. LAST LOCATION        !
!    SHOULD BE A DEFAULT LOCATION WITH      !
!    INFINITE QUEUE LENGTH.                 !
!-------------------------------------------!
      QARRAY;1-10/NS/1,1,0,0,0;
                   2,2,0,0,0;
                   3,5,0,0,0;
                   4,5,0,0,0;
                   5,2,1,5,0:
```

To send vehicles to queues in an orderly fashion, a vehicle queue length database is implemented. The number of AGVs allowed enroute or residing at any particular queue is controlled. Table 5 contains a sample queue length database.

Table 5: Queue Length Database

```
!-------------------------------------------!
!  QUEUE LENGTH DATABASE                    !
!     1 -  QUEUE LOCATION POINT             !
!     2 -  QUEUE SIZE LIMIT. AT LEAST ONE   !
!          QUEUE SHOULD BE INFINITE OR ALL  !
!          QUEUES SHOULD SUM TO NUMBER OF   !
!          AGVS IN SYSTEM.                  !
!     3 -  DYNAMIC - HOLDS CURRENT CONTENTS !
!-------------------------------------------!
       QLENGTH;1-10/NS/1,1,0;
                      2,1,0;
                      3,0,0;
                      4,0,0;
                      5,9,0:
```

Table 6 is a list of all points in the system where AGVs may pick, drop or queue. The simulation uses this to move the AGV transaction to other segments of logic in the program.

Table 6: Instruction Points

```
!-------------------------------------------!
!  LIST OF LEGAL PICK, DROP OR QUEUE POINTS !
!  WHERE AGV MAY RECEIVE PROGRAMMING        !
!  INSTRUCTIONS                             !
!-------------------------------------------!
       PRGLOC;1-10/NS/1;
                     2;
                     3;
                     4;
                     5:
```

2.6 Through-put Modeling

The purpose of an AGV system is to move loads from one stand to another. This system allows loads to be generated using an exponential rate specified by the analyst. The rate is stored in a constant called LOADRATE. The distribution of load pick-up stands is controlled by discrete-pdf 1. The load drop-off distribution is controlled by discrete-pdf 2. Loads are tracked through queues called PICKQ() and a *time-in-system* variable which monitors time from load creation until an AGV is programmed to pick it up. Complicated functions can be implemented by changing the discrete-pdfs.

3 EXAMPLE AGV SYSTEM

An AGV system model was created using the generic AGV simulator. Figure 1 depicts the general layout tested. Although the data used in this model has been contrived, real world data for a system such as this could be collected at the facility an AGV was to be installed. AGV vendors would aid their potential clients in the development of these throughput figures.

The goal of this simulation is to determine an AGV count to maximize through-put and prevent any loads from waiting for assignment too long. A further constraint to use no more than six AGVs. Idle time parking areas are limited and the client doesn't wish to have too many vehicles occupying floor space.

3.1 Operational Assumptions

The system operates according to the following rules:

1) *Loads arrive at either point 1 or 2 at an exponential rate with a mean of 35 seconds.*

2) *Loads at station 2 are high priority. Loads at station 1 have a regular priority.*

3) *Point five is a queuing position. AGVs are dispatched from here to park at either load pick-up point (1 or 2). When both pick-up queues are full, the AGVs are queued at five.*

4) *Loads are dropped at either point 3 or 4.*

5) *AGVs travel at 160 feet per minute.*

6) *Load pick-up times are 1 minute. Load drop-off times are also 1 minute.*

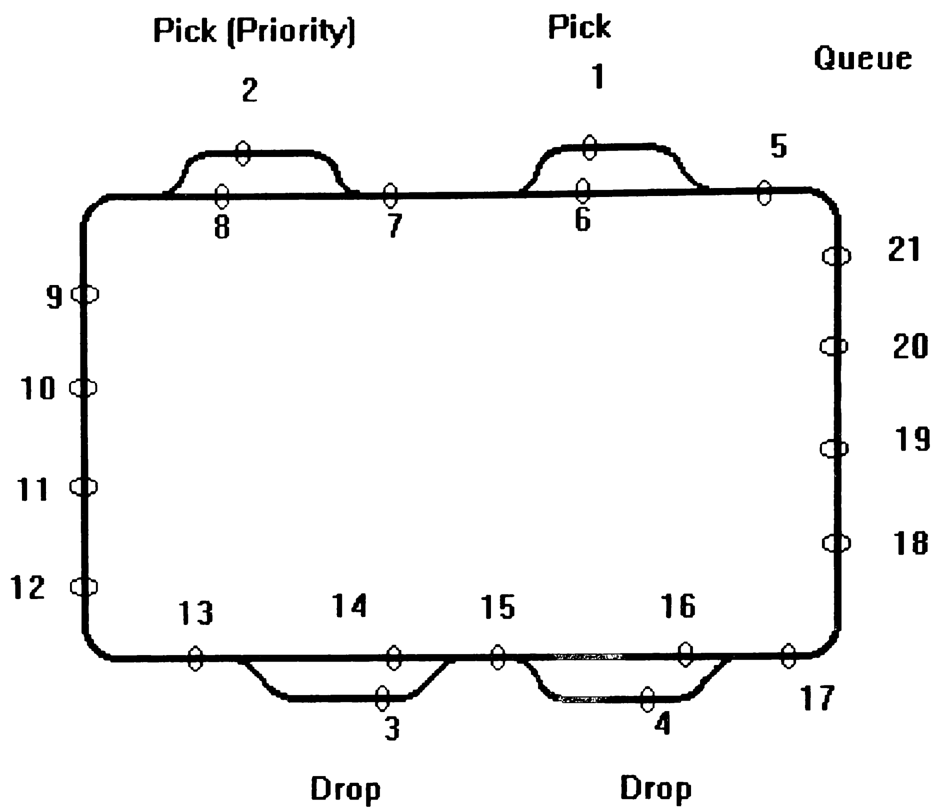

Figure 1 : Example AGV System Guidepath

7) *Vehicles maintain a separation of about ten feet when traveling. This distance is increased on corners.*

8) *Battery charging and maintenance is conducted off-shift.*

9) *The system runs for 1000 minutes.*

10) *The customer wishes to purchase no more than six AGVs.*

3.2 Results

The simulation was run with AGV counts of four, five and six. A transient period of 400 minutes was determined to exist. The replication method was used to gather statistics. Ten runs for each AGV count were completed and statistics gathered. Load time in system and pick-up queue lengths were tracked for the runs with 4, 5 and 6 AGVs. Tables 7, 8, and 9 list the results.

Table 7: Load Time in System (in seconds) vs. AGV Count

AGV Count	Load Time in System		
	Average	Maximum	S.D.
4	131.92	824.06	78.59
5	96.57	681.46	85.89
6	41.40	470.39	43.87

Table 8 : Pick-up Stand 1 Queue Length vs. AGV Count

AGV Count	Pick-Up 1 Queue Lengths			
	Average	Maximum	Last	S.D.
4	7.26	26	1	4.81
5	4.44	17	0	4.37
6	1.62	12	2	2.01

Table 9: Pick-up Stand 2 Queue Lengths vs. AGV Count

AGV Count	Pick-Up 2 Queue Lengths Average	Maximum	Last	S.D.
4	0.01	2	1	0.01
5	0.02	2	0	0.02
6	0.08	5	0	0.11

T-tests were performed to determine if any significant differences resulted in the model due to changes in vehicle count. Table 10 summarizes the findings.

Table 10: T-Test Results

Comparison	T-value	Conclusion
4 AGV vs. 5 AGV Load Time in System	129.4	Significant
4 AGV vs. 6 AGV Load Time in System	130.5	Significant
5 AGV vs. 6 AGV Load Time in System	95.3	Significant
4 AGV vs. 5 AGV Pick-up Queue 1 Average Length	5.2	Significant
4 AGV vs. 6 AGV Pick-up Queue 1 Average Length	6.3	Significant
5 AGV vs. 6 AGV Pick-up Queue 1 Average Length	3.4	Significant
4 AGV vs. 5 AGV Pick-up Queue 2 Average Length	2.6	Barely Significant
4 AGV vs. 6 AGV Pick-up Queue 2 Average Length	2.1	Not Significant
5 AGV vs. 6 AGV Pick-up Queue 2 Average Length	2.1	Not Significant

Note: T Critical Values for all tests is 2.1 (df=18, alpha=.05, 2-tail test)

As demonstrated, the six AGV system performs significantly better in all categories except for queue 2 lengths. This is the case because queue 2 was given high priority and queue 1 was the area more likely to be starved for AGV service.

4 LIMITATIONS

Although this AGV simulator is meant to be general purpose, it does not allow all the complexities of all AGV systems to be modeled. To make it more comprehensive, several other items can be added. The most important of these is a database for complex blocking. By adding a database of resources required by AGVs when traveling through particular line segments, complex intersection logic can be modeled. This would also allow AGV backups into spurs to be modeled.

Other additions to the system adding to its comprehensive ability to model most AGV systems should include, a gridlock avoidance database, battery charging modeling, ability to code (in database form) inhibiting conditions (e.g. boolean logic statements preventing queue or pick-up assignments when certain system states exist).

Work on the systems bringing loads to the AGV network should also be upgraded. Conveyor, AS/RS or manual load delivery logic should be made more comprehensive.

5 SUMMARY

The widespread use of simulation in the design stages of materials handling systems has warranted the development generic simulators capable of rapid modeling and testing of ideas. This project has provided a simple, unidirectional AGV simulator capable of modification through only database changes. An example system was developed using the simulator and tested in a project. The example system tested vehicle counts of four, five and six. The six vehicle system outperformed the other two vehicle counts. More importantly, the example system demonstrates the utility of having a generic AGV simulator.

REFERENCES

Pritsker, A.A.B. 1992. Simulation: The Premier Technique of Industrial Engineering. *Industrial Engineering* 7: 25-26.

Quinn, E.B. 1987. Discrete Event Computer Simulation of AGV Networks. *In Proceedings of AGVS '87*, ed. G. Hilty, 43-51. Materials Handling Institute Charlotte, North Carolina.

Sadowski, R. 1987. Computer Simulation of Automatic Guided Vehicles. *In Proceedings of AGVS '87*, ed. G. Hilty, 21-41. Materials Handling Institute, Charlotte, North Carolina.

Taha, H.A. 1988. *Simulation Modeling and SIMNET*. Englewood Cliffs : Prentice-Hall.

Taha, H.A. 1992. *Simulation with SIMNET II*. SimTech, Fayetteville, Arkansas.

AUTHOR BIOGRAPHY

ROGER MCHANEY is an Assistant Professor in the School of Business Administration at Kansas State University. He received his Ph.D. in Computer Information Systems and Quantitative Analysis from the University of Arkansas in 1995. His current research interests are in the areas of business applications of computers simulation, AGV system simulations and factors affecting computer simulation success. For eight years prior to returning to academics, Dr. McHaney worked as a simulation analyst for the Jervis B. Webb Company. He is author of a text published by the Academic Press entitled *Computer Simulation: A Practical Perspective*.