

## Research Article

# Analyzing the Diverging Diamond Interchange Using Discrete Event Simulation

Michael Anderson,<sup>1</sup> Bernard Schroer,<sup>2</sup> and Dietmar Moeller<sup>3</sup>

<sup>1</sup> Department of Civil Engineering, University of Alabama in Huntsville, Huntsville, AL 35899, USA

<sup>2</sup> Center for Management and Economic Research, University of Alabama in Huntsville, Huntsville, AL 35899, USA

<sup>3</sup> Department of Computer Science, University of Hamburg, Vogt-Keolln-Straße 30, 22527 Hamburg, Germany

Correspondence should be addressed to Michael Anderson, andersmd@uah.edu

Received 23 April 2012; Revised 19 October 2012; Accepted 20 October 2012

Academic Editor: Azah Mohamed

Copyright © 2012 Michael Anderson et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The diverging diamond interchange (DDI) can improve traffic flow by limiting the number of phases in the traffic signals and improve safety by eliminating left turns. A few installations of these interchanges have been constructed and there is great potential to construct more. In an effort to develop a methodology to evaluate these interchanges, this paper presents the development of a discrete event simulation model of the diverging diamond interchange (DDI). Specific emphasis is on using simulation to model the DDI, a description of the operation of the simulation model, and using simulation to understand the operation of the DDI. The paper concludes that the use of the simulation package allows for rapid evaluation of the DDI and demonstrates that this interchange design will not work in all locations.

## 1. Introduction

One of the major causes of accidents at interchanges is turning left into oncoming traffic. This is especially true on the overpasses where vehicles turn left to enter the interstate. One solution to this problem is the diverging diamond interchange (DDI). The DDI eliminates the left hand turns into free flowing oncoming traffic. The DDI uses traffic lights to allow vehicles to briefly shift to the left side of the road. The vehicles can then directly turn left on the access ramp without facing oncoming traffic.

The first diverging diamond interchange in the United States opened in 2009 in Springfield, Missouri. Figure 1 is a photograph of the Springfield DDI. No new bridges or elevated lanes were required. Initial results indicate that the new interchange reduced total crashes by 46%, left turn crashes were eliminated, and rear-end crashes down slightly [1]. The study also indicated that over eighty percent of the public expressed that the traffic flow had improved and traffic delays had decreased.

This paper presents the development of a discrete event simulation model of the diverging diamond interchange.

Specific emphasis is on the description of the operation of the simulation model and the use of simulation to understand the operation of the DDI. The hypothesis of the paper is to determine if a non-transportation simulation program can effectively model the DDI and simulate delays for a variety of test cases to demonstrate that the DDI is not the appropriate solution for all locations. The experimental design is to establish a DDI in the simulation program and develop cases to adjust traffic and monitor the response of the simulation.

The simulation model is written in ProcessModel [2]. ProcessModel was selected because of its ease of use and for its building block architecture. ProcessModel has the following building blocks: entities, activities, stores, resources, and labels. Entities, activities, and labels were used in constructing the DDI model. Entities simulate the arrival of vehicles. Activities simulate the processing, or movement, of the entities through the DDI. The action logic section of the activities was used to embed complex logic for controlling and routing vehicles through the DDI. The label blocks were used for displaying critical results during the running of the simulation to observe the operation of the DDI and for model verification and validation.



FIGURE 1: Diverging diamond interchange in Springfield, Missouri.

## 2. Previous Research

The first known diverging diamond interchanges were located in France. The diverging diamond interchange configuration was introduced to America in a paper by Chlewicki at the 2nd Urban Street Symposium held in Anaheim, California [3].

The Federal Highway Administration has modeled the Springfield, Missouri DDI using the Highway Driving Simulator to evaluate the human factor aspects of the proposed design [4]. The Federal Highway Administration has also released a technical brief covering four intersection designs and two interchange designs that offer substantial advantages of conventional at-grade intersections and grade-separated diamond interchanges [5].

Chilukuri et al. [1] have conducted a performance evaluation of the diverging diamond interchange at Springfield, MO. The evaluation assessed traffic operations, safety, and public perceptions. Other researchers have been studying various aspects of DDI including construction, operation and application [6, 7].

Anderson et al. [8] have developed an initial ProcessModel of the Diverging Diamond Interchange. This paper evaluated the following factors that affect traffic through a DDI: traffic volume, length of time traffic light is green, and the criteria for changing the traffic signal.

## 3. ProcessModel

A ProcessModel was developed of the DDI given in Figure 2. The interstate traffic travels East and West. The overpass traffic travels North and South. There are a total of twelve traffic routes through the DDI. The routes are from S to N, S to W, S to E, N to S, N to E, N to W, W to N, W to S, W to W, E to S, E to N, and E to E. The W to E and E to W interstate through traffic was not modeled since this traffic does not enter the DDI.

The ProcessModel has the following vehicle entities, one for each of the previous mentioned traffic routes.

Cars\_Arrivals\_from\_S  
Cars\_Arrivals\_from\_W  
Cars\_Arrivals\_from\_N

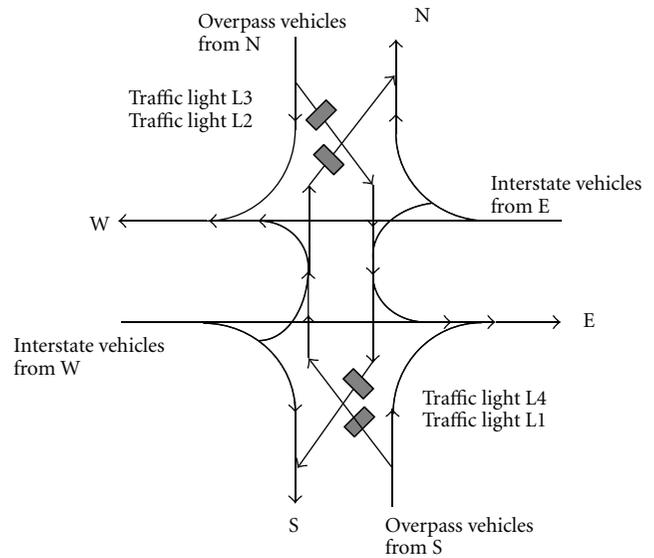


FIGURE 2: Diverging diamond interchange.

Cars\_Arrivals\_from\_E  
S\_Turn\_E  
S\_Turn\_W  
W\_Turn\_S  
W\_Turn\_W  
N\_Turn\_W  
N\_Turn\_E  
E\_Turn\_N  
E\_Turn\_E

The ProcessModel has the following global variables.

LIGHT $i$  ( $i = 1, 2, 3, 4$ ) values = GREEN or RED  
CARSL $i$  ( $i = 1, 2, 3, 4$ ) number of cars waiting at light  
XTIME $i$  ( $i = 1, 2, 3, 4$ ) clock time when light turns red  
DELTA $T_i$  ( $i = 1, 2, 3, 4$ ) time car has waited =  $CLOCK() * 60 - XTIME_i$   
NRCARS $i$  ( $i = 1, 2, 3, 4$ ) number of cars waiting at light  
LIGHTTIME minimum time before light changes  
MAXCARS minimum number of cars waiting before light changes

The ProcessModel has one entity attributes: NRTHRU $i$  ( $i = 1, 2, 3, 4$ ) serial number assigned to each entity. The ProcessModel has a number of label blocks for displaying the values of global variables during the execution of the simulation. These labels are used extensively for observing the operation of the model and for model validation and verification. The label blocks are as follows.

CARSL $i$  ( $i = 1, 2, 3, 4$ ) number of cars waiting at lights  
L $i$  ( $i = 1, 2, 3, 4$ )

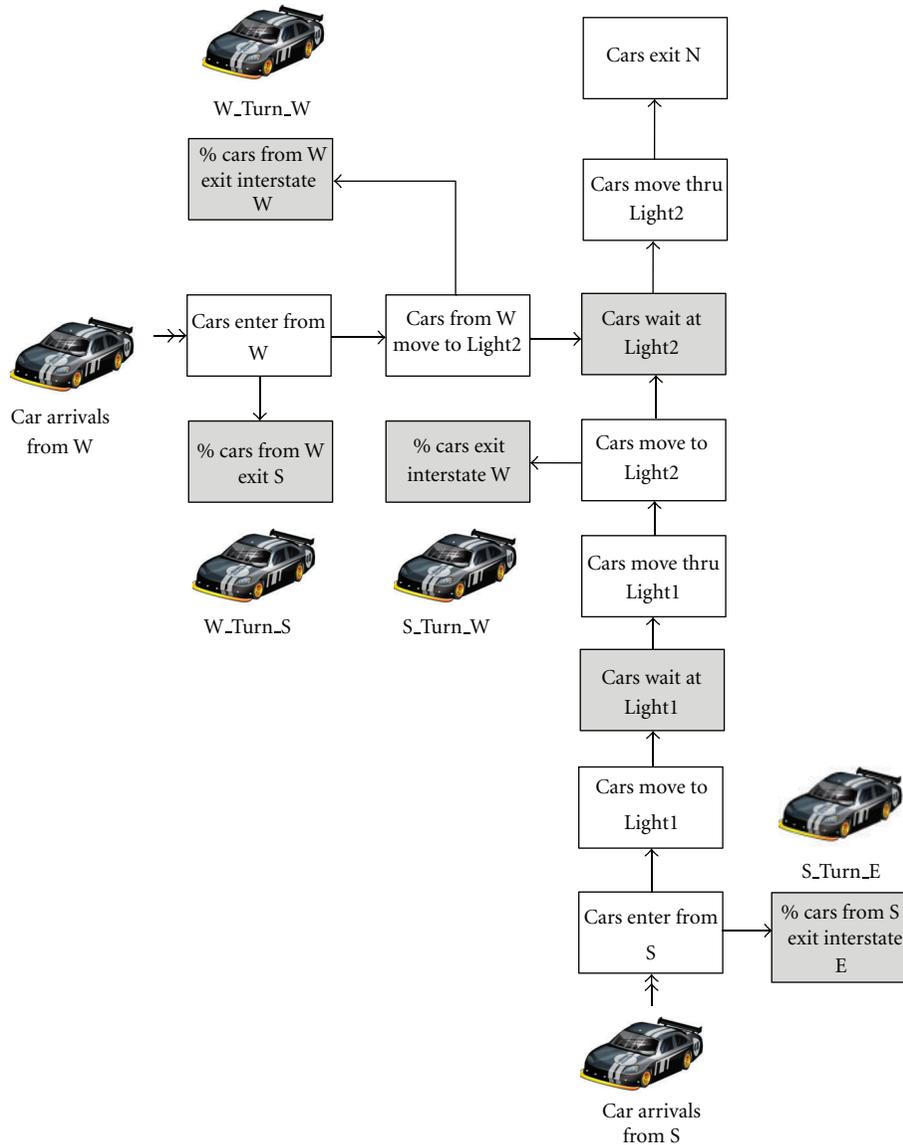


FIGURE 3: Traffic flow submodel.

DELTA $T_i$  ( $i = 1, 2, 3, 4$ ) time car has waited at lights  $L_i$  ( $i = 1, 2, 3, 4$ )

LIGHT $i$  ( $i = 1, 2, 3, 4$ ) values = GREEN or RED

#### 4. ProcessModel Traffic Flow Submodel

Figure 3 is the ProcessModel for the traffic flow submodel for vehicles entering the DDI from the south and west. The shaded blocks in Figure 3 are dummy activities and used only for routing entities and for adding logic statements. There is a similar traffic flow submodel for traffic entering the DDI from the north and east. The submodel simulates S to N, S to W, S to E, W to N, W to S, and W to W traffic.

The ProcessModel in Figure 3 also serves as an excellent flow chart for documenting the traffic through the DDI. Therefore, the traffic flow is as follows. Vehicles enter the model at activity named Cars\_Enter\_from\_S. The vehicle

entities are named Car\_Arrivals\_from\_S. The vehicle time at this activity is  $T(4,5,6)$  seconds. The dummy activity %\_Cars\_Exit\_Interstate\_E is for routing vehicles eastbound onto the interstate. No time is spent at this activity. These vehicle entities are renamed S\_Turn\_E so statistics can be collected on the amount of time these vehicles are at the DDI. These entities then exit ProcessModel.

The remaining vehicles continue northbound to activity named Cars\_Move\_to\_Light1. The mean vehicle time at this activity is  $T(4,5,6)$  seconds. Vehicles then move to activity named Cars\_Wait\_at\_Light1 where the vehicles wait until traffic Light1 turns green. This activity has no processing time. The logic at the activity is as follows.

```

INC NRCARS1,1
NRTHRU1 = NRCARS,1
INC CARSL1,1
    
```

```

DELTAT1 = CLOCK()*60-XTIME1
WAIT UNTIL LIGHT1 = GREEN
WAIT UNTIL LIGHT4 = RED

```

Global variable NRCARS1 is a counter of the vehicles waiting at the light. Entity attribute NRTHRU1 is a serial number assigned to the vehicles that start over at one every time the light changes. Global variable CARSL1 is a counter of the number of cars waiting at Light1 and is incremented by one. Global variable DELTAT1 determines the amount of time the vehicle entity has been waiting by subtracting the current clock time from XTIME1 which is the clock time when Light1 turned red. The vehicle entity will then wait until Light1 is green and Light4 is red.

Once Light1 is green vehicles will move to activity CarsMovethru\_Light1. This activity has a capacity of two indicating that only two vehicles can go through Light1 at a time. Other vehicles will have to wait. The logic at the activity is as follows.

```

WAIT UNTIL LIGHT1 = GREEN
TIME(T(5,6,7) SEC)
IF NRTHRU1 = 1 THEN TIME(4 SEC)
IF NRTHRU1 = 2 THEN TIME(4 SEC)
IF NRTHRU1 = 3 THEN TIME(4 SEC)
IF NRTHRU1 = 4 THEN TIME(4 SEC)
IF NRTHRU1 = 5 THEN TIME(3 SEC)
IF NRTHRU1 = 6 THEN TIME(3 SEC)
DEC CARSL1,1

```

The time through the traffic light is  $T(5,6,7)$  seconds. The first six vehicles waiting at the light require an acceleration time. The third through eight lines of the logic is for this acceleration time. The first four vehicles take an additional four seconds and the next two vehicles an additional three seconds to accelerate. The global variable CARSL1 is decremented by one whenever a vehicle exits the traffic light.

Vehicles then move to activity named Cars.Move\_to\_Light2. The vehicle time at this activity is  $T(4,5,6)$  seconds. The dummy activity %\_Cars\_Enter\_Interstate\_W is for routing vehicles westbound onto the interstate. No time is spent at this activity. These vehicle entities are renamed S\_Turn\_W so statistics can be collected on the amount of time these vehicles are at the DDI. These entities exit ProcessModel.

The remaining vehicles then move to activity named Cars.Wait\_at\_Light2 where the vehicles wait until traffic Light2 turns green. The activity has no processing time. The logic at the activity is as follows.

```

INC NRCARS2,1
NRTHRU2 = NRCARS2
INC CARSL2,1
DELTAT2 = CLOCK()*60-XTIME2
WAIT UNTIL LIGHT2 = GREEN
WAIT UNTIL LIGHT3 = RED

```

Global variable NRCARS2 is a counter of the vehicles waiting at the light. Entity attribute NRTHRU2 is a serial number assigned to the vehicles that start over at one every time the light changes. Global variable CARSL2 is a counter of the number of cars waiting at Light2 and is incremented by one. Global variable DELTAT2 determines the amount of time the vehicle entity has been waiting by subtracting the current clock time from XTIME2 which is the clock time when Light2 turned red. The vehicle entity will then wait until Light2 is green and Light3 is red.

Once Light2 is green vehicle entities will move to activity Cars\_Move\_thru\_Light2. This activity has a capacity of two indicating that only two vehicles can go through Light1 at a time. Other vehicles will have to wait. The logic at the activity is as follows.

```

WAIT UNTIL LIGHT2 = GREEN
TIME(T(5,6,7) SEC)
IF NRTHRU1 = 1 THEN TIME(4 SEC)
IF NRTHRU1 = 2 THEN TIME(4 SEC)
IF NRTHRU1 = 3 THEN TIME(4 SEC)
IF NRTHRU1 = 4 THEN TIME(4 SEC)
IF NRTHRU1 = 5 THEN TIME(3 SEC)
IF NRTHRU1 = 6 THEN TIME(3 SEC)
DEC CARSL2,1

```

The time through the traffic light is  $T(5,6,7)$  seconds. The first six vehicles waiting at the light require an acceleration time. The third through eight lines of the logic is for this acceleration time. The first four vehicles take an additional four seconds and the next two vehicles an additional three seconds to accelerate. The global variable CARSL2 is decremented by one whenever a vehicle exits the light.

This submodel also has vehicles exiting the interstate from the west. These entities are named Car\_Arrivals\_from\_W and move to activity Cars\_Enter\_from\_W. The vehicle time at the activity is  $T(4,5,6)$  seconds. The dummy activity %\_Cars\_from\_W\_Exit\_S is for routing vehicles south on the overpass. No time is spent at the activity. These vehicle entities are renamed W\_Turn\_S so statistics can be collected on the amount of time these vehicles are at the DDI. These entities exit ProcessModel.

The remaining vehicles move to activity Cars.from\_W\_Move\_to\_Light2. The mean vehicle time at this activity is  $T(4,5,6)$  seconds. Vehicles then move to activity named Cars.Wait\_at\_Light2 where the vehicles wait until traffic Light1 turns green. The activity has no processing time.

## 5. ProcessModel Traffic Intersection Submodel

Figure 4 is the ProcessModel for the traffic intersection submodel for traffic lights L1 and L4. Again there is a similar submodel for traffic lights L2 and L3.

A dummy entity named Item is created at time zero and will continue to loop through the logic during the

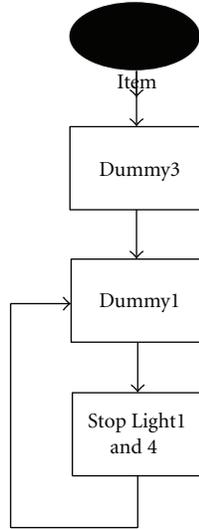


FIGURE 4: Traffic intersection submodel.

simulation. The following global variables are initialized at activity named Dummy.

```

LIGHT1 = RED
LIGHT4 = RED
XTIME1 = CLOCK()*60
XTIME2 = CLOCK()*60
  
```

The traffic lights are initially set to red. XTIME1 and XTIME2 are the clock times when the lights turn red. The logic at activity Stop\_Light1\_and\_4 is as follows.

```

WAIT UNTIL CARSL1 >= MAXCARS OR DELTAT1
>= LIGHTTIME
LIGHT4 = RED
XTIME4 = CLOCK()*60
LIGHT1 = GREEN
NRCARS1 = 0
TIME(90 SEC)
WAIT UNTIL CARSL4 >= MAXCARS OR DELTAT4
>= LIGHTTIME
LIGHT1 = RED
XTIME1 = CLOCK()*60
LIGHT4 = GREEN
NRCARS4 = 0
TIME(90 SEC)
  
```

This logic changes the traffic lights from red to green and vice versa. As the traffic light changes the vehicles in other submodels can move through the interchange. The entity Item waits until a minimum number of vehicles CARSL1 are greater or equal to a defined number of vehicles (MAXCARS) or until the vehicle has waited a minimum time of DELTAT1. Once this condition is satisfied, the traffic LIGHT1 is set to

TABLE 1: Input data for Baseline Run1.

Time between arrivals of vehicles	Value	
From W or E	15 sec	
From N or S	10 sec	
Vehicle routing		
From W to W or E to E	5%	
From W to S or E to N	50%	
From S to E or S to W	25%	
From N to W or N to E	25%	
Minimum time traffic light green	90 sec	
Minimum time before traffic light changes	30 sec	
Minimum number of cars waiting before traffic light changes	10 cars	
Activity	Time	Capacity
Vehicles enter from S	$T(4,5,6)$ sec	10
Vehicles move to Light1	$T(4,5,6)$ sec	20
Vehicles move through Light1	$T(5,6,7)$ sec	2
Vehicles move to Light2	$T(4,5,6)$ sec	20
Vehicles move through Light2	$T(5,6,7)$ sec	2
Vehicles exit N	$T(4,5,6)$ sec	10
Vehicles enter from W	$T(4,5,6)$ sec	10
Vehicles from W move to Light2	$T(4,5,6)$ sec	20
Vehicles enter from N	$T(4,5,6)$ sec	10
Vehicles move to Light3	$T(4,5,6)$ sec	20
Vehicles move through Light3	$T(5,6,7)$ sec	2
Vehicles move to Light3	$T(4,5,6)$ sec	20
Vehicles move through Light4	$T(5,6,7)$ sec	2
Vehicles exit S	$T(4,5,6)$ sec	10
Vehicles enter from E	$T(4,5,6)$ sec	10
Vehicles from E move to Light4	$T(4,5,6)$ sec	20

green and LIGHT4 is set to red. LIGHT1 will remain green for 90 seconds.

LIGHT1 will remain green until the minimum number of vehicles CARSL4 is greater or equal to a defined number of vehicles (MAXCARS) or until the vehicle has waited a minimum time of DELTAT4. Once this condition is satisfied the traffic LIGHT4 is set to green and LIGHT1 is set to red. The traffic LIGHT4 remains green for 90 seconds. After 90 seconds the entity Item branches back to activity Dummy1 and the above logic repeated.

## 6. Baseline Run1

The input data for Baseline Run1 are given in Table 1. The  $T(a,b,c)$  is a triangular distribution with parameters a, b, and c. Parameter “a” is the smallest value, “b” is the most likely value, and “c” is the largest value.

The mean vehicle times through the DDI, assuming no delays and no accelerations, are as follows.

- (i) S to N or N to S  $5 + 5 + 6 + 5 + 6 + 5 = 32$  seconds;
- (ii) S to W or N to E  $5 + 5 + 6 + 5 = 21$  seconds;
- (iii) W to W or E to E  $5 + 5 = 10$  seconds;
- (iv) W to S or E to N  $5$  seconds.

These times are computed by adding all the corresponding ProcessModel activity block mean times in Table 1.

TABLE 2: Input data for Baseline Run1 dummy activities.

Activity	Time	Capacity
% vehicles exit interstate E	0	1
Vehicles wait at Light1	0	1
% vehicles exit interstate W	0	1
Vehicles wait at Light2	0	1
% vehicles from W exit S	0	1
% vehicles from W exit interstate W	0	1
% vehicles exit interstate W	0	1
Vehicles wait at Light3	0	1
% vehicles exit interstate E	0	1
Vehicles wait at Light4	0	1
% vehicles from E exit N	0	1
% vehicles from E exit interstate E	0	1

TABLE 3: Results for Baseline Run1.

Vehicle route	Average time thru DDI (min)	Average delay (min)
S to N	1.23	0.67
S to E	0.08	0.0
S to W	0.84	0.47
N to S	1.23	0.67
N to W	0.08	0.00
N to E	0.83	0.47
W to S	0.08	0.00
W to N	0.80	0.44
W to W	0.16	0.00
E to N	0.08	0.00
E to S	0.80	0.44
E to E	0.16	0.00
Location of delays		
Vehicles from S waiting at L1		0.45
Vehicles from S waiting at L2		0.27
Vehicles from N waiting at L3		0.45
Vehicles from N waiting at L4		0.26

Table 2 lists the ProcessModel activities that are dummy activities and are used to route entities or add logic statements. As a result these activities do not have any processing times and have only a capacity of one.

The Baseline Run1 results are given in Table 3. Overpass vehicles going from S to N and from N to S had an average wait of 40 seconds. Vehicles going from S to W and N to E had an average wait of 28 seconds. Also interstate vehicles going from W to N and E to S had an average wait of 26 seconds.

As anticipated, all vehicle delays occurred at the traffic lights. The average delay at Light1 was 27 seconds, at Light2 16 seconds, at Light3 27 seconds, and at Light4 16 seconds. These values are typical delay values expected when evaluating single traffic lights using standard procedures outlined in the Highway Capacity Manual (HCM 2010).

TABLE 4: Experimental design.

Run	Description
Baseline Run1	See Tables 1 and 2
	Time between arrivals
	E or W 15 sec N or S 10 sec Min time traffic light green 90 sec
Run2	Time between arrivals
	E or W 12 sec N or S 8 sec Min time traffic light green 60 sec
	Time between arrivals
Run3	E or W 12 sec N or S 8 sec Min time traffic light green 30 sec
	Time between arrivals
	E or W 10 sec N or S 6 sec Min time traffic light green 60 sec
Run4	Time between arrivals
	E or W 10 sec N or S 6 sec Min time traffic light green 30 sec
	Time between arrivals
Run5	E or W 10 sec N or S 6 sec Min time traffic light green 30 sec

## 7. Experimental Design

In a previous study of the DDI [8] the following parameters were evaluated: traffic volume through the DDI, length of time traffic lights were green, and rule for changing traffic lights from green to red.

The results of this study were as follows.

- (i) Decrease in length of green traffic lights decreased average vehicle delays.
- (ii) Increase in vehicle traffic through DDI increased average vehicle delay.
- (iii) Criteria for changing light to green had no impact on average vehicle delay.

This previous study also identified several areas for further analysis. One area was the adding of acceleration logic for vehicles waiting at traffic lights. Another area was to determine the maximum vehicle throughput before congestion occurred beyond the normal waiting at traffic lights.

The Baseline Run1 results in Table 3 that included the addition of the acceleration data had very little impact as compared to the Baseline Run1 of the previous study [8]. One reason was that the acceleration was only added to the first six vehicles that were waiting. It was assumed that the other vehicles had already accelerated by the time they were moving through the traffic light.

In an attempt to try to determine a maximum vehicle throughput an experiment was defined that varied several of the parameters simultaneously (see Table 4).

TABLE 5: Results for Runs2–5—average times through DDI.

Vehicle Route	Average delay through DDI (minutes)				
	Run1	Run2	Run3	Run4	Run5
S to N	0.67	0.55	0.64	1.09	4.77
S to E	0.00	0.00	0.00	0.00	0.00
S to W	0.47	0.33	0.20	0.41	0.73
N to S	0.54	0.54	0.58	1.14	7.21
N to W	0.00	0.00	0.00	0.00	0.00
N to E	0.47	0.33	0.18	0.40	0.73
W to S	0.00	0.00	0.00	0.00	0.00
W to N	0.44	0.35	0.43	0.74	4.06
W to W	0.00	0.00	0.00	0.00	0.00
E to N	0.00	0.00	0.00	0.00	0.00
E to S	0.44	0.34	0.40	0.81	6.50
E to E	0.00	0.00	0.00	0.00	0.00

TABLE 6: Results for Runs2–5—average delays at traffic lights.

Traffic Light	Run1		Run2		Run3		Run4		Run5	
	Avg. delay (min)	Queue at end								
Light1	0.45	0	0.32	0	0.19	2	0.38	0	0.62	8
Light2	0.27	0	0.25	0	0.37	0	0.72	3	3.96	96
Light3	0.45	1	0.32	0	0.18	2	0.38	5	0.62	4
Light4	0.26	4	0.24	3	0.33	1	0.69	0	6.38	89

### 8. Simulation Results

Table 5 gives the average vehicle delays for each run. It appears that the average delays decreased for Run2 and decreased further for Run3. However, the delays began to increase for Run4 and then increase significantly for Run5. All the delays were at the traffic lights. The other traffic routes had no delays.

Table 6 gives the average delays at the traffic lights. The delays were the lowest for Run3 and averaged 12 seconds for Light1, 22 seconds for Light2, 11 seconds for Light3, and 20 seconds for Light4. On the other hand, the delays began to increase for Run4 and the system appears to become unstable for Run5, essentially reaching level of service F traffic conditions where the demand is exceeding the capacity of the interchange. The average delays were 36 seconds at Light1 and Light3, 4 minutes at Light2, and 6 minutes at Light4.

The large increase in delays at Light2 and Light4 for Run5 can be attributed to the increase in the arrival traffic from the west interstate going north through traffic Light2 and the east interstate going south going south through traffic Light4. The excessive queue lengths at Light2 and Light4 will continue until the demand drops below the initial input levels, as would happen naturally during typical operation. However, the amount of delay associated with passing through the interchange would make this a non-optimal solution.

Table 6 also gives the current queue contents at the traffic lights at the end of the simulation. Note that for Run5 96 vehicles were waiting at Light2 and 89 vehicles at Light4. Also

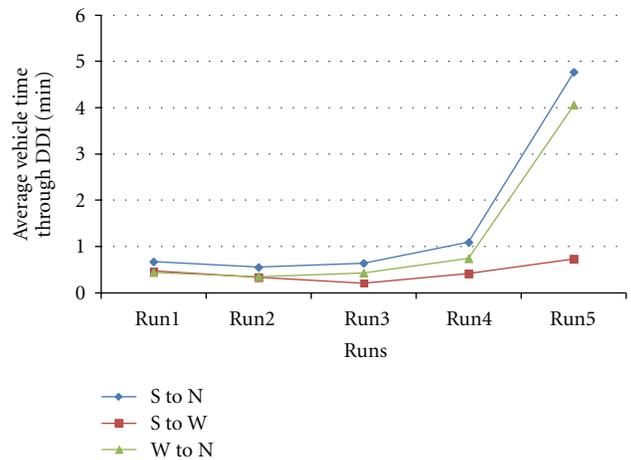


FIGURE 5: Average times through DDI.

for Run5 the queues increased to 8 vehicles for Light1 and 4 vehicles for Light4. This implies that all the vehicles that were waiting for the light to change were not able to move through the intersection when the light was green. As a result these queues will probably continue to increase if the simulation is extended.

The queue content at the street lights for the other runs were low and ranged from zero to five vehicles. These queues were not at the same traffic lights indicating that these were probably normal traffic buildups during the simulation before the traffic lights turned green.

Figure 5 is a plot of the average vehicle time through the DDI for traffic directions S to N, S to W, and W to N. The

other vehicles entering from the N and E had similar times. It appears that the times through the DDI were similar for Runs 1–3. However, once the mean times between arrivals were lowered to 10 seconds for W and E interstate traffic and to 6 seconds for N and S traffic, the vehicle times through the DDI and the corresponding delays increased considerably and probably signal that the system is becoming unstable.

## 9. Conclusions

The following conclusions are made. The modifications to the ProcessModel were rapidly made because of the modular construction. This construction would allow for quick analysis of other locations and evaluations in a timely fashion. It is intended that the simulation presented here would be used in combination with existing simulations of the existing conventional diamond interchange design and existing traffic volumes, essentially allowing the user to determine if a change in interchange structure will benefit the traffic in the area.

Of the five runs Run 3 had the smallest delays at the traffic lights. For Run 3 the time between arrivals was 12 seconds for E and W traffic and 8 seconds for N and S traffic. The traffic lights were green for a minimum of 30 seconds.

The simulation is probably becoming unstable starting with Run 4. For Run 5 the delays for S to N and for N to S traffic were between 5 and 7 minutes. The delays for W to N and E to S were between 4 and 6 minutes. Also the delays for S to W and N to E traffic were 50 seconds. Also the current queue contents for Run 5 were 96 vehicles at Light 2 and 89 vehicles at Light 4 which is another indicator that the system is becoming unstable. The other traffic routes had no delays because these routes did not have traffic lights.

Overall, the diverging diamond interchange can be shown through simulation to be a good potential solution; however, it is not the proper solution in all situations and care must be taken to ensure that the interchange is constructed in the appropriate locations for success.

## Acknowledgments

This research was sponsored by the US Department of Transportation, Federal Transit Administration.

## References

- [1] V. Chilukuri, S. Siromaskul, M. Trueblood, and T. Regan, "Diverging diamond interchange performance evaluation (I-44 & Route 13)," Tech. Rep., HDR Engineering, Organizational Results Research Report for Missouri Department of Transportation, 2011.
- [2] ProcessModel, *Users Manual*, ProcessModel Corporation, Provo, Utah, USA, 1999.
- [3] G. Chlewicki, "New interchange and intersection designs: the synchronized split-phasing intersection and the diverging diamond interchange," in *Proceedings of the 2nd Urban Street Symposium*, Anaheim, Calif, USA, July 2003.
- [4] V. Inman, J. Williams, R. Cartwright, B. Wallick, P. Chou, and M. Baumgartner, "Evaluation of the diverging diamond

interchange," TECHBRIEF FHWA-HRT-07-048, U.S. Department of Transportation, Federal Highway Administration, Washington, DC, USA, 2010, FHWA contact: J. Bared, 202-493-3314.

- [5] W. Hughes and R. Jagannathan, "Double crossover diamond interchange," TECHBRIEF FHWA-HRT-09-054, U.S. Department of Transportation, Federal Highway Administration, Washington, DC, USA, 2010, FHWA contact: J. Bared, 202-493-3314.
- [6] H. Xu, H. Liu, Z. Tian, and W. Zhang, "Control delay calculation at diverging diamond interchanges," *Transportation Research Record: Journal of the Transportation Research Board*, no. 2257, pp. 121–130, 2011.
- [7] G. Chlewicki, "Should the diverging diamond interchange always be considered a diamond interchange form?" *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2223, pp. 88–95, 2011.
- [8] M. Anderson, B. Schroer, and D. Moeller, "Simulation of the diverging diamond interchange," UAH Research Report, University of Alabama in Huntsville, Huntsville, Ala, USA, 2011.

