0. Model assumptions

0.1 Arrival and departure event at an intersection

0.2 Velocity of vehicle

0.3 Type of vehicles

0.4 Behavior of vehicles at sections and intersections.

**1. Input analysis**

1. Arrival rate of vehicle
2. Velocity of vehicle
3. Length of vehicle
4. Probability of right or left turn at an intersection

**1.1 Arrival rate of vehicle**

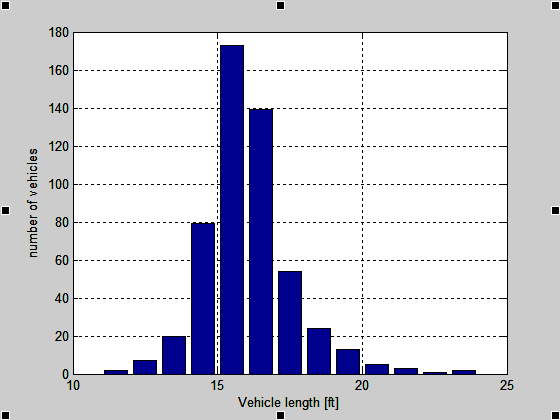
We first investigate how many vehicles arrive at or depart from each intersection. Fig.2 represents number of vehicles that enter the zone of interest from intersections (top) and leave from the zone (bottom). As shown above, more than 83% of vehicles came from the first intersection(Originating zone iD is 101,102,123) or fifth intersection(Originating zone ID is 113, 114, 115). Further, more than 86% of vehicles depart from first(Destination zone ID is 201, 202, 223) or fifth intersection(Destination zone ID is 213, 214, 215). From this result, we focus on the probability of the arrival event at the first and fifth intersection. The inter-arrival time of vehicles at the first intersection is shown in Fig.x. we assume that the distribution of time intervals of the arrival events follows an exponential distribution and estimated the rate parameter at 0.6 from the empirical data. As shown in Fig.x, the number of samples are not large enough to estimate the distribution. Thus, we assume that probability of the arrival of vehicle is uniformly distributed and the probability of occurrence is selected so that the total number of vehicles that arrived at the zone of interest is close the empirical data. Further discussion about the rate of arrival event is shown in Appendix B.

**1.2 Velocity of vehicle**

The average traveling time under a certain traffic signal pattern would depend on the velocity of the vehicle. In addition, distribution of the velocity would influences the congestion condition of the road and it would affect the travel time. Therefore, velocity of vehicle should be modeled as a random variable to precisely predict the traveling time. Since the length of an intersection is shorter than that of road between intersections(i.e. section), we only consider the velocity when a vehicle is driving through a section. Further, as empirical data includes velocity of vehicles when they stop waiting green signal, we exclude data if velocity is less than 1 ft/sec. A histogram in Fig.X illustrates a distribution of velocity of vehicles at section 2. We assume that distribution of velocity of vehicles follows a normal distribution and select the standard deviation so that the average traveling time of vehicles in simulation to be close to those in the empirical data.

**1.3 Length of vehicle**

We calculate average travel time considering the length of a line of vehicles waiting at an intersection because we consider length of vehicle Further,. Therefore, the length of the vehicle should be an input parameter of the simulation. Since average length of vehicle depends their types, we first count the number of vehicles per their type. According to the given empirical data, vehicles are classified into motorcycles, automobiles, tracks. Since number of automobiles are larger than others (number of automobile is 738, motorcycle is 1, track is 14), we only consider the length of automobiles. Fig.x shows the histogram of the length of vehicles. The mean and standard deviation of the length of the vehicles are 16.05 feet and 2.87 feet. We assume that the distribution of the length of the vehicles follows a normal distribution.



**1.4 Probability of right or left turn at an intersection**

As noted in section 1.1, more than 85 percent of vehicles leave from northernmost and southernmost intersections. Thus, the travel time of the vehicles that leave intermediate intersections may have less influence on the average traveling time over all vehicles. However, if destinating zones are constrained in northernmost and southernmost intersections, the road will be more crowded than it actually is and the difference of congesting condition of the road influences the average traveling time of other vehicles because probability of spill over will increase and a line of intersection should be longer. Thus, we model a departure of vehicles at the intermediate sections and obtain the probability of the departure event at each intersection from the empirical data by distinguishing the direction of vehicles.

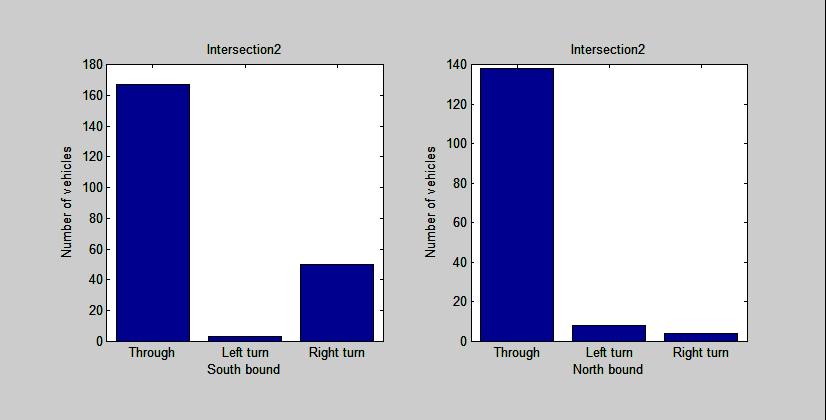
Fig. x shows the number of vehicles that go through, turn left and turn right at the second intersection.

Fig.x. Number of vehicles that go through and turn at the second intersection(left:southbound vehicles, right:northbound vehicles).

**Appendix A. Event oriented simulation**

**1 Overview**

We also build an event oriented simulation program for our better understanding of course materials in addition to the process based program shown in the above chapters,. This program is built based on the same model assumptions as the process oriented program. For this purpose, we also build a conceptual model of the simulator by using graphical modeling language (System Modeling Language, SysML). Block definition diagram(BDD) defines components of the program and sequence diagram defines interactions between the components. In our event oriented simulation program, components shown in BDD are implemented as a class.In this chapter, we show how our event oriented simulation program works by using a block diagram and sequence diagram of SysML.

**2. Block definition diagram**

Fig.A1 shows BDD of the event oriented simulation program. Each rectangle(block) represents a component of the simulation model. Value properties and operations of components are shown in compartments of the block. Arrows with white triangle head express a generalization relationship and specialized component is illustrated at the arrowheads. Arrows with white diamond head express an association relationship and referenced components are shown at the arrowheads. Though we also build a class which record behavior of vehicles during simulation and draw figure based on them, we do not put them to make the diagram concise.

**2.1 Scheduler class**

Scheduler class manages a future event list. The future event list is a linear list and sorted according to the time when the event should be executed. Other classes such as event class or Simulation class can add new event by calling addEvent operation. If there are events whose scheduled time are same as the new event, Scheduler class compares priorities of events and insert the new event before the existing events in the future event list if the new event has higher priority. After an event is executed, the event is excluded from the future event list.

**2.2 Simulation class**

Simulation class calls events and manages simulation time. First, this class retrieve an event which is listed the top of the event list. Then, the class advance simulation time by the time when the event should be executed and execute the event. This process is repeated until the simulation time reaches an end time.

**2.3 Intersection class**

Intersection class has queue, signal color, state of occupancy of an intersection as a state variable. Behavior of vehicles depends on the state variables and Event class shown later calls operations to change the state variables. For example, when vehicles arrives an intersection, they are listed in the queue and the vehicles can be dequeued only when the signal color is green and the intersection is empty.

**2.4 Event class**

The simulation program has four event class that inherit Event class. SignalEvent is an event where traffic light changes from red to green or green to red. ArrivalEvent enqueues vehicles and DeparetureEvent dequeue vehicles and change the state of occupancy to occupied state considering with the traffic light. In ServiceCompleteEvent set the state of occupancy as empty state when vehicle leaves the intersection.

**2.5 Event owner class**

In the simulation program, we consider behavior of multiple entities such as vehicles and traffic signals. To manage their behavior and state variables, all events are associated with EventOwner who has unique ownerID.

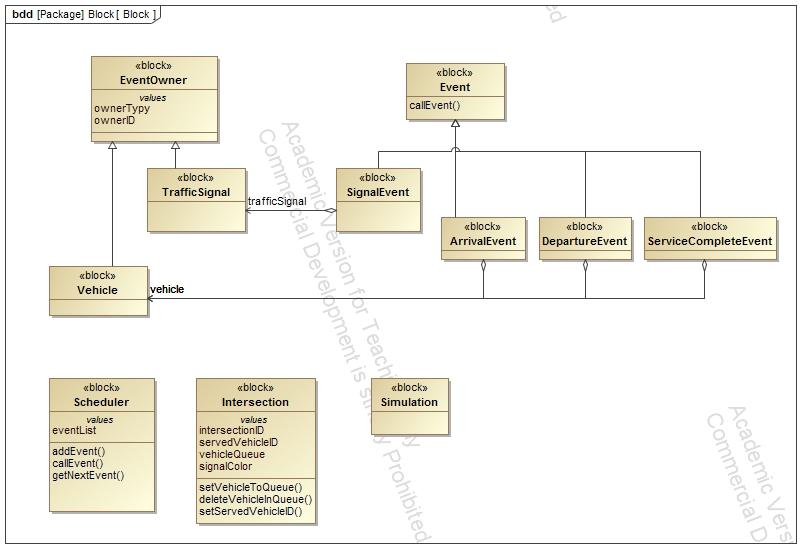


Fig. A1 Block definition diagram of the simulation program

**3. Sequence Diagram**

We introduce a simple scenario to show how our event oriented program works. In this scenario , we assume that there are only two vehicles going through the same direction and there is only one intersection. When the first and second vehicles arrive at the intersection, the traffic signal is red. Therefore, they have to wait until the signal turns green. When signal turns green, the first vehicles starts to pass through the intersection and after the first vehicle leave the intersection, the second vehicle starts. A sequence diagram which illustrates the scenario is shown in Fig. A2. In Fig. A2, instances of the blocks defined in Fig. 3 are illustrated as rectangles and interactions among them are represented by solid arrows. To make diagram concise, reply messages from receivers are omitted. Time flows from top to bottom of the diagram.

When the simulation starts, Simulation class add arrival event of the first vehicle(1) and signal event(2) to the future event list. Then, it get next event(3) (in this case, returned event is the arrival event of the first vehicle) and advance time by the time when the event should be executed(4). Simulation class also calls ArrivalEvent(5). ArrivalEvent class add it’s owner (i.e. vehicle) to queue at an intersection(6). Then the ArrivalEvent class add arrival event of the second vehilce(7) and DepartureEvent of the first vehicle(8). When Simulation class calls DepartureEvent of the first vehicle(9), the first vehicle does not cross the intersection because the traffic light is red. Thus, the DepartureEvent schedule the next DepartureEvent for the first vehicle(10). Simulation class calls ArrivalEvent of the second vehicle(11) that is scheduled at (7) and the ArrivalEvent add the second vehicle to the queue. After the second vehicle arrives, Simulation class calls SignalEvent(13) that is add to the future event list at (2). SignalEvent changes signal color from red to green(14) and schedule next SignalEvent where the traffic light turn green to red(15). The DepartureEvent of the first vehicle is called when the traffic light is green(16) and the event calls operation of the Intersection class to dequeue the first vehicle(17) and change the state of occupancy to occupied state. Further, the event schedule ServiceCompleteEvent for the first vehicle(19) and DepartureEvent for the second vehicle(20). After the state of occupancy becomes empty state by ServiceCompleteEvent(21), the second vehicle leaves from the intersection(22).

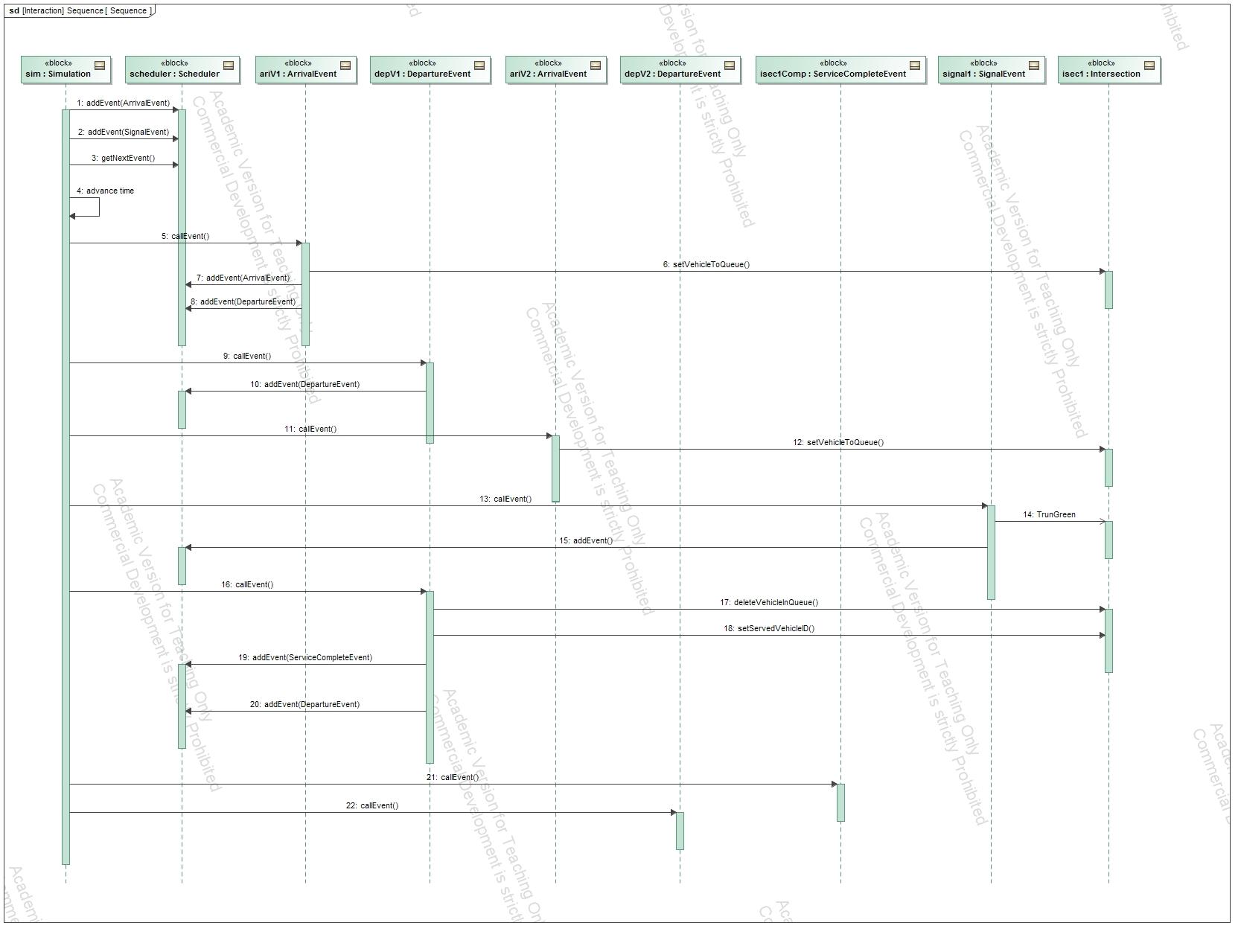


Fig. A2 Sequence diagram of simulation process

Fig. A3 represents time history signal color and queue at an intersection. In Fig. A3, horizontal line represents simulation time. Green and red circles illustrates color of traffic light at the time. Numbers in black shows IDs of vehicles listed in the queue and numbers in red represents an ID of a vehicle who is passing through the intersection. In this simulation, vehicle arrives at every four seconds and it takes four seconds for vehicle to pass through the intersection. The first vehicle(ID is one) and the second vehicle arrive at 4 sec and 8 sec respectively and they wait until 9 sec when the traffic light turns green. The first vehicle occupies the intersection from 9 sec to 12 sec and then leaves from the intersection. The second vehicle goes to the intersection at 13 sec and leave at 16 sec.

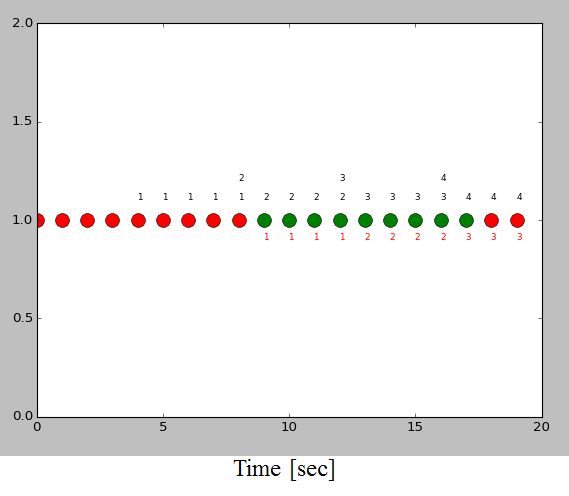


Fig.A3. Time history of signal color and queue at an intersection

**Appendix B.**

**B.1 Arrival rate of vehicle**

As mentioned in section 2.1.2 and 4.1, we assume that the arrival event of vehicle is a stationary stochastic process and the inter-arrival time of vehicle follows the exponential distribution. However, the assumption is not valid in some intersections.

Fig.B2 shows a time history of arrival events of vehicles at originating zone 114 obtained from the empirical data.The horizontal axis represents the observation time and vertical axis shows the number of vehicles that arrived at the observation time. As shown, arriving rate changes periodically at every one hundred seconds. This means that the arrival event in the originating zone 114 is not a stationary stochastic process. We guess that this periodical change of arrival rate is caused by northern traffic signals of originating zone 114. Thus, by considering the change of the distribution, we could make a simulation model much closer to the observation.

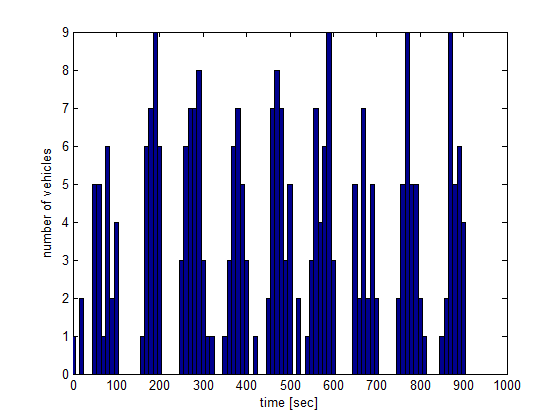


Fig. B2. Time history of arrival event in empirical data.

Fig.B3 illustrates a histogram (blue bar) of inter-arrival time at originating zone 114 from the empirical data. As shown in the figure, the distribution has the peak at not zero but between two to three seconds. Therefore, the empirical data does not give us enough evidence that the distribution of inter-arrival time follow the exponential distribution. To examine other alternatives, we conduct a Chi-Square Good Fitness Test under the assumption that the distribution of the inter-arrival time follow the gamma distribution. In Fig.B3, blue line illustrates a probability distribution function of gamma distribution with maximum likelihood estimation from the empirical data (shape parameter k is 5.517, scale parameter theta is 0.4892). The result of Chi-Square Good Fitness Test shows that the null hypothesis is not rejected with 95% confidence. Therefore, in so far as originating zone 114, it would be better to use the gamma distribution as probability of the arrival event.

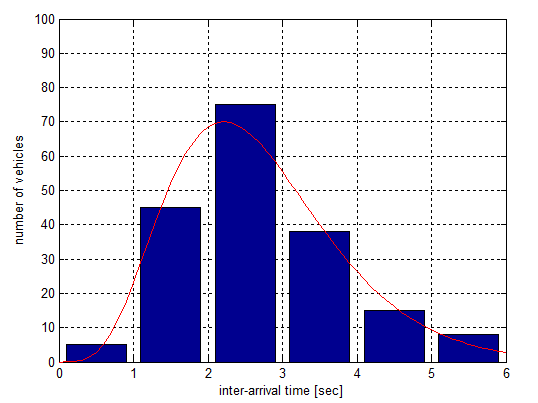


Fig. B3. Inter-arrival time of vehicles.

**B.2 Velocity of vehicle**

As mentioned in section 4.2, we assume that distribution of the velocity follows the normal distribution and we select the parameter so that an average travel time of the simulation to be close to that of the empirical data. However, velocity of vehicle in the empirical data seems uniformly distributed between and as shown in Fig.B4. We build random variable generator which yields velocity of vehicle by using inverse function of cumulative distribution function. Fig.B5 shows a histogram of cumulative distribution function and red line represents approximating curve. The approximating curve is fifth order polynomial function and obtained by a least square algorithm. We also build the random variable generator for other intersections in a similar manner and conduct simulation. However, we do not use the random variable generator in our simulation study since the average of velocities yielded from the random variable generator become smaller than the because of overfitting.

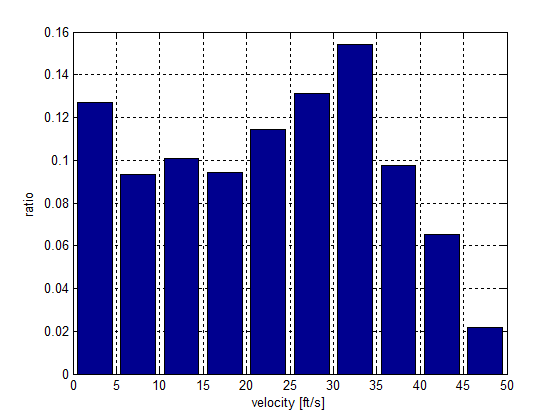


Fig. B4. Distribution of velocity of vehicles from the empirical data.

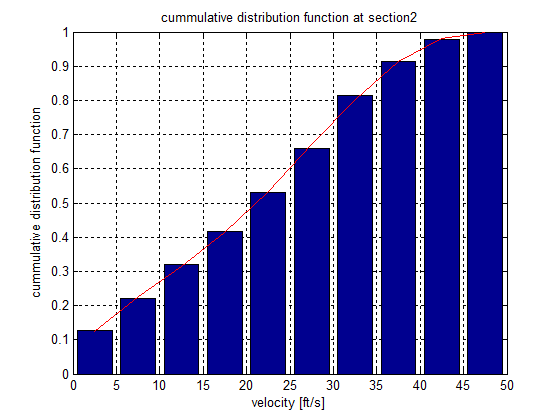


Fig. B5. Cumulative distribution function of velocity