PARAMICS Plugin Document – Loop Data Aggregator

Lianyu Chu Henry X. Liu Will Recker

PATH ATMS Center
University of California, Irvine

Plugin Compatibility: V4
Release date: 2/7/2005

522 Social Science Tower
Irvine, CA 92697-3600
URL: http://www.its.uci.edu/
Table of Contents

Table of Contents ............................................................................................................ 2
1. Introduction ................................................................................................................. 3
2 Plugin implementation ................................................................................................. 4
   2.1 Aggregation method .............................................................................................. 4
   2.2 Pseudo code .......................................................................................................... 5
   3.1 Preparation of the “loop_control” file ................................................................... 6
   3.2 Loading plugin ...................................................................................................... 7
   3.3 Text file outputs .................................................................................................... 7
   3.4 Error checking ...................................................................................................... 8
4. PROGRAMMER capabilities ..................................................................................... 9
   4.1 Interface functions ............................................................................................... 9
   4.2 How to use interface functions in other plugins .................................................... 10
5 Technical Supports ..................................................................................................... 11
   5.1 Release notes ....................................................................................................... 11
   5.2 Comparison with the loop data aggregator plugin of Quadstone ......................... 11
   5.3 Extending Capabilities ......................................................................................... 11
   5.4 FAQ ..................................................................................................................... 12
   5.5 Contact information ............................................................................................ 12
1. Introduction

PARAMICS can output two types of loop detector data for analysis:

- Point loop data, including flow, speed, headway, occupancy, and acceleration of a vehicle, and
- Link loop data, including flow, average speed, density, lane use, and lane changing on a link.

Point data is gathered at every time step when an individual vehicle passes over the loop; link data analyses the traffic data over a link, where loops locate, at a user-defined time period. However, many ATMIS applications demand point traffic data, but in an aggregated manner over user-defined time intervals, e.g. 30 seconds.

The objective of this plugin is to emulate the outputs of real-world data collection from induction loops in PARAMICS. It is implemented through gathering point loop data at each time step of simulation and then aggregating at any time interval specified by users. The gathered data can be raw data or smoothed data in term of user’s choice. Aggregated loop data (including volume, occupancy, and speed) can be output to text files, and can be also accessed by interface functions defined in this plugin.
2 Plugin implementation

2.1 Aggregation method

In the real world, most detectors are loop detectors. A loop detector station generally has multiple loop detectors and each loop detector covers a lane. In PARAMICS, a detector can cover all lanes or just cover a lane. This plugin outputs aggregated detector data in term of a detector station. The aggregated data outputs include not only aggregated data of each lane but also the grouped data of the detector station.

In the real world, loop detectors are used to report volume and percent occupancy. In the simulation, besides volume and percent occupancy, speed can also be obtained from simulation because it is a basic element of simulation. As a result, this plugin will be used to aggregate traffic volume, percent occupancy and speed data.

The aggregated volume is defined as the number of vehicles passing the detector during last time interval. The aggregated speed is the average of speeds of passing vehicles during last time interval. If at the aggregation time, a vehicle is just on a loop, it is counted as a passed vehicle for aggregation.

Percent occupancy is defined as the percentage of time of a loop occupied by vehicles. However, the occupancy obtained from PARAMICS (via API function loop_occupancy()) is time occupancy, which is calculated based on vehicle length, loop detector length, and vehicle speed. Therefore, we need to convert from time occupancy to percent occupancy:

$$\text{OCC} = \sum_{i=1}^{N} \frac{\text{OCC}(i)}{TT}$$

where OCC(i) is the time occupancy of vehicle i; N is the total number of vehicles having passed the detector during last time interval; TT is the interval of aggregation. If at the aggregation time, a vehicle is just on a loop, the duration the vehicle is on the detector (this value can be obtained from simulation) is used for aggregation.

Based on aggregated data of each lane, the grouped aggregated data (including grouped volume, occupancy and speed) of this detector station are also calculated. The grouped volume represents the total number of vehicles having passed the detector station (including several detectors, each lane has one detector) during last time interval, which can be expressed as

$$N_i(t) = \sum_{j=1}^{N} N_{i,j}(t)$$

where i is the index of the detector station; j is the loop index at detector station i; n is the total number of lanes, or loops at detector station i; $N_{i,j}(t)$ is the number of vehicles passing loop j of detector station i during time interval $(t-1, t)$.

The grouped occupancy represents the average percent occupancy of a detector station during last time interval, which can be expressed as
\[ O_j(t) = \left( \frac{1}{n} \sum_{j=1}^{n} \sum_{k=1}^{n} (TO_{i,j,k}(t)) \right) / TT \]

where \( i \) is the index of the detector station; \( j \) is the loop index at detector station \( i \); \( n \) is the total number of lanes, or loops at detector station \( i \); \( k \) is the vehicle index; \( N_{i,j}(t) \) is the number of vehicles passing loop \( j \) of detector station \( i \) during time interval \((t-1, t)\); \( TO_{i,j,k}(t) \) is the time occupancy of vehicle \( k \) passing loop \( j \) at detector station \( i \) during time interval \((t-1, t)\).

The grouped speed represents the average speed of a detector station during last time interval, which can be expressed as

\[ V_i(t) = \frac{1}{n} \left( \frac{1}{N_{i}(t)} \sum_{j=1}^{n} \sum_{k=1}^{n} V_{i,j,k}(t) \right) \]

where \( i \) is the index of the detector station; \( j \) is the loop index at detector station \( i \); \( n \) is the total number of lanes, or loops at detector station \( i \); \( k \) is the vehicle index; \( N_{i,j}(t) \) is the number of vehicles passing loop \( j \) of detector station \( i \) during time interval \((t-1, t)\); \( V_{i,j,k}(t) \) is the loop speed when vehicle \( k \) passes loop \( j \) at detector station \( i \) during time interval \((t-1, t)\).

### 2.2 Pseudo code

The control logic is given in the following pseudo codes:

1. Initialization of loop data aggregator plugin, including reading “loop_control” file, opening output files, memory allocation, and other initial settings;
2. At every time step, PARAMICS overload API function: “vehicle_detector()” when a vehicle traverses on or passes a loop. If a vehicle passed a detector, the occupancy and speed of the vehicle are accumulated. The following exceptional cases need to be handled in order to obtain the correct occupancy value:
   - (1) Unexpected incorrect value of occupancy from simulation, which may happen when a loop is placed at a location near the start or the end of a link;
   - (2) More than one vehicle are on a loop at the same time;
   - (3) One vehicle stays over a loop for more than certain time period, which may happen when an incident or congestion appears;
   - (4) One vehicle is just on a loop at the time of aggregation.
3. At every time step, PARAMICS overload API function: “net_action()”. For detector = 1:n
   `{ 
   If it is the time to calculate and report the aggregated data of a loop
   { 
   Calculate count, average speed, and percent occupancy of a detector.
   Calculate grouped count, occupancy, and speed of all detectors at a detector station
   Output these data to output files and the interface function
   }
   }

3.1 Preparation of the “loop_control” file

“loop_control” is the input file of the loop data aggregator plugin. This file should be put to the same directory as any other network files. An example of “loop_control” file is shown as follows:

<table>
<thead>
<tr>
<th>field</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>detector count</td>
<td>42</td>
</tr>
<tr>
<td>report cycle</td>
<td>30</td>
</tr>
<tr>
<td>activation time</td>
<td>06:00:00</td>
</tr>
<tr>
<td>deactivation time</td>
<td>10:00:00</td>
</tr>
<tr>
<td>gather smoothed data</td>
<td>no</td>
</tr>
<tr>
<td>output to files</td>
<td>yes</td>
</tr>
</tbody>
</table>

name 405n0.6ml

gather interval 00:00:30

name 405n0.93fr

gather interval 00:00:60

... 

There are two parts in the file. The first part is the general information about data aggregation.

1. The first row of the file shows the number of detectors that are required to do the aggregation operation.

2. The second row specifies the polling / report cycle of data aggregation. The unit is seconds. This cycle is corresponding to the time interval of real-world loop data collection. A typical value of the “report cycle” is 30 seconds. Basically, this polling cycle is not related to aggregated data outputs. It is used for ATMS applications, such as adaptive ramp metering, that are based on aggregated loop data.

3. The next two rows specify the activation time and deactivation time of the loop data aggregation.

4. The fifth row specifies whether to gather smoothed loop data (including speed, occupancy). If “no”, raw data will be gathered. There are two kinds of loop data that can be provided by PARAMICS at each time step, raw data or smoothed data. Smoothed refers to a value \( t_N^s \) at time-step \( N \) smoothed using the expression:

\[
t_N^s = (1 - p)t_{N-1}^s + pt_N
\]
where \( t_N \) is the current value and \( p \) is the co-efficient of smoothing.

5. The sixth row specifies whether to output aggregated loop data to files. If say “yes”, a file, generally named as “\( XYZ.txt \)” (“\( XYZ \)” is the name of the corresponding loop), will be generated for storing the aggregated volume, occupancy and speed data based on the gather interval specified in the second part of this control file.

The second part of the file contains the information of each loop detector, including the name of detector and the time interval that loop data are aggregated and reported to text files. There is a blank row between the information of any two loops. The name of a loop can be found in the “detectors” file, which is one of network files. The time interval to aggregate loop data can be different for different detectors.

3.2 Loading plugin

The names of this plugin files are:

- `loop_agg.dll`: Modeller Plugin
- `loop_agg-p.dll`: Processor Plugin

Please make sure these two plugin files are in the Paramics root directory. If not, please copy these two files to the directory. Then, please prepare a file named “programming.modeller” (using Modeller) or “programming.simulator” (using Processor or batch mode simulation) in the directory of your network. The content of the file is the name of the plugin, i.e. `loop_agg.dll` or `loop_agg-p.dll`. Please check the example network if you have any problem.

Then, you can load the simulation network together with this plugin. Run simulation and then you will obtain the aggregated loop data outputs if you enable the option “output to files”.

3.3 Text file outputs

If “output to files” is “yes”, aggregated loop data will be output to text files. Each detector specified in the “loop_control” file has its own output file. These output files can be found in the subdirectory:

```
network/Log/run-xxx
```

where network is the name of the current working directory, and xxx is a three-digit sequence number.
During the simulation process, aggregated detector data are continuously calculated, and then immediately stored to the output text file. Each output file has several fields, whose definitions are shown as follows:

| Time stamp, grouped volume, grouped occupancy, grouped speed, volume of lane 1, occupancy of lane 1, average speed of lane 1, volume of lane 2, occupancy of lane 2, average speed of lane 2, ..., volume of lane n, occupancy of lane n, average speed of lane n |

For right hand driving, lane 1 is the inside lane (the leftmost lane, it might be the HOV lane if applicable). Lane n is the outside lane (the rightmost lane). The unit of the speed output is miles per hour. The percent occupancy value is shown in the format of “0.094”, which represents the percent occupancy of 9.4%. Figure 1 shows an example of the output file.

![405n06m.txt - Notepad](image)

Figure 1 Output file of loop data aggregator plugin

3.4 Error checking

This plugin is easy to use if “loop_control” is prepared correctly. If any mistakes happened in the “loop_control” file, the plugin will be disabled. The report window of PARAMICS will show whether this plugin is working.

If the plugin is not working, you may need to check if there is any error in “loop_control”. This plugin generates a file named “Log-loop.txt” under the network directory, which can be used to check if the “loop_control” file has been understood by this plugin correctly.
4. PROGRAMMER capabilities

4.1 Interface functions

This plugin provides two interface functions. The first one is used to obtain the current polling / report cycle of data aggregation, defined in the second row of “loop_control”. If an ATMS application, such as an adaptive ramp metering control, is based on aggregated loop data, this interface can provide the polling cycle of aggregated loop data.

\[ \text{int } uci\_\text{loop\_agg\_interval}(\text{void}) \]

Return Value: The polling cycle of aggregated loop data.
Parameters: None

Another interface function is used for querying the aggregated loop data of the latest time interval at a detector station. The aggregated loop data includes grouped volume, average occupancy and average speed, and lane-based volume, average occupancy and average speed.

\[ \text{LOOPAGG } uci\_\text{loop\_agg}(\text{int index}) \]

Return Value: The aggregated detector data of a loop detector
Parameters: index: the network-wide index number for a loop detector

LOOPAGG is a structure that has the following definition:
\[
\text{typedef struct}\ \text{loopagg}\ \text{LOOPAGG};
\text{struct loopagg} \\
\{
\text{int} \quad \text{index}; \\
\text{float} \quad \text{time}; \\
\text{int} \quad \text{g\_vol}; \\
\text{float} \quad \text{g\_occ}; \\
\text{float} \quad \text{g\_spd}; \\
\text{int} \quad \text{lane}; \\
\text{int} \quad \text{*vol}; \\
\text{float} \quad \text{*occ}; \\
\text{float} \quad \text{*spd}; \\
\};
\]

where

index is the network-wide index for the detector;
time is the time stamp for the calculation of aggregation, decided by the gather interval of each loop;
g_vol is the total traffic volume passing all lanes of a detector station within last time interval;
g_occ is the average occupancy of all lanes of a detector station;
g_spd is the average speed of all vehicles passing all lanes of a detector station;
lane is the total number of lanes at the detector station;
*vol, *occ, *spd are pointers for recording values of volume, occupancy...
and average speed of each lane of a detector station. Most items in this structure can be found in the output file of aggregated loop data, whose fields are defined in Table 2.

4.2 How to use interface functions in other plugins

The interface functions can be called by other plugins. The following setting is required:

1. In the workspace of your plugin that wants to use the interface function, specify the library file “loop_agg.lib” of this plugin as an input object/library module. The path of “loop_agg.lib” should be specified as well.

2. Make sure that loop aggregator plugin is specified before the plugin that will use the interface function in the “plugin” file (V3, located in “plugins\windows” under the PARAMICS installed directory), or the “programming” file (V4, located at the network directory).

3. Specify the prototypes of interface functions at the beginning of your plugin:
   `_declspec(dllexport) LOOPAGG uci_loop_agg(int index);
   `_declspec(dllexport) int uci_loop_agg_interval(void);`
# Technical Supports

## Release notes

Compare to the plugin used in PARAMICS version 3, this plugin has the following modification:

1. In the “loop_control” file, the fifth row, “gather smoothed data:” is changed to “gather smoothed data”.
2. In the “loop_control” file, the sixth row, “output to files:” is changed to “output to files”.
3. The definitions of the second row “report cycle” and the second row in the second part “gather interval” have changed. “report cycle” of V4 plugin takes the meaning of “gather interval” of V3 plugin and “gather interval” of V4 plugin takes the meaning of “report cycle” of V3 plugin.

## Comparison with the loop data aggregator plugin of Quadstone

<table>
<thead>
<tr>
<th>Measurements</th>
<th>Quadstone</th>
<th>PATH ATMS Center</th>
</tr>
</thead>
<tbody>
<tr>
<td>Occupancy output</td>
<td>Time occupancy</td>
<td>Percent occupancy</td>
</tr>
<tr>
<td>Output files</td>
<td>A file only includes a lane’s data, or the grouped data.</td>
<td>Grouped data and lane-based data are in the same file for a detector</td>
</tr>
<tr>
<td>Traffic count</td>
<td>During highly congested condition, the count has errors.</td>
<td></td>
</tr>
</tbody>
</table>

## Extending Capabilities

This plugin can output more parameters, such as gap, flow, by some minor modifications of the source codes. In PARAMICS, the definition of flow is

\[
Flow = \frac{1}{Headway}
\]

and the definition of gap is

\[
Gap = \frac{1}{Headway - occupancy}
\]

There is another MYSQL version of Loop Data Aggregator plugin. MYSQL database is used for storing aggregated loop data. All aggregated loop data since the beginning of simulation can be accessed through querying the database.
5.4 FAQ

1. Grammar of input files

Unlike the parser system of PARAMICS, which allow flexible grammars and comments (i.e. ##), the format of the input file of this plugin is rigid and thus any problem in the file may cause the plugin not work well. Our recommendation for users is that the input file of the example network of this plugin is a good starting point to make your own input file in order to avoid editing problems.

5.5 Contact information

Any comments and suggestions are welcome. Please contact us at the email address: lchu@translab.its.uci.edu.