

PARAMICS Plugin Document – SATMS ramp metering control

Xuegang (Jeff) Ban, Lianyu Chu

California Center for Innovative Transportation
University of California, Berkeley
&
California ATMS Testbed
University of California, Irvine

Plugin Compatibility: V5
Release Date: 10/01/2006

2105 Bancroft Way, Suite 300
Berkeley, California, CA 94720-3830
URL: <http://www.calccit.org>

Table of Contents

Table of Contents	1
1. Introduction.....	2
2. Plugin implementation	3
2.1 Algorithm description	3
2.2 Development framework	4
2.3 Pseudo Codes	5
2.3.1 Pseudo codes for RATES.....	5
2.3.2 Pseudo codes for SIGNAL.....	6
2.3.3 Input Parameters	7
3. Step-by-step user manual	8
3.1 Adding detectors	8
3.2 Preparation of the “satms_control” file.....	8
3.3 Loading plugin	10
3.4 Output file	10
3.5 Error checking.....	10
4. Technical Supports.....	11
4.1 Release notes.....	11
4.2 Contact information	11

1. Introduction

The purpose of this plugin is to allow users to use the SATMS (Semi Actuated Traffic Metering System) ramp metering control of Caltrans in the Paramics simulation environment. This plugin is developed based on source codes of SATMS, provided by Caltrans.

2. Plugin implementation

2.1 Algorithm description

SATMS can be setup as either demand-capacity based traffic responsive control or pre-timed control. Metering rates are updated every 30 seconds. For traffic responsive metering, metering rates vary according to different times of a day, traffic conditions, and priority setups. There are six priority options for determining the metering rate, namely (from priority high to low), metering controlled in field at controller (**FMNL**), TOC requested rate (**TMNL**), rate requested from PSO (**PMNL**), rates requested from TMC subject to time-of-day (TOD) minimums (**CORM**), local mainline traffic responsive rate (**LMRATE**), and TOD table rate (**TDRATE**). The TOC requested rate (TMNL) may be set up as the System Wide Adaptive Ramp Metering (SWARM) rate. Obviously, whenever a higher priority option is valid, it will be taken as the actual metering rate. However, there are also some exceptions. A flow chart of the logic for determining metering rate is given in Figure 1.

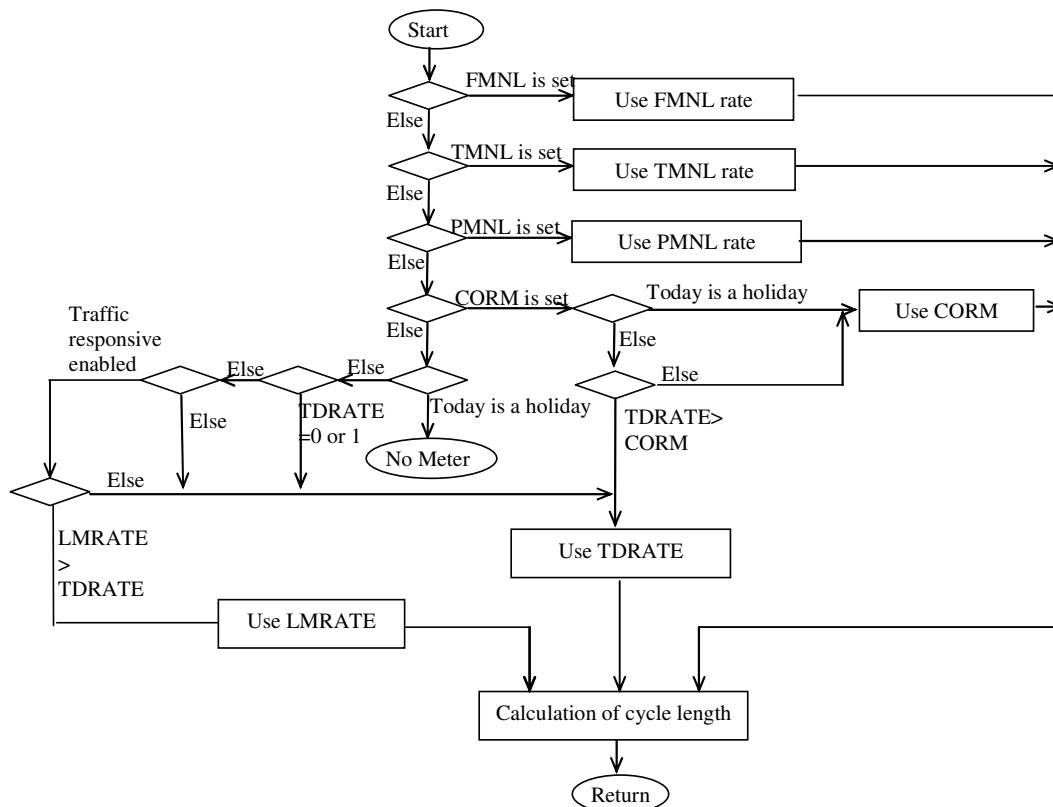


Figure 1 Flow Chart to Determine the Metering Rate

For any SATMS-controlled meter, the TOD rates need to be setup indicating the lower-bound metering rates that are used at a given period of time. The first four most top priorities (FMNL, TMNL, PMNL, and CORM) are optional.

To compute the traffic responsive metering rate, 1-minute mainline occupancy and 3-minute mainline volume data are used and compared (against the critical values). The computation procedure can be described as follows.

```
If current average 1-minute mainline occupancy >= 9/5*CROCC
    LMRATE = 0;
Else
    If current 3-minute average mainline volume >= CRVOL
        LMRATE = 0;
    Else
        LMRATE = (# of active mainline lanes)*[CRVOL - (3-min Avg Vol)]/3
        If resultant LMRATE is huge (>255)
            LMRATE = 1;
        End
    End
End
```

Here CROCC is the critical 1-minute mainline occupancy and CRVOL is the critical 3-minute mainline volume. Note that the unit of LMRATE is vehicles/minute.

2.2 Development framework

Figure 2 illustrates our hierarchical framework to develop advanced ramp-metering algorithm plugins in PARAMICS.

The SATMS plugin is built based on two basic plugin modules, i.e., ramp metering controller and loop data aggregator. The on-ramp signals in the simulation network are controlled by the ramp metering plugin, through which metering rates can be queried and set by the SATMS plugin. The loop data aggregator emulates the real-world loop data collection, typically with a thirty-second polling interval, and broadcast the latest loop data to the dynamic memory. At each time increment, the SATMS plugin accesses the dynamic memory and obtains the required loop data through interface functions provided by the loop data aggregation plugin. Then the metering rate for the next control interval is calculated based on the SATMS algorithm. The new metering rate is finally sent back to the ramp controller plugin for implementation.

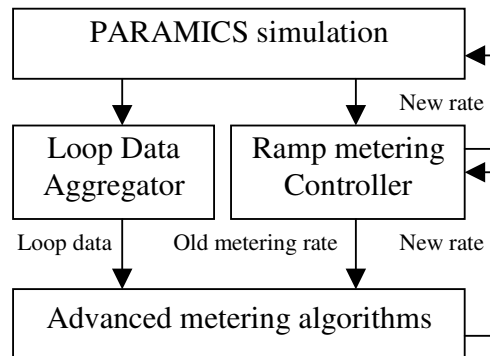


Figure 2 The hierarchical approach for the development of advanced metering algorithms

2.3 Pseudo Codes

We implement two major subroutines for the SATMS algorithm: RATES and SIGNAL. The RATES subroutine is to compute and update the metering rate, which is called every 30 seconds. The SIGNAL subroutine is to set the signal phase and timing, which is executed every 1/10 second.

2.3.1 Pseudo codes for RATES

The pseudo codes for the RATES subroutine are as follows:

1. Fetch current TOD rate from the TOD table
2. Communicating with ram metering API and loop data integrator API in order to obtain related up-to-date traffic information, in particular the 3-min average mainline volume and 1-min average mainline occupancy.
3. Compute the traffic responsive metering rate using the method in Section 2.1.
4. Determine current metering rate as follows (also shown in the flow chart in Figure 1):

```

If (the FMNL rate is set) {
    Use the FMNL rate
}else If (the TMNL rate is set) {
    Use the TMNL rate
}else If (the PMNL rate is set) {
    Use the PMNL rate
}else If (the CORM rate is set) {
    If (today is a holiday || CORM > TDRATE){
        Use the CORM rate
    }else{
        use the TDRATE rate
    }
}else{
    if (today is a holiday){
        no metering is needed
    }else{
        if(TDRATE ==0 || TDRATE == 1 || pre-timed ||
  
```

```

                (traffic-responsive && TDRATE > LMRATE)){
                    use TDRATE
                }else{
                    use LMRATE
                }
            }
        }
    }
}

```

5. Compute the metering cycle length.

2.3.2 Pseudo codes for SIGNAL

The pseudo codes for the SIGNAL subroutine are as follows:

RED:

```

Show 2 seconds minimum red
If(pre_time | demand call | demand error | stay_in_green is set){
    Start next cycle by going to GREEN
}else{
    stay in red
}

```

GREEN:

```

Show 2 seconds minimum green
If(stay_in_green is set){
    Stay in green
}else if(pre_time | max green is done | demand is satisfied){
    Start to show red by going to RED
}else{
    Stay in green
}

```

Notes:

1. The above codes are for cases that the metering rate is always greater than 0. If the metering rate changes from 0 to a positive value, it will start the metering from either black or flashing. In this case, a first green will be shown followed by a first yellow phase. The durations for the first green and yellow phases can be configurable. The default values are 3 seconds for both. After these two phases, the meter will go to normal metering. On the other hand, if the metering rate changes from a positive value to 0, a last green can be displayed for a configurable duration. But the default value is currently set as 0, indicating that no last green will be shown. After the last green phase (if set), the meter will be shut down (showing black).
2. The above pseudo codes also work only for the “1 vehicle per green” scenario. For “2 or more vehicles per green” cases, a yellow phase may be needed if the platoon yellow time is nonzero. In this case, the yellow phase can be set as 1.5 – 6 seconds if no platoon is detected or 3 – 6 seconds if platoon is detected. A platoon is detected if the displayed green time for a cycle is longer than 7 seconds.

3. A stay_in_green flag will be set if the current metering rate is 1. This can be triggered if (1) any of FMNL, TMNL, PMNL, or CORM is set to 1, (2) current TOD rate is set as 1, (3) the computed traffic responsive rate is too large (> 15 vehicles per minute per lane).
4. A queue override flag will be set if queue detectors are occupied for more than a pre-defined threshold time, default as 5 seconds. The cycle length of the metering will be automatically set as 4 seconds (900 vehicles per hour) once this queue override flag is set.

2.3.3 Input Parameters

The most important input parameters for the SATMS metering logic are the critical mainline volume and occupancy and TOD table rates for each meter. The 3-minute critical mainline volume and 1-minute critical occupancy are directly used by SATMS to compute the traffic responsive metering rate. Since for most meters the highest four priorities (FMNL, TMNL, PMNL, and CORM) will not be set, the actual metering rate will be determined solely by the traffic responsive rate and TOD table rates.

Other parameters, such as minimum green time, queue override settings, etc., are also important inputs to SATMS. Most of them are configurable, as will be shown in more details in Section 3.

3. Step-by-step user manual

3.1 Adding detectors

SATMS needs four types of detectors. The first one is located on the mainline freeway, immediately upstream of the entrance ramp so that metering rate can be properly computed from the volume and occupancy collected at the mainline detectors.

The on-ramp detector is used for counting total number of vehicles entering freeway from entrance ramps. The demand and passage detectors (i.e. corresponding to the check-in and check-out detectors) are used for the operation of on-ramp signals. The demand detector employs to initiate green and the passage detector employs to return the signal to red.

If one wants to implement SATMS with a queue override strategy, he/she may need to add a queue detector. Please see the document of on-ramp queue control plugin for how to use that plugin and how to add a queue detector.

3.2 Preparation of the “satms_control” file

The “satms_control” file contains all necessary information required by the SATMS plugin: locations of the ramp, detector settings, metering rate settings, and TOD rates. The file is in an XML format for ease of understanding and transferring. An example of the file is given as follows:

```
<Metering_Control>
  <Meter>
    <Config>
      <Node>65</Node>
      <Location>SandCanyon</Location>
      <Up_Link>107:65</Up_Link>
      <Down_Link>65:64</Down_Link>
      <Mainline_Detector>405n2.99ml</Mainline_Detector>
      <Demand_Detector>405n2.99orb</Demand_Detector>
      <Passage_Detector>405n2.99ora</Passage_Detector>
      <Queue_Detector>405n2.99orspill2</Queue_Detector>
      <Meter_Enable>1</Meter_Enable>
      <Demand_Recall>1</Demand_Recall>
      <Passage_Recall>0</Passage_Recall>
      <Queue_Enable>1</Queue_Enable>
    </Config>
  <SATMS>
    <Setting>
      <Critical_Vol>1600</Critical_Vol>
      <Critical_Occ>0.20</Critical_Occ>
    </Setting>
  </SATMS>
</Meter>
</Metering_Control>
```

```

        <Min_Meter_Rate>360</Min_Meter_Rate>
        <Max_Meter_Rate>1800</Max_Meter_Rate>

        <Vehicle_Per_Cycle>1</Vehicle_Per_Cycle>
        <Max_Green>6</Max_Green>
        <Queue_Cycle_Length>4</Queue_Cycle_Length>
        <Queue_Threshold>5</Queue_Threshold>
        <Metering_Type>2</Metering_Type>
    </Setting>
    <Plan>
        <Start_Hour>14</Start_Hour>
        <Start_Minute>0</Start_Minute>
        <End_Hour>14</End_Hour>
        <End_Minute>30</End_Minute>
        <TOD_rate>480</TOD_rate>
        <Active_Mainline_Lanes>4</Active_Mainline_Lanes>
    </Plan>
    <Plan>
        <Start_Hour>14</Start_Hour>
        <Start_Minute>30</Start_Minute>
        <End_Hour>15</End_Hour>
        <End_Minute>30</End_Minute>
        <TOD_rate>720</TOD_rate>
        <Active_Mainline_Lanes>4</Active_Mainline_Lanes>
    </Plan>
    <Plan>
        <Start_Hour>15</Start_Hour>
        <Start_Minute>30</Start_Minute>
        <End_Hour>16</End_Hour>
        <End_Minute>30</End_Minute>
        <TOD_rate>960</TOD_rate>
        <Active_Mainline_Lanes>4</Active_Mainline_Lanes>
    </Plan>
    <Plan>
        <Start_Hour>16</Start_Hour>
        <Start_Minute>30</Start_Minute>
        <End_Hour>18</End_Hour>
        <End_Minute>00</End_Minute>
        <TOD_rate>1200</TOD_rate>
        <Active_Mainline_Lanes>4</Active_Mainline_Lanes>
    </Plan>
    </SATMS>
</Meter>
</Metering_Control>

```

The file is self-explanatory for most of the element definitions. In particular, each meter should have a corresponding “Meter” element, which has two sub-elements: “Config” and “SATMS”. “Config” sets the configuration of the meter, including metering node, upstream and downstream links, and needed detectors (mainline, demand, passage, and/or queue). It can also enable or disable the metering, demand and passage call, and queue override control by setting the value of the corresponding element to “1” or “0”. Note that at current version, queue override control will be automatically set if queue detector is defined explicitly in this element.

The “SATMS” element contains one “Setting” sub-element and multiple “Plan” sub-elements. “Setting” defines meter-specific parameters, such as mainline critical volume and occupancy, min and max metering rates, max green, vehicle per cycle, and queue override control parameters. One should note that metering rates (min or max) are for all the metering lanes (in the unit of vehicles per hour), while mainline critical volume is for each lane and in the unit of vehicles per hour per lane. The “Metering_Type” sub-element explicitly sets the metering as “disabled” (0), or “pre-timed” (1) or “traffic responsive” (2).

Each “Plan” element defines a time-of-day (TOD) rate. It has the starting and ending hours and minutes, the TOD rate (vehicles per hour), and the number of active mainline lanes. All the “Plan” sub-elements of a “SATMS” element constitute the TOD table for the specific meter.

3.3 Loading plugin

The names of this plugin files are satms.dll. It can be used under both Modeller and Processor or batch mode. The SATMS plugin depends on detector data aggregator plugin. In the “programming” file, the two plugins should be written as follows:

```
loop_agg.dll
satms.dll
```

3.4 Output file

At the end of every time interval when a new metering rate is calculated, the plugin will output metering rate and other related information to a file named “Log-SATMS.txt”. It 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.

3.5 Error checking

If any mistakes happened in the “satms_control” file or the other supporting plugin, i.e. detector data aggregator, this plugin will be disabled. The report window of PARAMICS will show whether this plugin is working.

4. Technical Supports

4.1 Release notes

This plugin is based on STAMS version 3. It is developed and tested using Paramics v5.1. Incompatibilities may exist with lower versions of Paramics.

4.2 Contact information

Any comments and suggestions are welcome. Please contact us at the email address: xban@berkeley.edu.