

PARAMICS Plugin Document – ALINEA ramp metering control

Lianyu Chu

PATH ATMS Center
University of California, Irvine

Plugin Compatibility: V4
Release date: 3/20/2003

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 Algorithm description.....	4
2.2 Development framework.....	4
2.3 Pseudo codes.....	5
3 Step-by-step user manual.....	6
3.1 Adding detectors.....	6
3.2 Preparation of the “alinea_control” file.....	6
3.3 Loading plugin.....	8
3.4 Output file.....	9
3.5 Error checking.....	9
3.6 Calibration of the ALINEA algorithm.....	10
4 Technical supports.....	12
4.1 Release notes.....	12
4.2 Future development.....	12
4.3 FAQ.....	12
4.4 Contact information.....	12

1. Introduction

The ALINEA algorithm, proposed by Papageorgiou et al, is a local feedback ramp metering policy. The algorithm attempts to maximize the mainline throughput through maintaining a desired (near critical) occupancy on the downstream mainline freeway.

This plugin is to implement the ALINEA ramp-metering algorithm in PARAMICS.

2 Plugin implementation

2.1 Algorithm description

The local adaptive ramp-metering algorithm we use is ALINEA (Papageorgiou, et al, 1990 and 1991). ALINEA is a local feedback ramp metering policy, which attempts to maximize the mainline throughput by maintaining a desired occupancy on the downstream mainline freeway. As shown in Figure 2, two detector stations are required for the implementation of the ALINEA algorithm. The first loop detector is located on the mainline freeway, immediately downstream of the entrance ramp, where the congestion caused by the excessive traffic flow originated from the ramp entrance can be detected. The second loop station is on the downstream end of the entrance ramp, and used for counting the on-ramp volume.

For an on-ramp under ALINEA control, its metering rate during time interval $(t, t + \Delta t)$ is calculated as:

$$r(t) = \tilde{r}(t - \Delta t) + K_R \cdot (O^* - O(t))$$

where Δt is the update cycle of ramp metering implementation; $\tilde{r}(t - \Delta t)$ is the measured metering rate of the time interval of $(t - \Delta t, t)$; $O(t)$ is the measured occupancy of time interval $(t - \Delta t, t)$ at the downstream detector station; K_R is a regulator parameter, used for adjusting the constant disturbances of the feedback control; O^* is the desired occupancy at the downstream detector station. The value of O^* is typically set equal to or slightly less than the critical occupancy, or occupancy at capacity, which can be found in the volume-occupancy relationship.

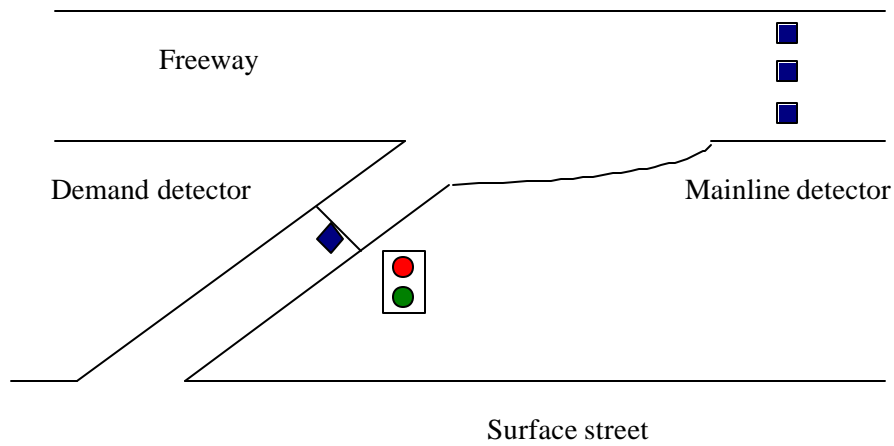


Figure 2 Detector layout for the implementation of the ALINEA algorithm

2.2 Development framework

Figure 1 illustrates the hierarchical development framework of advanced ramp-metering algorithm plugins in PARAMICS. The advanced ramp-metering algorithm plugin is built on top of 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 other plugin modules. 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 advanced ramp-metering algorithm plugin can access 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 advanced ramp-metering algorithm. The new metering rate is finally sent back to the ramp controller plugin for implementation.

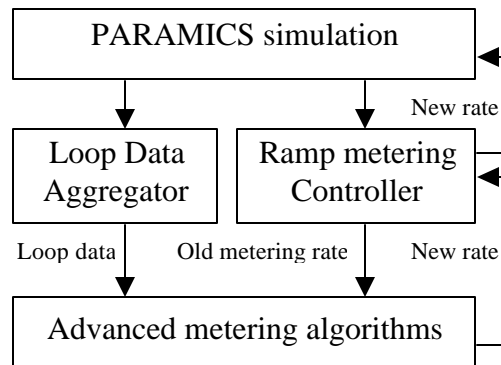


Figure 1 The hierarchical approach for the development of advanced metering algorithms

2.3 Pseudo codes

We implement the ALINEA algorithm plugin with the following pseudo codes:

1. Communicating with ramp metering API and loop data aggregator API in order to obtain related up-to-date traffic information and historical metering rates.
2. Calculating the next metering rate based on formula 1.
3. Metering rate restriction
An on-ramp volume restriction, which requires the implemented metering rate to be limited within some pre-defined maximum and minimum values, is also included in this API.
4. HOV adjustment
The ideal ramp-metering algorithm should control all vehicles entering freeway from entrance ramps. If there is a HOV preferential lane on the entrance ramps, the HOV traffic volume can either be considered or not considered in the calculation of future metering rate.
5. Sending its computed metering rate to the ramp metering API for implementation.

3 Step-by-step user manual

3.1 Adding detectors

ALINEA needs two detector stations. The first one is located on the mainline freeway, immediately downstream of the entrance ramp, where the congestion caused by the excessive traffic flow originated from the ramp entrance can be detected.

The second one is located on the downstream end of the entrance ramp, and used for counting the on-ramp volume. You can use the demand detector (i.e. check-in detector), which is installed to the network for the vehicle presence in the ramp metering plugin.

If you want to implement ALINEA with a queue override strategy, you 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 “alinea_control” file

The “alinea_control” file includes all necessary information required by this ALINEA plugin. Some of inputs should be calibrated based on the modeled network before implementation. An example of the file is as follows:

```

total number of alinea controlled ramps is 7
checking control file          yes
metering rate update interval 30
algorithm activation time      06:00:00
algorithm deactivation time    09:00:00
report metering rate          yes

ramp                           33
mainline detector              405n0.93ml-ds
on-ramp detector               405n0.93orb
HOV                            0
control type                   1
desired occupancy              0.13
regulator                      70.0
rate restriction                300 1200

...

ramp                           95
mainline detector              405n5.55ml-ds

```

on-ramp detector	405n5.55orb
HOV	1
control type	1
desired occupancy	0.20
regulator	70.0
rate restriction	240 900
ramp	92
mainline detector	405n5.74ml-ds
on-ramp detector	405n5.74orb
HOV	0
control type	1
desired occupancy	0.20
regulator	70.0
rate restriction	400 1500

“metering rate update cycle” is the time interval of metering rate calculation.

The option of “checking control file” is used for checking if there are any mistakes in the control file. If “yes”, this API will print out the information obtained from “alinea_control” file.

If “report metering rate” is yes, the metering rates of ALINEA controlled ramps will be output (every update cycle) to a file named “ALINEA-rampRate.txt” under the “Log” directory.

During the time period between “algorithm activation time” and “algorithm deactivation time”, the ramp-metering algorithm is activated.

The second part of “alinea_control” is about each ALINEA-controller ramps. “mainline detector” generally corresponds to the detector placed on the downstream of an on-ramp. It can be specified as an upstream detector but the performance of this algorithm may be deteriorated.

“on-ramp detector” generally corresponds to the demand detector shown in Figure 2. The purpose of this detector is to count the total vehicles entering freeway from an on-ramp. If the specified on-ramp detector cannot be found in the “detectors” file, this API will not work.

On the row of “HOV” (number of HOV lane at the location of on-ramp detector), if the HOV lane is modeled as a separated link or there is no HOV lane, please write down 0 on this row. If there is at least one HOV lane on the in-ramp, and the HOV and SOV lanes are modeled on the same on-ramp link, there are two conditions, HOV bypass and HOV metering. If “HOV” is specified as one (or more than one), vehicles entering freeway from the HOV lane are un-metered and the calculation of the future metering rate will not consider HOV vehicles. This is the HOV bypass situation, whose example is the first

ramp shown in the above example. If “HOV” is specified as 0, this is the HOV metering situation. Vehicles entering freeway from HOV lanes will be considered in the calculation of the future metering rate. The example of this case is the second ramp shown in the above example.

“control type” means one-car-per-green or multiple-car-per-green. There are three possible values, 1, 2, 3. This information will be used for calculating the correct metering cycle under a certain control type.

“desired occupancy” and “regulator” are two parameters. “desired occupancy” has the format like “0.20”, but it represents 20%.

“rate restriction” include a minimum rate and a maximum rate, which are actually the two boundaries that metering rate can vary. Its unit is vehicle per hour.

3.3 Loading plugin

The names of this plugin files are:

 alinea.dll: Modeller Plugin

 alinea-p.dll: Processor Plugin

The ALINEA plugin depends on other two plugins, ramp metering control and loop data aggregator. These two plugins should be specified earlier than this plugin in the “plugins” or “programming” file, i.e.:

 loop_agg.dll

 ramp_controller.dll

 alinea.dll

If you want to implement ALINEA with a queue override strategy, you will need to add on-ramp queue control plugin to the “plugins” or “programming” file with the following sequence in order to give the on-ramp queue control strategy higher priority than ALINEA:

 loop_agg.dll

 ramp_controller.dll

 queue_control.dll

 alinea.dll

In addition, in order to correctly load and run this plugin, please satisfy the following requirements:

(1) For ramp metering control plugin: on-ramp signals controlled by the ALINEA algorithm should be specified in the “ramp_control” file. For example, the correspondent on-ramp signal 33 needs to be specified in the “ramp_control” file:

on-ramp signal	33
name	405N & ICD 1 @ 0.93
demand detector	405n0.93orb
number of control plans	2
from 6:0 to 9:0	METER_ON with 1 veh per 6 sec
from 15:0 to 19:0	METER_ON with 1 veh per 6 sec

(2) For the loop data aggregator plugin: all loops involved in the “alinea_control” file should be specified in the “loop_control” file for aggregated data collection. In addition, the “report cycle” in the “loop_control” file should be the same as the “metering rate update interval” specified in the second row of “alinea_control” file. For example, all relevant loops of ramp 33 needs to be specified in the “loop_control” file:

detector count	42
report cycle	30
activation time	06:00:00
deactivation time	10:00:00
gather smoothed data	no
output to files	yes

...

name 405n0.93ml
gather interval 00:00:30

name 405n0.93orb
gather interval 00:00:30

...

3.4 Output file

If “report metering rate” in the “alinea_control” file is specified as “yes”, the applied metering rate of all ALINEA controlled on-ramps will be output (every update cycle) to a file named “moe-ALINEA.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 “alinea_control” file or the other two supporting plugins, i.e. loop data aggregator and ramp metering control, this plugin will be disabled. The report window of PARAMICS will show whether this plugin is working.

Through enabling the option: “checking control file” in the “alinea_control” file, you can check if there is any error in the “alinea_control” file.

3.6 Calibration of the ALINEA algorithm

In order to maximize the performance of ALINEA metering control, parameters of the ALINEA algorithm need to be calibrated and optimized. Basically, ALINEA has four parameters to be calibrated, including the location of the downstream detector station, the desired occupancy of the downstream detector station, the update cycle of metering rate, and a constant regulator parameter K_R .

The following is a summary of currently applied parameter values:

- (1) The desired occupancy can be equal to or around the occupancy value at capacity, which can be also found in the volume-occupancy diagram. Various values ranging from 0.18 to 0.31 have been found in previous applications.
- (2) The regulator K_R , used for adjusting the constant disturbances of the feedback control, is found to perform well in the real-life experiment when it is set to 70. Simulation results are considered to be insensitive for a wide range of values.
- (3) The downstream detector should be placed at a location where the congestion caused by the excessive traffic flow originated from the ramp entrance can be detected. In reported implementations, this site is located between 40 m and 500 m downstream of the on-ramp nose.
- (4) The update cycle of ramp-metering rate is also variable with the range from 40 seconds to 5 minutes. In theory, if the value is small, the location of the downstream detector station should be close to the entrance ramp. Otherwise, there is a risk of congestion built-up in the interior of the stretch from the ramp nose to the downstream detector.

These calibrated parameters of ALINEA can be obtained based on reported practices (such as the regulator) and your own network (such as the desired occupancy). The recommended values are shown in Table 1. Since the current loop aggregation cycle is 30 seconds, the update cycle is set to 30 seconds in order to quickly feedback the variation of mainline traffic to the ramp control.

Table 1 Calibrated parameters for the ALINEA algorithm

Calibrated parameters	Calibrated values
Location of downstream detector station	60 m

Desired occupancy	18%
Update cycle of the metering rate	30 seconds
Regulation parameter K_R	70 veh / hour

4 Technical supports

4.1 Release notes

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

1. We delete the row “time period to accumulate detector data”.
2. We delete the queue override algorithm embedded in the previous version of this plugin. In the “alinea_control” file, you need to delete the row “queue detector”.

4.2 Future development

As a promising ramp metering algorithm, ALINEA has been considered to adapt to various conditions. For example, Caltrans has placed detectors to the upstream. Researchers have worked a prediction method that can use upstream detector to predict the traffic condition at downstream.

In addition, ALINEA has experienced some minor modifications in order to have a better control of traffic. For example, the calculation of metering rate of next control interval is based on the measured metering rate using on-ramp detector. It may happen that last interval has no vehicles but the current interval will have a lot. Based on the current algorithm, the calculated metering rate will be a smaller value, which will face problems to release these in-coming vehicles to the freeway.

The proposer of ALINEA, Papageorgiou, has had found these problems and had solutions. We will integrate these new things to this plugin in the future.

4.3 FAQ

1. We want to implement ALINEA with a queue override strategy. If the “plugins” or “programming” file specifies the on-ramp queue control plugin later than ALINEA plugin, what will happen?

The on-ramp queue control will not be functional because you give ALINEA higher priority than the on-ramp queue control strategy.

4.4 Contact information

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