Fuel control in Wialon

Oleg Zharkovsky

Deputy Head of Technical Consulting
Wialon Trainers Team Lead, Gurtam

TelematicsX
Partner conference

Agenda

1. All **fuel** settings with **fuel** chart samples
2. **Fuel** consumption by rates
3. **Fuel** traffic
4. **Fuel**, **fuel**, **fuel**...
5. MOAR **FUEL**!!!

Ways to control fuel

No sensors (rates)

– no thefts detection

– low accuracy

+ cheap

Fuel Level Sensors (FLS)

+ fillings/thefts detection

– low consumption accuracy

– must be calibrated

Fuel Consumption Sensors (FCS)

– no fillings/thefts detection

+ high consumption accuracy

Ways to control fuel

No sensors (rates)

– no thefts detection
– low accuracy
+ cheap

Fuel Level Sensors (FLS)

+ fillings/thefts detection
– low consumption accuracy
– must be calibrated

Fuel Consumption Sensors (FCS)

– no fillings/thefts detection
+ high consumption accuracy

combine
FLS & FCS

Ways to control fuel

No sensors (rates)

− no thefts detection
− low accuracy
+ cheap

Fuel Level Sensors (FLS)

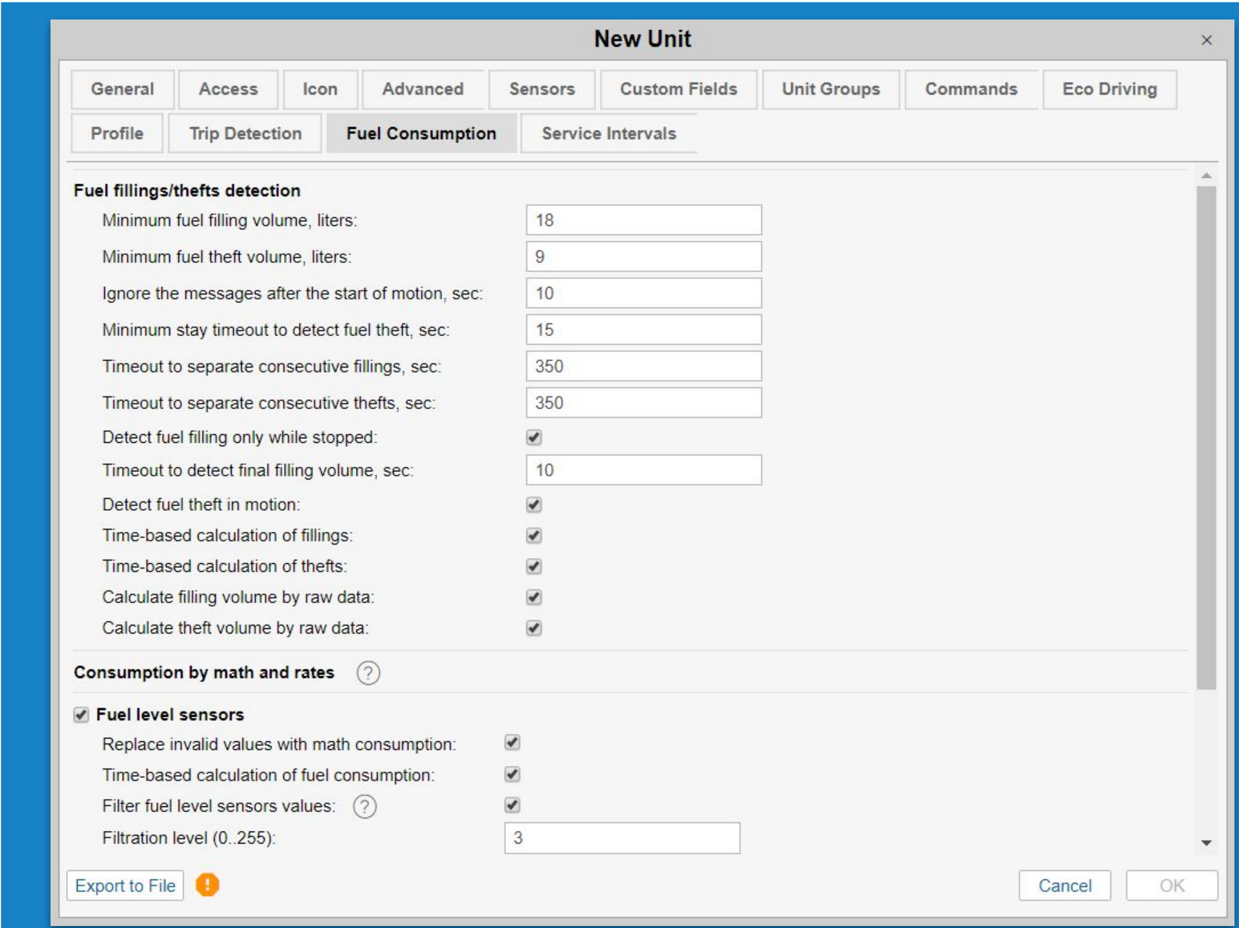
+ fillings/thefts detection
− low consumption accuracy
− must be calibrated

Fuel Consumption Sensors (FCS)

− no fillings/thefts detection
+ high consumption accuracy

combine
FLS & FCS

FLS settings in Wialon



Fuel fillings/thefts detection

Minimum fuel filling volume, liters:	18
Minimum fuel theft volume, liters:	9
Ignore the messages after the start of motion, sec:	10
Minimum stay timeout to detect fuel theft, sec:	15
Timeout to separate consecutive fillings, sec:	350
Timeout to separate consecutive thefts, sec:	350
Detect fuel filling only while stopped:	☑
Timeout to detect final filling volume, sec:	10
Detect fuel theft in motion:	☑
Time-based calculation of fillings:	☑
Time-based calculation of thefts:	☑
Calculate filling volume by raw data:	☑
Calculate theft volume by raw data:	☑

Consumption by math and rates ?

☑ **Fuel level sensors**

Replace invalid values with math consumption:	☑
Time-based calculation of fuel consumption:	☑
Filter fuel level sensors values: ?	☑
Filtration level (0..255):	3

Sensor settings (including calibration procedure)

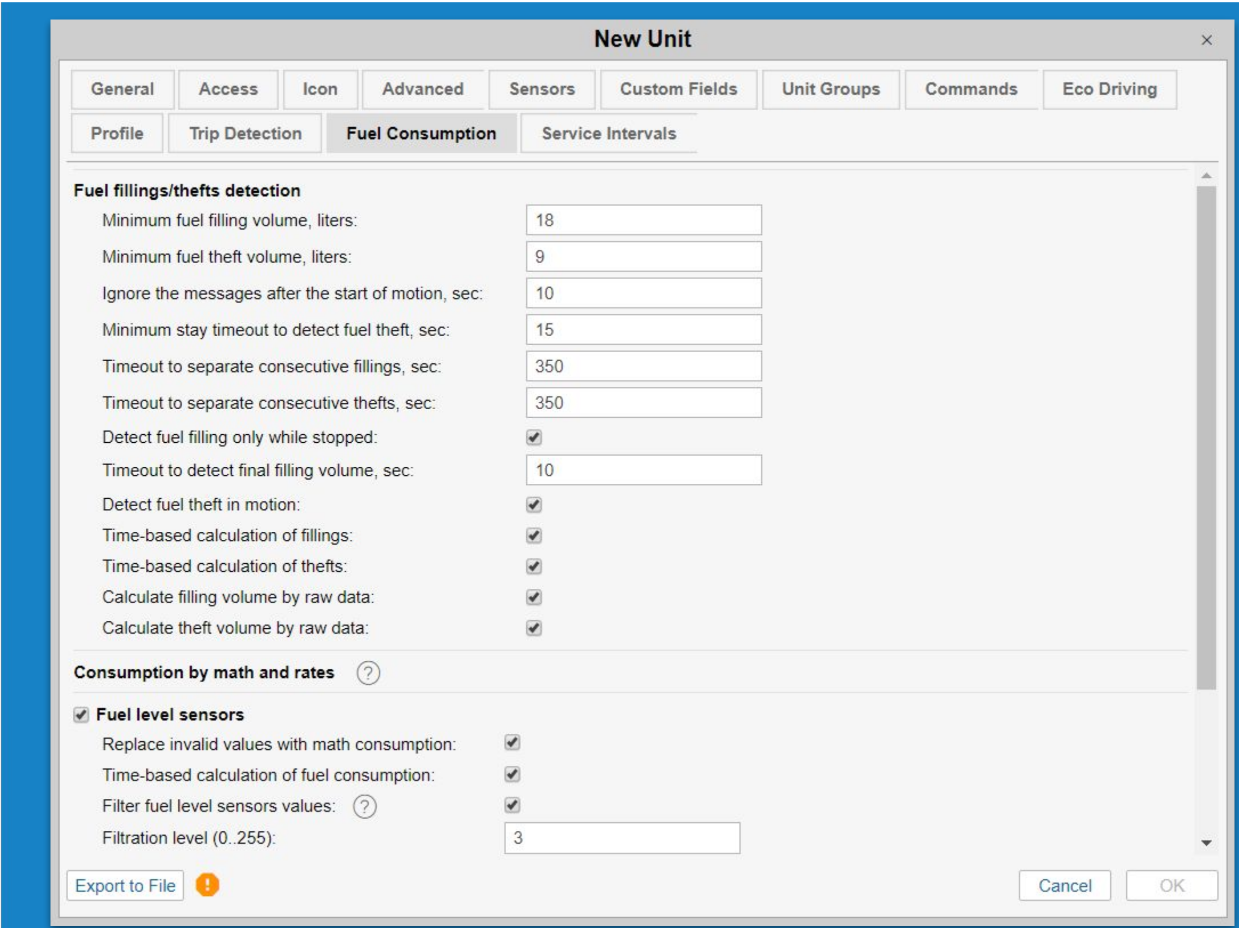
+

17 settings on tab “Fuel Consumption”

+

Mathematical model of correct consumption

FLS settings in Wialon



Fuel fillings/thefts detection

Minimum fuel filling volume, liters: 18

Minimum fuel theft volume, liters: 9

Ignore the messages after the start of motion, sec: 10

Minimum stay timeout to detect fuel theft, sec: 15

Timeout to separate consecutive fillings, sec: 350

Timeout to separate consecutive thefts, sec: 350

Detect fuel filling only while stopped: ☑

Timeout to detect final filling volume, sec: 10

Detect fuel theft in motion: ☑

Time-based calculation of fillings: ☑

Time-based calculation of thefts: ☑

Calculate filling volume by raw data: ☑

Calculate theft volume by raw data: ☑

Consumption by math and rates ?

☑ **Fuel level sensors**

Replace invalid values with math consumption: ☑

Time-based calculation of fuel consumption: ☑

Filter fuel level sensors values: ? ☑

Filtration level (0..255): 3

Sensor settings (including calibration procedure)

+

17 settings on tab “Fuel Consumption”

+

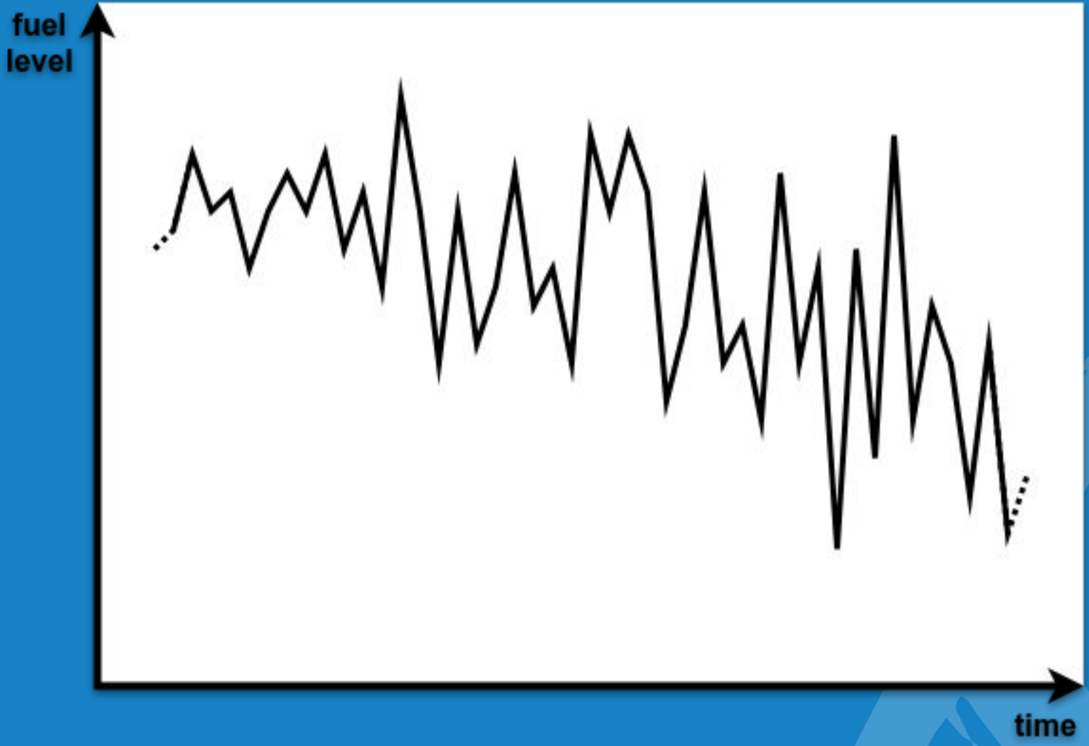
Mathematical model of correct consumption

Filtration

Fuel level sensor data always has fluctuations.

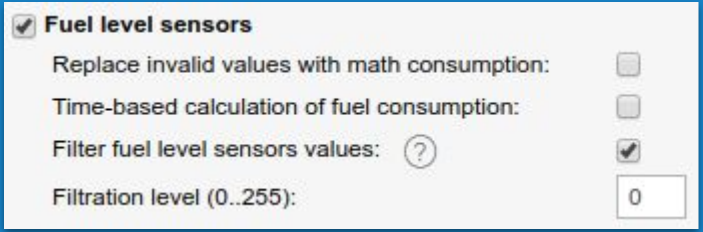
What affects fluctuations?

- engine vibrations
- tank volume and shape
- rough road
- movement at a slope
- impurities in the fuel
- wrong calibration
- power fluctuations
- messages sending frequency



Filtration

Let’s add a small filtration.



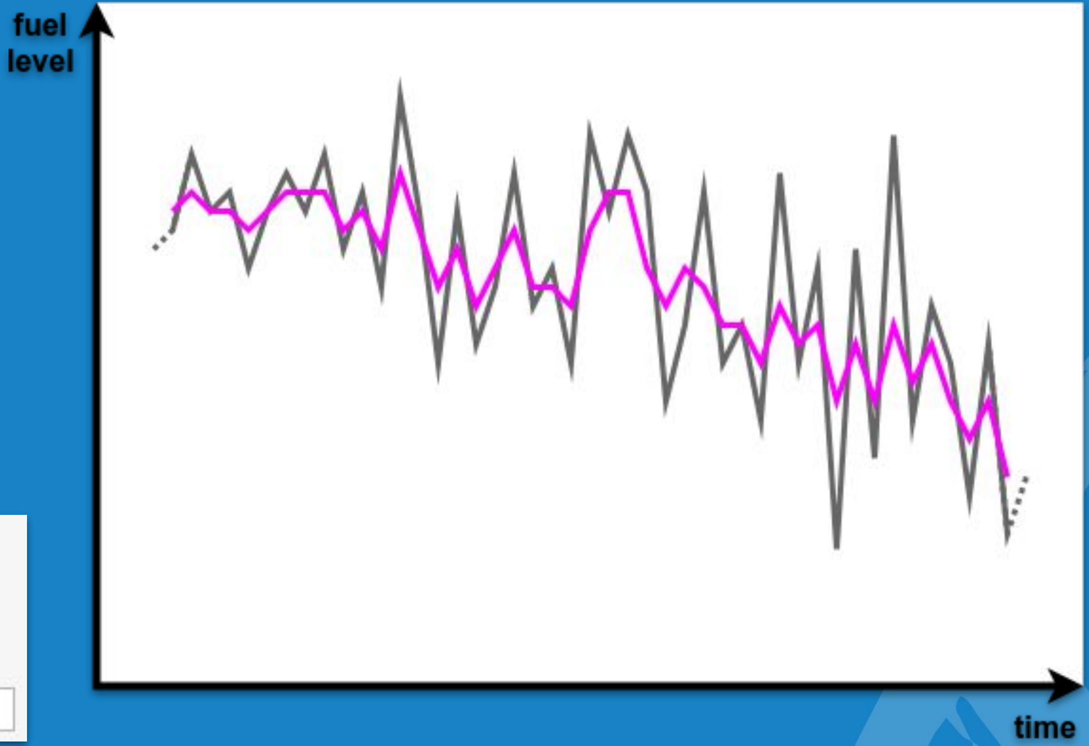
☑ **Fuel level sensors**

Replace invalid values with math consumption: □

Time-based calculation of fuel consumption: □

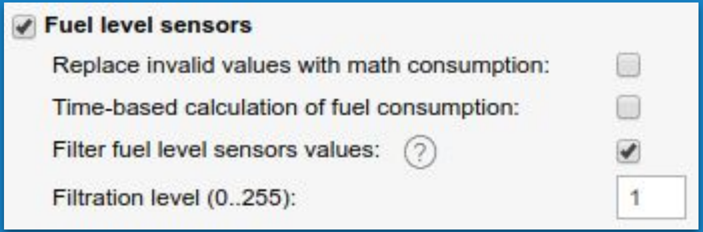
Filter fuel level sensors values: ⓘ ☑

Filtration level (0..255): 0



Filtration

That’s not enough.
Let’s add more filtration.



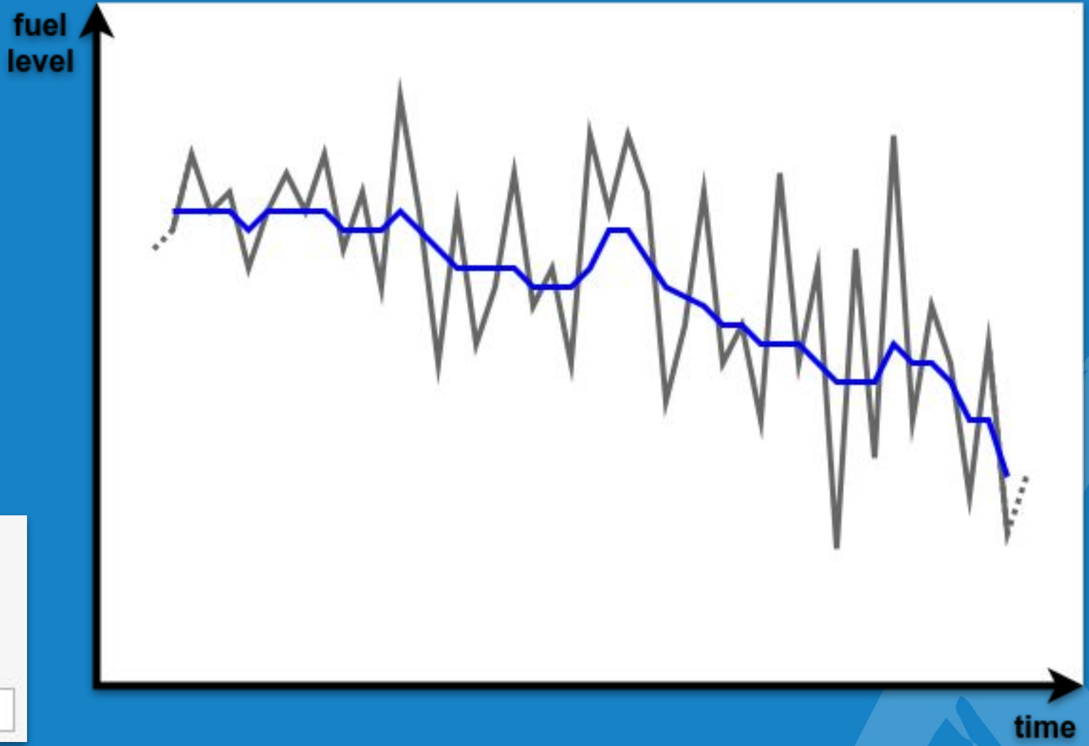
☑ **Fuel level sensors**

Replace invalid values with math consumption: □

Time-based calculation of fuel consumption: □

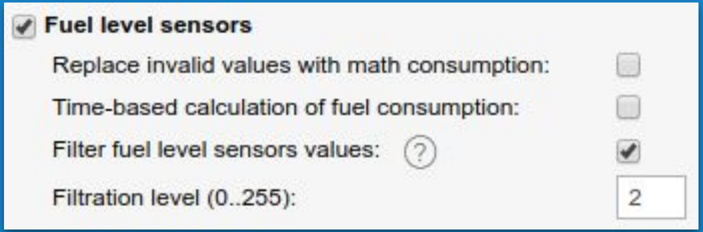
Filter fuel level sensors values: ? ☑

Filtration level (0..255): 1



Filtration

It seems like that’s OK now!



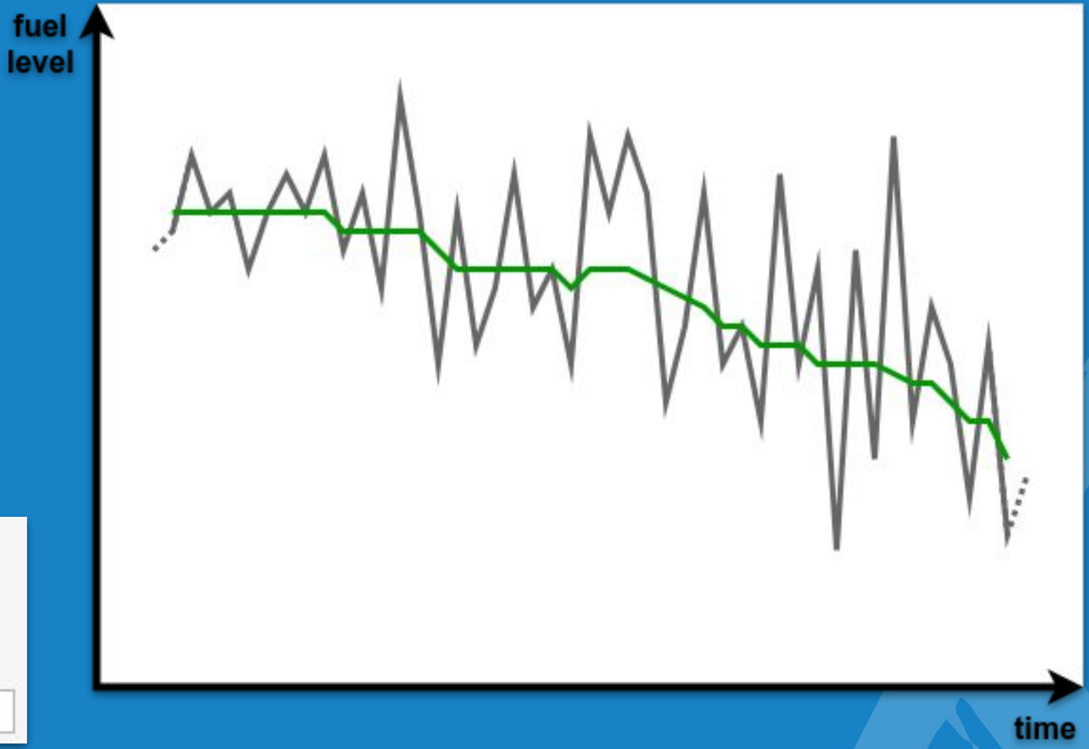
☑ **Fuel level sensors**

Replace invalid values with math consumption: ☐

Time-based calculation of fuel consumption: ☐

Filter fuel level sensors values: ? ☑

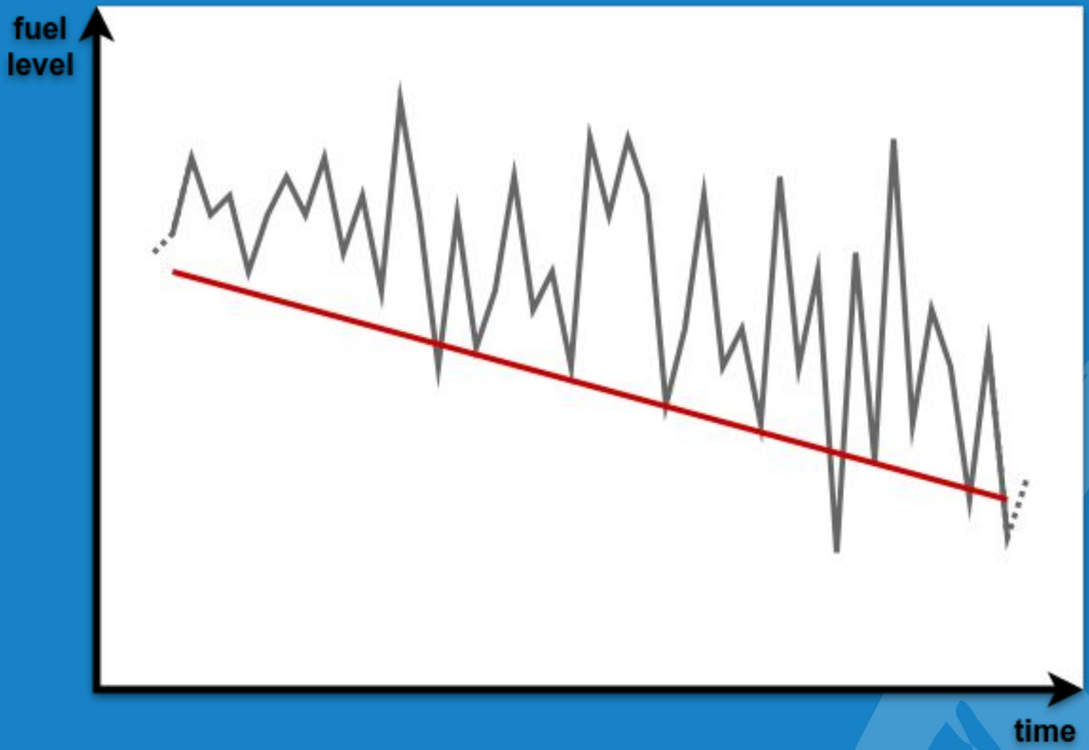
Filtration level (0..255): 2



Filtration

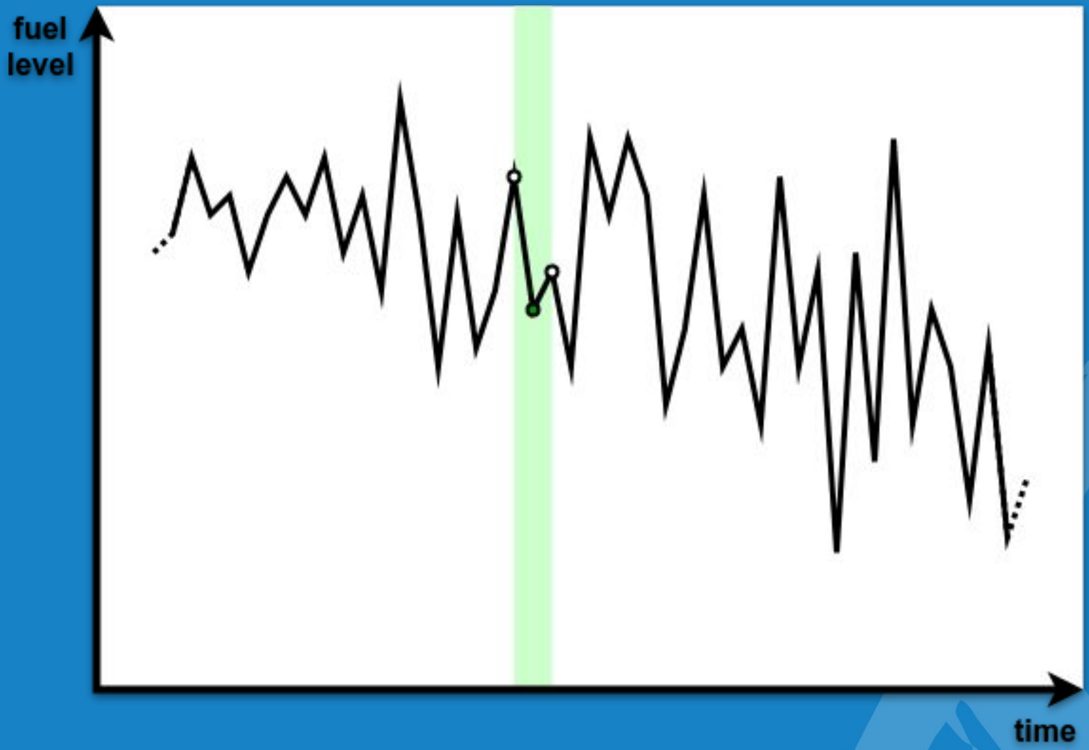
Don’t think bigger filtration level is always better. Because at some level it will affect incoming data too much. Output data will be smoothed, but wrong.

Usually we don’t recommend to use filtration level more than 7-10.



Filtration

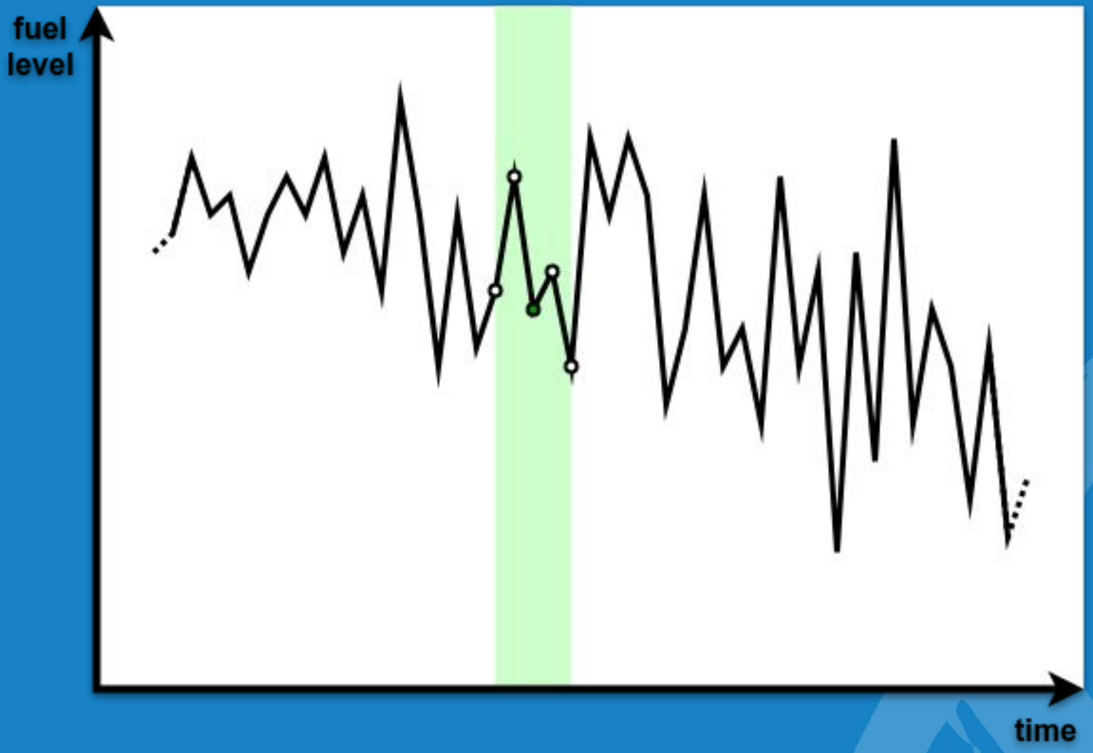
We use median filtration in Wialon. You can set it up with a help of filtration level. The bigger level you set – the wider sliding window is used to calculate smoothed value in the middle.



Filtration

We use median filtration in Wialon. You can set it up with a help of filtration level. The bigger level you set – the wider sliding window is used to calculate smoothed value in the middle.

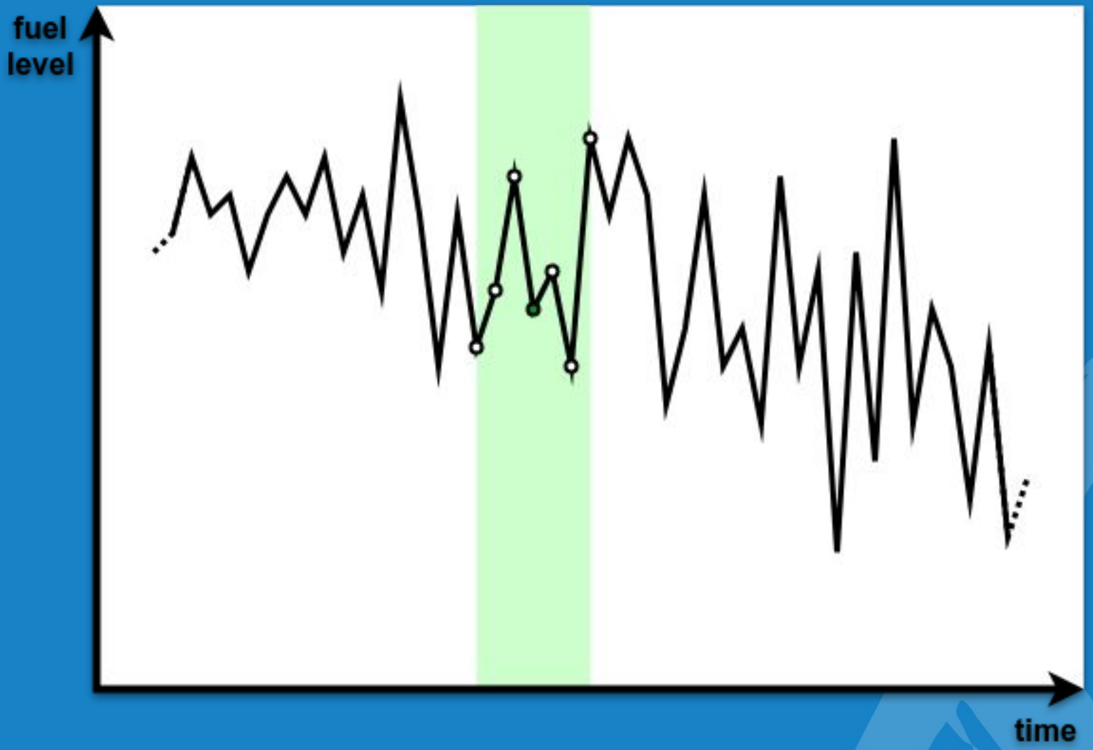
Let’s make it wider…



Filtration

We use median filtration in Wialon. You can set it up with a help of filtration level. The bigger level you set – the wider sliding window is used to calculate smoothed value in the middle.

Let’s make it wider…
Now it’s OK

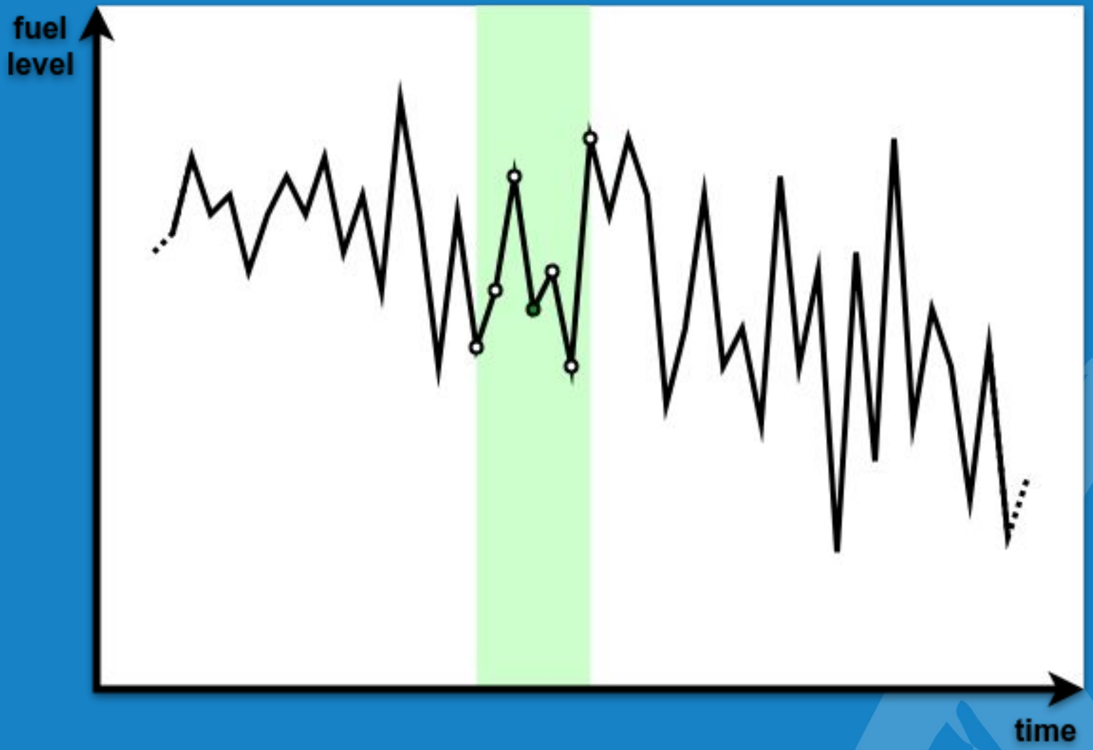


Filtration

We use median filtration in Wialon. You can set it up with a help of filtration level. The bigger level you set – the wider sliding window is used to calculate smoothed value in the middle.

Let's make it wider…
Now it's OK

And then it slides ↔

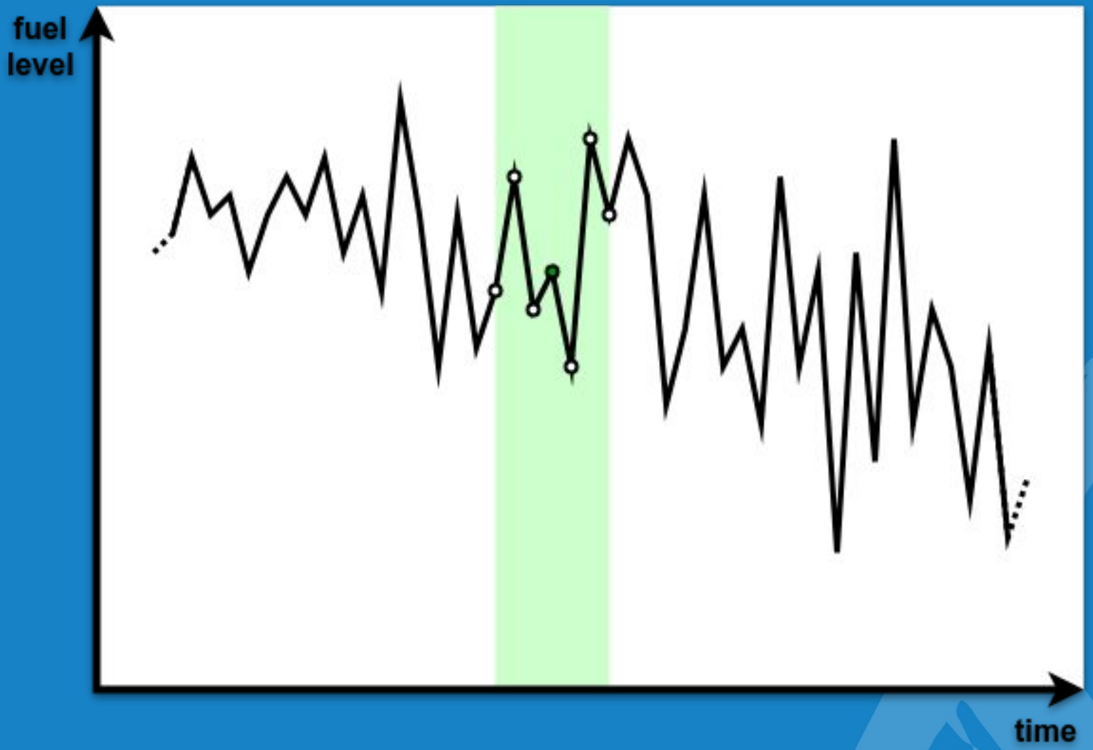


Filtration

We use median filtration in Wialon. You can set it up with a help of filtration level. The bigger level you set – the wider sliding window is used to calculate smoothed value in the middle.

Let’s make it wider…
Now it’s OK

And then it slides ↔

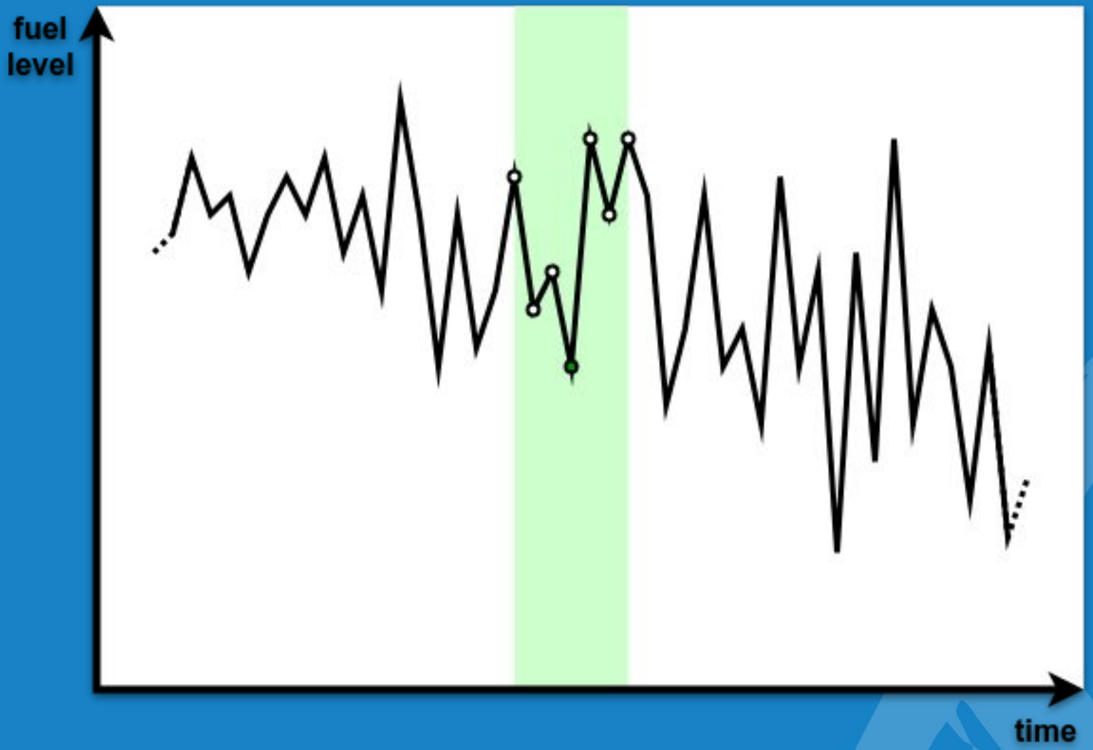


Filtration

We use median filtration in Wialon. You can set it up with a help of filtration level. The bigger level you set – the wider sliding window is used to calculate smoothed value in the middle.

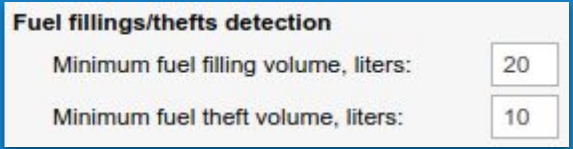
Let’s make it wider…
Now it’s OK

And then it slides ↔



Fillings/Thefts

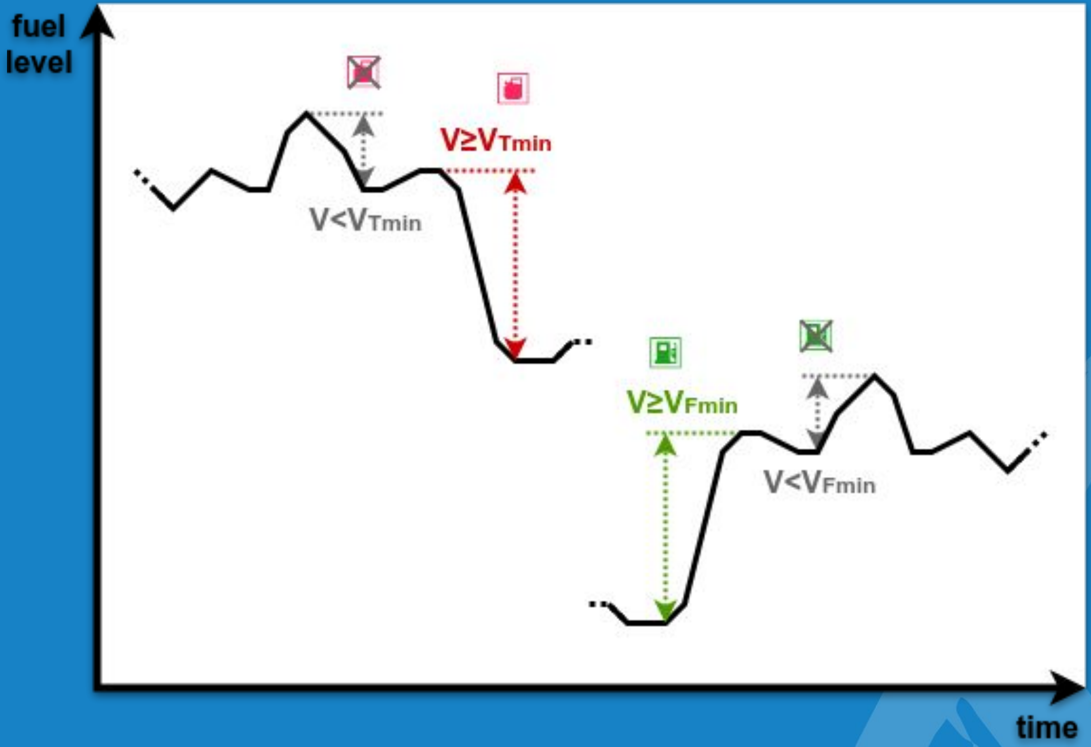
Even after filtration there will be some fluctuations left. So we need to separate these fluctuations from fillings and thefts with a help of rough filter – minimum fuel filling and theft volume (V_{Fmin} & V_{Tmin}).



Fuel fillings/thefts detection

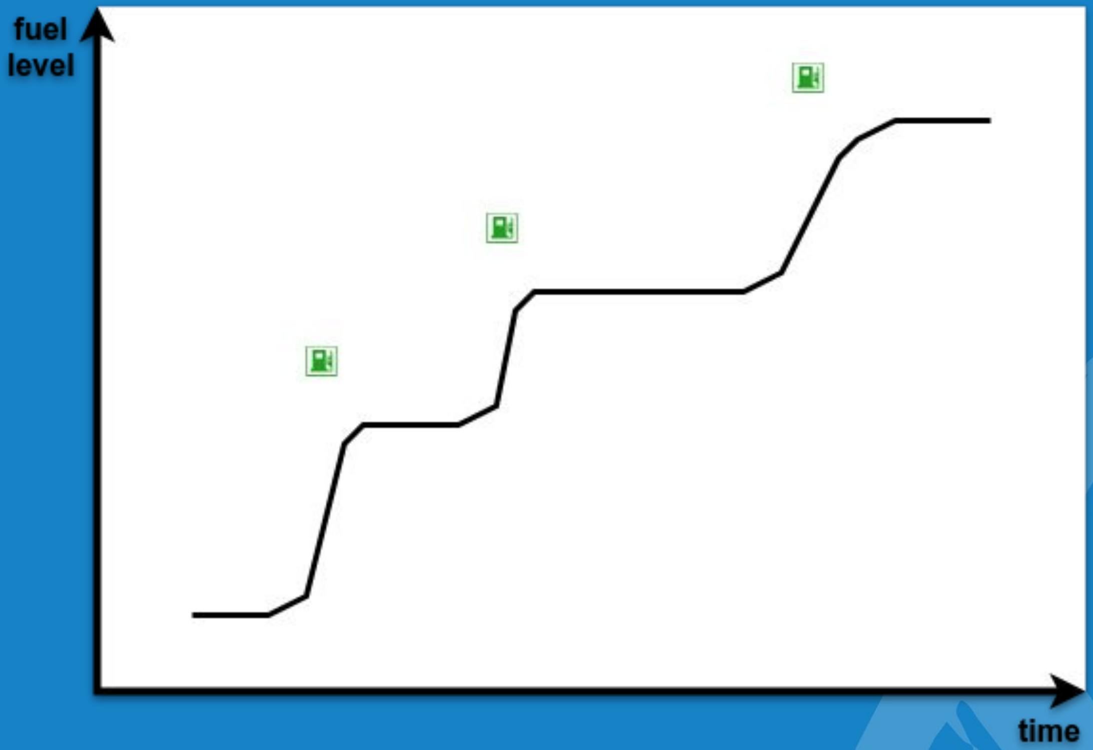
Minimum fuel filling volume, liters: 20

Minimum fuel theft volume, liters: 10



Fillings/Thefts

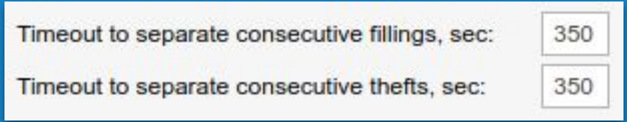
Sometimes fillings/thefts are done in several parts (or fuel just goes in/out by portions).



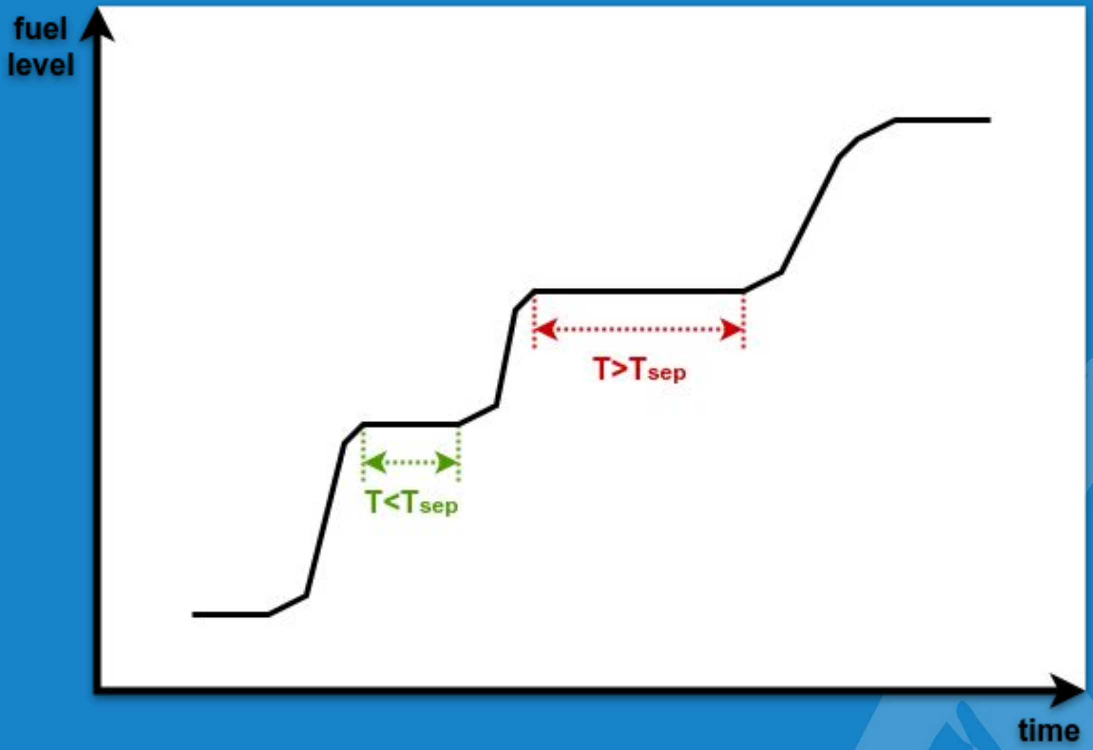
Fillings/Thefts

Sometimes fillings/thefts are done in several parts (or fuel just goes in/out by portions).

You can use timeout (T_{sep}) to separate or to merge them into one.



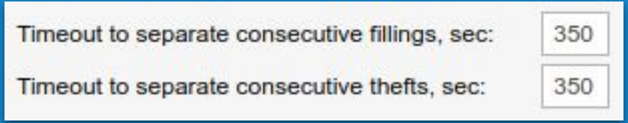
Timeout to separate consecutive fillings, sec:	350
Timeout to separate consecutive thefts, sec:	350



Fillings/Thefts

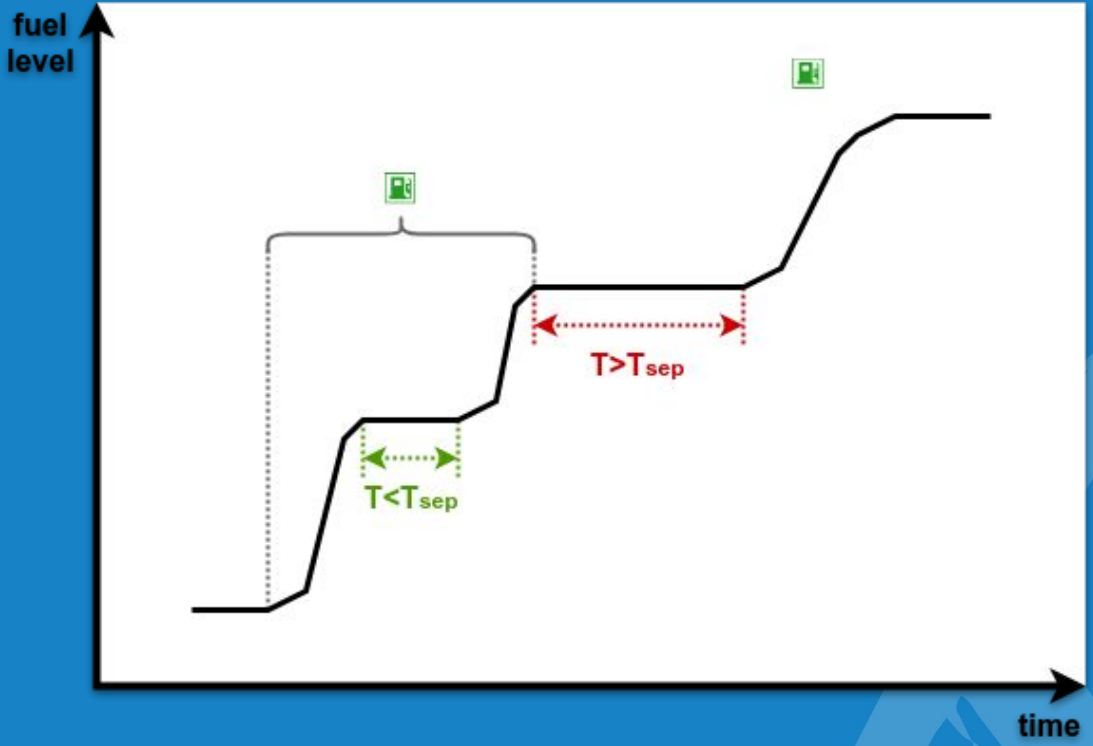
Sometimes fillings/thefts are done in several parts (or fuel just goes in/out by portions).

You can use timeout (T_{sep}) to separate or to merge them into one.

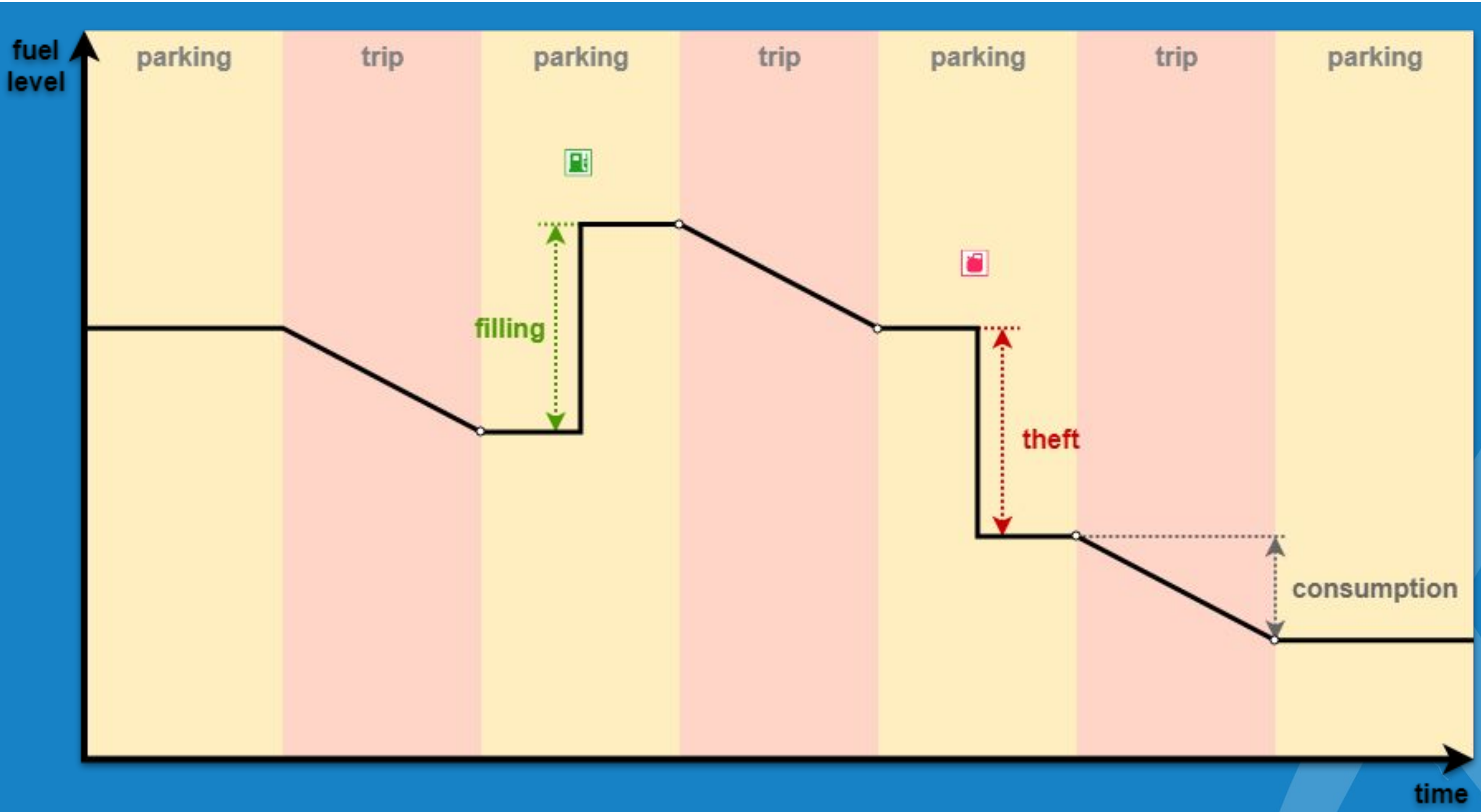


Timeout to separate consecutive fillings, sec: 350

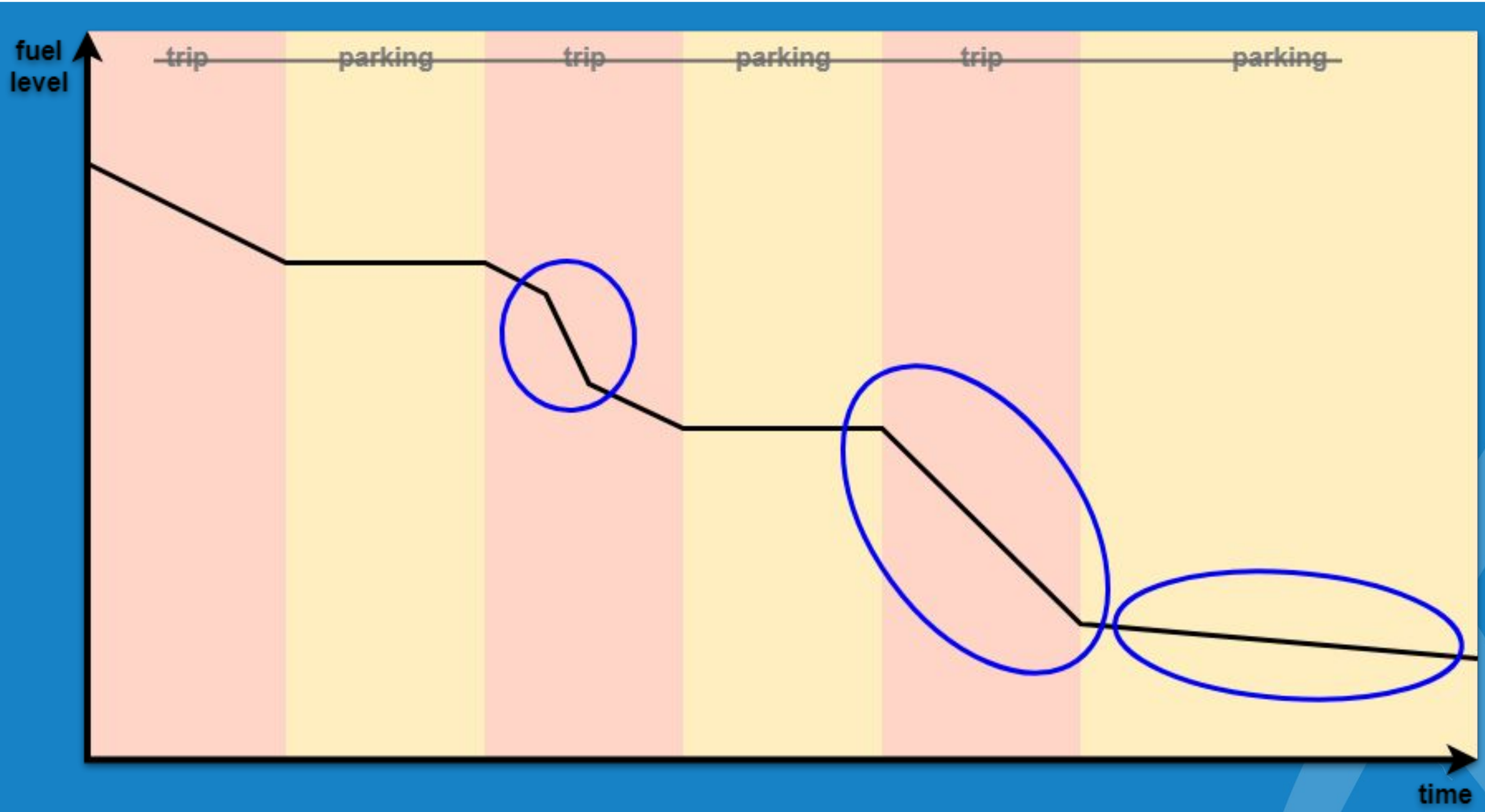
Timeout to separate consecutive thefts, sec: 350



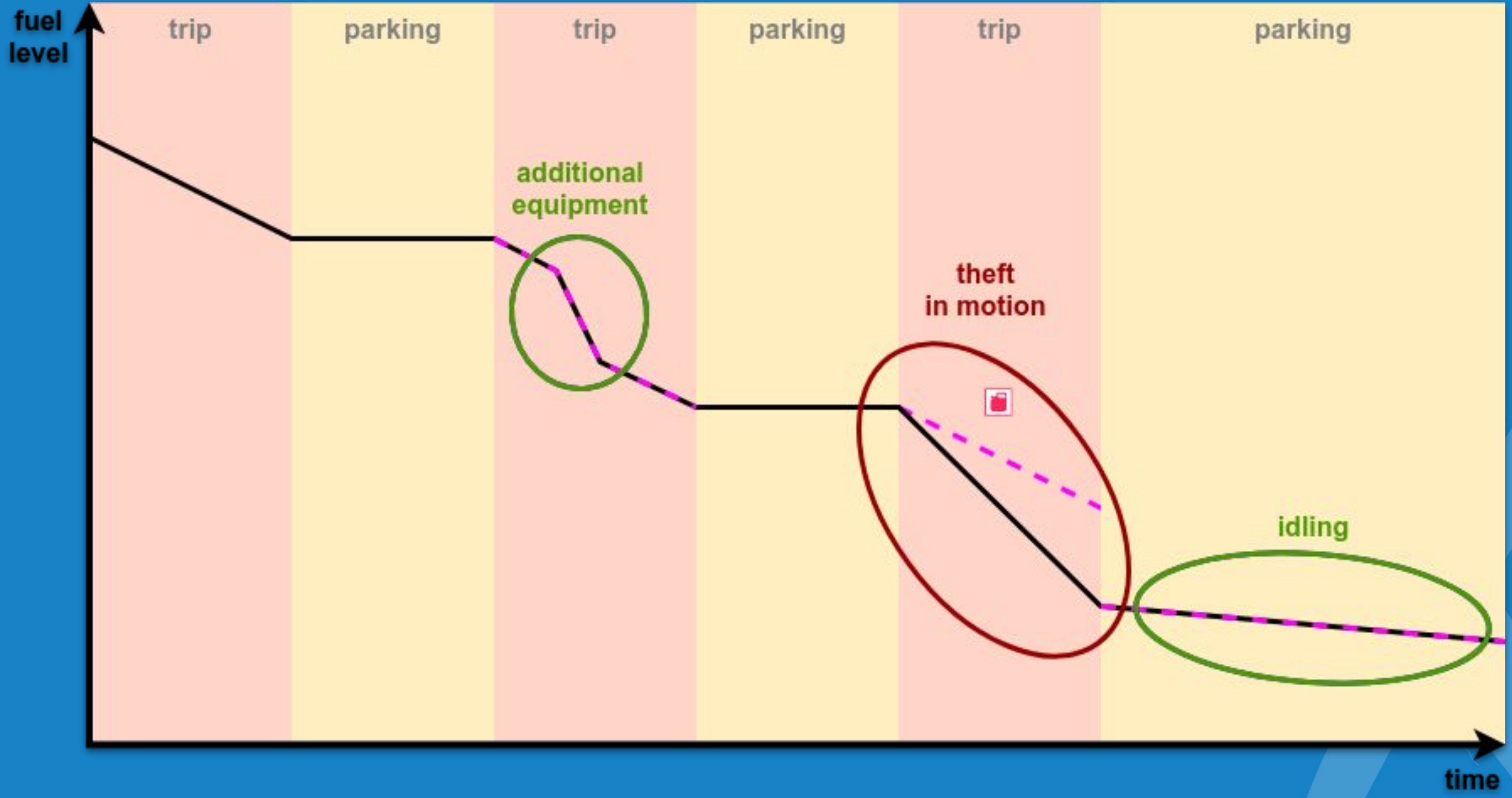
Mileage-based fuel algorithm



Time-based fuel algorithm



Time-based fuel algorithm

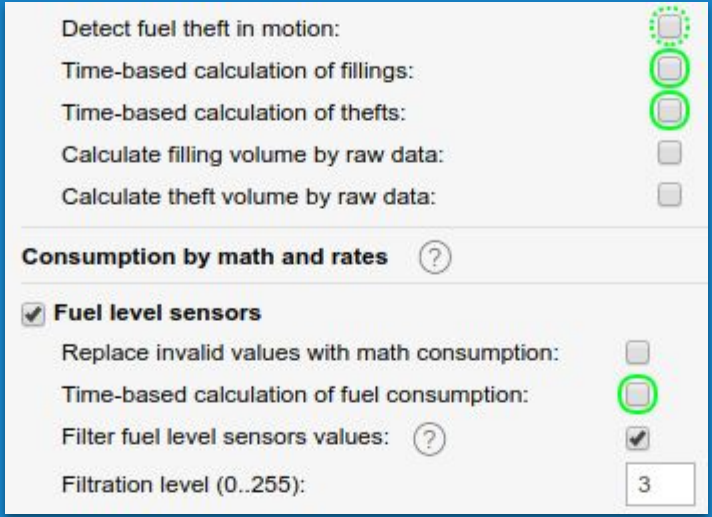


Mileage-based vs Time-based fuel algorithms

Mileage-based

Used by default, easy to configure.
Suits for:

- main part of moving units



Detect fuel theft in motion: □

Time-based calculation of fillings: □

Time-based calculation of thefts: □

Calculate filling volume by raw data: □

Calculate theft volume by raw data: □

Consumption by math and rates ?

☑ **Fuel level sensors**

Replace invalid values with math consumption: □

Time-based calculation of fuel consumption: □

Filter fuel level sensors values: ? ☑

Filtration level (0..255): 3

Time-based

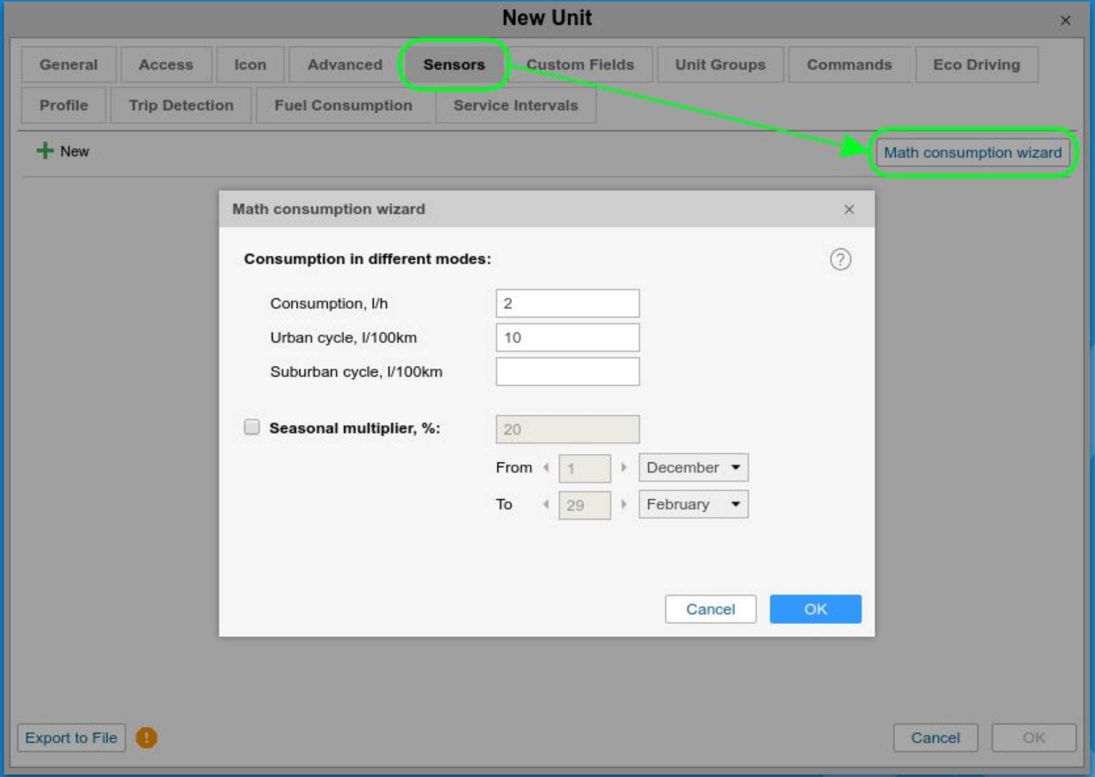
A little bit harder to configure.
Suits for:

- stationary units
- units with long intervals of idling
- units with additional equipment affecting consumption
- drivers smart enough to make a theft during movement

Mathematical model of consumption

To use time-based algorithm you should create a mathematical model of correct consumption. It can include as many factors (sensors) as you want.

Rough math model (based on speed and season) can be created in several clicks with a help of Math Consumption wizard.



Math consumption wizard

Consumption in different modes:

Consumption, l/h: 2

Urban cycle, l/100km: 10

Suburban cycle, l/100km:

□ Seasonal multiplier, %: 20

From 1 December

To 29 February

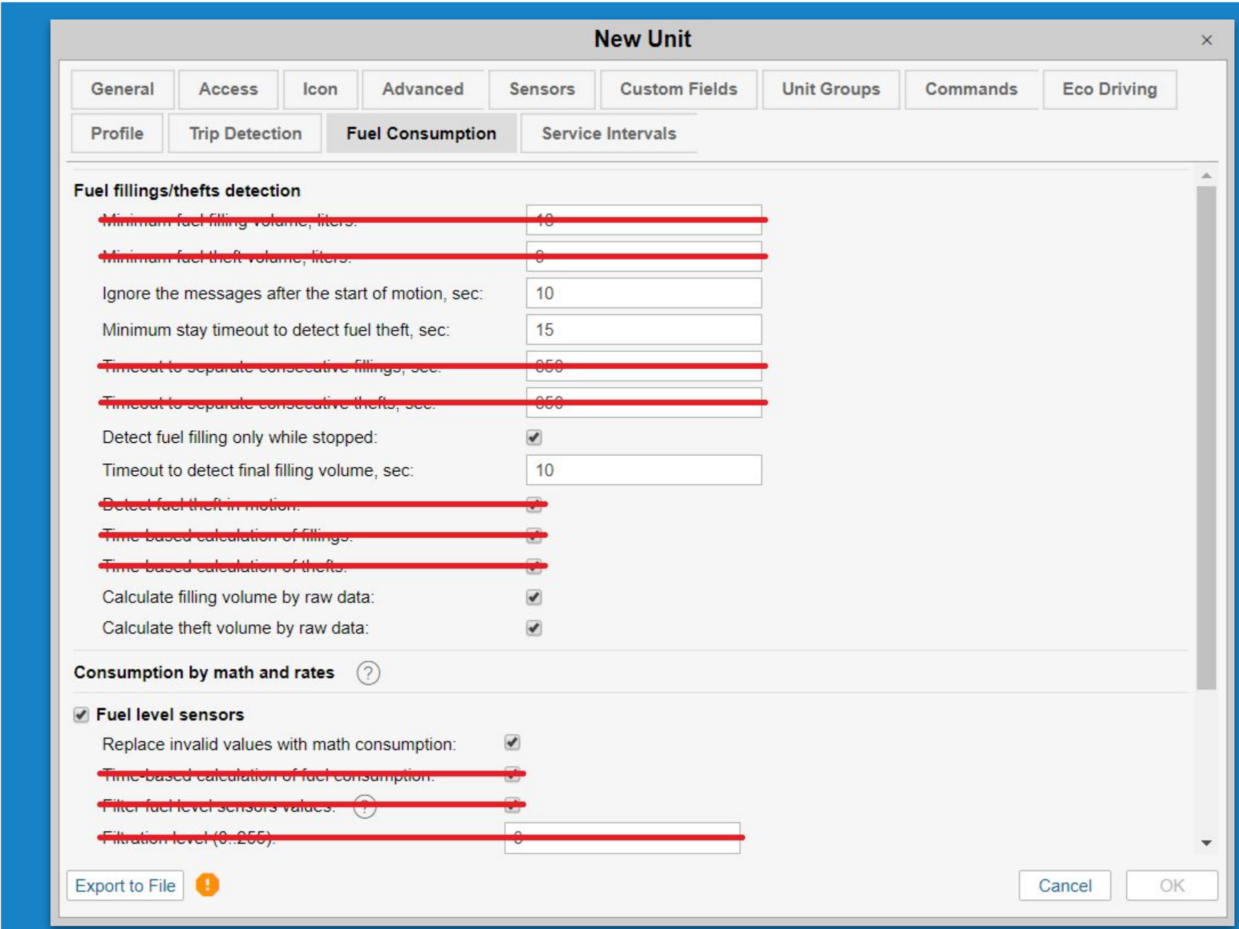
Cancel OK

Mathematical model of consumption

To use time-based algorithm you should create a mathematical model of correct consumption. It can include as many factors (sensors) as you want.

Rough math model (based on speed and season) can be created in several clicks with a help of Math Consumption wizard.

FLS settings in Wialon



Fuel fillings/thefts detection

~~Minimum fuel filling volume, liters: 10~~

~~Minimum fuel theft volume, liters: 0~~

Ignore the messages after the start of motion, sec: 10

Minimum stay timeout to detect fuel theft, sec: 15

~~Timeout to separate consecutive fillings, sec: 850~~

~~Timeout to separate consecutive thefts, sec: 850~~

Detect fuel filling only while stopped: ☑

Timeout to detect final filling volume, sec: 10

~~Detect fuel theft in motion: ☑~~

~~Time-based calculation of fillings: ☑~~

~~Time-based calculation of thefts: ☑~~

Calculate filling volume by raw data: ☑

Calculate theft volume by raw data: ☑

Consumption by math and rates ?

☑ **Fuel level sensors**

Replace invalid values with math consumption: ☑

~~Time-based calculation of fuel consumption: ☑~~

~~Filter fuel level sensors values: ? ☑~~

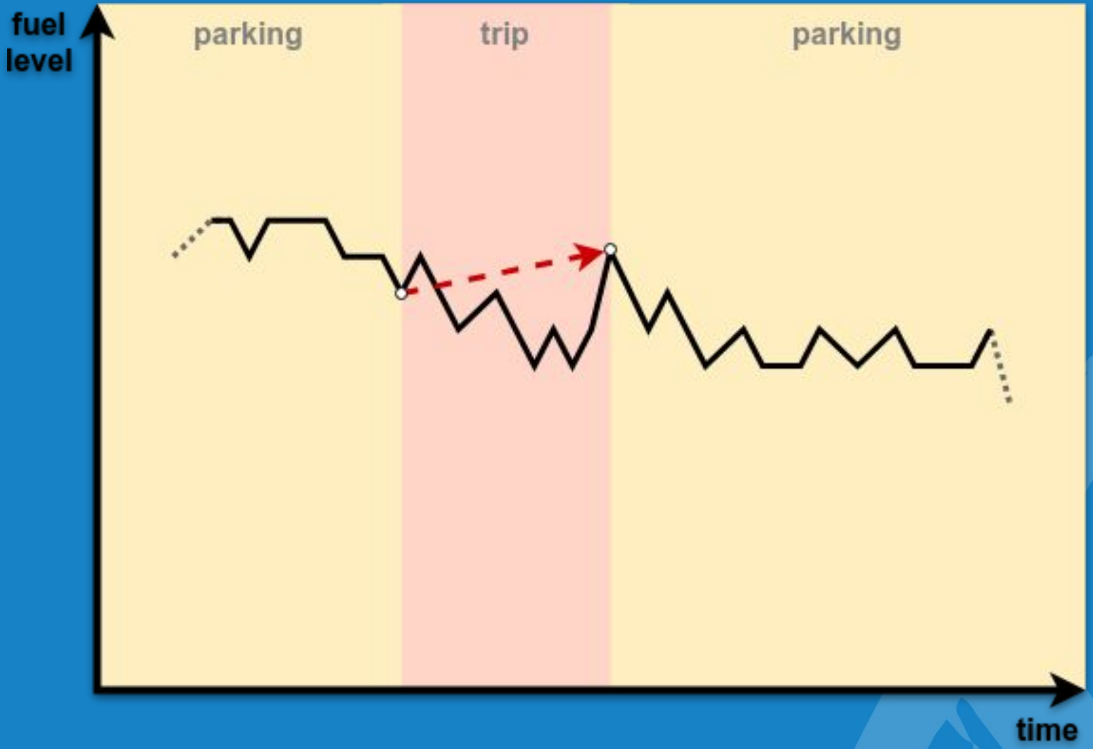
~~Filtration level (0..255): 0~~

10 most commonly used options are already discussed!

7 to go...

Math consumption

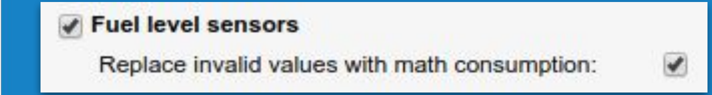
Sometimes due to fluctuations fuel level in the end of the interval (e.g. trips) is higher than in the beginning. Consumption has 0 value in such a situation.



Math consumption

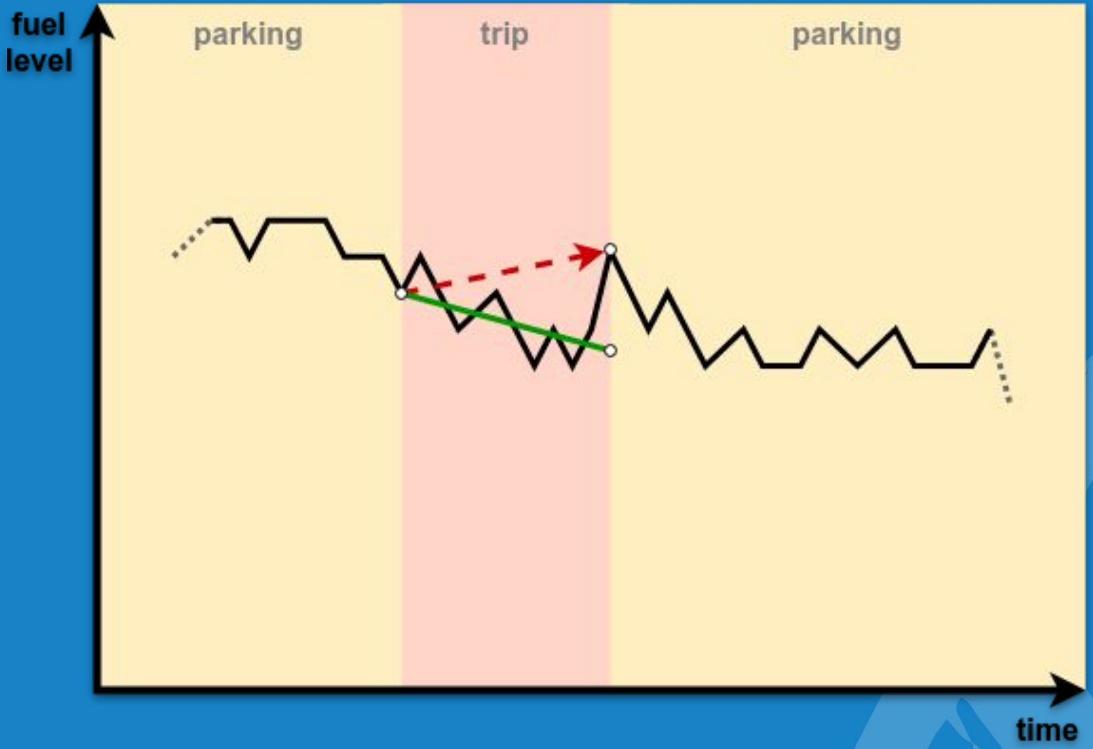
Sometimes due to fluctuations fuel level in the end of the interval (e.g. trips) is higher than in the beginning. Consumption has 0 value in such a situation.

But math model can help us once again to calculate the consumption.



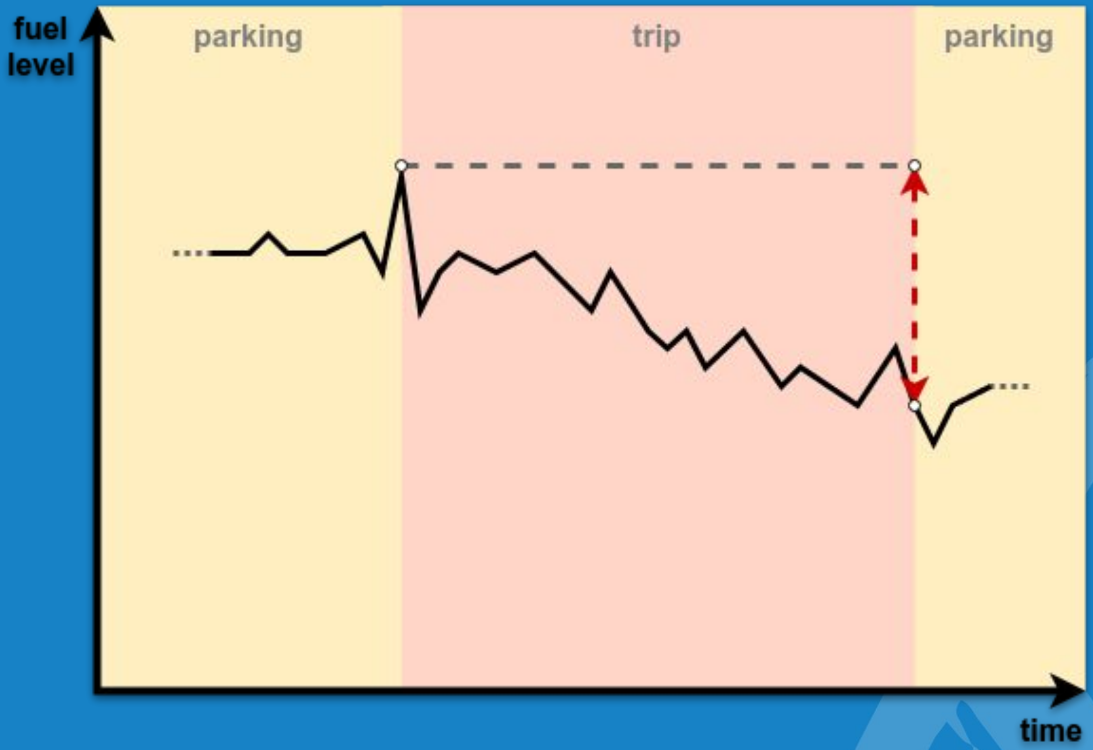
☑ **Fuel level sensors**

Replace invalid values with math consumption: ☑



Time filters

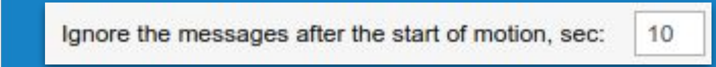
As you can see, we've got troubles with fluctuations during the trips, especially in the beginning/end of the them. But we've got options to filter them out also.



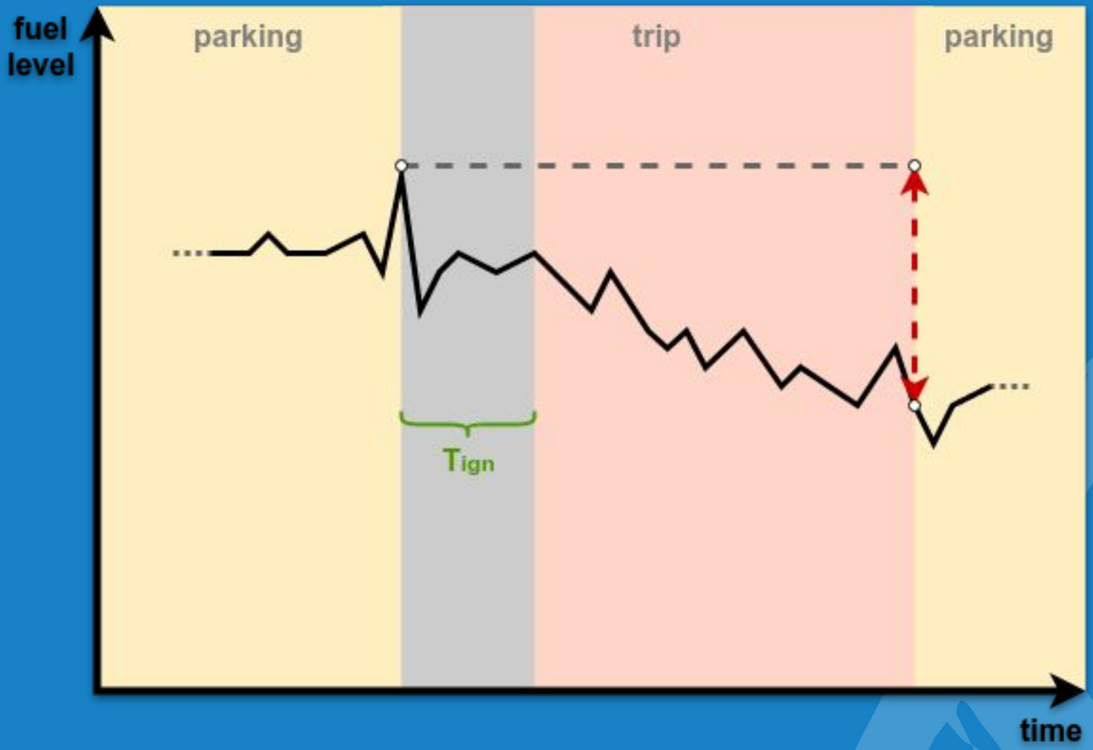
Time filters

As you can see, we’ve got troubles with fluctuations during the trips, especially in the beginning/end of the them. But we’ve got options to filter them out also.

With the following option we can ignore interval after the start of the motion.



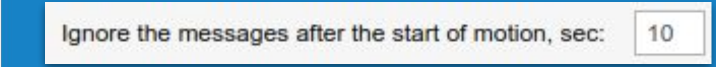
Ignore the messages after the start of motion, sec: 10



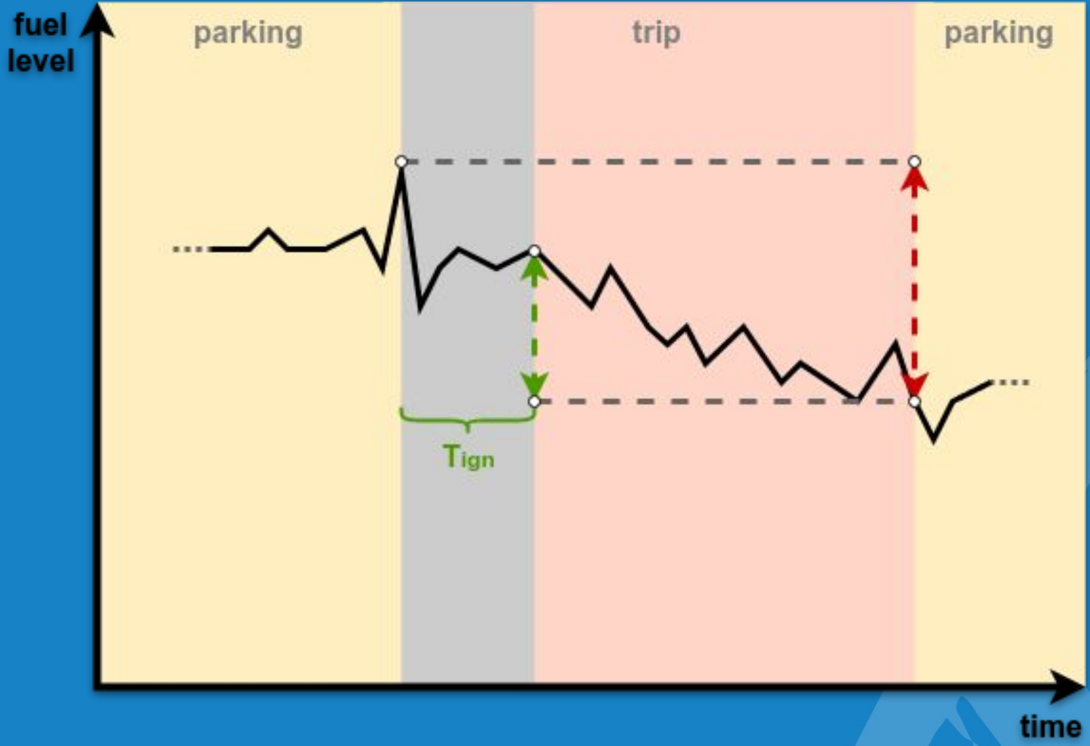
Time filters

As you can see, we’ve got troubles with fluctuations during the trips, especially in the beginning/end of the them. But we’ve got options to filter them out also.

With the following option we can ignore interval after the start of the motion.

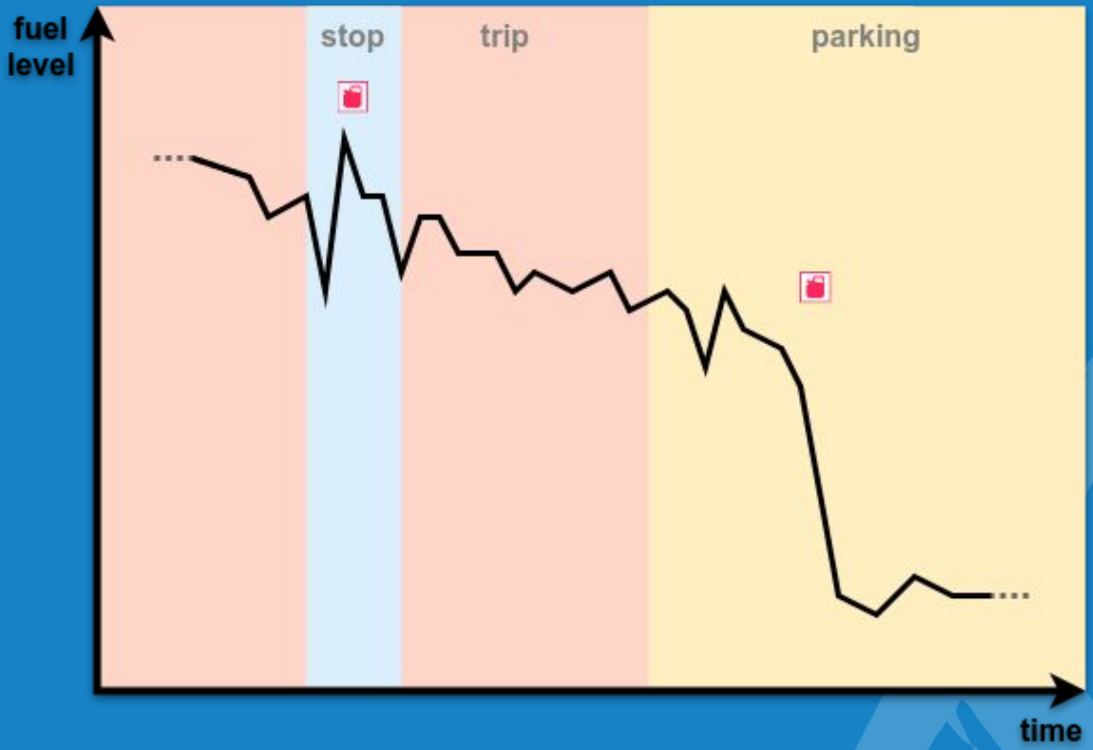


Ignore the messages after the start of motion, sec: 10



Time filters

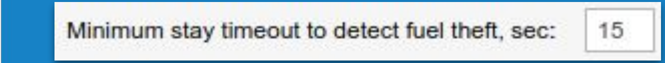
As you can see, we've got troubles with fluctuations during the trips, especially in the beginning/end of the them. But we've got options to filter them out also.



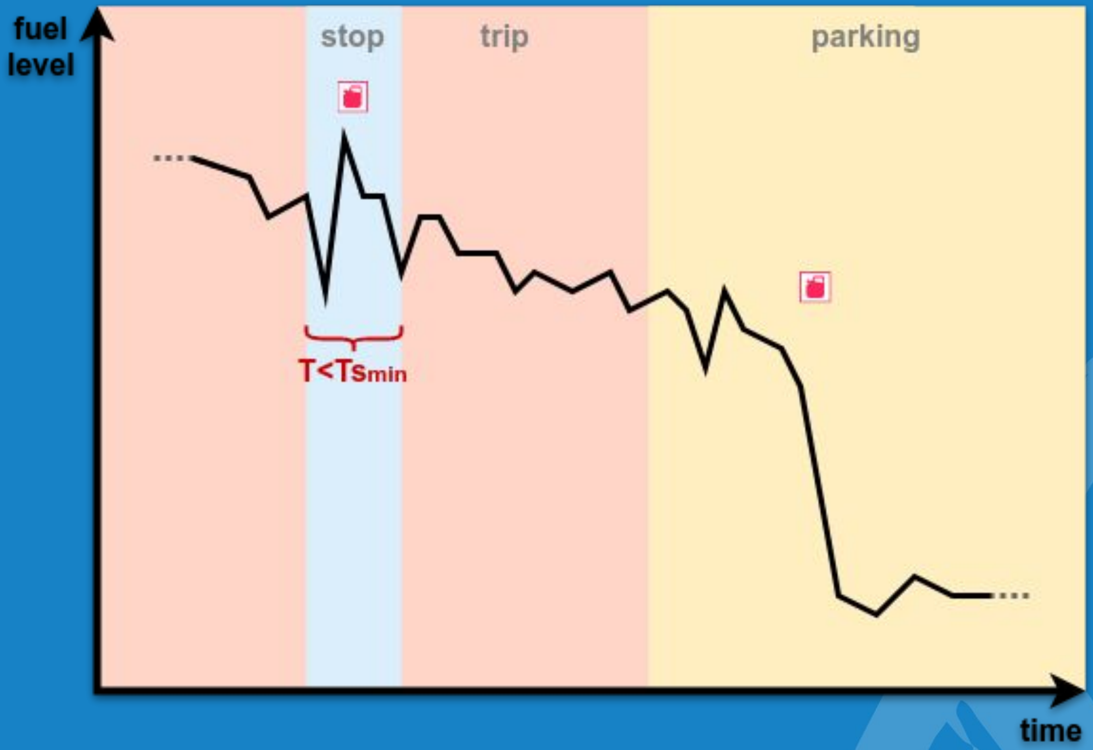
Time filters

As you can see, we’ve got troubles with fluctuations during the trips, especially in the beginning/end of the them. But we’ve got options to filter them out also.

With the following option we can stop analyzing small stops for thefts detection.



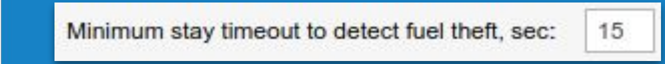
Minimum stay timeout to detect fuel theft, sec: 15



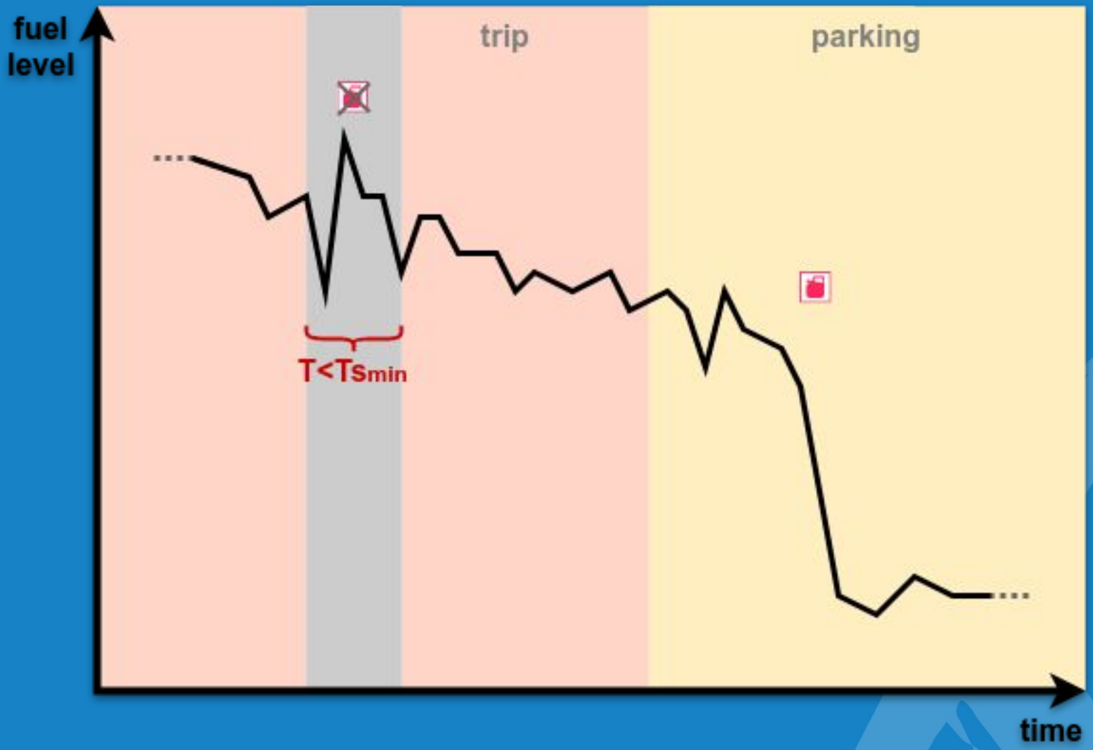
Time filters

As you can see, we’ve got troubles with fluctuations during the trips, especially in the beginning/end of the them. But we’ve got options to filter them out also.

With the following option we can stop analyzing small stops for thefts detection.

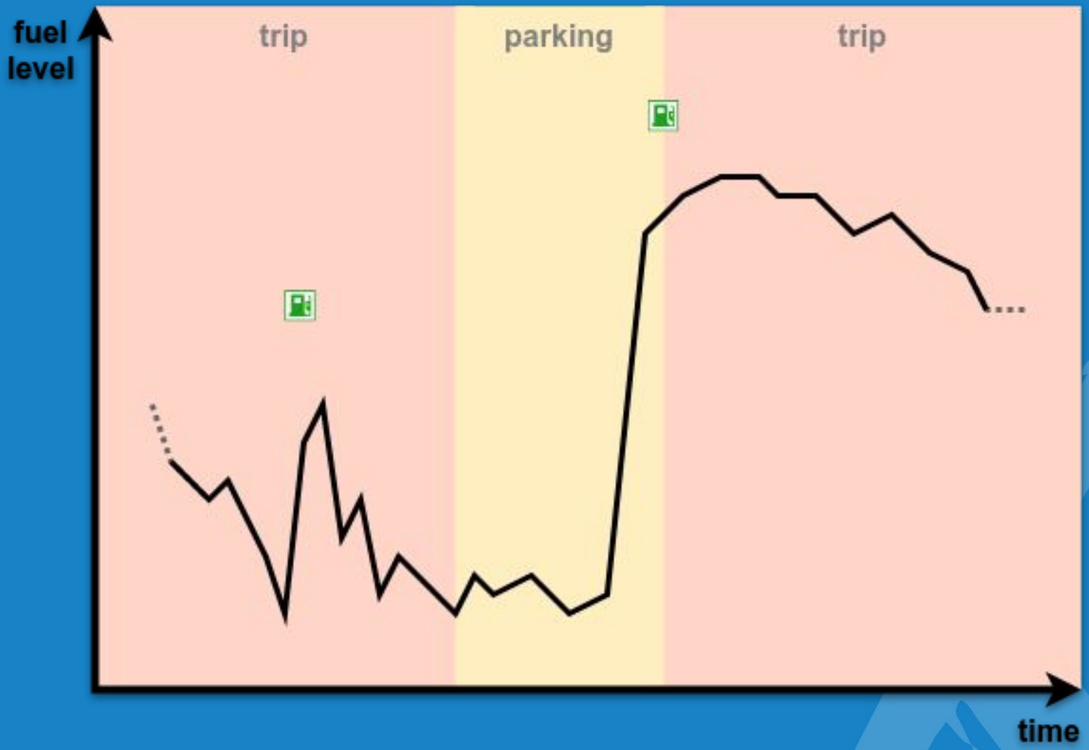


Minimum stay timeout to detect fuel theft, sec: 15



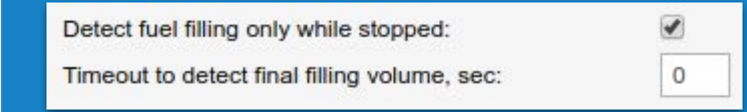
Time filters

Usually fillings detection is done for the whole FLS data, but the following option can exclude movement.



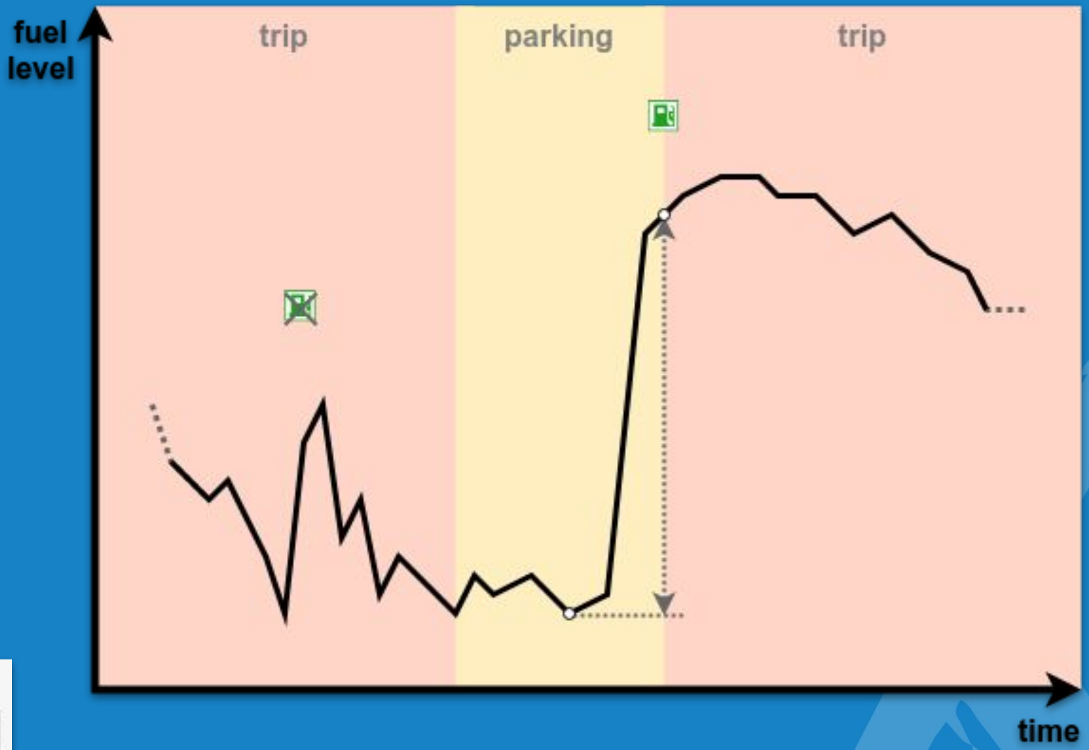
Time filters

Usually fillings detection is done for the whole FLS data, but the following option can exclude movement.



Detect fuel filling only while stopped: ☑

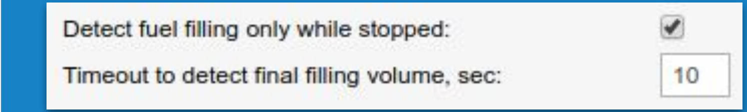
Timeout to detect final filling volume, sec: 0



Time filters

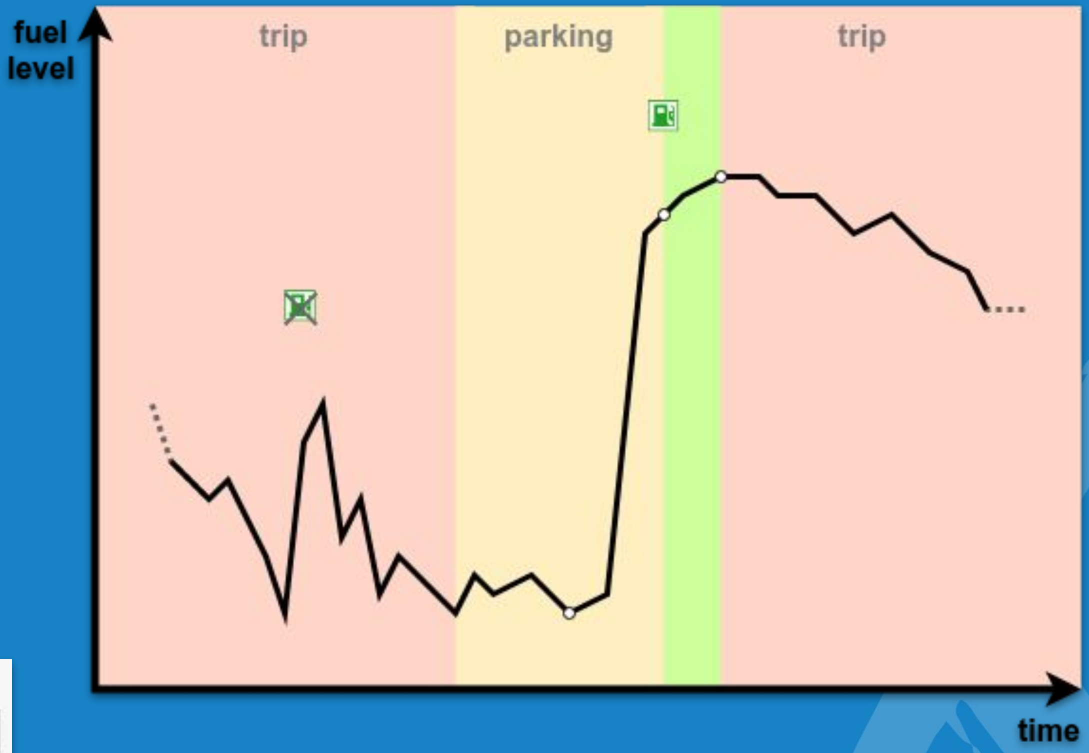
Usually fillings detection is done for the whole FLS data, but the following option can exclude movement.

Still some FLS are inertial, and their values changes with a delay. The following option can delay detection of final filling volume to compensate it.



Detect fuel filling only while stopped: ☑

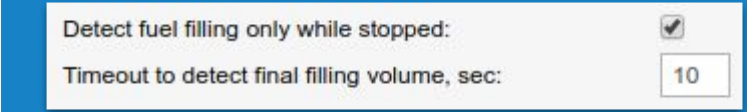
Timeout to detect final filling volume, sec: 10



Time filters

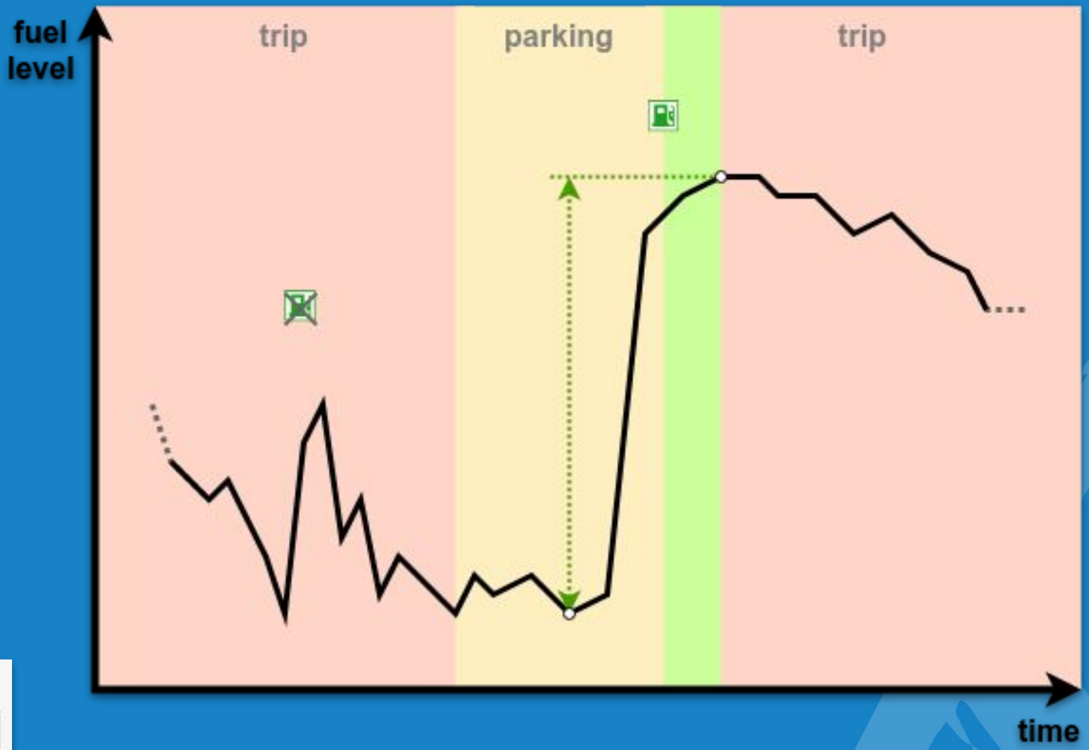
Usually fillings detection is done for the whole FLS data, but the following option can exclude movement.

Still some FLS are inertial, and their values changes with a delay. The following option can delay detection of final filling volume to compensate it.



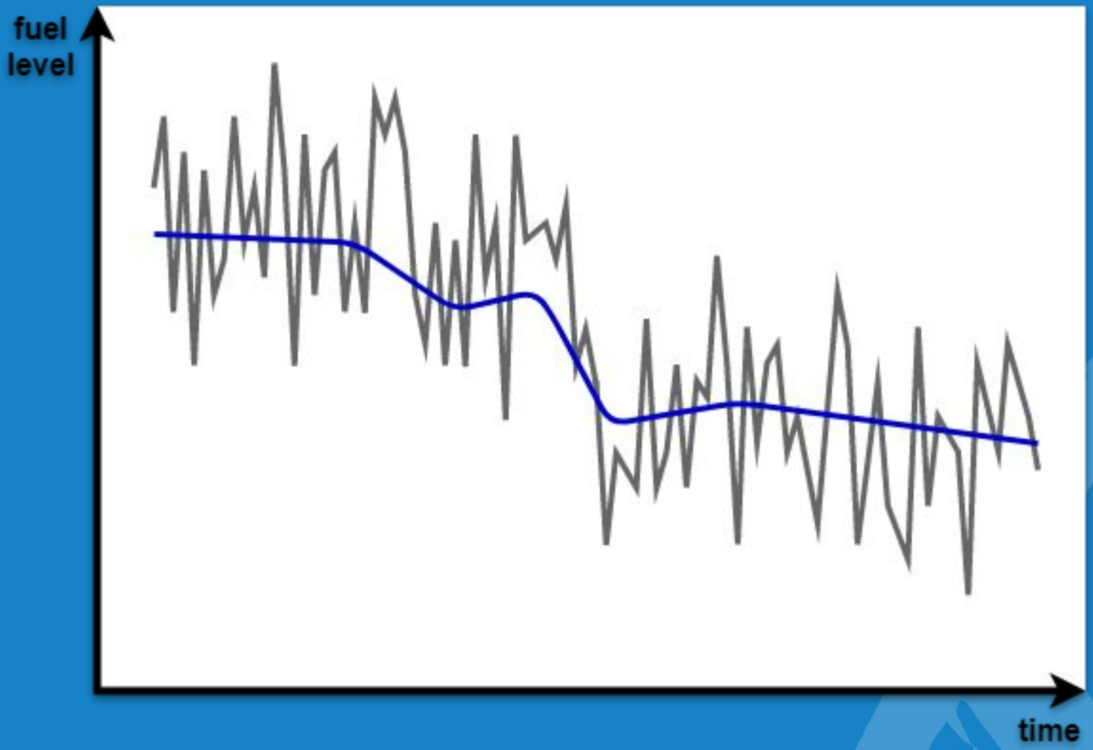
Detect fuel filling only while stopped: ☑

Timeout to detect final filling volume, sec: 10



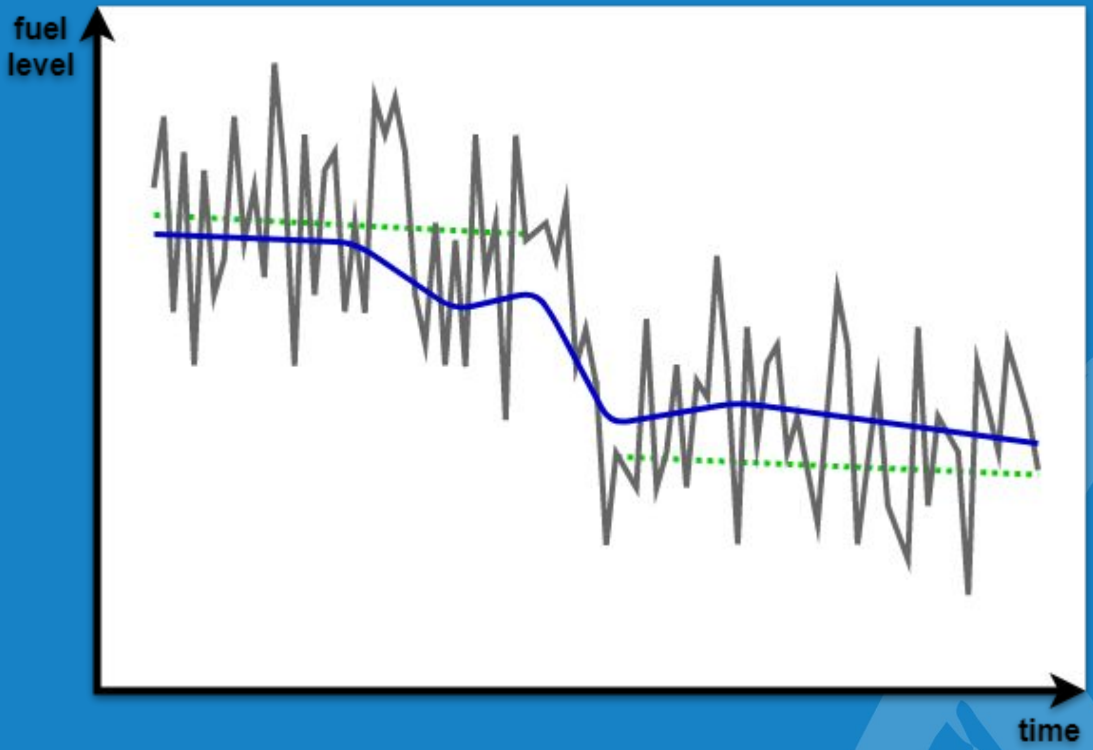
Fillings/Thefts

Sometimes filtration affects the data so much, that you have to combine processed and raw data to analyze filling/theft correctly.



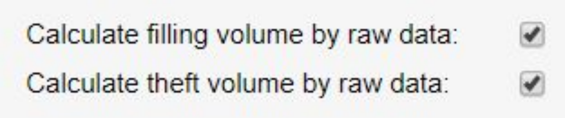
Fillings/Thefts

Sometimes filtration affects the data so much, that you have to combine processed and raw data to analyze filling/theft correctly.



Fillings/Thefts

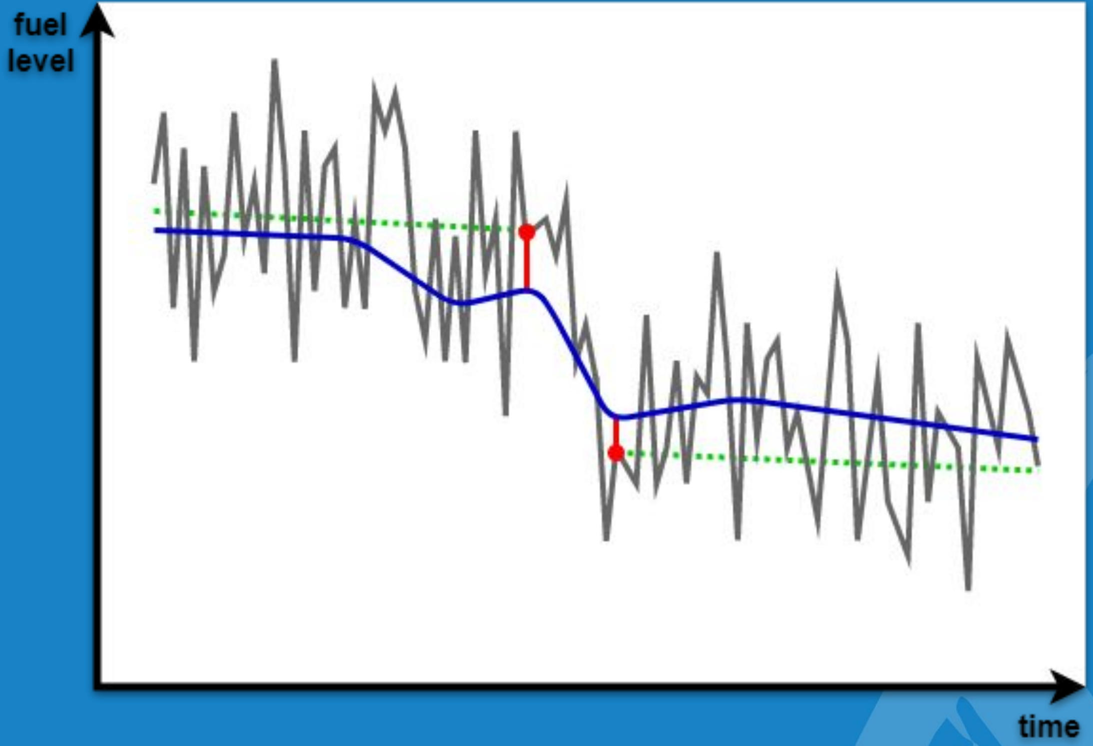
Sometimes filtration affects the data so much, that you have to combine processed and raw data to analyze filling/theft correctly.



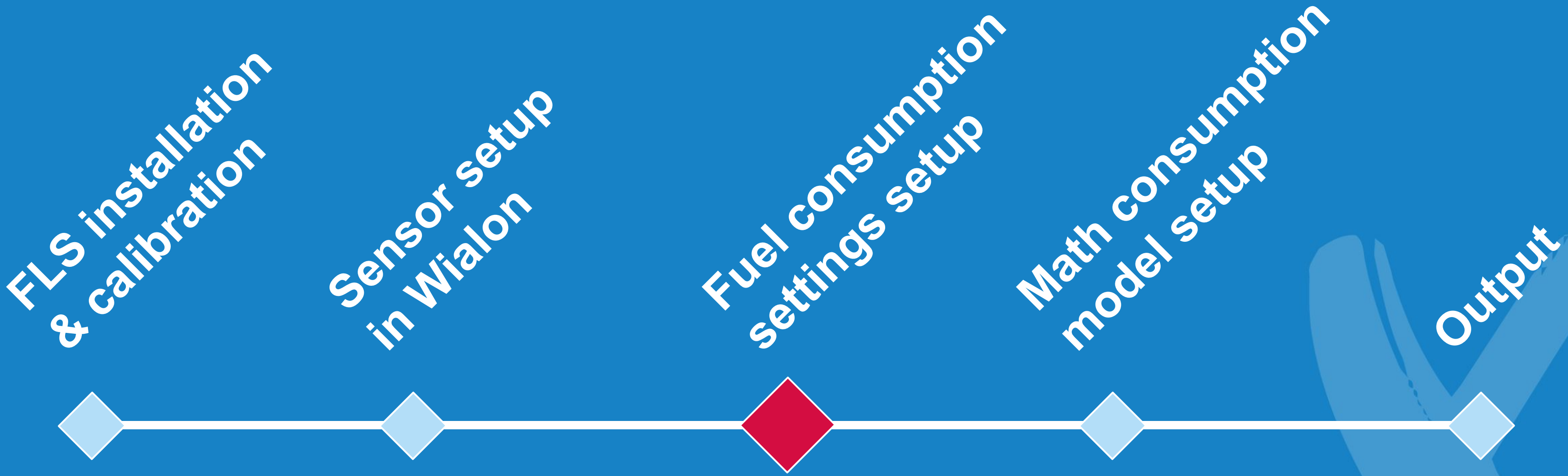
Calculate filling volume by raw data: ☑

Calculate theft volume by raw data: ☑

The fact of filling/theft will be detected with filtered data, while volume will be taken from the raw data.



Steps to use FLS



FLS installation & calibration

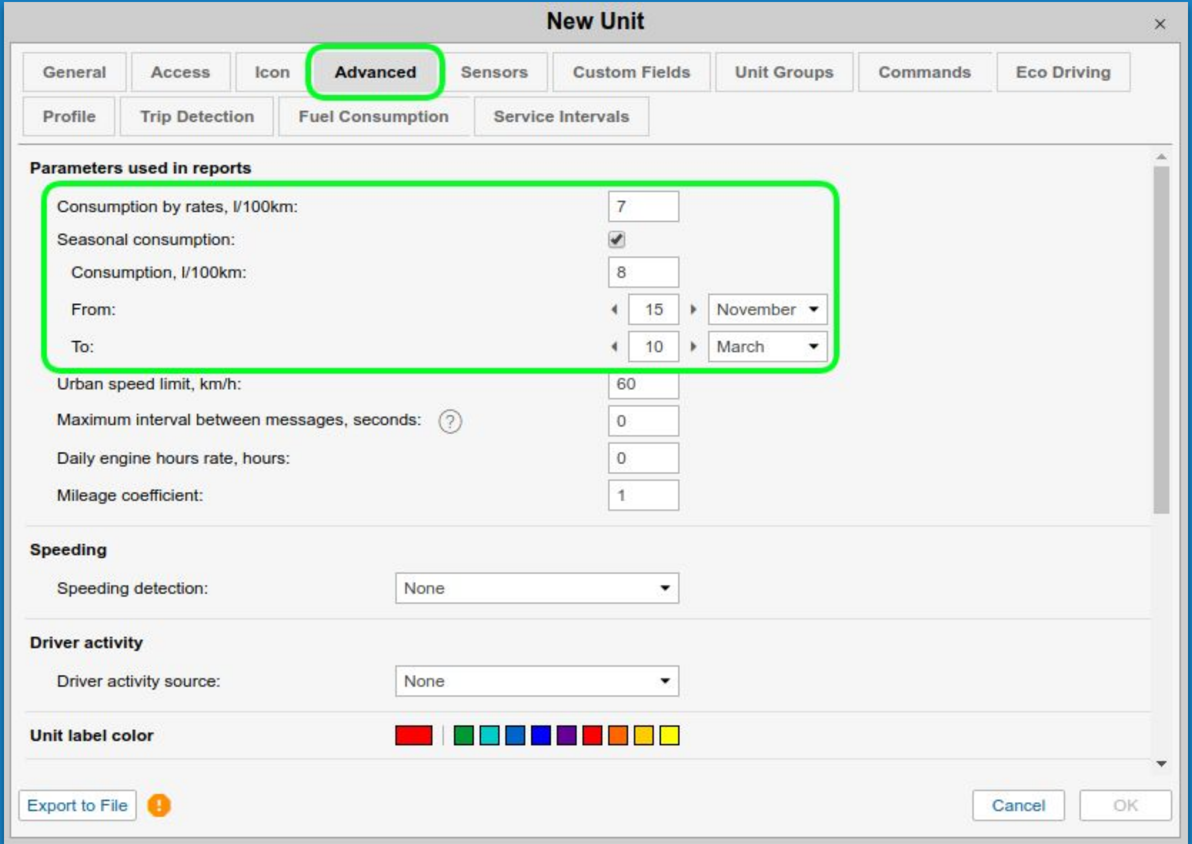
Sensor setup in Wialon

Fuel consumption settings setup

Math consumption model setup

Output

Consumption by rates



Parameters used in reports

Consumption by rates, l/100km: 7

Seasonal consumption: ☑

Consumption, l/100km: 8

From: 15 November

To: 10 March

Urban speed limit, km/h: 60

Maximum interval between messages, seconds: 0

Daily engine hours rate, hours: 0

Mileage coefficient: 1

Speeding

Speeding detection: None

Driver activity

Driver activity source: None

Unit label color

🟥 | 🟩 🟦 🟦 🟦 🟪 🟥 🟧 🟨 🟨

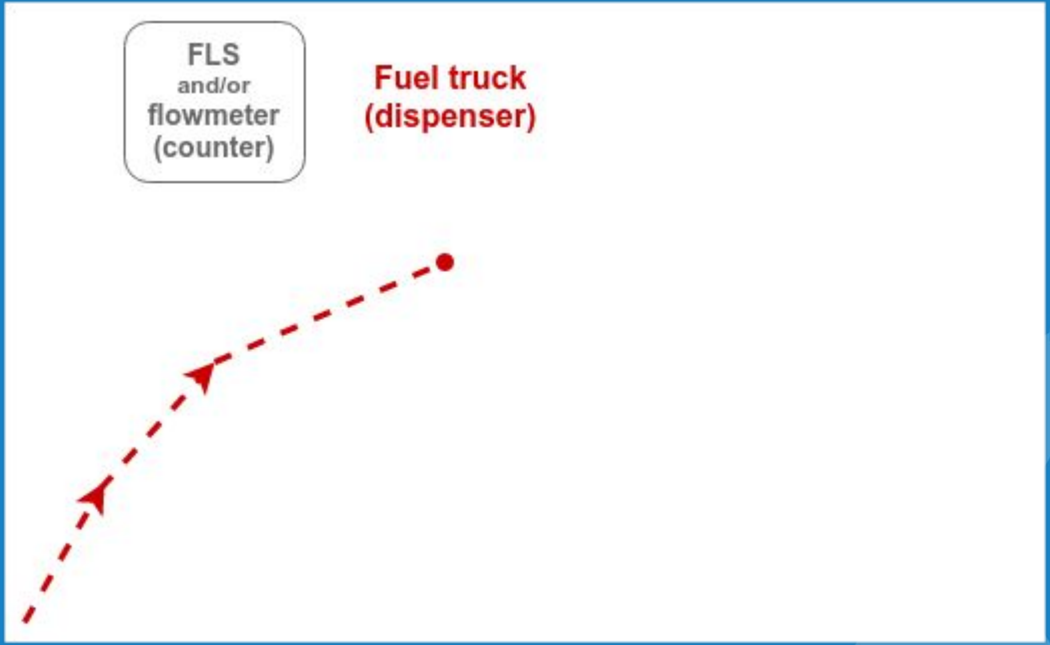
I have mentioned possibility to take account of fuel with no sensors.

And the easiest way to do it is to use rates on the tab “Advanced”. Rate will be multiplied with mileage to calculate consumption.

We have added seasonal rate to it, so you can have 2 different rates depending on season.

Fuel traffic

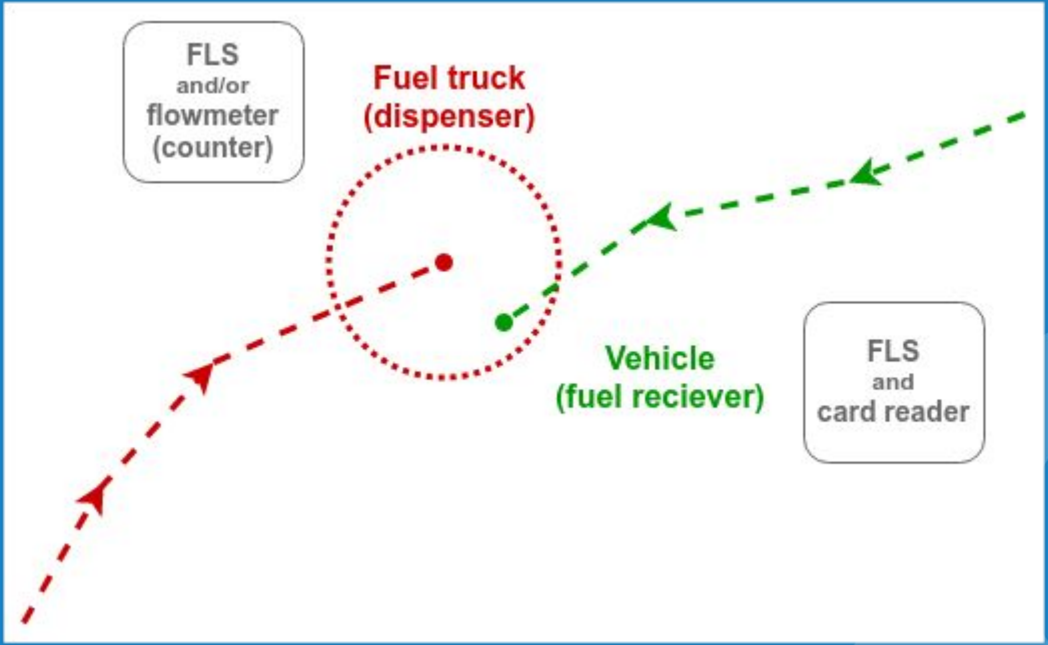
This table analyzes fuel data for fillings, thefts and intervals of counter sensor operation.



Fuel traffic

This table analyzes fuel data for fillings, thefts and intervals of counter sensor operation.

You can add a filter by units for it. In such a case system can look for filling of units nearby. As a result in one table line you'll see the volume of fuel dispensed and received, as well as the difference and driver name.



Zharkovsky Oleg
Deputy Head of Technical Consulting
Wialon Trainers Team Lead, Gurtam

training@gurtam.com

Thank you for your attention!

Zharkovsky Oleg
Deputy Head of Technical Consulting
Wialon Trainers Team Lead, Gurtam

Telematics
Partner conference