README.docx

Jelle Haasnoot	**&**	**Tim de Jong**
Jhaasnoot@tacx.nl		**tss_dejong@outlook.com**
(+31) 6 13 29 49 29		**(+31) 6 36 19 94 96**

SimulANT+ Log Analyzer

Contents

Introduction

This README is created to help the user of the *SimulANT+ Log Analyzer* in his or her use of the program.

The application was created by Jelle Haasnoot & Tim de Jong on July 26th 2018.

First will be explained what the collected Log-files should look like, and how they should be acquired to optimize the functionality of the analyser-app. Then the usage of those files in the app will be explained.

Collecting Log Files

The SimulANT+ Log Analyzer takes four Log-files. These Log-files are created with SimulANT+ 2.2.0. The four Log-files will need to be done as follows:

File 1) **High Power Program**

The first Log-file to be created is the high power program. This will be used to acquire the upper range limit of the trainer.

In SimulANT+ 2.2.0, select the FE-C (Fitness Equipment Control) simulator. In the appearing screen, at the bottom, select the *Turn On* button. If the right trainer isn't selected outright, select *Turn Off*, and then try again. If, after a while, it still didn't work, the user can, in Channel Parameters, give the ANT-ID him- or herself.

Once the right trainer is connected, send a new user configuration using the *Send User Configuration* button. The only values that need changing are the *Bike Weight*, this needs to be changed from 200 to 150, and the wheel diameter, this needs to be changed from 70 to 67.9. This means that the actual bike weight will be 7.5 kg.

To make sure the trainer is operating at its absolute braking limits, power of 2500W. This is done by selecting the *SendTargetPower* button and changing the power value from 0 to 10000, since it needs to be written in terms of 0.25W.

Now, for the cycling part, be sure to <u>accelerate smoothly</u>. It can be tough, especially in the low speed ranges, but it is quite imperative for a smooth and reliable result. The downside to ANT+ is, it only gives 2 messages per second. An acceleration from 0 W to as-hard-as-you-can W should take about 90 seconds in a proper file. If the user has to change gears, try to do so smoothly.

File 2) **Low Power Program**

The second Log-file to be created is the low power program. This will be used to acquire the lower range limit of the trainer.

The work order is almost exactly the same as with the *High Power Program*, with one main difference; the power value should be set to 0. Slow acceleration is, again, preferable. As is not changing gears, but if it is really necessary, do so smoothly.

File 3) **Constant Velocities Program - Trainer**

The third Log-file to be created is the constant velocities program. This will be used to acquire the internal frictions and resistance the trainer has.

Once the trainer is again connected, and the user data configuration is sent, a new type of training will be activated. Select the *SendTargetPower* button aigain to send a power goal to the trainer. The value should be 0 W.

This file is called the constant velocities program because the rider has to ride as constantly as possible on one velociy; <u>12 m/s</u>, or <u>43.2 km/h</u>. The readout of the velocity of the trainer is visible in the SimulANT+ 2.2.0 screen above the buttons. The readout is automatically expanded when selecting the FE-C simulator. When collecting data for this file, <u>DO NOT change gears</u>. Remember what sprocket you are using and how many teeth it has, both at the crank and rear wheel, and remember this.

File 4) **Constant Velocities Program - Sensor**

To verify the trainer, and whether it collects data correctly, an external power / cadence sensor is used. Make sure the sensor records data at the cranks, and not the rear wheel, otherwise the application does not work correctly in estimating the accuracy of the trainer.

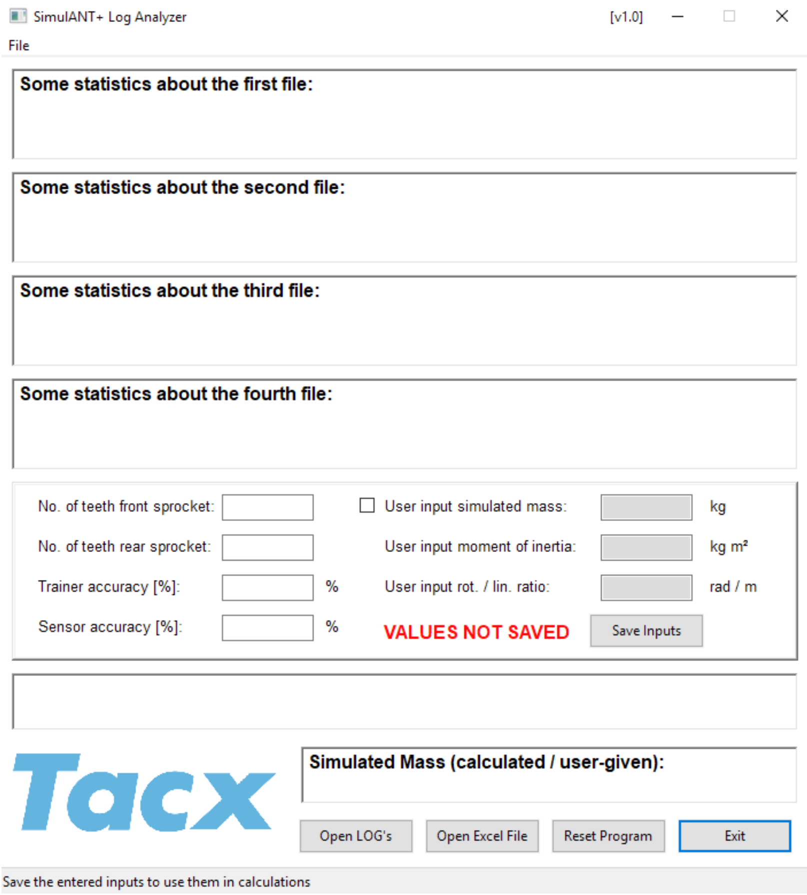
This file can be collected simultaneously with File 3, the only thing that is needed is another ANT+ dongle. For the first dongle, the user needs to select the FE-C simulator (as in previous files), but for the new dongle, the Bike Power Display simulator is needed.

If the user is confident the power sensor is correctly calibrated (if not, use third-party software to do so), the test can be started. Make sure that <u>both files are logging data</u>. Don't worry about the time difference in starting the logging in both simulators, that is compensated for in the application.

The files created above should be imported in this order into the *SimulANT+ Log Analyzer.*

Using the application

When the *SimulANT+ Log Analyzer* is opened, the following screen appears (after the welcome-message):



The fields

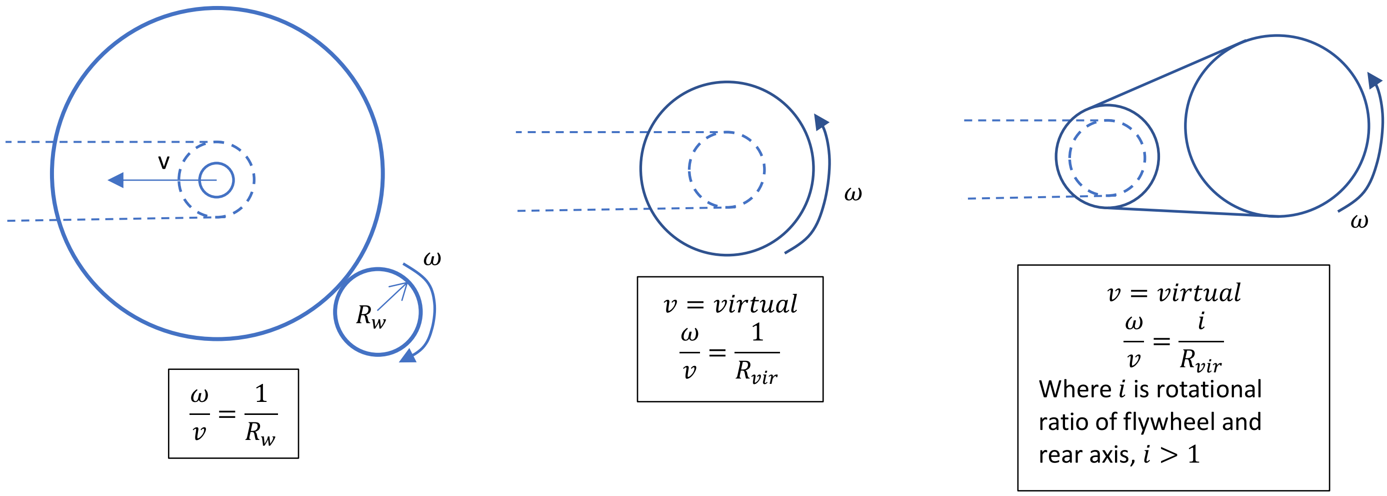
There is a total of 7 fields on the screen which can be filled out. This is how they work, and what they are used for:

- No. of teeth front sprocket:
  Because a cadence sensor is used on the cranks, the velocity needs to be calculated from the ratio between the used sprocket at the cranks and at the rear wheel. This is why it was necessary to not change gears while collecting the data.
- No. of teeth rear sprocket:
  See previous point, now for the rear wheel sprocket.
- Trainer accuracy [%]:
  The accuracy of the trainer is used to, instead of a single value, give the range the trainer's measurements should be in, according to the factory accuracy. In other words, when a trainer gives a certain value, this does not mean that, in reality, that value is correct. But the value is at least guaranteed to be in the area given by the accuracy. This is represented visually in the Excel file, which will be discussed more in-depth in the next chapter.

- Sensor accuracy [%]:
  See previous point for conceptual explanation. What goes for trainers and their respective sensors, also goes for sensors. Though these accuracies are significantly higher than those from trainers.

- User input simulated mass [kg]:
  When it appears that for some reason the application does not give an accurate estimation of the simulated mass of the trainer (which will be the case in multiple instances, due to internal control or friction errors), the user can give one. This value will then be used to give the operating range of the trainer based on that fixed value. The downside of this method is that it is now the manufacturer's responsibility to correctly relay the simulated mass.
- User input moment of inertia [kg m^2]:
  When the simulated mass is unavailable to the user, but the moment of inertia of the flywheel is, the 'User input simulated mass' field can be left empty, and both this and the following field can be filled out. The moment of inertia of just the rotating flywheel should be filled out, to simplify using this method.
- User input rot.(ational velocity) / lin.(ear velocity) [rad / m]:
  To obtain the simulated mass from the moment of inertia, one more parameter is needed. The ratio between the rotational (or angular) velocity of the flywheel and the linear velocity of the bicycle is needed.

  In other words, when a cyclist moves forward, his or her wheels move in a certain relation relative to the ground. There are three possible situations with most trainers. These, and the respective formulas to the required ratio, are given in the sketches below:



Be advised that where the rear wheel is not used in the trainer, the value given to SimulANT+ is very important. This is given in the *SendUserConfiguration* option, where the wheel diameter can be set. It is important to double-check that this was set to 0.679 meters. R_{vir} (virtual radius) would in this case be equal to 0.3995 m.

The buttons

There are five clickable buttons on the screen, these are their respective functions:

- Save Inputs:
  The fields given in the previous section should be filled out before using this button. This button saves the text values internally, so they can be used in future calculations. As the red / green text on-screen indicates, the values are either not saved or saved. When they are saved, they can be overwritten by simply using the *Save Inputs* again.
- Open LOG's: Opens four consecutive dialog screens in which the user is asked to select the log-files he or she wants to open. Navigate to the respective locations of the log-files and select them in the order as given in the last chapter.

The caption in the upper bar of the dialog screens also hints at which file to select now. After selecting, the user will be prompted with a dialog screen which asks the user to give a name for the Excel to be created. Note: <u>All previous existing Excel-files with the same name, in the same directory will be overwritten</u>.

It is also important that any running Excel windows need to be closed. The application only has permissions to write when the Excel is not running.

- Open Excel File: Opens the Excel file created after analysing the log-files which were selected. Excel needs to be installed on the user's PC. If no log-files have been selected yet, this will appear as a warning message.
- Reset Program: though the developers tried their best to solve all the bugs and errors, it is possible that one will still occur. When this happens, the entire program can be reset by using this button.
- Exit: Exit the program. All changes are lost (they are also not save-able to begin with, only the created Excel-file is).

The checkbox

The checkbox which says *User Input Simulated Mass* next to it has the following function: it allows the user to give a simulated mass or an inertia of the trainer they are verifying.

It is understandable that not all the parameters are available to the user. The *SimulANT+ Log Analyzer* also has the capability to make a (very) rough estimation of the simulated mass of the trainer. Rough, because multiple steps of linearization are applied, ANT+ only gives two messages per second and data is easily influenced by deviations like internal (dynamic) friction and calibration issues.

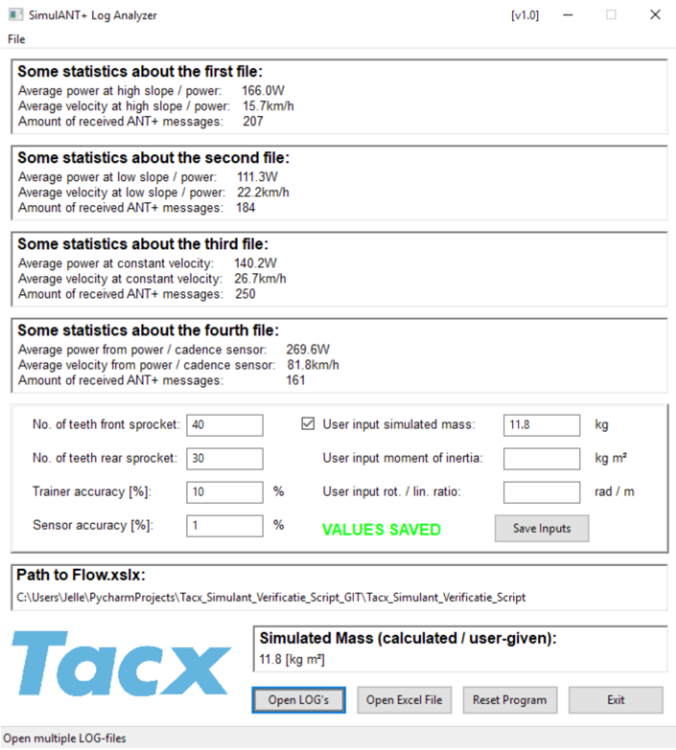
If the user wants the program to make this estimation, the checkbox can be left unchecked. If the calculations produce a simulated mass which is not realistic (negative / very high), the user can either redo the test with new Log-files, or use another simulated mass beforehand.

The File-menu

At the top of the screen, the *File*-menu is visible. In this menu, the user has the option to view the *About*-screen, which displays some basic information about the program. The user also has the ability to open the Log-files from here, but either way works. The same goes for the *Exit* button.

After selecting files

When all files have been selected, the now empty boxes with *Path to directory first selected LOG-file:* and *Some statistics about the first file:* will be filled with what their titles imply. The path will appear for each of the four selected files, to allow the user to double-check if the right file has been selected. See the following screen-shot:



The statistics which are shown for each file are to also help the user in confirming the right file has been selected. The duration, average velocity and average power will be displayed. The user can then confirm that the training from the respective file was the one that was ridden.

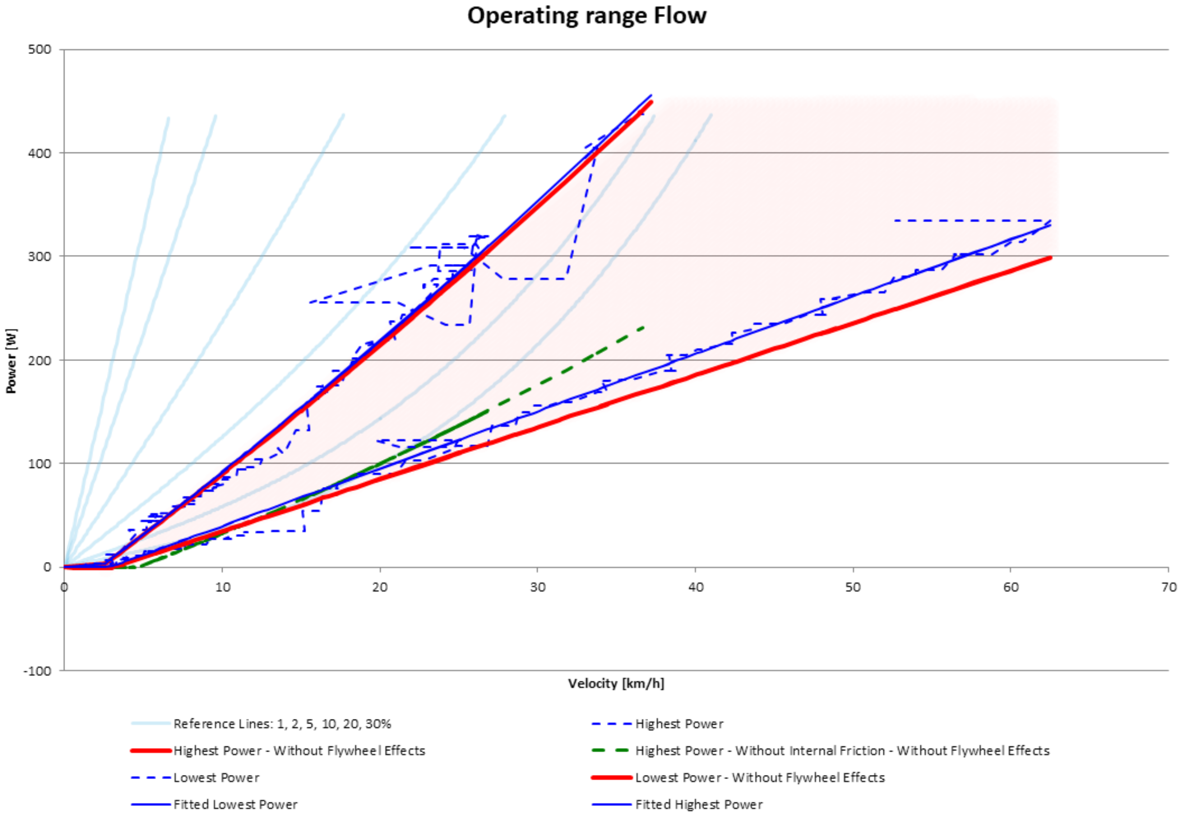
The box above the Tacx-logo will be filled with the path to the Excel-file which was created. If, for some unknown reason, the *Open Excel* button does not work, the Excel-file can be opened manually by navigating to this location.

The final space to be filled is the one which displays the simulated mass. Here, as implied, the simulated mass, which was either estimated by the program, or given by the user, will be displayed.

The created Excel-file

The elements used in the created Excel-file will be elaborated here. When opening the created Excel-file, there are three graphs visible. The first contains the power data from all tests, and is used to display the operating range of the tested trainer. The power here is a function of velocity. The second graph, below the first, displays and compares the power readings from both the trainer and the external power sensor, as functions of time. The third and final graph shows the velocity readings from both the trainer and the sensor, as a function of time. The importance of not changing gears and remembering the sprockets used comes up here, because the cadence readings are converted to linear velocity with those values.

Graph 1:



In total, seven data lines are plotted. A short summary:

- Highest Power – blue, dashed: The raw power data from the first file.
- Fitted Highest Power – blue, solid: The fit through the raw power data from the first file. The program decides what shape this fit will have: based on the minimal error between the original data either a linear or parabolic fit will be made.
- Highest Power – Without Flywheel Effects – red, solid: The power needed to accelerate only the flywheel is subtracted from the fit of the highest gradient power. This line thus illustrates what power is needed just to overcome the braking power and internal friction of the trainer.
- Highest Power – Without Internal Friction – Without Flywheel Effects: green, dashed: This line shows the power needed to overcome the braking power only; all other effects have been subtracted.

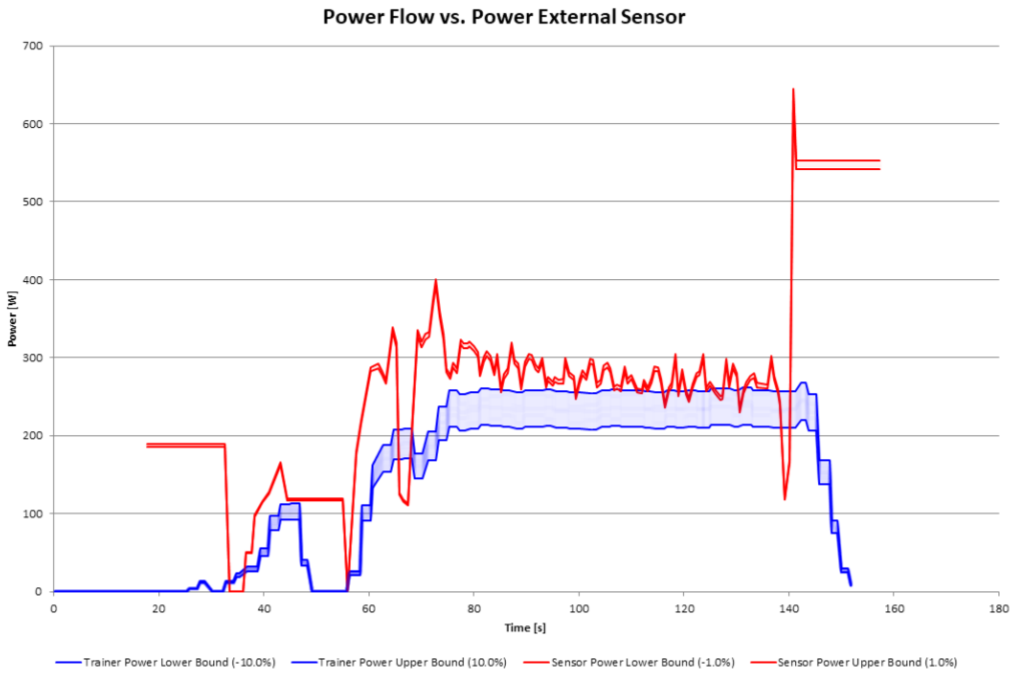
The colouring and explanation of the other three plots is the same as above, except they apply to the low-gradient training. Also, the plot for the exclusive braking power is left out, since this would produce a horizontal line for the low power program.

The semi-transparent lines indicate the power needed when an actual person would be riding at this velocity, at this slope. From right to left, the slopes are 1, 2, 5, 10, 20 and 30%.

The area between the two red lines is the final operating range of the respective trainer. The larger this area (the steeper the high-power line, the flatter the low-power line), the wider is the operating range of this trainer.

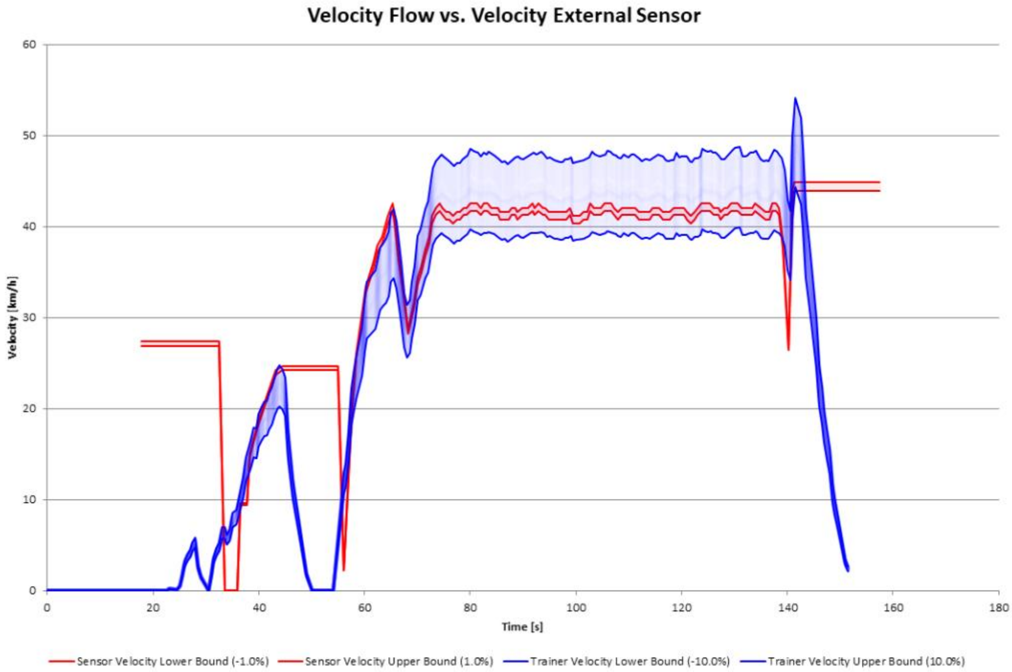
The chart is placed on the first worksheet, *Charts*. The second worksheet, *Data*, contains the raw data from each file. These can be reviewed by the user, if so desired.

Graph 2:



Here, there are four lines plotted. The blue ones represent the upper and lower bound of the region the power measurement is guaranteed (by the manufacturer) to be in of the trainer. The red one represents the power sensor.

Graph 3



Here, there are also four lines plotted. Now the lines represent the velocity measurements from both the trainer (blue) and the sensor (red).

We hope to have given a complete overview of the functionality of the program, but, if not, please contact us or Tacx via the information given on the first page.