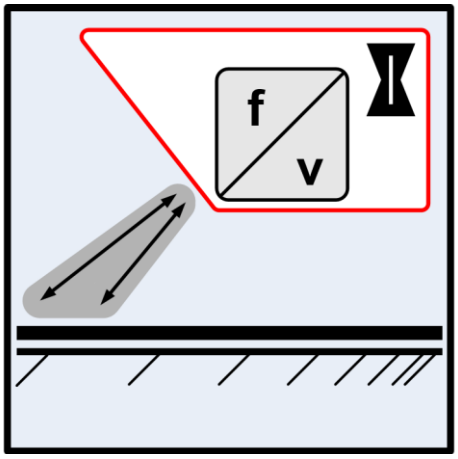
Qualification plan SmaRT for DRS05

Tool for inclinometer method



Author: Rainald Koch / ET
Reviewer: Arndt Niepenberg / ET
Status: Released

Change Index

Ver	Date	Description of Change	Author
1.1	2023-02-18	Translated from German version.	R. Koch

Content

1 This document

The “SmaRT Radar Tool” is for measuring the mounting angle of radar sensors DRS05 ([BA]) and for calculating the corresponding calibration factors ([Schu]), to be multiplied in the on-board unit with the DRS05 output (speed and distance increments). Obviously, the calibration factor, cos(45°)/(cos(45° + “nose down” angle), is “data” in the sense of [EN50657] item 3.1.44 (T3 tool).

This document provides information to cover Sect. 6.7.4 of [EN50657] and plans qualification tests (6.7.4.4).

2 References

[Schu] = Training Material - Measuring the mounting angle_V2.1.pdf

[EN50657] = EN 50657:2017

[BA] = Instruction Handbook DRS05/3 (AB1773)

3 Tool classification and selection

SmaRT consists of two components:

- the “SmaRT App” for Android smartphones and
- the “SmaRT phone holder” as alignment tool.

There are two variants of SmaRT phone holders for the small DRS05 housing and the DRS05/S1, respectively. They allow the alignment of the smartphone parallel and antiparallel to the direction of travel – measurements in both directions eliminate offsets and diminish the impact of local deviations (dirty surface, differential nonlinearity). A video in the app shows the handling of the tool (covers [EN50657] 6.7.4.3).

The SmaRT App uses the acceleration sensor in the smartphone to obtain its orientation relative to the direction of gravity. Beyond this function of COTS inclinometers, the app guides the user. It indicates the direction to the device connector for the intended orientation of the next measurement. It asks the user for the train/cab number and lets them take a photo to document the orientation of the radar relative to the train (for identification of the mounting position in case of two radars per cab) and of the train relative to the track (in case of non-negligible track inclination). That allows in a later verification step to detect sign and assignment errors. The result page presents measurement from several trains or dates with a flexible search function. This paragraph and Sects. 5 and 9 cover [EN50657] 6.7.4.2 (Tool selection).

4 Qualification of the app

[EN 50657] 6.7.4.4 says (emphasis and ellipsis by RKo):

> For each tool in class T3, evidence shall be available that the output of the tool conforms to the specification of the output or failures in the output are detected. Evidence may be based on the same steps necessary for a manual process as a replacement for the tool, […]

Exactly that is what has been done repeatedly during the development of the app and what we intend to repeat with customer personnel in the field.

The app is shown to be correct with respect to several possible errors related to signs (±), scale factors, and offsets. In addition, we demonstrate functions of the app to avoid mistakes in the procedure.

4.1 Sign I

A DRS05 pitched "nose down" (steeper microwave beam) tends to underestimate the speed, because the Doppler effect is based on the radial velocity component of the train speed (in beam direction). This underestimation would have to be compensated by a calibration factor larger than 1. For a nose-up pitch, the calibration factor must be < 1.

If there is a visibly pitched radar available, take a measurement with SmaRT and compare the resulting calibration factor with said expectation. In case of a failed expectation, it is likely that the adjacent hint "to connector" has been ignored.

If the the pitch of the radar is not obvious by eye, place the smartphone on a slope. Form your expectation by imagining the radar with the connector on the indicated side.

Document both the expectation and the result.

4.2 Sign II

The sensor measures the pitch relative to the horizontal plane, but we need the pitch relative to the direction of movement, which differ in case of a track with significant slope. The app hints at this effect and allows to enter the track slope as magnitude and sign; the latter unmistakably via buttons with an uphill or downhill looking radar.

Check that the slope is used with the correct sign. Expectation: The beam of a radar looking uphill is steeper relative to the track than relative to the horizontal, so that the calibration factor increases over the one calculated without track slope.

4.3 ACC_Y → angle → calibration factor

If the smartphone is held in the hand with the long axis tilted by ±45°, an angle with magnitude of about 45° should be displayed. The correction factor must diverge towards 45° "nose down", be $1/\sqrt{2} \approx 0.707$ for 45° "nose up", and correspond to the linear approximation (1 − 0.0175·"nose-up angle") for small angles. Note that we are concerned here with errors of the implementation, not with the accuracy of the sensor (more on this in Sect. 8).

4.4 Waiting for a stable signal

As with commercial inclinometers, the measurement starts automatically when the signal is stable and is discarded and restarted if the signal fluctuates before the end of the measurement period. The primary purpose of this function is ease of use. It also contributes a little to safety: An incorrectly placed SmaRT phone holder might wiggle. The function also prevents the use of a smartphone with a very noisy acc signal, but this did not happen in tests with several different models.

Checking the function: Starting a measurement with the smartphone held in the hand should even be difficult with both forearms resting on the table.

4.5 Offset

Offsets occur in the sensor itself, due to its skewed mounting in the smartphone, and due to the angle between the smartphone and the contact surface of the holder. The total offset is eliminated by a pair of measurements with a 180° rotation in between – the difference contains the inclination but not the offset. The app does not start "Step II" if it does not detect a rotation of at least 90°. This is to prevent measuring without rotation, which would eliminate the inclination instead of the offset and calculate the calibration factor from the latter.

Two things have to be checked:

4.5.1 Ensure the rotation

Attempting to start the second measurement by waiting calmly must lead to an abort, as must turning by >90° and back again.

4.5.2 Implementation of the difference

Increase the offset by fixing the smartphone in the holder with a slight tilt in the longitudinal direction. The calibration factor determined for a radar sensor should not be affected (before-after difference < 0.1 %). If affected, the fixation was not stable or the differential nonlinearity of the sensor is too large.

4.6 Measure against offset drift

The app limits the time between the two measurements (end to end) to 30 s. On timeout, an error message appears and the first measurement is discarded. The event is not logged.

Check: Delay the end of the second measurement and start a new measurement after the error message. It must start with "Step I".

5 Qualification of the SmaRT phone holders

Undeniable advantages of using the SmaRT Radar Tool over commercially available inclinometers are that personnel do not have to handle three things (inclinometer, template for alignment, and writing utensils) and that the SmaRT phone holder has contact points rather than surfaces, so there is little likelihood of hitting local surface irregularities.

A potential drawback is the less rigid connection of the sensor to the contact points/surfaces. Identified weak points:

- Clamping of the smartphone. The angle between it and the SmaRT phone holder must not change due to expected jerk between the two measurements by more than 0.05°. The uncertainty here is in "expected jerk".
- Deformation of the SmaRT phone holder. "Expected manual forces" (contact pressure and rotational torque, especially in the pitch direction) must also not change this angle by more than 0.05°.

The participants in the qualification test should therefore estimate the limits of what jerk and moment, respectively, can reasonably be expected, whereby the difficulty lies in “deliberately refraining from intent”.

6 Competition of tools

Depending on further measures in the OBU, gross errors in determining the correction factor must be so rare that they should not actually occur in a field test. However, experience with the previous method (using a COTS inclinometer) indicates the opposite. So it can be instructive to let both procedures compete against each other, up to a simulated OBU input.

Since the error rate decreases with practice, several untrained test persons should participate if possible, while the number of measurements per person need not be large, perhaps four measurements per procedure (two big and two small DRS05).

Half of the subjects use the COTS inclinometer first, the other half start with the SmaRT radar sensor tool. Then they switch.

7 Programming the SmaRT App

The development environment (Android Studio) and the compiler for the used programming language (Kotlin) are very suitable for the development of an Android app and have been tested a million times. All functions that can directly influence the value of the correction factor are either integrated into the API/framework (input of values via text fields, accessing the accelerometer signal, allocation of memory, formatting of the result as a string) or very simple (averaging, rule of three, cosine function). Together with the tests according to Sects. 4 and 8, this covers [EN50657] 6.7.4.7/8.

8 Qualifying individual smartphones

The following tests have been performed with several smartphones during development, with always positive results. However, since the smartphone market is large and open, problematic models cannot be excluded with sufficient certainty. The first three tests are quick and should be carried out with every new device. Linking to this document in the SmaRT App covers [EN50657] 6.7.4.3.

The phyphox app is better suited for the qualification of smartphones than the SmaRT App, as it displays time histories graphically and exports raw data.

8.1 Orientation of the coordinate system

The Android API defines that the Y axis is parallel to the longer edge of the device and oriented so that in normal posture, the acceleration due to gravity is displayed with positive values.

8.2 Gain of the Acc sensor

If aligned vertically, Acc_Y should be offset ± 9.8 m/s² (+ upright, − upside down), so the difference is 2 · 9.8 m/s² = 19.6 m/s². This has been well fulfilled in all tested smartphones so far, apparently subject to factory or dynamic calibration.

Scaling errors in the order of 2% are not critical for moderate pitch of radar mountings, since it is a second-order error. E.g., a slope of 4° results in a correction of 7 %, which is wrong by 0.28 % if the scaling is wrong by 4 % (0.07 · 0.04 = 0.0028).

8.3 Cross sensitivity

In case of significant roll angle, the cross sensitivity is relevant. Estimation as in 8.2 above: A roll angle of 4° at a cross sensitivity of 4 % causes an error of the correction factor of 0.28 %.

The cross sensitivity has two causes, the micromechanics of the MEMS sensor and its alignment to the DRS05 – hence the SmaRT phone holder as alignment tool.

The cross sensitivity can be measured as the slope of the function ACC_Y versus ACC_X at a pure lateral tilt, i.e., rotation about the DRS05 longitudinal axis defined sufficiently precisely by the lateral edges of its flange. So, put the smartphone in the holder, the holder correctly aligned on a DRS05, the DRS05 with its flange on a table and then tilt everything around one flange edge, slowly, until ACC_X = 5 m/s² (30°). An observed change in ACC_Y of e.g. 0.1 m/s² would mean a cross sensitivity of 0.1 / 5 = 2 %.

8.4 Offset drift

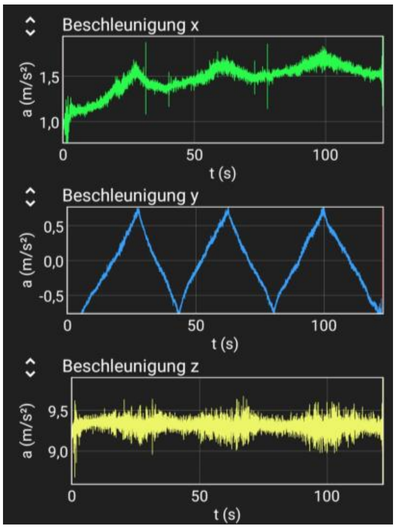
Offset drift is harmful if it occurs quickly, between the two measurements (before / after rotation). This is conceivable in case of a rapid temperature change, tested by connecting to a quick-charger during recording with phyphox. The result was harmless.

Jumps in the offset are conceivable if the holder's jaws are greasy (low static friction) and the holder is pushed roughly to the radar for the second measurement.

8.5 Differential nonlinearity

In view of the sensor principle, only the A/D conversion is conceivable as a source of differential nonlinearity (DNL). This would manifest itself as non-occurrence or less frequent occurrence of some codes. In extreme cases, several adjacent codes are affected. Noise reduces the influence of DNL on the averaged result if the noise amplitude is larger than the gaps. The app employs a high data rate to ensure sufficient noise.

Check: Record an ACC_Y sweep over the interesting range of pitch, e.g., 9.8 m/s² · tan(±4°) ≈ ±0.7 m/s², export and display as a histogram. Out of hand, a slow sweep is difficult to perform smoothly. Fluctuations in sweep rate average out with multiple back-and-forth sweeps. The adjacent phyphox screenshot illustrates the procedure.



The following figure shows the histogram for this measurement, result: no code gaps.

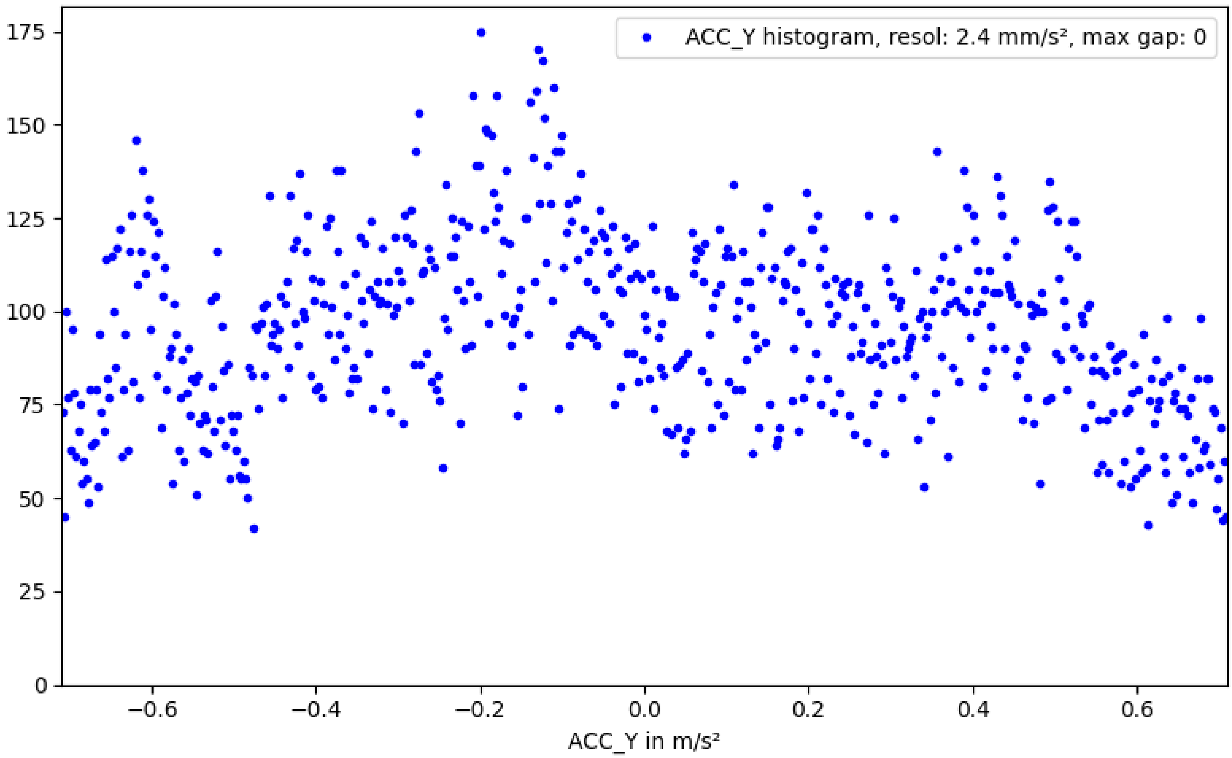


Fig. 2 from the Python script: histogram over ACC_Y – no code gaps.

9 App updates

The source code of the app is subject to version control (git). For foreseeable changes and extensions to the SmaRT App, the necessary qualification measures are also foreseeable. The following subsections serve as examples for analyses complying with [EN50657] 6.7.4.11.

There is no obligation to perform an update. However, an update may be necessary if a newer smartphone is used or an update of the operating system requires this. It is up to the customer to decide which version to use in which project (exporting [EN50657] Sect. 6.7.4.10, Configuration Management).

Several versions can be installed in parallel. The version number can be viewed on the "About Us" page (behind the ⓘ button). Measurement results are stored separately for each version. Only the latest version is available in Google Play. Older versions can be obtained as apk files from Deuta.

9.1 Multilingual user interface

The user interface is currently in English. Other EU languages and Chinese will be required in the foreseeable future.

The extension does not affect the validity of the qualification tests planned above. Of course, each language version of the GUI must be verified by native speakers.

9.2 Layout details

For particularly long smartphones, the used display area should be limited.

Such changes do not affect the validity of the qualification tests planned above.

9.3 Format of the K factor

On a new page "Settings" one should be able to set the number of decimal places (3 to 5, factory setting: 4) as well as the decimal separator (comma, point, language-specific or not at all, factory setting: language-specific).

Does not affect the validity of the qualification tests planned above. Test all options.

10 Safe input of the K-factor into the on-board unit

[EN50657] 6.7.4.1, last paragraph:

> The selected tools should be able to work together. In this context, this means tools work together when the output products of one tool have suitable content and format for automatic input to a subsequent tool, thereby reducing the possibility of human error in the post-processing of intermediate results.

Here is a proposed method that keeps humans to perform the input, but uses a safe return channel:

1. OBU: Select the "K-factor input" function and in it, if two radar sensors are connected, the channel to which the factor should be applied. Ideally, the OBU will present the same icons as the app to select the channel: the radar sensor's viewing direction is either parallel or antiparallel to the cab.
2. SmaRT App: Select the desired measurement on the results page, taking into account the train number/cab, the channel, date and time. The app displays the K-factor, the photo showing the orientation of the radar sensor and a “Validate QR” button.
3. OBU: Manual input of the K-factor. The OBU then presents the train number/cab it knows for sure, the previously set channel, the K-factor and a transaction number (TAN), all in the form of a QR code on the display.
4. SmaRT App: Pressing [Validate QR] switches to the camera to capture the QR code. The data in it is compared with that of the selected measurement. If there is a match, the current time is entered in a column "validated OBU upload" of the result list and a TAN is displayed in a large font size, otherwise an error message is displayed.
5. OBU: The TAN is entered manually to inform the OBU about the success.

The procedure only requires a display as hardware, so it should usually be feasible. The app should perhaps enforce a configurable input format for train number/cab so that the match with the content of the QR code does not fail due to format variations.

Appendix: Python script to Sect. 8.5

```
""" Untersuchung von ACC-Sensordaten auf differenzielle
Nichtlinearität im Bereich 0 ± RNG m/s².
"""
import numpy as np
import matplotlib.pyplot as plt

FNAM = '2023-01-30 144435_DNL_A50.csv'
RNG = 0.71 # Für die Neigungsmessungen ausreichender Bereich.

def get_cols(fnam): # Lies PhyPhox-Daten.
    data = np.genfromtxt(
        fnam,
        delimiter = ',', # Wie exportiert.
        names     = True,
        usecols   = (1,2,3),
        dtype     = 'f4,f4,f4')
    if not data.dtype.names == ('Acceleration_x_ms2',
                                'Acceleration_y_ms2',
                                'Acceleration_z_ms2'):
        raise NotImplementedError
    return data.view('f4').reshape((-1,3)).T

acc = get_cols(FNAM)
median = np.median(acc, axis=1); median[1] = 0
fig, ax = plt.subplots(nrows=3, ncols=1, sharex=True, figsize=(16,12),
                       gridspec_kw={'hspace':0.})
for i in range(3):
    ax[i].plot(acc[i], 'o', markersize=3, alpha=.5,
               label=None if i else FNAM)
    ax[i].set_ylim(median[i] - RNG, median[i] + RNG)
    ax[i].set_xlim(-.1)
ax[0].legend()
fig.tight_layout()
fig.show()

diff = np.diff(sorted(set(acc[1][np.abs(acc[1]) < RNG])))
resol = np.median(diff)
max_gap = int(diff.max()/resol) - 1
n, bins = np.histogram(acc[1], bins=len(diff)+1, range=(-RNG, RNG))
mid = .5*(bins[:-1] + bins[1:])

fig, ax = plt.subplots(figsize=(8,5))
ax.plot(mid, n, 'b.', label=f'ACC_Y histogram for {FNAM}, '
    f'resol: {resol*1E3:.1f} mm/s², max gap: {max_gap}')
ax.set_xlabel('ACC_Y in m/s²')
ax.set_xlim(-RNG, RNG)
ax.set_ylim(0)
ax.legend()
fig.tight_layout()
fig.show()
```