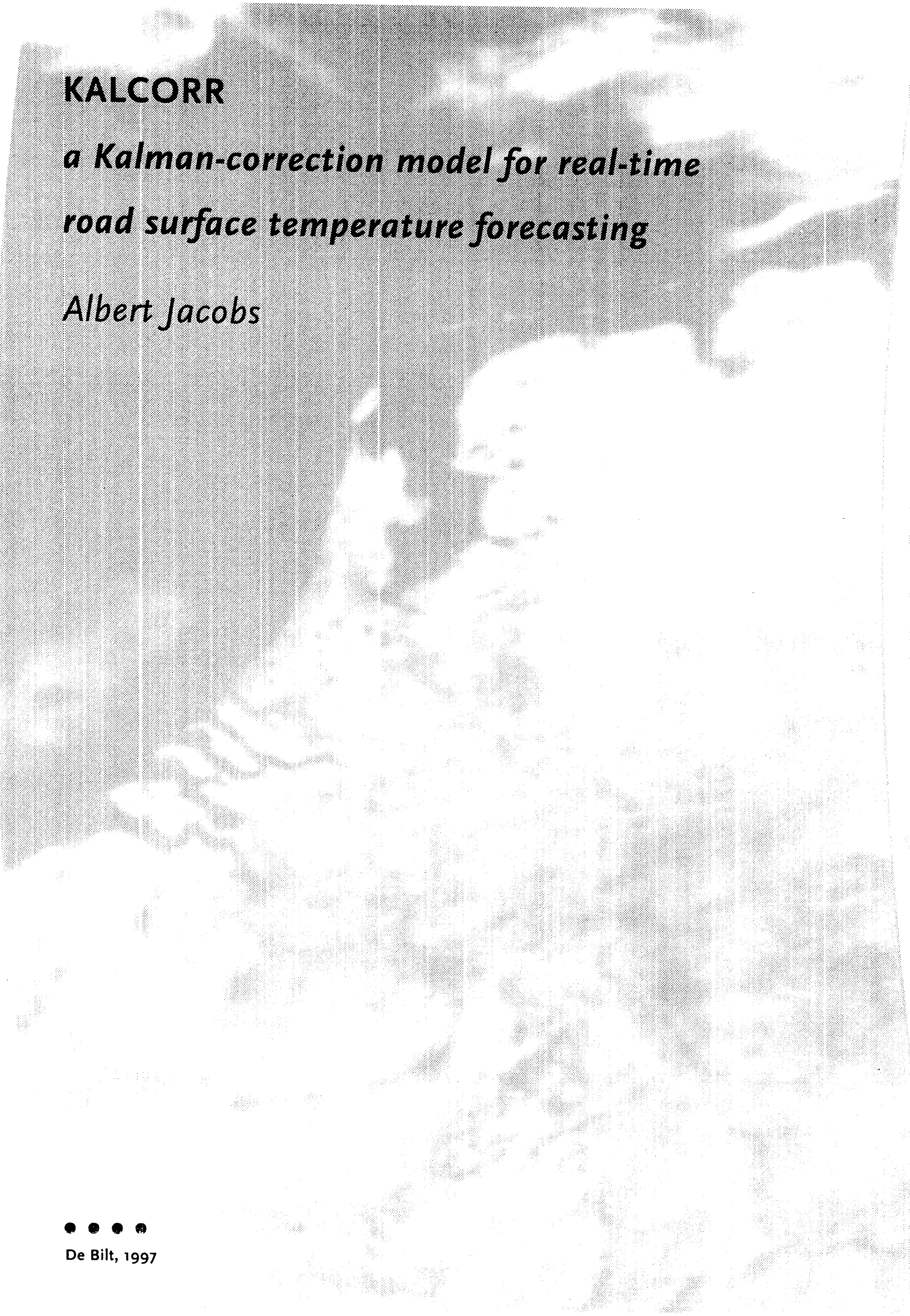
Technical report; TR-198

KNMI



KALCORR

a Kalman-correction model for real-time road surface temperature forecasting

Albert Jacobs

Koninklijk Nederlands Meteorologisch Instituut

De Bilt, 1997

Technical report = technisch rapport; TR-198

De Bilt, 1996

PO Box 201
3730 AE De Bilt
The Netherlands
Telephone +31.30.220 69 11
Telefax +31.30.221 04 07

Authors: Albert Jacobs

UDC: 551.501.45
551.509.323
625.8
ISSN: 0169-1708
ISBN: 90-369-2119-8

● ● ● ●

KALCORR: A Kalman-correction model for real-time road surface temperature forecasting.

by

Albert Jacobs,
Applications and Modelling Division KNMI,
January 1997

Contents

1 Introduction

KALCORR is a statistical post-processing model which we have especially designed to forecast road surface temperatures up to 24 hours ahead. The model is fully automatic and self-optimising towards an accurate temperature prediction. The latter is accomplished by introducing Kalman-filter techniques.
KALCORR is an important tool in the control of icy and slippery roads, and as such, part of the "Weer en Verkeer" (WenV) system of KNMI. The model generates site-specific road surface temperature forecasts each hour of the day. Together with other site-specific information such as the dewpoint temperature at 2m, the cloud amount and type, the direction and the speed of the wind, and the amount and probability of precipitation, a forecast of the state of the road surface can be made. Here the main conditions are dry, wet or slippery. In this context, **KALCORR** is part of a decision support system, and improves the ability to control the condition of the road surface.

The model is purely statistic, and is based on 2m-temperature forecasts from the high resolution limited area model HIRLAM, and recent observations from a specific site, at the location of a sensor.[1]
HIRLAM is a Numerical Weather Prediction (NWP) model, and heart of the Automatic Production Line (APL). Hirlam produces meteo forecasts 48 hours ahead, every 6 hours at $00^h, 06^h, 12^h$, and 18^h GMT.[2] The reader should be aware, that all the times in this report are in GMT. For the purpose of road surface temperature forecasting the direct model output of the 2m-temperature from a nearby Hirlam location is used as a first guess for the expected road surface temperature. In this sense the 2m-temperatures, that are extracted from the Hirlam timeseries, act as a background model in our statistical approach.
Persson [1990] already pointed out that NWP models are often systematically biased w.r.t. 2m-temperature. The magnitude of the bias depends linearly on the forecasts itself, and varies with the geographical circumstances of the site, the time of the season, the time-lag w.r.t. the last available observations, and the forecasting time of the day. Based on these results, and generalising towards road surface temperature forecasting, we use a statistical correction formula of the form

$$\begin{aligned} T_{\text{corr}}(t;t_f) &= X_0(t;t_f) + X_1(t;t_f) * T_{\text{2m}}(t) + X_2(t;t_f) * T_{\text{obs}}(t_a) + \\ &\quad X_3(t;t_f) * T_{\text{obshh}}(t_a). \end{aligned} \tag{1.1}$$

---

[1] To be more precise, rather than forecasting the temperature of the road surface itself, we produce temperature forecasts of the roadsite at the location of the sensor, where the observations are done. This sensor is located a few mm. below the road surface.

[2] With the available computer power these forecasts are available at successively 04^h, 10^h, 16^h, and 22^h GMT.

In this statistical model T_{corr} is the corrected road surface temperature, $T_{2\text{m}}$ is the 2m-temperature from Hirlam, T_{obs} is the last available observation at the specific site, T_{obshh} is the last available observation at hh-hour, t is the forecast time of the day, t_a is the actual calling time of **KALCORR**, and $t_f = t - t_a$ is the time-lag w.r.t. the last available observation, also called the length of the forecast interval. In our application we fix hh to $hh = 12^h$.

The regression coefficients X_0, X_1, X_2 and X_3, in model (1.1) are time-varying and depend on t_f. For constant X_0, X_1, X_2 and X_3, (1.1) is a classical linear regression model. It's purpose is to correct $T_{2\text{m}}$ at each time t, in order to produce a better road surface temperature T_{corr}. In classical linear regression the history of observations for each specific site determines optimal values for the regression coefficients, in order to eliminate the bias, and to mimimise the standard deviation of the forecasts over one or more seasons. These optimal values are subsequently used to produce corrected forecasts in following seasons. This method is rather static and does not take into account changes in the background model, seasonal variations, and variations in the geographical circumstances of that specific site. Optimal values for the regression coefficients for one period may not be optimal a couple of weeks later. Therefore a better policy would be to refresh these coefficients at least once a year, or even better every day, in order to meet the different kinds of model and/or environmental changes.

In the operational statistical model a more or less dynamic approach is introduced. The time-dependent regression coefficients are updated on-line, each time when new 2m-temperature forecasts and road surface temperature observations are available. The adaption of the model coefficients is based on a Kalman-filter technique. The automatic adaption method is recursive. To start with we have to make an initial choice for the values of the regression coefficients. Because of this uncertainty, it takes at least one month before the corrected forecasts are of proper quality.
The Kalman-filter is part of the operational forecasting scheme, and consists of a Kalman-assimilation module where the regression-coefficients are updated based on recent road surface temperature observations, and a Kalman-correction module where the Hirlam 2m-temperature is corrected by means of (1.1), and the most recent values of the regression coefficients X_0, X_1, X_2, and X_3.
A similar method based on the VAISALA-model, as a physical background model, was described in TR-180, cf. Jacobs [1995]. For the interested reader, a detailed outline of the Kalman-filter can also be found in that report. New in our present approach is the inclusion of T_{obshh}, with $hh = 12^h$, in order to cope with the poor forecast quality at afternoon hours (see also Appendix D). A thorough description of the operational road surface temperature forecast module **KALCORR**, as part of the APL, and the embedding of this module in the WenV-system, will be the subject of this report.

2 The operational forecast scheme of KALCORR in the APL

In the APL, road surface temperature forecasts 24 hours ahead are produced each hour, for various districts in The Netherlands (cf. appendix E), and within each district for a number of sensors, mostly 3 to 6. Corrected forecasts are produced, based on the last available 2m-temperatures at the nearest and most suitable gridpoint. The Hirlam 2m-temperatures are refreshed 4 times a day, and available at successively $06^h, 10^h, 16^h$, and 22^h GMT, approximately 4 hours after a Hirlam forecast run has been started. The last available roadsurface temperature observations are supplied each hour by the clients of the WenV-system. These clients are the Departments of the Regional and Provincial Waterworks (Waterstaten). Periodically, 4 times a day at $00^h, 06^h, 12^h$, and 18^h GMT, the observations are incorporated in the regression model (1.1), and the regression coefficients are updated. Note that in this context, site specific information like the road-type, and the geographical circumstances of the environment, are somehow included in the hourly measurements of the sensor. Therefore we can say that the physical properties of the site are, in some sense, modelled implicitly by our statistical model.
Hirlam produces 4 different sets of timeseries, originated from resp. the $00^h, 06^h, 12^h$ and the 18^h runs. It is a well-known fact that each set has its own statistical characteristics in terms of bias and standard deviation in the error of the 2m-temperature forecasts. Consequently, a forecast classification in our systematic error analysis is necessary, and results daily in 4 different Kalman-filters, one for each set of Hirlam runs. Each of these filters computes the regression coefficients for Hirlam forecasts 36 hours ahead,[3] and calling times t_a varying from $t_a = t_v$ until $t_a = t_v + 12$, with t_v the time of the last available Hirlam forecast. Hirlam forecasts older than 12 hours are not accepted for forecast correction. Each hour **KALCORR** produces an effective 24 hour road surface temperature forecast which is extracted from the 36 hour ahead timeseries, and this one is handed out to the clients.
Figure (2.2) gives a clear view of the daily operational runscheme of **KALCORR**, and the interconnection with other modules in the WenV-system. In the next section, the various modules in the WenV-system, and the connection of these modules with **KALCORR**, are described in some more detail. A description of the database files can be found in Appendix C, and the filter-files are described in some more detail in Appendix D.

The WenV-system, and as such also **KALCORR**, is initialised by calling the script **init_wenv.psc**. This script builds the subdirectory tree for each client and sensor, for each Hirlam location, computes the initial restart filter file filter.ini, and builds the database

---

[3] In this pratical application a forecast 36 hours ahead is sufficient in order to produce an effective forecast of 24 hours ahead.

files for the actual month. For an overview of the WenV directory tree in the APL, we refer to Appendix A. The initialisation scheme is given in figure (2.1). Besides the daily operational runscheme, we have two subschemes, cf. also figure (2.3). The first one removes old files in the WenV-system, builds the new database files on the first day of each month, keeps track of the status of each module, and runs the verification program 4 times a month. The second subscheme is specially meant for WenV clients, and produces forecast files in a by the client specified format, the so called vfiles.

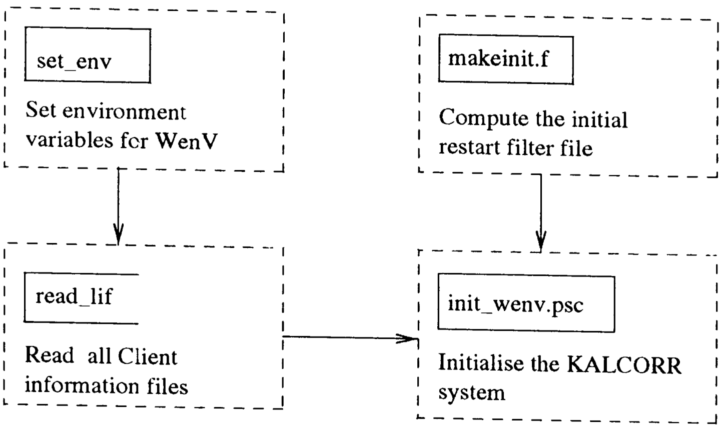


Figure 2.1: The initialisation scheme of the WenV-system and **KALCORR** in the APL.

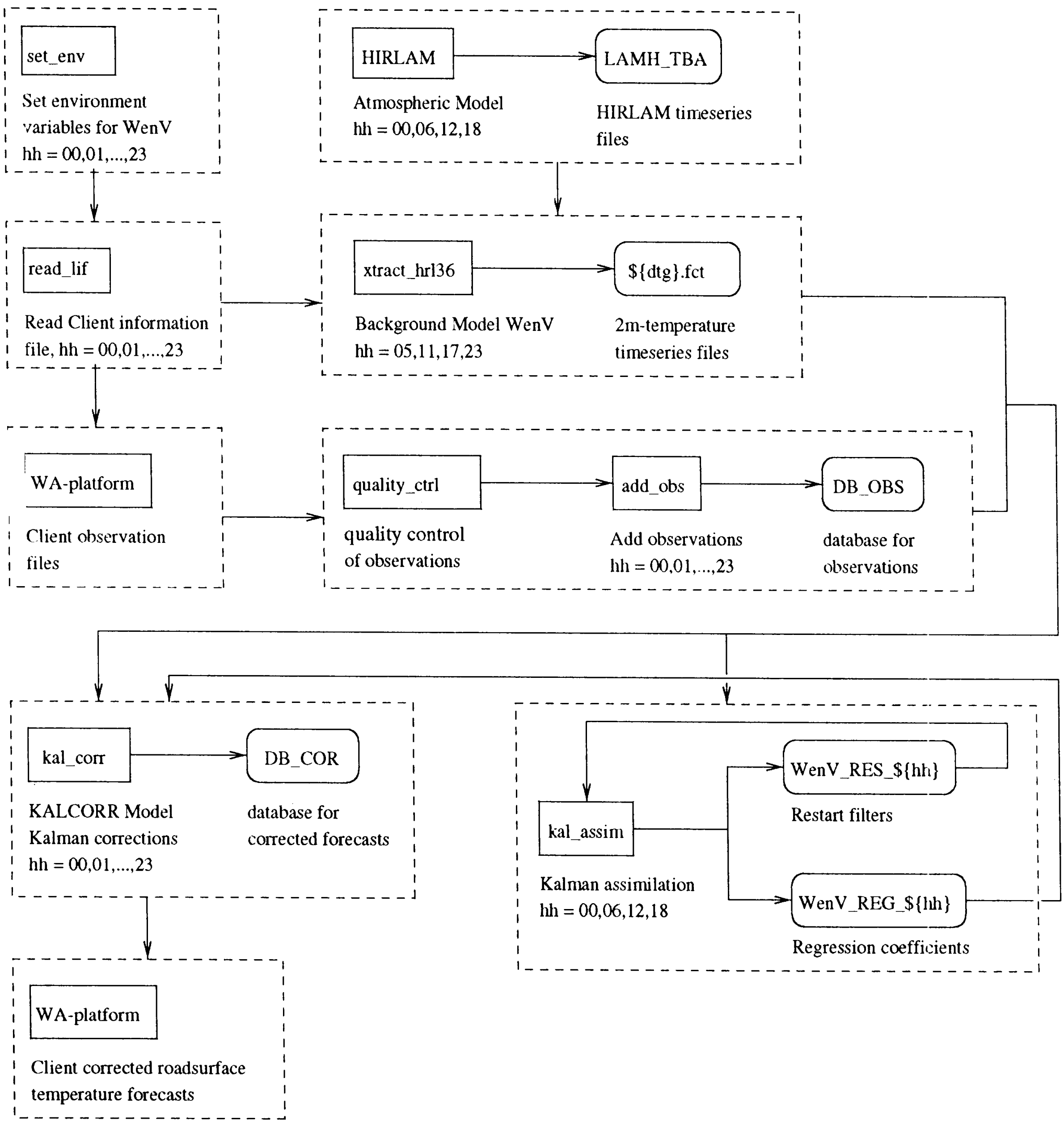


Figure 2.2: The daily operational runscheme of **KALCORR** in the APL.

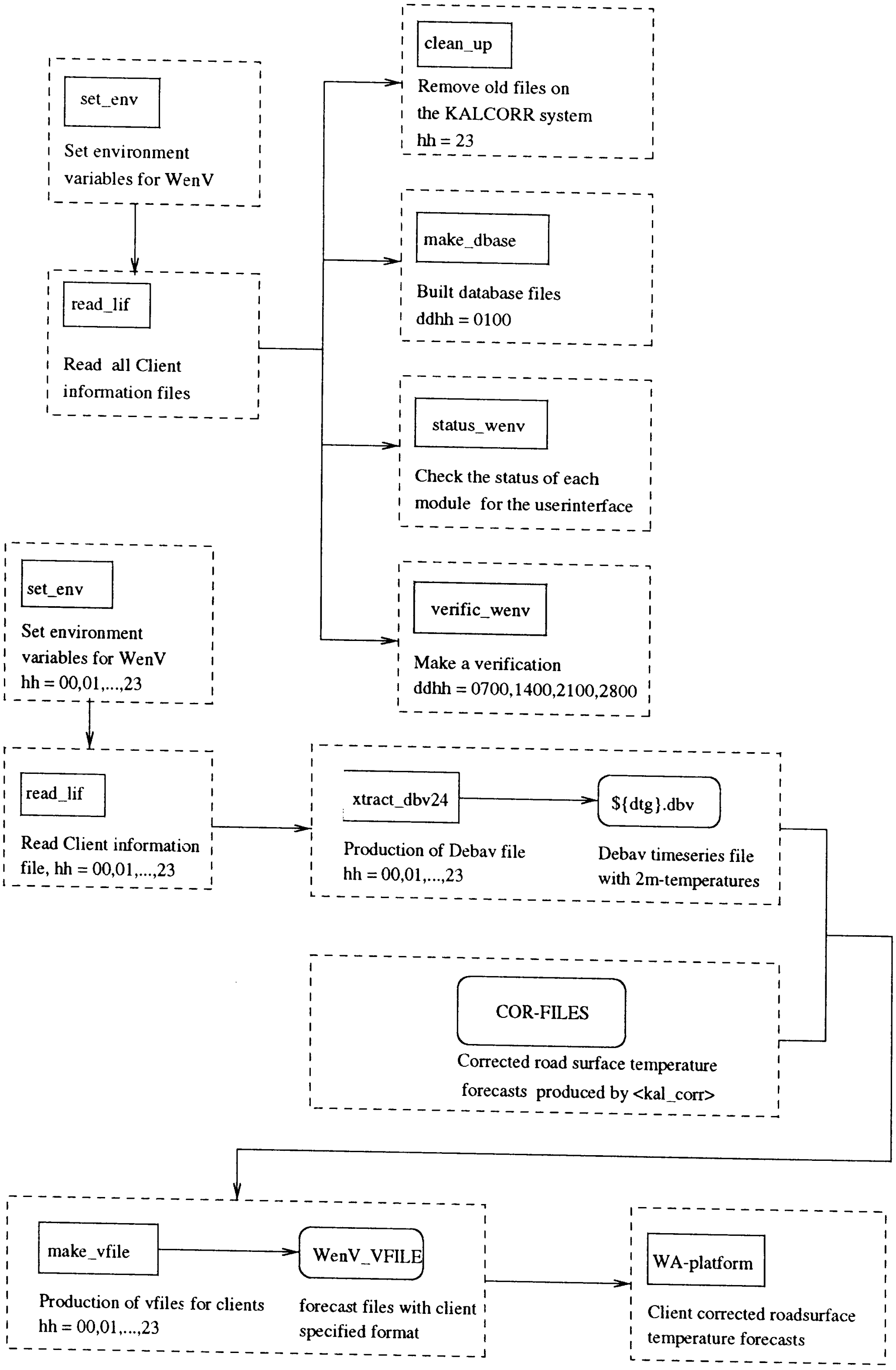


Figure 2.3: The operational subschemes of the WenV-system and **KALCORR**.

3 A description of the WenV-modules

The operational WenV-system consists of various interconnected modules. **KALCORR**, the heart of the system, is one of them. Each of these modules has its own well defined task, and is called by the WenV supervisor script <**oper_wenv.psc**> at predefined times. The supervisor script also keeps track of the status of these modules, and writes a status-report after each operational run.

Each of the separate modules is called with an identifier $actdtg, the actual calling time of the module. This identifier is used to name the intermediate and final results of each model run, and to combine the model-forecasts with the correct observations, restart filters, and so on. Therefore, this identifier acts as a necessary timelabel.
From figure (2.2) it is obvious that the WenV-system is divided into two separate parts which are connected at various points. There is a general part that sets the environment variables for the WenV-system, reads the client information files, extracts a 36-hour 2m-temperature forecast from the Hirlam timeseries, the LAMH_TBA files, checks if there are new observations available, and if so, puts them in the observation database. This general scheme is connected to the Kalman-filter routines, where observations are assimilated into the linear regression model, new regression coefficients and restart filters are produced 4 times a day, and a road surface temperature forecast correction is computed each hour of the day and exported to the clients of the WenV-system. In the remaining of this section we will discuss the general, and the specific Kalman-filter modules in some more detail. Furthermore, we will discuss the operational submodules, cf. figure (2.3), to some extend.

3.1 The initialisation-scheme

The WenV-system is initialised by the script <**init_wenv.psc**>. This script reads all location information files, and builds the WenV-directory tree for each district and sensor, for each Hirlam location, and it computes the initial restart filter file filter.ini, by calling the fortran routine <**makeinit**>, cf. also figure (A.1) in Appendix A. Furthermore the databases for the observations and corrections are initialised, by calling <**make_dbase**>, if they do not yet exist.

3.2 The general runscheme

Before we can call the **KALCORR** module in the WenV-system, we start the system by defining specific locations in the WenV-directory tree, cf. figure (A.1). We do this by running the script <**set_env**>, which sets the environment variables for the system. The general runscheme consists of two main actions. Both actions are controlled by client dependent variables such as the location-name, the location-code, the district-code, the number of sensors, the Hirlam location-name and location-code, and the status of each sensor, i.e. ON or OFF. These client information variables are set by running the script <**read_lif**>.

Now for each client two main actions are performed:

1. Extraction of model-forecasts: <**xtract_hrl36**>

   If at hh-hour, new Hirlam timeseries-files LAMH_TBA are available, we extract a 36-hour forecast of the 2m-temperature, the dewpoint temperature, the surface temperature, precipitation, cloud-cover, and the wind-direction and wind-speed. The extraction of the model-forecasts from the binary files is performed by <**tsfplot**>, and <**hirlam36**> writes a 36-hour forecast, including timelabels to the file \${dtg}.fct, with \$dtg = yymmddhh, the timelabel of the corresponding LAMH_TBA file. We put the resulting file \${dtg}.fct in the modeldata directory /hirlam/\${stat_code}, with \${stat_code} the Hirlam-location code.

2. Collection of observations: <**add_obs**>

   On the WA-platform, recent observed road surface temperatures are stored in ascii-files, named WenV_OBS_\${loc_code}_\${sens_code}_\${dtg}. Here \${loc_code} is the location-code of the sensor (an integer of 4 digits), \${sens_code} is the number of the sensor (an integer of 2 digits), and \${dtg} is the timelabel of the last available observation (dtg = yymmddhh). Each of these files contains observed road surface temperatures at hh, hh-1, and hh-2 hours. Each hour, the script <**add_obs**> checks if there are new observations available on the WA-platform. Available observations of road surface -and depth-temperatures, are added to the monthly database file with observations, i.e. the binary file WenV_OBS_\${yymm}_SW. Before the observations are accepted, they are checked on their lowerbounds and upperbounds, and on the deviation w.r.t. an average observation over all locations, i.e. a simple quality control is performed first.

3.3 The Kalman-filter runscheme

After setting the environment variables, the extraction of the 2m-temperature timeseries-files, and the collection and storage of observations, we run the Kalman-filter scheme, which consists of two actions. The first one, Kalman-correction, is performed each hour of the day, and the second action, Kalman-assimilation, takes place periodically 4 times a day at fixed hours.

1. Kalman-correction: <**kal_corr**>

   Each hour, after storage of actual observations in the observation database, the most recent model-forecasts \${dtg}.fct, with \$dtg $\leq$ \$actdtg, are corrected by means of the linear regression model (1.1). The regression model uses the last available observation at the actual calling time \$actdtg, which is extracted from the observation database, and the last observed roadsurface temperature at hh-hour, also from the database. As a default, we take hh = 12. If the observation at \$actdtg, i.e. *tobs(\$actdtg)*, does not exist, we extract the most recent prog at \$actdtg, i.e. *tcor(\$actdtg)*, from the correction database. Progs older than 12 hours are not accepted for forecast correction. Model forecasts older than 12 hours are also not accepted for forecast purposes. Note that in both cases, a corrected road surface temperature forecast cannot be produced, and an error message is written to the status information file of the specific run. If model forecasts **\${dtgfct}.fct**, with \$dtgfct $\leq$ \$actdtg, recent observations (or progs) **actobs**, and **tobs12**, are available, we compute a corrected road surface temperature forecast by means of formula (1.1). The resulting corrections, a 24-hour forecast, are stored in the correction database, a binary file named WenV_COR_\${yymm}_SW. In the regression formula we use regression coefficients that are 48 hours old, and these are stored in the subdirectory /kalman/\${dis_code}\${sens_code}, in the binary file WenV_REG_\${hhfct}_SW. Here \${dis_code} is the district-code of the location, \${sens_code} is the sensor number, and \${hhfct} is the hour at which the model-forecast file \${dtgfct}.fct was produced. If the file with regression coefficients does not exist, forecast correction is not possible.

2. Kalman-assimilation: <**kal_assim**>

   Periodically, at hh = 00,06,12 and 18, this script extracts a 36-hour history of observations from the observation database, and puts them in a file. The observations are combined with model-forecasts in the file \${assdtg}.fct, \$assdtg = \$actdtg - 36 hours, and \$actdtg the actual calling time of this routine. The comparison is done

by assimilating the observations in the linear regression model (1.1), and updating the regression coefficients by means of a Kalman-filter. The filter is initialised with starting conditions contained in the file WenV_RES_${hhres}_SW, where $hhres is the hour-label of the previous mentioned model forecast file.
After the assimilation, the newly computed regression-coefficients are written to the file WenV_REG_${hhreg}_SW, with $hhreg = $hhres, and the restart values for the next run are contained in WenV_RES_${hhres}_SW.

3.4 The operational subscheme

The operational subscheme checks the status of each WenV-run, removes old files on the system, builds the new databases for the observations, and corrections at the first day of the month, at ddhh = 0100, and writes a verification report for each client, once every 7 days. The second subscheme produces client specific vfiles.

1. Remove old files on the system: <**clean_up**>

   This script removes old files on the WenV-system, removes old crontab-information, log-information, and status-information. Model-forecast files, and files with roadsurface temperature corrections are removed after 5 days.

2. Built database files: <**make_dbase**>

   Each month at ddhh = 0100, we built the observation database, and the correction database. The district-codes, and location-codes, necessary for initialisation of the databases, are read from the location information files (∗.lif). The number of sensors in the databases is fixed to $nsens = 6.

3. Check the status of the main modules: <**status_wenv**>

   This script checks the existence of various files on the WenV-system, and checks the status of the Kalman-correction, and the Kalman-assimilation. The results are written to the file **status_wenv.${dis_code}**, with ${dis_code} the two-letter district-code.

4. Make a verification: <**verific_wenv**>

   This script makes a verification for each district, and sensor. The results are written to the file veri_wenv.${dis_code}.

5. Extraction of DEBAV timeseries: <**xtract_dbv24**>

   Each hour Debav timeseries containing information about the present weather are produced by means of the script <**xtract_dbv24**>. A 24 hour forecast from the Debav database is written to the file ${dtg}.dbv, with $dtg = yymmddhh the corresponding timelabel. The resulting file is stored in the data directory /debav/${dbv_code}, with ${dbv_code} the Debav location-code.

6. Production of vfiles: <**makevfile**>

   For the clients of the WenV system the corrected road surface temperatures are stored in vfile format and returned to the clients.

Acknowledgements

The author wishes to thank Kees Kok, Kees Lemcke and Hans de Vries for reading most part of the first version of this report. Furthermore the development of the WenV-system in the APL has been greatly supported by stimulating discussions with Kees Lemcke and Hans de Vries.

References

[1] Jacobs, A.J.M., *Een Kalman-correctieschema voor de wegdektemperatuurverwachtingen van het VAISALA-model.*
Technisch rapport KNMI, TR-180, De Bilt, 1995.

[2] Kok, C.J., en C.L.M. van Vliet, *Gids ten behoeve van de weersverwachting tijdens de WK94.*
Memorandum WM-94-13 (ongepubliceerd manuscript), KNMI, De Bilt, 1994.

[3] Maybeck, P.S., *Stochastic models, estimation, and control, Vol. I.*
Academic Press, New York, 1979.

[4] Persson, A., *Kalman filtering - a new approach to adaptive statistical interpretation of numerical meteorological forecasts.*
ECMWF newsletter, no. 46 (1989); p. 16-20.

[5] Rayer, P.J., *The Meteorological Office Forecast Road Surface Temperature Model.*
The meteorological magazine, Vol. 116, no. 1379, 1987, p. 180-191.

[6] Thornes, J.E., *The prediction of ice formation on motorways.*
Ph.D. thesis, University of London, 1984.

A The WenV directory tree

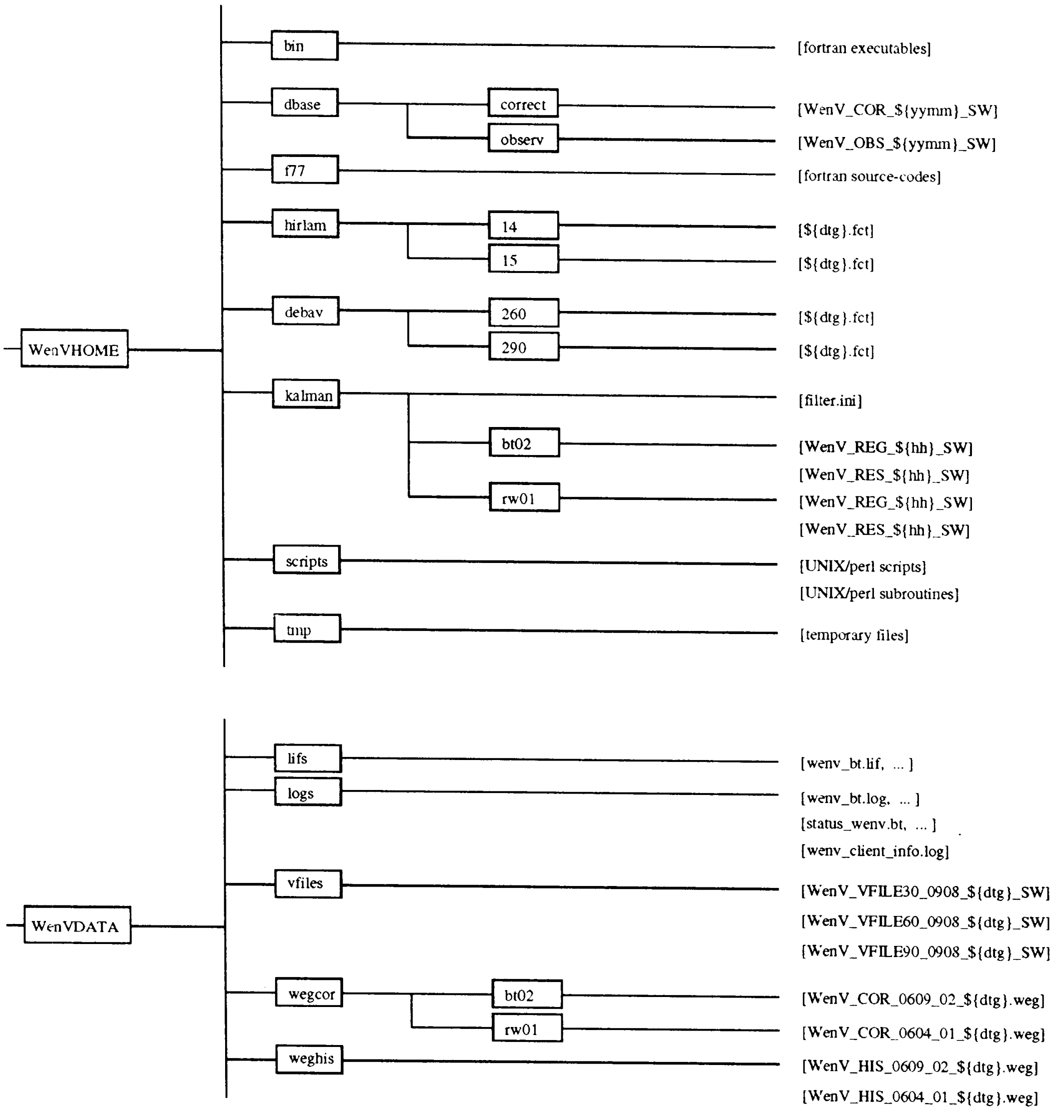


Figure A.1: The WenV directory tree in the APL.

B Filenames in the WenV-system

In this appendix we give a list of all important files in the WenV-system.
Here **dtg** = **yymmddhh** is the date-time group of the WenV-runs, where for **yy**, **mm**, **dd**, and **hh** respectively the year, month, day and hour of the actual run have to be substituted.
Furthermore we use **dis** for the district-code, **loc** for the location-code, **sens** for the sensor-code, and **stat** for the Hirlam location-code (cf. also appendix A and F).

wenv_${dis}.lif	Location information file (ascii)
LAMH_TBA_${dtg}00_00000_TW	Hirlam single-level timeseries file (binary)
DEBAV_${dtg}.S01	Debav timeseries file (ascii)
${dtg}.fct	WenV model-forecast file
${dtg}.dbv	Debav timeseries file for station (ascii)
WenV_OBS_${loc}_${sens}_${dtg}	Observation file from WA-platform (ascii)
WenV_OBS_${yymm}_SW	Observation database (binary)
WenV_COR_${yymm}_SW	Correction database (binary)
WenV_RES_${hh}_SW	Restart-file for Kalman-filter (binary)
WenV_REG_${hh}_SW	File with regression-coefficients (binary)
WenV_${dis}.log	Extended log-information for WenV runs (ascii)
status_wenv.${dis}	Status information for WenV runs (ascii)
wenv_client_info.log	General log-info for reading lif-files (ascii)
WenV_COR_${loc}_${sens}_${dtg}.weg	Corrected forecasts for WA-platform (ascii)
WenV_HIS_${loc}_${sens}_${dtg}.weg	History of observations for WA-platform (ascii)
WenV_VFILE**_${loc}_${dtg}_SW	Vfile for client (binary)

C A description of the WenV database-files

The WenV database files are called WenV_OBS_${yymm}_SW for the hourly observations, and WenV_COR_${yymm}_SW for the hourly corrections, both during the month **yymm**. These database files contain the observations and corrections for a selected set of locations. For the winter-season 1996–1997, these locations are given in table E.2. For each location we have defined 6 sensors, some of them are active and some of them are not, but 6 is the maximum (cf. Appendix F). For the initialisation of the databases, for the storage and extraction of data, we have written various fortran application programs:

Application programs for the observation database

inidbobs	Initialisation of observation database
putdbobs	Storage of observations in database
getdbobs	Extraction of observations from database
gacdbobs	Extraction of actual observation, $T_{\mathrm{obs}}(\$actdtg)$, from database
ghhdbobs	Extraction of $T_{\mathrm{obshh}}(\$dtghh)$, \$dtghh $\leq$ \$actdtg, from database

Application programs for the correction database

inidbcor	Initialisation of correction database
putdbcor	Storage of corrections in database
getdbcor	Extraction of corrections from database
gprdbcor	Extraction of prog, $T_{\mathrm{cor}}(\$actdtg)$, from database

D A description of the WenV filter-files

The Kalman-filter file WenV_RES_${hhres}_SW, is the restart-filter for the next Kalman-update, that takes place 24 hours later. The file contains the covariance-matrix P for the regression coefficients X_0, X_1, X_2 and X_3. This covariance-matrix is a combination of:

(a) The covariance of the model-error in the time-evolution equation for the regression coefficients, i.e. the matrix Q,

(b) The covariance of the error in the observation equation – the statistical model equation (1.1) – i.e. r with $r \geq 0$. Note that r is a scalar function, and $\sqrt{r}$ is the standard deviation for the uncertainty in the measurements, and the choice for our statistical model (1.1).

A more precise description of the model covariances and their computation, can be found in Jacobs [1995], TR-180.

The covariances depend on the actual calling-time t_a, and the length of the forecast interval t_f, i.e. the time-lag w.r.t. the last available observation (cf. also the Introduction section). For each calling time t_a, we accept recent model-forecasts for correction, unless they are older than 12 hours. This implies the restriction $t_v \leq t_a \leq t_v + 12$, where t_v is the hour at which a recent model-forecast is produced. In other words, the time-lag w.r.t. the last available model-forecast file is 12 hours at maximum.

In the WenV-system we produce model-forecasts 36 hours ahead, and therefore $\max(t_f) = 36 - (t_a - t_v)$. The restart-filter file has the following format:

$$t_f \rightarrow \left\{ \begin{array}{cccc} t_a = t_v & t_a = t_v + 1 & \cdots\cdots & t_a = t_v + 12 \\ X_0, X_1, X_2, X_3 & & & \\ P & & & \\ Q & & & \\ r & & & \end{array} \right. \tag{D.1}$$

where t_f varies in each column from $t_f = 0$ to $t_f = 36 - (t_a - t_v)$. Note that the differences between t_a and t_v are computed in hours, therefore we work with their corresponding dtg's instead.

The file WenV_REG_${hhreg}_SW, which contains the regression-coefficients for the Kalman-filter, has the same format, accept that the covariances P, Q, and r, are left out.

Some notes on the initial values for the iteration-process can be found in Jacobs [1995], TR-180. The initial conditions are

$$X_0 = X_1 = X_3 = 0, \qquad X_2 = \begin{cases} 1 & \text{if } t_f = 0 \\ 0 & \text{otherwise} \end{cases} \tag{D.2}$$

for the regression-coefficients, and for the covariances

$$P_0 = \begin{pmatrix} 0.7 & 0.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 \end{pmatrix}, \qquad Q = \begin{pmatrix} 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 \end{pmatrix}, \tag{D.3}$$

and for the error covariance in the observation equation we take $r \equiv 0.25$.
The presence of system-noise in the time-evolution equation for the regression coefficients is still subject of further research and is therefore neglected in the present version of the WenV-system. In this version we also neglect the T_{obshh} term in the regression model until we have some more experience with the time it takes for the model to spin-up with this additional predictor.

E A list and a map of the WenV locations

In this appendix we give a map and a list of all the WenV-locations in the Netherlands. Table (E.2) gives the names and location-codes of all locations. The list is suplemented with the location-coordinates, and the corresponding, most near and suitable, Hirlam location-codes. The Hirlam location-codes, location-names, and coordinates, can be found in table (E.1). Furthermore, figure (E.1) shows the GMS[4] locations for the WenV-system, and the Hirlam locations at the correct positions on the map.

Nr.	Location name	Hirlam coord. LAT	Hirlam coord. LON
16.	Rotterdam	51.73	04.48
19.	Stadskanaal	52.85	06.78
20.	Drachten	53.01	06.00
21.	Medemblik	52.68	04.96
22.	Kampen	52.53	05.74
23.	Ommen	52.37	06.52
24.	Oldenzaal	52.21	07.29
25.	Uithoorn	52.20	04.72
26.	Amersfoort	52.05	05.49
27.	Zevenaar	51.90	06.26
28.	Tilburg	51.58	05.25
29.	Venraij	51.43	06.01
30.	Vlissingen	51.39	03.49
31.	Roosendaal	51.25	04.25
32.	Beek/Zuid-Limburg	50.95	05.76

Table E.1: Hirlam locations, and location-codes for WenV.

---

[4] GMS means "Gladheids-meldsysteem"

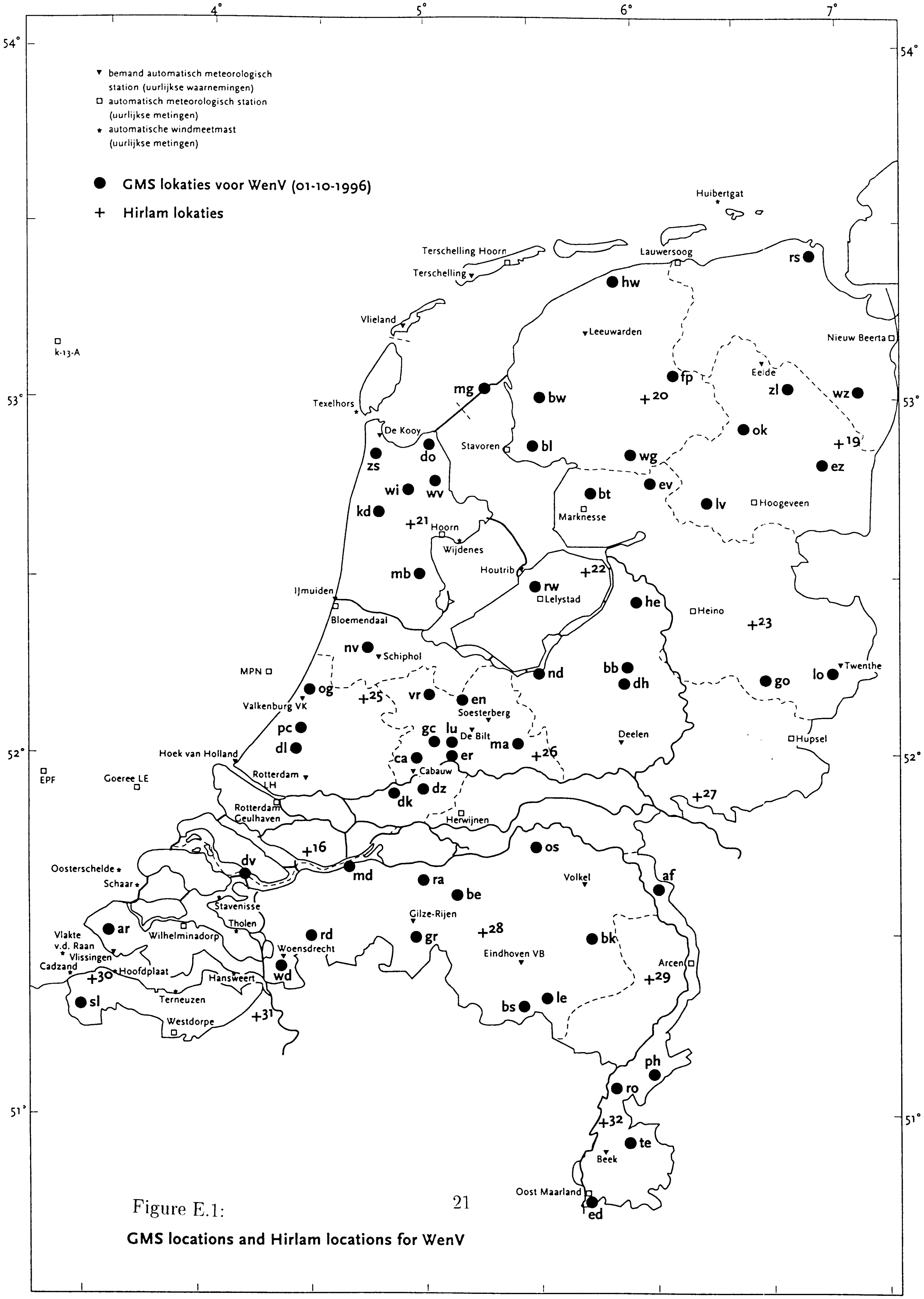


Figure E.1:

GMS locations and Hirlam locations for WenV

Nr.	Location name	Location codes		Amersf. Coord.		LAT	LON	Hirlam LOC.nr.
				X:	Y:			
1.	Lonneker	0501	lo	258.0	474.0	52:14’43”	6:53’46”	24
2.	Goor	0505	go	237.0	472.0	52:13’51”	6:35’17”	23
3.	Eesveen	0512	ev	205.0	537.0	52:49’ 8”	6: 7’46”	20
4.	Winkel	0534	wi	122.0	529.0	52:44’54”	4:53’56”	25
5.	Middenbeemster	0542	mb	125.5	506.0	52:32’31”	4:57’10”	25
6.	Nieuw-Vennep	0553	nv	107.0	475.0	52:15’43”	4:41’ 4”	25
7.	Vreeland	0556	vr	127.5	470.5	52:13’22”	4:59’ 6”	26
8.	West Fr. Vaart	0521	wv	129.5	535.0	52:48’10”	5: 0’34”	25
9.	Middelgronden	0533	mg	150.5	565.0	53: 4’22”	5:19’14”	20
10.	Oss	0908	os	163.0	416.5	51:44’17”	5:30’13”	28
11.	Runderweg	0604	rw	165.0	503.5	52:31’12”	5:32’ 6”	22
12.	Bant	0609	bt	180.5	531.0	52:46’ 0”	5:45’56”	22
13.	De Hucht	1402	dh	189.0	466.5	52:11’12”	5:53’ 6”	26
14.	Nulde	1406	nd	165.0	475.8	52:16’16”	5:32’ 3”	26
15.	Heerde	1412	he	198.0	492.0	52:24’55”	6: 1’11”	26
16.	De Zouwe	0361	dz	128.5	439.0	51:56’23”	5: 0’ 8”	26
17.	De Kaai	0358	dk	114.7	426.4	51:50’00”	4:45’56”	16
18.	Galecoppersbrug	0205	gc	135.0	452.5	52: 3’41”	5: 5’45”	26
19.	Maarn	0208	ma	153.2	453.0	52: 3’59”	5:21’41”	26
20.	Eemnes	0214	en	144.2	472.1	52:14’16”	5:13’46”	26
21.	Cabouw (Lopik)	0252	ca	125.0	442.5	51:58’16”	4:57’ 4”	26
22.	Zijpersluis	0529	zs	111.3	536.4	52:48’51”	4:44’22”	25
23.	Roosendaal	0711	rd	88.9	393.9	51:31’52”	4:26’ 6”	31
24.	Moerdijkbrug	0713	md	103.3	414.2	51:42’54”	4:38’22”	16
25.	Woensdrecht	0721	wd	73.5	382.5	51:25’36”	4:12’56”	31
26.	Eijsden	0817	ed	179.0	309.5	50:46’35”	5:43’41”	32
27.	Afferden	0830	af	197.7	405.7	51:38’22”	6: 0’16”	29
28.	Posterholt	0836	ph	201.0	348.0	51: 7’15”	6: 2’41”	32
29.	Roosteren	0852	ro	185.8	343.3	51: 4’46”	5:49’38”	32
30.	Ten Esschen	0853	te	192.8	324.2	50:54’27”	5:55’30”	32
31.	Gilze-Rijen	0951	gr	126.2	395.6	51:32’59”	4:58’21”	28
32.	Leende	0955	le	166.5	372.8	51:20’43”	5:33’10”	28
33.	Borker & Schaft	0963	bs	159.0	367.8	51:18’ 2”	5:26’42”	28
34.	Bakel	0965	bk	181.0	390.3	51:30’ 8”	5:45’44”	29
35.	Arnestein	1004	ar	34.3	391.5	51:30’ 2”	3:38’56”	31
36.	De Hevel	1005	dv	77.7	412.0	51:42’00”	4:16’00”	16
37.	Sluis	1014	sl	15.5	370.5	51:18’27”	3:23’12”	31
38.	Oranjekanaal	1102	ok	231.8	545.5	52:53’32”	6:31’44”	19
39.	Lageveen	1103	lv	220.2	521.8	52:40’51”	6:21’ 7”	22
40.	Zuidlaren	1111	zl	240.0	567.2	53: 5’ 9”	6:39’23”	19
41.	Emmen-Zuid	1115	ez	252.8	531.3	52:45’40”	6:50’12”	19
42.	Frieschepalen	1152	fp	208.9	572.0	53: 7’59”	6:11’35”	20
43.	Roodeschool	1155	rs	247.0	605.0	53:25’26”	6:46’18”	19
44.	Winschoterzijl	1160	wz	269.0	574.5	53: 8’46”	7: 5’30”	19
45.	Bolsward	1202	bw	165.0	563.2	53: 3’23”	5:32’13”	20
46.	Wolvega	1207	wg	193.0	550.0	52:56’12”	5:57’11”	20
47.	Balk	1217	bl	163.8	543.0	52:52’30”	5:31’ 6”	20
48.	Holwerd	1231	hw	189.0	598.5	53:22’21”	5:53’55”	20
49.	Delft	0555	dl	87.0	443.0	51:58’20”	4:23’52”	16
50.	Prins Clauspl.	0554	pc	85.5	453.0	52: 3’43”	4:22’27”	16
51.	Beekbergen	1401	bb	189.5	466.8	52:11’19”	6: 1’26”	26
52.	Everdingen	0362	er	134.0	441.0	51:57’29”	5: 4’56”	26
53.	Raamsdonkveer	0905	ra	120.0	410.5	51:40’59”	4:52’53”	28
54.	Berkel Enschot	0912	be	138.5	399.0	51:34’50”	5: 8’58”	28
55.	Koedijk	0528	kd	110.7	522.5	52:41’21”	4:43’57”	25
56.	Oegsgeesterk.	0551	og	92.0	467.5	52:11’35”	4:27’58”	25
57.	Den Oever	0532	do	131.5	549.5	52:55’59”	5: 2’17”	25
58.	Lunetten	0207	lu	139.0	452.0	52: 3’25”	5: 9’15”	26

Table E.2: WenV locations, coordinates and Hirlam-locations.

F The WenV location information file

All client information is contained in the location information file wenv_${dis_code}.lif, with ${dis_code} the district-code of the client. This file controls the complete WenV-system. Besides road surface information, such as the precise coordinates of the location and the physical type of the surface, the lif-file defines the number, and the status of the sensors, adds suitable Hirlam locations for the production of model forecasts, Debav locations for the production of vfiles, and defines unique codes that identify data from different sources. Table (F.1) gives an example of such a lif-file.

```
Location Information File

Dienstkring            :   Apeldoorn
Locatie                :   De Hucht
Lettercode             :   dh
Weg                    :   RW1
Wegdek                 :   ZOAB(1996)
Km paal                :   78.100
Amersfoortse co        :   189.0 466.5
Lat/Lon coord          :   52 11 12 / 5 53 6

Location-name          =   DeHucht
Location-code          =   1402
District-code          =   dh
Number-of-sensors      =   6
Hirlam Station-name    =   Amersfoort
Hirlam Station-code    =   26

Status of sensors:

Status-of-sensor-01    =   ON
Status-of-sensor-02    =   ON
Status-of-sensor-03    =   ON
Status-of-sensor-04    =   ON
Status-of-sensor-05    =   ON
Status-of-sensor-06    =   ON

Production of vfiles:

vfile-prod             =   ON
Debav Station-name     =   Twente
Debav Station-code     =   290
```

Table F.1: The Location Information File.